

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte de Proyecto de Integración: B2B Tetra Pak - Alpura
Modalidad: Experiencia Profesional

Primera versión

Trimestre 14-O

Alumno: Daniel López Martínez
Matrícula: 207205239
Correo electrónico: dalopezmartinez@gmail.com

Empresa: Ganaderos Productores de Leche Pura, S.A.P.I de C.V.
Departamento: Informática
Sección: Desarrollo Oracle

Jefe Inmediato: Eduardo Tovar López
Puesto: Jefe de Desarrollo Oracle
Nivel Académico: Licenciatura
Correo electrónico: etovar@alpura.com

8 de diciembre de 2014

Yo, Eduardo Tovar López, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la 12 Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Nombre y firma del jefe directo

Yo, Daniel López Martínez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Daniel López Martínez

Nombre y firma del alumno

Tabla de contenido

Resumen	4
I. Introducción	5
1. Antecedentes	6
2. Periodo	6
3. Objetivo general	6
4. Objetivos específicos.....	6
5. Justificación.....	6
6. Marco Teórico	7
6. Descripción del trabajo.....	12
6.1 Recepción de requisiciones	14
6.1.1 Tabla de interface	17
6.1.2 Tabla de datos derivados	18
6.1.3 Disparador	19
6.1.4 Validación de líneas después del disparador	23
6.1.5 Importación de requisiciones mediante BPEL SOA	26
6.1.6 Obtención del estatus del concurrente de Importación de Requisiciones.....	29
6.1.7 Notificación de mensajes por medio de correo electrónico.....	31
6.2 Envío de orden de compra.....	35
6.3 Confirmación de orden de compra	36
6.4 Notificación de Entrega.....	36
Referencias	37

Resumen

El proyecto de Alpura tiene como finalidad la implementación de un esquema comercio electrónico de negocio a negocio o Business to Business (B2B) para el procedimiento de compra de material a Tetra Pak, reduciendo tanto recursos como tiempo al procedimiento que actualmente se está realizando. El proceso se describe a continuación:

El usuario de Alpura encargado de la tarea de administrar el material proveniente de Tetra Pak se conectará al portal de dicho proveedor para generar, dentro del mismo, una requisición de material necesario dentro de la compañía. Una vez registrada la transacción, Tetra Pak enviará la información de la misma a Alpura, donde a partir de ello se crea la requisición interna de forma automática y queda disponible para su aprobación. El usuario de Alpura, por medio de su sistema de planificación de recursos empresariales (ERP, por sus siglas en inglés, enterprise resource planning) la E-Business Suite de Oracle (EBS), aprueba la requisición y se genera una orden de compra y de forma automática se notifica a Tetra Pak y se queda en espera de su confirmación. Tetra Pak envía un nuevo mensaje a Alpura donde notifica la recepción de orden de compra con el detalle de la transacción. A partir de ello Tetra Pak enviará el material a las instalaciones de Alpura, donde se realizará la recepción del mismo, completando así el proceso.

Toda la parte de análisis y requerimientos se realizaron por la parte funcional de Alpura, entregando a los desarrolladores los requerimientos documentados de acuerdo a las políticas de Oracle, también conocidos como MD050, para comenzar la implementación.

El desarrollo requirió diferentes tecnologías y lenguajes para su implementación, entre las que se encuentran PL/SQL por medio de procedimientos almacenados de base de datos, disparadores, procedimientos calendarizados. A su vez se crearon pantallas para la EBS (Oracle forms) también desarrolladas mediante PL/SQL. Para generar una parte del desarrollo se utilizó el paradigma de arquitectura orientada a servicios (SOA, por sus siglas en inglés, service oriented architecture) por medio del lenguaje de ejecución de procesos de negocio con servicios web (BPEL, por sus siglas en inglés, business process execution language).

El proyecto fue desarrollado en la instancia de pruebas y el viernes 31 de octubre de 2014 liberado a la Instancia de producción, por lo que ya se encuentra operando.

I. Introducción

El proyecto se implementó en la empresa Ganaderos Productores de Leche Pura, S.A.P.I de C.V., mejor conocida como Alpura.

Alpura es una empresa de productos lácteos 100% mexicana que emplea aproximadamente a 5,000 trabajadores directos y otros 7,000 indirectos, entre sus 123 ranchos propios, promotoría y distribución, lo que representa una generación de más de 12,000 fuentes de trabajo.

La planta de Alpura está ubicada en Autopista México-Querétaro Km. 37, Parque Industrial Cuamatla, Cuautitlán Izcalli, Estado de México, C.P. 54730.

Algunas de las tareas asignadas a la sección de desarrollo Oracle son implementar sistemas con lenguajes de programación como Oracle PL/SQL y Java para la EBS de Oracle, utilizando tecnologías como Oracle Forms, Oracle Application Framework (OAF), Application Development Framework (ADF) y BPEL servidores con Linux Oracle.

Los desarrollos se realizan de acuerdo a las necesidades de cada uno de los usuarios de las diferentes áreas de la compañía, siendo aprobadas y analizadas por la parte funcional (expertos en los módulos que componen la EBS) correspondiente, la cual plasma en documentos los requerimientos para ser entregados e inicie su implementación.

1. Antecedentes

El proyecto surge de la necesidad de mejorar el proceso de generación de requisiciones y órdenes de compra con Tetra Pak. Anteriormente el ciclo de trabajo no estaba automatizado, generando más trabajo del usuario, mayores tiempos de espera, tiempo de retardo, entre otros problemas, y todo ello derivaba en pérdidas económicas. Además se buscaba darle el enfoque B2B.

La implementación del proyecto ayuda considerablemente a reducir todos los factores mencionados con anterioridad y permite llevar un control más ordenado de este tipo de transacciones. De igual forma marca la pauta para poder llevar al mismo esquema otros procedimientos internos de Alpura.

2. Periodo

El proyecto fue implementado en la Planta de Alpura, con una duración de desarrollo de aproximadamente tres meses. Actualmente se encuentra operando. El 1 de agosto de 2014 comenzó su análisis y posteriormente la implementación.

3. Objetivo general

El objetivo del proyecto es automatizar el proceso con el que anteriormente se contaba para el procedimiento de generación de requisición de material y órdenes de compra con el proveedor Tetra Pak. Para ello se analiza el proceso y se decide que la mejor forma de llevarlo a cabo es un cambio a un esquema B2B.

Este proyecto será el ejemplo para modificar varios otros procedimientos internos al mismo modelo.

4. Objetivos específicos

Particularmente el proyecto tiene como finalidad reducir tiempos del proceso, los cuáles eran extensos y dependían mucho de los usuarios. Al implementar un esquema de tipo B2B se gana rapidez y seguridad para las comunicaciones, una integración más directa de los datos de la transacción, la posibilidad de generar más transacciones en menor tiempo; todo ello se buscaba.

Como consecuencia de los anteriores factores y como uno de los principales objetivos se requiere reducir el costo de recursos.

5. Justificación

Después de poco más de un mes en que el proyecto ya está funcionando en la instancia productiva claramente se puede observar que los objetivos han sido alcanzados; el procedimiento automatizado ha mejorado considerablemente en tiempos de ejecución, ha facilitado el manejo de

los usuarios, ha disminuido el error humano, ha agilizado trámites. Asimismo ha mejorado la comunicación con el proveedor. Como consecuencia de todo lo anterior se está alcanzado el objetivo primordial, la reducción de costos.

6. Marco Teórico

B2B (Negocio a negocio)

Por sus siglas en inglés business to business. Se refiere a las transacciones comerciales realizadas de manera electrónica. Principalmente utilizan la tecnología EDI.

Algunas de las ventajas de trabajar con B2B para las empresas implicadas son:

- Rapidez y seguridad de las comunicaciones
- Integración directa de los datos de la transacción en los sistemas informáticos de la empresa
- Posibilidad de recibir mayor número de ofertas o demandas, ampliando la competencia
- Despersonalización de la compra con lo que se evitan posibles tratos de favor
- Abaratamiento del proceso: menos visitas comerciales, proceso de negociación más rápido, etc. Por tanto, los compradores pueden pedir una reducción de precios en virtud del menor coste de gestión, o los vendedores incrementar su margen comercial.

Solo por establecer un ejemplo de referencia, el B2B aplica a la relación entre un fabricante y el distribuidor de un producto y también a la relación entre el distribuidor y el comercio minorista pero no a la relación entre el comerciante y su cliente final [Business-to-business].

EDI (Intercambio electrónico de datos)

Por sus siglas en inglés electronic data interchange; es la transmisión de datos estructurada por medios electrónicos. Se usa para transferencia de información de un sistema a otro. Algunos de los formatos que de utilizan son EDIFACT, XML, etc.

Albaranes, facturas, órdenes de compra y otros documentos comerciales electrónicos pueden tramitarse directamente desde la computadora de la empresa emisora a la de la empresa receptora, con gran ahorro de tiempo y evitando muchos errores, propios de la comunicación tradicional en papel.

Ejemplos de aplicaciones compatibles con EDI son BizLayer, MIC2000 ERP+, IBM WebSphere Datapower Appliances, etc [Intercambio electrónico de datos].

EDIFACT (Intercambio electrónico de datos para la administración, comercio y transporte)

Por sus siglas en ingles, electronic data interchange for administration, commerce and transport. Es un formato estándar de la ONU para el intercambio de datos electrónicos a nivel mundial. El sistema EDI permite el intercambio (envío y recepción) de documentos comerciales por vía telegráfica [EDIFACT].

La Figura 1 muestra un ejemplo de EDI para una requisición de materiales.

```
UNA:+.? 'UNB+UNOC:3+2742:ZZ+GPL860521FW8:ZZ+141202:1946+83'  
UNH+1+DESADV:D:96A:UN:EAN005'  
BGM++0207477434'DTM+17:20141202:102'  
DTM+137:20141202:102'RFF+ON:1465198-8'  
NAD+SU+++2742'NAD+DP+++14015'NAD+BY+++14015'CP+1'PAC+23'MEA+PD+AAD+KGM:17255.789'CP+2+1'PAC+1'PCI+33E'GIN+BJ+300000006411970070'  
LIN+1'PIA+1+DIENTC0206:BP::92'QTY+12:35430:pz'RFF+ON:1465198-19:20'RFF+ZZZ:F'CP+3+1'PAC+1'PCI+33E'GIN+BJ+300000006411980147'  
LIN+2'PIA+1+DIENTC0207:BP::92'QTY+12:43525:pz'RFF+ON:1465198-19:30'RFF+ZZZ:F'CP+4+1'PAC+1'PCI+33E'GIN+BJ+300000006410140283'  
LIN+3'PIA+1+DIENTC0216:BP::92'QTY+12:11940:pz'RFF+ON:1465198-7:20'RFF+ZZZ:F'CP+5+1'PAC+1'PCI+33E'GIN+BJ+300000006414320360'  
LIN+4'PIA+1+DIENTC0079:BP::92'QTY+12:29490:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+6+1'PAC+1'PCI+33E'GIN+BJ+300000006414320285'  
LIN+5'PIA+1+DIENTC0079:BP::92'QTY+12:46875:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+7+1'PAC+1'PCI+33E'GIN+BJ+300000006414320353'  
LIN+6'PIA+1+DIENTC0079:BP::92'QTY+12:38250:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+8+1'PAC+1'PCI+33E'GIN+BJ+300000006414320315'  
LIN+7'PIA+1+DIENTC0079:BP::92'QTY+12:46920:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+9+1'PAC+1'PCI+33E'GIN+BJ+300000006414320391'  
LIN+8'PIA+1+DIENTC0079:BP::92'QTY+12:39015:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+10+1'PAC+1'PCI+33E'GIN+BJ+300000006414320384'  
LIN+9'PIA+1+DIENTC0079:BP::92'QTY+12:19600:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+11+1'PAC+1'PCI+33E'GIN+BJ+300000006414320414'  
LIN+10'PIA+1+DIENTC0079:BP::92'QTY+12:7820:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+12+1'PAC+1'PCI+33E'GIN+BJ+300000006417430011'  
LIN+11'PIA+1+DIENTC0169:BP::92'QTY+12:33900:pz'RFF+ON:1465198-29:10'RFF+ZZZ:P'CP+13+1'PAC+1'PCI+33E'GIN+BJ+300000006417430226'  
LIN+12'PIA+1+DIENTC0169:BP::92'QTY+12:33900:pz'RFF+ON:1465198-29:10'RFF+ZZZ:P'CP+14+1'PAC+1'PCI+33E'GIN+BJ+300000006417430509'  
LIN+13'PIA+1+DIENTC0169:BP::92'QTY+12:31555:pz'RFF+ON:1465198-29:10'RFF+ZZZ:P'CP+15+1'PAC+1'PCI+33E'GIN+BJ+300000006414320377'  
LIN+14'PIA+1+DIENTC0079:BP::92'QTY+12:14700:pz'RFF+ON:1465198-21:10'RFF+ZZZ:F'CP+16+1'PAC+1'PCI+33E'GIN+BJ+300000006417430516'  
LIN+15'PIA+1+DIENTC0169:BP::92'QTY+12:4080:pz'RFF+ON:1465198-29:10'RFF+ZZZ:P'CP+17+1'PAC+1'PCI+33E'GIN+BJ+300000006417430431'  
LIN+16'PIA+1+DIENTC0169:BP::92'QTY+12:33895:pz'RFF+ON:1465198-29:10'RFF+ZZZ:P'CP+18+1'PAC+1'PCI+33E'GIN+BJ+300000006414481238'  
LIN+17'PIA+1+DIENTC0216:BP::92'QTY+12:6780:pz'RFF+ON:1465198-24:10'RFF+ZZZ:F'CP+19+1'PAC+1'PCI+33E'GIN+BJ+300000006414480224'  
LIN+18'PIA+1+DIENTC0216:BP::92'QTY+12:33900:pz'RFF+ON:1465198-24:10'RFF+ZZZ:F'CP+20+1'PAC+1'PCI+33E'GIN+BJ+300000006414481177'  
LIN+19'PIA+1+DIENTC0216:BP::92'QTY+12:32040:pz'RFF+ON:1465198-24:10'RFF+ZZZ:F'CP+21+1'PAC+1'PCI+33E'GIN+BJ+300000006414530738'  
LIN+20'PIA+1+DIENTC0187:BP::92'QTY+12:27120:pz'RFF+ON:1465198-23:20'RFF+ZZZ:F'CP+22+1'PAC+1'PCI+33E'GIN+BJ+300000006414530745'  
LIN+21'PIA+1+DIENTC0187:BP::92'QTY+12:27120:pz'RFF+ON:1465198-23:20'RFF+ZZZ:F'CP+23+1'PAC+1'PCI+33E'GIN+BJ+300000006410210306'  
LIN+22'PIA+1+DIENTC0210:BP::92'QTY+12:33615:pz'RFF+ON:1465198-8:10'RFF+ZZZ:F'CP+24+1'PAC+1'PCI+33E'GIN+BJ+300000006410210313'  
LIN+23'PIA+1+DIENTC0210:BP::92'QTY+12:13490:pz'RFF+ON:1465198-8:10'RFF+ZZZ:F'UNT+219+1'UNZ+1+83'
```

Figura 1. Ejemplo de un EDI para generar una requisición de materiales.

EBS de Oracle

Por sus siglas en inglés, Oracle E-Business Suite, es la suite más completa de aplicaciones empresariales globales totalmente integradas. Es un ERP. Actualmente se trabaja y se desarrolla sobre la versión 12.1.3 en Alpura.

La EBS permite manejar diversos procesos de negocios. Esto es por medio de módulos; por ejemplo finanzas, recursos humanos, contabilidad, cuentas por cobrar, cuentas por pagar, etc.

Tiene incorporadas diversas tecnologías para su funcionamiento, de base de datos, RDBMS, PL/SQL, Java, .NET, HTML, XML; también Oracle Forms Server, Oracle Reports Server, Apache Web Server, Oracle Discoverer, Jinitiator y Sun's Java, entre otras.

Concurrente de Oracle

Es el medio que tienen los usuarios para ejecutar procesos dentro de la EBS. Los Programas concurrentes son de uso común dentro de Oracle E-Business Suite como un mecanismo para ejecutar un archivo ejecutable. Permiten que los usuarios pasen parámetros al ejecutable, lo que permite que se comporte de diferentes maneras.

En Oracle EBS R2 existen varios tipos de programas concurrentes que puedes crear y ejecutar, los cuales son:

- Oracle Reports: esta opción es utilizada para ejecutar reportes desarrollados utilizando Reports Builder.
- Procedimientos almacenados PL/SQ: esta opción es utilizada para ejecutar procedimientos almacenados en la Base de Datos.
- Host script: esta opción es utilizada comúnmente para ejecutar secuencias de comandos Shell, es comúnmente utilizada para realizar acciones sobre el sistema operativo tales como copiar o mover un archivo, etc.
- Procedimientos almacenados Java: como su nombre lo indica se trata de archivos ejecutables escritos en el lenguaje de programación Java y almacenados en la base de datos, utilizar procedimientos almacenados siempre será la mejor manera de realizar operaciones dentro del motor, pero hay ciertas actividades que con Java se las realizaría mejor, especialmente cuando se trata de interactuar con el sistema operativo o con una base de datos de terceros.
- SQL *Plus: Se utiliza para ejecutar una secuencia de comandos SQL *Plus o bloque anónimo, se obtendrá el mismo resultado como si se lo estaría ejecutando en la herramienta SQL Plus de Oracle.

- Programa concurrente Perl: Se utiliza para ejecutar programas escritos en CGI Perl. CGI significa Common Gateway Interface y Perl es el más común lenguaje para escribir scripts CGI. Se utiliza para scripting, programación web interfaces / desarrollo, y es ideal para el análisis [Creando Programas Concurrentes en EBS R12, Aplicación Personalizada].

Funcional

Dentro de Alpura un Funcional se encarga del análisis de los procesos internos para su mejora y mantenimiento. Si el usuario tiene alguna necesidad que no se cubre por la EBS de Oracle el funcional realiza el análisis de la misma, realiza la documentación necesaria (MD050) para pasarla al departamento de desarrollo y comience su implementación.

El funcional trabaja muy de cerca con los usuarios, está pendiente de las necesidades e incidencias que puedan presentarse en cada una de las actividades diarias y de acuerdo a esa labor y análisis genera propuestas y soluciones de mejoras o nuevos proyectos para su posterior implementación.

SOA (Arquitectura orientada a servicios)

Por sus siglas en inglés, service oriented architecture, es un paradigma de arquitectura para diseñar y desarrollar sistemas distribuidos. Las soluciones SOA han sido creadas para satisfacer los objetivos de negocio las cuales incluyen facilidad y flexibilidad de integración con sistemas legados, alineación directa a los procesos de negocio reduciendo costos de implementación, innovación de servicios a clientes y una adaptación ágil ante cambios incluyendo reacción temprana ante la competitividad.

Permite la creación de sistemas de información altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios (comúnmente pero no exclusivamente servicios web), lo cual facilita la interacción entre diferentes sistemas propios o de terceros [Compartiendo Tips de Oracle].

BPEL SOA (Lenguaje de ejecución de proceso de negocios)

Por sus siglas en inglés, business process execution language, es un lenguaje basado en XML que permite a los Servicios Web interconectar y compartir datos en una Arquitectura Orientada a Servicios (SOA).

En Oracle su implementación se realiza mediante la IDE de desarrollo JDeveloper [BPEL, Business Process Execution Language].

MD050

Es un documento funcional que se escribe para dar requerimientos al desarrollador de acuerdo a las necesidades del usuario. Contiene todas las especificaciones que la implementación deberá considerar.

Se trabaja de acuerdo a las políticas de Oracle. Estos documentos incluyen una descripción general del proceso, un diagrama de flujo, el detalle de cada paso, la forma en que el desarrollador debe obtener la información requerida y el resultado que se espera.

Cuando durante el proceso de implementación o ya operando el desarrollo surgen incidencias o cambios que deben aplicarse, el funcional genera un Anexo, mismo que tiene las mismas características del documento.

Los MD050 deben pasar varios filtros una vez creados, y ya listos son firmados por todas las áreas implicadas y responsables.

Requisición de Material

Es una petición de material de acuerdo a las necesidades de la empresa. Para realizar la requisición en la EBS de debe ingresar al módulo de compras y generarla por medio de la aplicación. En el presente desarrollo la requisición se genera por medio del desarrollo, invocando los programas estándar.

En Oracle es un documento electrónico que se utiliza para realizar un pedido de un bien o servicio que contiene o puede contener algunas de las siguientes características: lugar y fecha, nombre del solicitante, departamento del solicitante, centro de costos, nombre del artículo, código o número de artículo, cantidad solicitada, proveedor, fecha de entrega, entre otras.

Orden de Compra

Una orden de compra es un documento comercial y oficial emitido por el comprador al proveedor, indicando los tipos, cantidades y precios acordados para los productos o servicios del proveedor proporcionará al comprador.

En la EBS se crea la orden de compra. En particular, para este proyecto se crea a partir de una requisición de material.

Algunas de las líneas que contienen las órdenes de compra son: lugar y fecha de emisión, nombre y número de orden del comprobante, nombre y domicilio, datos de la empresa que imprime el documento, detalles de los productos solicitados, entre otros.

Organización de inventarios

Es un equivalente a secciones dentro de Alpura. Por ejemplo, el área encargada de procesar la leche corresponde a una organización y el área responsable de procesar yogurt corresponde a otra organización. Cada una tiene un código asignado por la aplicación al momento de su creación.

Internamente se relacionan entre sí por diversas actividades, como transporte de material de una organización a otra y viceversa. Cada una cumple con una función en específico.

Existen organizaciones para diversas tareas, como la que se encarga de la logística del transporte y otra que administra pagos y préstamos a los socios.

Juego de Valores

Como su nombre lo indica es una lista de valores. Oracle usa el valor establecido en la lista para validar los datos y asegurarse de que los datos correctos se introduzcan en los campos.

En la EBS se pueden escoger los siguientes tipos de lista: lista desplegable, lista de valores y lista de valores.

Los juegos de valores se llenan de acuerdo a la necesidad del usuario y del proceso. Algunas ocasiones se utilizan para colocar constantes. Si en alguna ocasión se debe realizar un cambio en dichos valores el desarrollador no tendría que modificar su código, el usuario o el funcional encargado del juego de valores hace la modificación en él y es transparente el proceso [Juego de valores].

6. Descripción del trabajo

La visión general del proceso se describe a continuación:

Un usuario de Alpura, específicamente del departamento de compras, ingresa al portal de Tetra Pak para generar una requisición de material necesario. A continuación Tetra Pak envía la información de la misma a Alpura en forma de EDIFACT, la cual es interpretada y almacenada en una tabla de interface para una previa validación de los datos; en seguida se genera la Requisición dentro del módulo de Compras de la EBS de Oracle. De igual forma se notifica por medio de correo electrónico a los usuarios involucrados en el proceso, puesto que el siguiente paso es, en base a la Requisición generada crear una Orden de Compra.

La creación de la Orden de Compra la realiza el usuario también por medio de la EBS. Una vez que este proceso se completa, se detecta que se generó una Orden de Compra para Tetra Pak y se deposita en la tabla de interface donde es leída, mapeada en forma de EDIFACT y enviada a Tetra Pak.

Para confirmar que Tetra Pak recibió la información de forma exitosa responde con otro EDIFACT, el cual también se interpreta, se deposita en la tabla de interface y se notifica por medio de correo electrónico a los involucrados.

Una vez que los pasos anteriores son completados, Tetra Pak, en un determinado tiempo envía otro EDIFACT, en esta ocasión con la información del material que va a enviar a la planta de Alpura. El mensaje también es interpretado, guardado en la tabla de interface y se notifica por correo electrónico a los involucrados.

Para este momento, se queda en espera de que el material arribe a la planta de Alpura; una vez que esto ocurre se recibe por medio de la EBS para registrar el dato en el sistema, donde se cierra el ciclo de este proceso.

La interpretación y el mapeo de los EDIFACT están delegados a otra área de la dirección de sistemas, y se realiza por medio de Delta, software que permite la hacer todo el mapeo de estas operaciones.

De acuerdo al proceso descrito, el desarrollo se puede dividir en cuatro fases, y a cada uno le corresponde un MD050:

1. **Recepción de requisiciones:** comienza cuando por medio de Delta se inserta en la tabla de interface la información que el usuario de Alpura ingresó como requisición de material. Esta se identifica como un tipo de transacción “ORDERS” y un tipo de documento “RECEPCION”.
2. **Envío de orden de compra:** una vez que el usuario de Alpura, basado en la requisición anterior, genera una orden de compra, el desarrollo coloca en la tabla de interface la información de la misma. El tipo de mensaje guardado es “ORDERS” con un tipo de documento “ENTREGA”.
3. **Confirmación de orden de compra:** a través de Delta se recibe el mensaje corroborando que Tetra Pak recibió la información de la orden de compra. Aquí se inserta en la tabla de interface un tipo de transacción “ORDRSP”.
4. **Notificación de entrega:** una vez que Tetra Pak tiene el material listo envía un nuevo mensaje, Delta lo mapea a la tabla de interface con un tipo de transacción “DESADV”. La información queda en sistema lista para cuando llegue de manera física el material y poder realizar la recepción. La Figura 2 muestra el proceso completo.

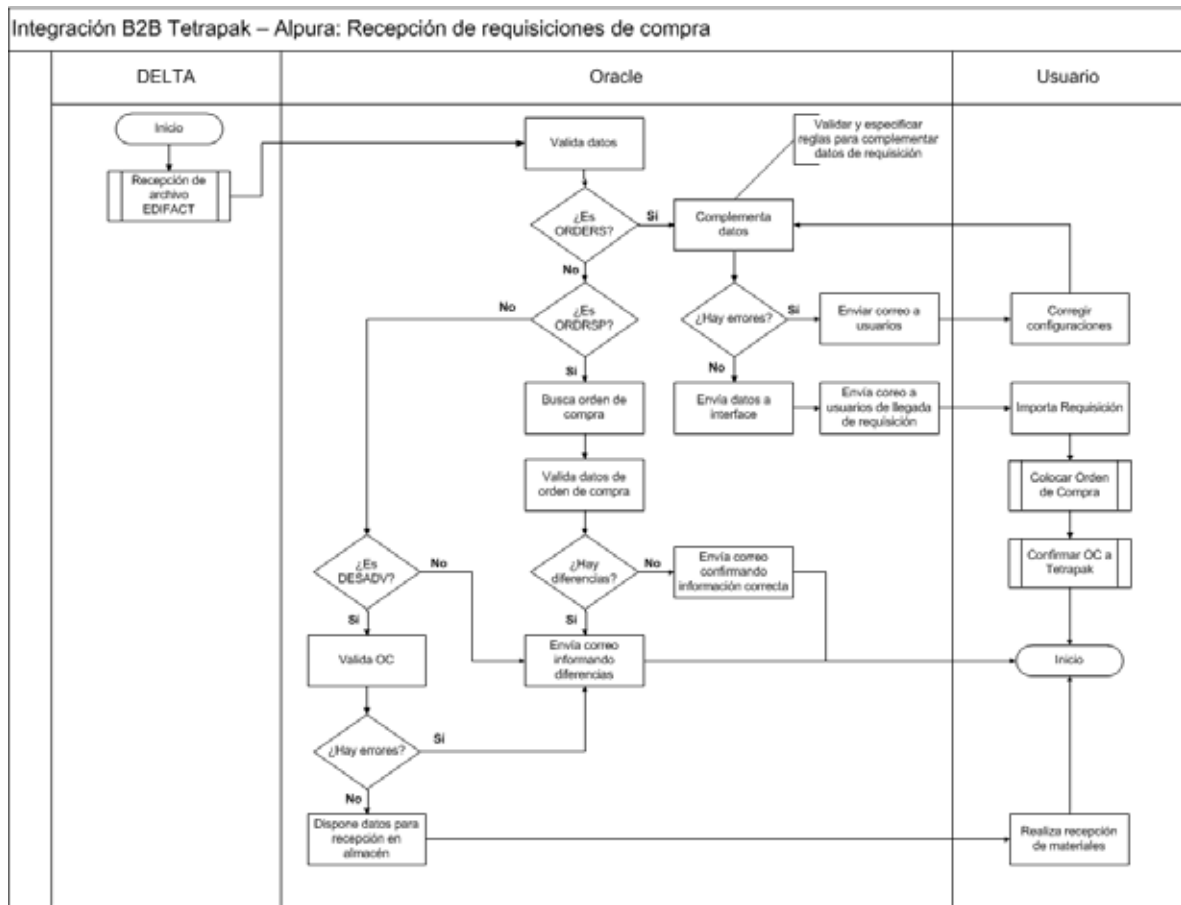


Figura 2. Proceso general del desarrollo B2B Tetra Pak - Alpura

En las siguientes páginas se detalla cada una de las fases del proyecto.

6.1 Recepción de requisiciones

Una vez que el usuario procesa en el portal de Tetra Pak la requisición de materiales, Tetra Pak envía la misma en forma de EDIFACT, mismo que es interpretado por Delta y dejado en la tabla de interface (ver sección 6.1.1 Tabla de interface para más detalle) como tipo de transacción “ORDERS”.

En esta parte empieza a actuar el desarrollo; se creó un disparador para detectar cuando se inserta un tipo de transacción “ORDERS”. En él empiezan las validaciones de los datos ingresados. La información que envía Tetra Pak no es suficiente para generar una requisición en el módulo de compras de la EBS de Oracle, por lo que el resto de los registros se deben llenar de acuerdo a otra tabla más, la tabla de datos derivados (ver sección 6.1.2 Tabla de datos derivados). Esta tabla es administrada por un usuario a través de una pantalla creada con Oracle Forms Builder.

Los datos derivados dependen de dos campos que vienen en el mensaje de Tetra Pak, el RFC (Campo RFC en la tabla de interface) de Alpura y el código de la Organización de Alpura que solicitó el material (Campo DELIVER_TO_LOCATION_ID). A partir de ello se realiza una consulta a la tabla de datos derivados, obteniendo el resto de la información requerida. En el

mensaje de Tetra Pak también son obligatorios datos como la cantidad de material (QUANTITY), el identificador del lote solicitado (BATCH_ID), el número de pedido interno de Tetra Pak (HEADER_ATTRIBUTE8), el número de línea de la requisición (HEADER_ATTRIBUTE9, esto porque pueden ser varias líneas las que conforman la requisición), el código del artículo solicitado (ITEM_SEGMENT1), la unidad de medida del artículo, entre otros. Si la derivación de datos no se completó de forma exitosa o alguno de los datos obligatorios es nulo la línea es marcada con estatus de error (“ERROR EN VALORES DERIVADOS O CALCULADOS”); por el contrario, si las validaciones son correctas y los datos se derivaron con éxito la línea es marcada con un mensaje de aprobación (“VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE”). El detalle del disparador se encuentra en la sección

Se creó un procedimiento programado de base de datos (DBMS Scheduler) que se encarga de correr cada dos minutos invocando un procedimiento almacenado dentro de un paquete que tiene como función detectar líneas en la tabla de interface con el mensaje de éxito o de error de la validación de recepción de requisiciones.

Si el procedimiento detecta que existieron errores en las validaciones se encarga de notificar a los usuarios involucrados en el proceso por medio de un correo electrónico. Si detecta que todo salió como se esperaba marca las líneas para el proceso de importación a la EBS de Oracle (“IMPORT REQUISITION SOA”). Este siguiente paso se desarrolló por medio de BPEL SOA (ver sección 6.1.5 Importación de requisiciones mediante BPEL SOA para más detalle), donde se leen todas las líneas con dicho estatus y se llena una tabla de interface de la aplicación. Después se procede a mandar llamar un concurrente de la EBS para generar la requisición. Esto último es la forma en que debe realizarse una importación de requisición dentro de la EBS.

Una vez que se manda llamar con éxito el concurrente, se procede a verificar el estatus de finalización del mismo. Existen dos opciones, que termine con éxito o con error. Un ejemplo de error podría ser porque uno de los artículos que se requiere importar no pertenece a la organización de inventarios a la que se intenta agregar o no está configurado. En ambos casos, éxito o error, se notifica por medio de correo electrónico a todos los involucrados. El proceso en general se ilustra en la Figura 3.

Integración B2B Tetra pak – Alपुरa: Recepción de requisiciones de compra

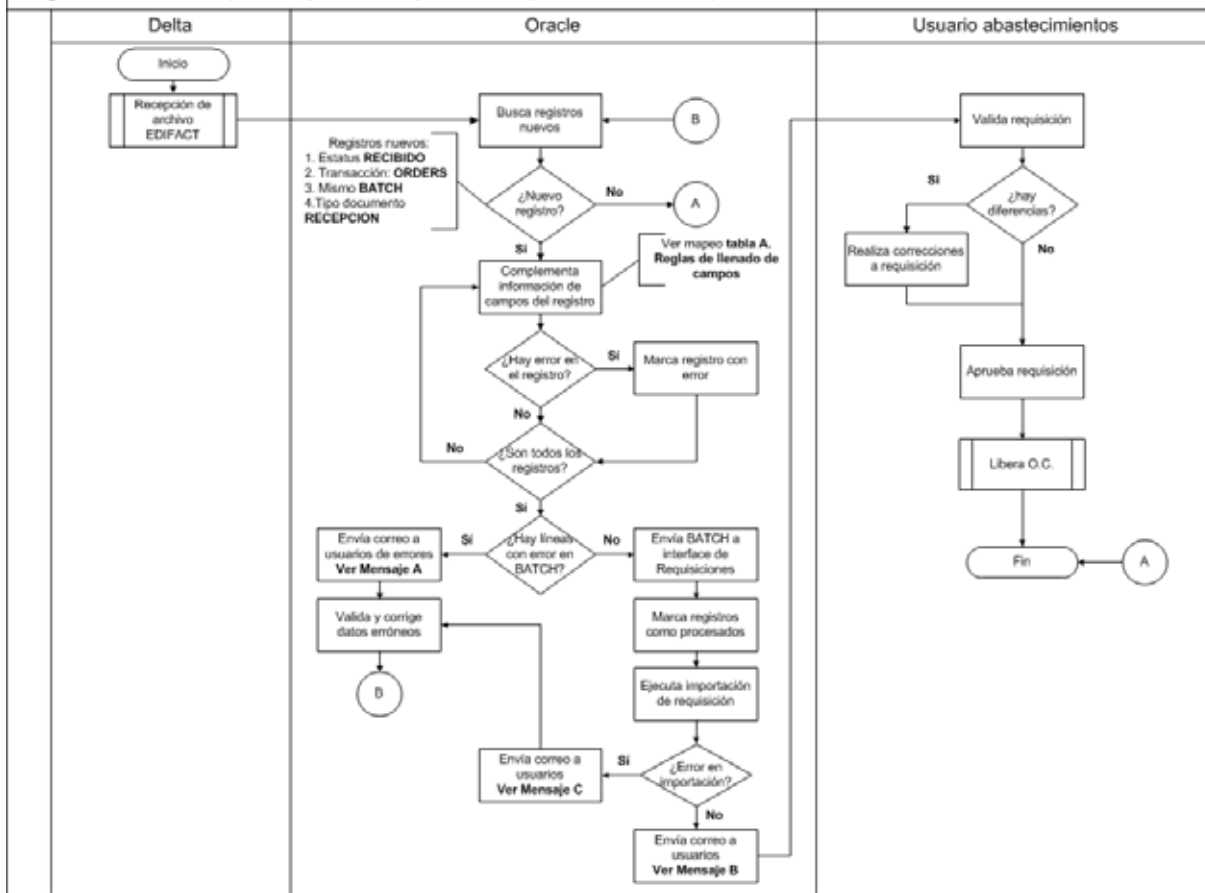


Figura 3. Proceso de Recepción de Requisiciones

A continuación se detalla cada una de las partes del proceso de Recepción de Requisiciones.

6.1.1 Tabla de interface

Descripción de la tabla de interface

Nombre: ALP_TMP_REQUISITIONS_INTERFACE

NOMBRE DE LA COLUMNA	DESCRIPCIÓN DEL CAMPO	TIPO DE DATO
INTERFACE_ID	Campo llave. Identificador único.	NUMBER
TRANSACCION	Campo donde se indica que tipo de documento se entrega. Puede ser ORDERS, DESADV o ORDRSP	VARCHAR2 (25)
ORDEN_COMPRA	Numero de orden de compra	VARCHAR2 (25)
RFC	RFC de Alpura	VARCHAR2 (15)
INTERFACE_SOURCE_CODE	Código del origen de la interface de datos	VARCHAR2 (30)
SOURCE_TYPE_CODE	Constante definida en configuración (VENDOR)	VARCHAR2 (25)
REQUISITION_TYPE	Constante definida en configuración (PURCHASE)	VARCHAR2 (25)
DESTINATION_TYPE_CODE	Constante definida en configuración (INVENTORY)	VARCHAR2 (25)
QUANTITY	Cantidad de piezas solicitadas	NUMBER
UNIT_PRICE	Precio unitario	NUMBER
AUTHORIZATION_STATUS	Estado de la autorización del registro. Constante definida en configuración (INCOMPLETE)	VARCHAR2 (25)
BATCH_ID	Numero de Batch de documento	NUMBER
NOTE_TO_APPROVER	Notas para el aprobador de orden de compra	VARCHAR2 (100)
PREPARER_ID	ID del usuario que genera la requisición	NUMBER
HEADER_DESCRIPTION	Descripción de la cabecera de la requisición. Cadena: B2B Requisition Tetra pak dd-mmm-yyyy	VARCHAR2 (100)
HEADER_ATTRIBUTE_CATEGORY	Constante definida en configuración (B2B Requisition)	VARCHAR2 (100)
HEADER_ATTRIBUTE8	Número de pedido dentro del punch-out de Tetra pak	VARCHAR2 (100)
HEADER_ATTRIBUTE9	Número de línea de la requisición	VARCHAR2 (100)
NOTE_TO_BUYER	Notas para el comprador	VARCHAR2 (100)
ITEM_SEGMENT1	Código del artículo requerido	VARCHAR2 (40)
CHARGE_ACCOUNT_ID	ID de cuenta contable de cargo	NUMBER
UOM_CODE	Código de la Unidad de medida del artículo	VARCHAR2 (5)
REFERENCE_NUM	ID del número de coproducción - CP-nn	VARCHAR2 (25)
DESTINATION_ORGANIZATION_ID	ID de la organización de inventarios de entrega	NUMBER
DELIVER_TO_LOCATION_ID	ID de la dirección de entrega	NUMBER
DELIVER_TO_REQUESTOR_ID	ID del usuario que requirió el material	NUMBER
DELIVERY_NUMBER	ID del proveedor propuesto	VARCHAR2 (25)
LOTE	Lote del material entregado	VARCHAR2 (25)
SUGGESTED_VENDOR_ID	ID del proveedor propuesto	NUMBER
SUGGESTED_VENDOR_SITE_ID	ID del site del proveedor propuesto	NUMBER
NEED_BY_DATE	Fecha de necesidad del material requerido	DATE
DOCUMENT_DATE	Fecha de documento que se está recibiendo	VARCHAR2 (25)
CURRENCY_CODE	Moneda de cotización	VARCHAR2 (10)
ORG_ID	ID de la entidad legal	NUMBER
STATUS	Estado del registro	VARCHAR2 (1000)
ERROR_DESCRIPTION	Descripción del error generado	VARCHAR2 (1000)
TIPO_DOCUMENTO	Determina si es un documento de ENTREGA ó RECEPCION	VARCHAR2 (25)
USER_PREPARER	Username del usuario de Alpura	VARCHAR2 (150)
GLN	GLN del proveedor	VARCHAR2 (15)
ES_ULTIMA_ENTREGA	Bandera de especificación de última entrega	VARCHAR2 (12)
SSCC	Número identificador de Tetra Pak	VARCHAR2 (18)
CREATED_BY	ID de usuario que crea el registro	NUMBER
CREATION_DATE	Fecha/hora de creación del documento	DATE
LAST_UPDATED_BY	ID Usuario que hizo última actualización	NUMBER
LAST_UPDATE_DATE	Ultima fecha/hora actualización	DATE
LIN_SELECCIONADA	Marca de recepción	VARCHAR2 (2)
QUANTITY2	Cantidad recibida de material	NUMBER

Tabla 1. Tabla de interface

6.1.2 Tabla de datos derivados

Tabla de datos derivados

Tabla: ALP_TETRA_DATOS_DERIVADOS

NOMBRE DE LA COLUMNA	DESCRIPCIÓN DEL CAMPO	TIPO DE DATO
DATO_ID	Campo llave. Identificador único.	NUMBER
RFC	RFC de Alpura	VARCHAR2 (15)
DELIVER_TO_LOCATION_ID	ID de la dirección de entrega	NUMBER
ORG_ID	ID de la entidad legal	NUMBER
INTERFACE_SOURCE_CODE	Código del origen de la interface de datos	VARCHAR2 (30)
DELIVER_TO_REQUESTOR_ID	ID del usuario que requirió el material	NUMBER
PREPARER_ID	ID del usuario que genera la requisición	NUMBER
DESTINATION_ORGANIZATION_ID	ID de la organización de inventarios de entrega	NUMBER
SOURCE_TYPE_CODE	Constante definida en configuración (VENDOR)	VARCHAR2 (25)
REQUISITION_TYPE	Constante definida en configuración (PURCHASE)	VARCHAR2 (25)
DESTINATION_TYPE_CODE	Constante definida en configuración (INVENTORY)	VARCHAR2 (25)
AUTHORIZATION_STATUS	Estado de la autorización del registro. Constante definida en configuración (INCOMPLETE)	VARCHAR2 (25)
NOTE_TO_APPROVER	Notas para el aprobador de orden de compra	VARCHAR2 (100)
HEADER_DESCRIPTION	Descripción de la cabecera de la requisición. Cadena: B2B Requisition Tetra pak dd-mmm-yyyy	VARCHAR2 (100)
NOTE_TO_BUYER	Notas para el comprador	VARCHAR2 (100)
CHARGE_ACCOUNT_ID	ID de cuenta contable de cargo	NUMBER
CURRENCY_CODE	Moneda de cotización	VARCHAR2 (10)
HEADER_ATTRIBUTE_CATEGORY	Constante definida en configuración (B2B Requisition)	VARCHAR2 (100)
SUGGESTED_VENDOR_SITE_ID	ID del proveedor propuesto	NUMBER
GLN	GLN del proveedor	VARCHAR2 (15)
TIPO_DOCUMENTO	Determina si es un documento de ENTREGA ó RECEPCION	VARCHAR2 (25)
ACUERDO_COMPRA	Constante.	NUMBER
VERSION	Constante.	NUMBER
CREATED_BY	ID de usuario que crea el registro	NUMBER
CREATION_DATE	Fecha/hora de creación del documento	DATE
LAST_UPDATED_BY	ID Usuario que hizo última actualización	NUMBER
LAST_UPDATE_DATE	Ultima fecha/hora actualización	DATE

Tabla 2. Tabla de datos derivados

La tabla es administrada por el usuario. En caso de que por cuestión operativa existiera algún tipo de cambio el usuario podría realizarlo.

La forma de administrar la pantalla es por medio de una pantalla de Oracle Forms. Se creó específicamente con ese fin. El resultado final de la pantalla se muestra en la Figura 4.

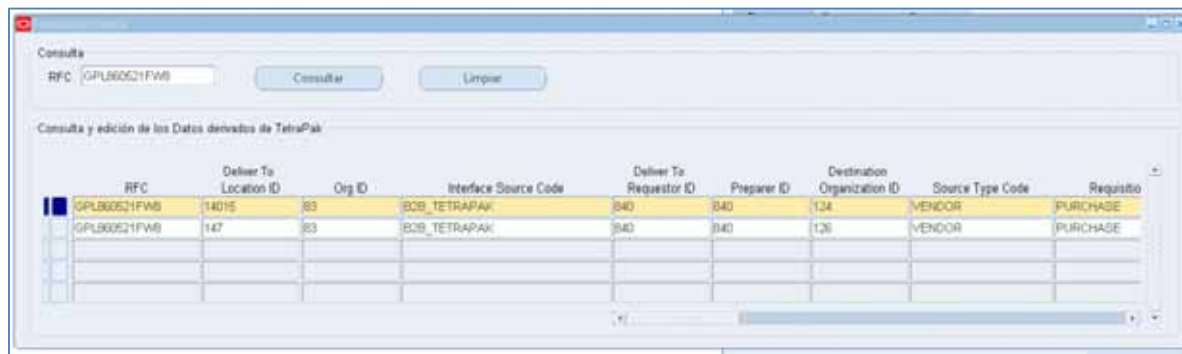


Figura 4. Pantalla de Oracle Forms para administración de la tabla de datos derivados

6.1.3 Disparador

La especificación del disparador para recepción de requisiciones se muestra a continuación.

Creación del disparador y declaración de variables que se utilizarán:

```
create or replace
TRIGGER "APPS"."ALP_TETRA_TMP_INTERFACE" BEFORE INSERT ON
ALP_DELTA.ALP_TMP_REQUISITIONS_INTERFACE
FOR EACH ROW
DECLARE

--Declaramos variables
v_seq                NUMBER;
v_interface_source_code  VARCHAR2 (15) ;
v_source_type_code     VARCHAR2 (15) ;
v_batch_id            NUMBER ;
v_destination_type_code VARCHAR2 (15) ;
v_authorization_status VARCHAR2 (25) ;
v_requisition_type     VARCHAR2 (20) ;
v_preparer_id         NUMBER ;
v_charge_account_id    NUMBER ;
v_destination_organization_id NUMBER ;
v_deliver_to_requestor_id NUMBER ;
v_deliver_to_location_id NUMBER ;
v_note_to_buyer       VARCHAR2 (50) ;
v_need_by_date        DATE ;
v_suggested_vendor_id NUMBER ;
v_suggested_vendor_site_id NUMBER ;
v_header_description  VARCHAR2 (50) ;
v_currency_code       VARCHAR2 (8) ;
v_note_to_approver    VARCHAR2 (50) ;
v_rfc                 VARCHAR2 (15) ;
v_org_id              NUMBER ;
v_created_by          NUMBER ;
v_header_attribute_category VARCHAR2 (100) ;
v_document_date       DATE ;
v_tipo_documento      VARCHAR2 (100) ;

v_error_flag          VARCHAR2 (10) ;
v_cadena_error        VARCHAR2 (5000) ;

v_valida_articulo     NUMBER ;
v_valida_cuenta       NUMBER ;
v_valida_uom_code     NUMBER ;
v_valida_requisitante NUMBER ;
```

Se genera una secuencia para la llave primaria, se llenan datos de auditoría como la fecha de creación y el usuario de creación. Se asignan a la tabla.

```
BEGIN
--Generamos un nuevo número de secuencia
BEGIN
SELECT seq_tmp_req_id.NEXTVAL
INTO v_seq
FROM dual;

v_created_by := fnd_profile.value('USER_ID');
IF v_created_by IS NULL THEN
:NEW.CREATED_BY := 0;
:NEW.LAST_UPDATED_BY := 0;
ELSE
:NEW.CREATED_BY := v_created_by;
:NEW.LAST_UPDATED_BY := v_created_by;
END IF;
:NEW.CREATION_DATE := SYSDATE;
:NEW.LAST_UPDATE_DATE := SYSDATE;
EXCEPTION WHEN OTHERS THEN
v_error_flag := 'ERROR';
v_cadena_error := 'v_seq: ' || SQLERRM;
END;
:NEW.INTERFACE_ID := v_seq;
```

Si el tipo de transacción que se insertó es “ORDERS”, “ORDRSP” y “DESDAV” continúa. En caso contrario llena el campo de la STATUS y ERROR_DESCRIPTION con leyendas de que los tipos de transacción no existen; es decir, una validación global. Ahora empiezan las validaciones y asignaciones puntuales del tipo de transacción ORDERS. Para que el disparador continúe el campo de TIPO_DOCUMENTO debe venir en nulo. Una vez pasados esos filtros se obtienen los valores derivados, basados en el RFC y el DELIVER_TO_LOCATION_ID (Id de la organización a donde se va a entregar el material). Los guardamos en variables temporales para después mapearlas.

```

--Validamos que sea un valor permitido para transacción
IF TRIM(:NEW.TRANSACCION) IN('ORDERS', 'ORDRSP', 'DESDAV') THEN
-----
IF :NEW.TRANSACCION = 'ORDERS' AND :NEW.TIPO_DOCUMENTO IS NULL THEN

    v_error_flag      := NULL;

    --Obtenemos los valores derivados
    -----
BEGIN
    SELECT  org_id
           , interface_source_code
           , deliver_to_requestor_id
           , preparer_id
           , destination_organization_id
           , source_type_code
           , requisition_type
           , destination_type_code
           , authorization_status
           , note_to_approver
           , header_description
           , note_to_buyer
           , charge_account_id
           , currency_code
           , header_attribute_category
           , suggested_vendor_site_id
           , tipo_documento
    INTO    v_org_id
           , v_interface_source_code
           , v_deliver_to_requestor_id
           , v_preparer_id
           , v_destination_organization_id
           , v_source_type_code
           , v_requisition_type
           , v_destination_type_code
           , v_authorization_status
           , v_note_to_approver
           , v_header_description
           , v_note_to_buyer
           , v_charge_account_id
           , v_currency_code
           , v_header_attribute_category
           , v_suggested_vendor_site_id
           , v_tipo_documento
    FROM    alp_tetra_datos_derivados
    WHERE   rfc = :NEW.rfc
    AND    deliver_to_location_id = :NEW.deliver_to_location_id;
EXCEPTION WHEN OTHERS THEN
    v_error_flag      := 'ERROR';
    v_cadena_error    := v_cadena_error || ' || VALORES_DERIVADOS: ' || SQLERRM;
END;

```

Como podemos observar, gran parte de los datos se derivan. Todos ellos son necesarios para generar una requisición por medio de la EBS de Oracle.

Si existiera algún error, como por ejemplo que el RFC no coincida, o que el DELIVER_TO_LOCATION_ID venga nulo o alguna de otro tipo generará una excepción, misma

que es atrapada y se guarda en la variable de error V_ERROR_FLAG, y la descripción del error y de posibles errores posteriores se van concatenando en la variable V_CADENA_ERROR.

El siguiente paso es la asignación de datos, estos que han sido obtenidos deben guardarse en la tabla de interface.

```

--Asignamos los valores a la tabla
BEGIN
:NEW.INTERFACE_SOURCE_CODE      := v_interface_source_code;
:NEW.SOURCE_TYPE_CODE          := v_source_type_code;
:NEW.REQUISITION_TYPE          := v_requisition_type;
:NEW.DESTINATION_TYPE_CODE     := v_destination_type_code;
:NEW.AUTHORIZATION_STATUS      := v_authorization_status;
:NEW.NOTE_TO_APPROVER          := v_note_to_approver || ' ' ||
TO_CHAR(SYSDATE, 'DD-MON-YYYY');
:NEW.PREPARER_ID               := v_preparer_id;
:NEW.HEADER_DESCRIPTION        := v_header_description || ' ' ||
TO_CHAR(SYSDATE, 'DD-MON-YYYY');
:NEW.HEADER_ATTRIBUTE_CATEGORY := v_header_attribute_category;
:NEW.NOTE_TO_BUYER             := v_note_to_buyer || ' ' || TO_CHAR(SYSDATE,
'DD-MON-YYYY');
:NEW.CHARGE_ACCOUNT_ID         := v_charge_account_id;
:NEW.DESTINATION_ORGANIZATION_ID := v_destination_organization_id;
:NEW.DELIVER_TO_REQUESTOR_ID   := v_deliver_to_requestor_id;
:NEW.SUGGESTED_VENDOR_SITE_ID := v_suggested_vendor_site_id;
:NEW.CURRENCY_CODE             := v_currency_code;
:NEW.ORG_ID                    := v_org_id;
:NEW.TIPO_DOCUMENTO           := v_tipo_documento;
:NEW.UOM_CODE                  := LOWER(:NEW.UOM_CODE);
:NEW.BATCH_ID                  := SUBSTR(:NEW.HEADER_ATTRIBUTE8,
(LENGTH(:NEW.HEADER_ATTRIBUTE8) - 7), LENGTH(:NEW.HEADER_ATTRIBUTE8));

EXCEPTION WHEN OTHERS THEN
v_error_flag := 'ERROR';
v_cadena_error := v_cadena_error || ' ' || ASIGNACION_VALORES: ' ' || SQLERRM;
END;

```

Nuevamente, si existe un error al momento de tratar de hacer la asignación se concatena la descripción del error.

A continuación se realizan dos validaciones, la cantidad de material no puede ser menor a cero; también se requiere que algunos de los valores que manda Tetra Pak en el EDIFACT sean obligatorios.

```

--Validamos que la cantidad sea positiva
IF :NEW.QUANTITY < 0 THEN
v_error_flag := 'ERROR';
v_cadena_error := v_cadena_error || ' ' || QUANTITY: ' ' || 'La cantidad es un
valor menor a 0';
END IF;

--Validamos que los valores que están insertando no sean nulos
IF :NEW.TRANSACCION          IS NULL OR
:NEW.RFC                    IS NULL OR
:NEW.QUANTITY                IS NULL OR
:NEW.BATCH_ID                IS NULL OR
:NEW.HEADER_ATTRIBUTE8      IS NULL OR
:NEW.HEADER_ATTRIBUTE9      IS NULL OR
:NEW.ITEM_SEGMENT1          IS NULL OR
:NEW.UOM_CODE                IS NULL OR
:NEW.DELIVER_TO_LOCATION_ID IS NULL OR
:NEW.SUGGESTED_VENDOR_ID    IS NULL OR
--:NEW.ORG_ID                IS NULL OR
:NEW.GLN                     IS NULL OR
:NEW.USER_PREPARER           IS NULL OR
:NEW.DOCUMENT_DATE          IS NULL OR
:NEW.NEED_BY_DATE           IS NULL
THEN
v_error_flag := 'ERROR';
v_cadena_error := v_cadena_error || ' ' || 'VALORES OBLIGATORIOS NULOS';
END IF;

```

Ahora, en caso de que las condiciones no se cumplan se concatena el error.

Ahora se realizan tres validaciones más. El código de artículo que viene contenido debe existir en la organización a la que se va a mapear la requisición, la unidad de medida del mismo debe coincidir y la cuenta contable a la que se va a realizar el cargo de la generación de la orden de compra debe existir y estar activa.

```
--Validamos que el artículo exista en la organización
BEGIN
  SELECT  inventory_item_id
  INTO    v_valida_articulo
  FROM    apps.mtl_system_items_b
  WHERE   segment1 = :NEW.ITEM_SEGMENT1
  AND     organization_id = v_destination_organization_id;
EXCEPTION WHEN OTHERS THEN
  v_error_flag := 'ERROR';
  v_cadena_error := v_cadena_error || ' || ' || 'EL ARTICULO NO EXISTE PARA LA
ORGANIZACION';
END;

--Validamos la unidad de medida
BEGIN
  SELECT  inventory_item_id
  INTO    v_valida_uom_code
  FROM    apps.mtl_system_items_b
  WHERE   segment1 = :NEW.ITEM_SEGMENT1
  AND     organization_id = v_destination_organization_id
  AND     UPPER(primary_uom_code) = UPPER(:NEW.UOM_CODE);
EXCEPTION WHEN OTHERS THEN
  v_error_flag := 'ERROR';
  v_cadena_error := v_cadena_error || ' || ' || 'LA UNIDAD DE MEDIDA NO
CORRESPONDE CON EL ARTICULO';
END;

--Validamos la cuenta contable
BEGIN
  SELECT  code_combination_id
  INTO    v_valida_cuenta
  FROM    apps.gl_code_combinations
  WHERE   code_combination_id = v_charge_account_id
  AND     enabled_flag = 'Y';
EXCEPTION WHEN OTHERS THEN
  v_error_flag := 'ERROR';
  v_cadena_error := v_cadena_error || ' || ' || 'LA CUENTA CONTABLE NO EXISTE O
NO ESTÁ ACTIVA';
END;
```

Como siguiente paso se debe validar que los campos de PREPARER_ID y DELIVER_TO_REQUESTOR_ID (son datos que envía Tetra Pak con referencia al usuario que realizó la transacción en su portal) se encuentren configurados en un juego de valores. Esto para validar y tener el control de que el procedimiento se realiza solamente por los usuarios pertinentes.

Por último se revisa la bandera de error. Si en alguna de todas las validaciones ocurrió algo el estatus se deja como error (“ERROR EN VALORES DERIVADOS O CALCULADOS”), y en el campo de descripción del error se tiene concatenado el detalle, ya sea que ocurrió uno solo o varios. Caso contrario, cuando ninguna de las validaciones generó error o excepción la bandera de error se encuentran en nulo; se procede a llenar estatus valor exitoso (“VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE”). En caso de que ocurra una excepción que no fue cachada por los procesos individuales se envía un email al desarrollador.

```

--Validamos que el preparer_id y deliver_to_requestor_id se encuentre en la lista
de valores
BEGIN
--Preparer
select fvs.flex_value_set_id
INTO v_valida_requisitante
from apps.fnd_flex_value_sets fvs
, apps.fnd_flex_values fv
, apps.fnd_flex_values_tl fvt
where fvs.flex_value_set_id = fv.flex_value_set_id
and fv.FLEX_VALUE_ID = fvt.FLEX_VALUE_ID
and fvt.language = 'ESA'
and fvs.flex_value_set_name = 'ALP_B2B_USERS'
AND fv.enabled_flag = 'Y'
AND flex_value = :NEW.PREPARER_ID;

--Requestor
select fvs.flex_value_set_id
INTO v_valida_requisitante
from apps.fnd_flex_value_sets fvs
, apps.fnd_flex_values fv
, apps.fnd_flex_values_tl fvt
where fvs.flex_value_set_id = fv.flex_value_set_id
and fv.FLEX_VALUE_ID = fvt.FLEX_VALUE_ID
and fvt.language = 'ESA'
and fvs.flex_value_set_name = 'ALP_B2B_USERS'
AND fv.enabled_flag = 'Y'
AND flex_value = :NEW.DELIVER_TO_REQUESTOR_ID;

EXCEPTION WHEN OTHERS THEN
v_error_flag := 'ERROR';
v_cadena_error := v_cadena_error || ' ' || ' ' || 'EL PREPARER_ID O EL
DELIVER_TO_REQUESTOR_ID NO SE ENCUENTRA EN EL JUEGO DE VALORES CORRESPONDIENTE';
END;

--Si existió error en el procedimiento lo guardamos
IF v_error_flag = 'ERROR' THEN
:NEW.STATUS := 'ERROR EN VALORES DERIVADOS O CALCULADOS';
:NEW.ERROR_DESCRIPTION := v_cadena_error;
--Si no existió error alguno marcamos el estatus como exitoso
ELSIF v_error_flag IS NULL THEN
:NEW.STATUS := 'VALORES DERIVADOS Y CALCULADOS INSERTADOS
EXITOSAMENTE';
END IF;

ELSE
NULL;
END IF;

--Si TRANSACCION no se encuentra en ninguno de los valores esperados
ELSE
:NEW.STATUS := 'ERROR EN VALORES DERIVADOS O CALCULADOS';
:NEW.ERROR_DESCRIPTION := v_cadena_error || ' ' || 'El valor del campo TRANSACCION no es
válido.';
END IF;
EXCEPTION WHEN OTHERS THEN
alp_custom.alp_test_send_email.
send_mail ('desarrollador@alpura.com',
'aworkflow@alpura.com',
'Exception trigger ALP_INV_TETRA_INTERFACE para tabla
ALP_TMP_REQUISITIONS_INTERFACE (PROD 139)',
SQLERRM,
'172.108.1.74',
25);
END:

```

6.1.4 Validación de líneas después del disparador

Una vez que el diparador deja las líneas marcadas con estatus “ERROR EN VALORES DERIVADOS O CALCULADOS” ó “VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE” pueden ser leídas por el procedimiento de validación de líneas. Este procedimiento es ejecutado cada dos minutos de forma automática mediante una calendarización de la base de datos. El código de dicho proceso se detalla a continuación:

```

BEGIN
DBMS_SCHEDULER.CREATE_JOB (
  job_name      => 'JOB_IMPORTACION_VALIDA_STATUS',
  job_type     => 'PLSQL_BLOCK',
  job_action    => ' BEGIN
                    alp_tetrapack.valida_estatus_lineas;
                    EXCEPTION WHEN OTHERS THEN
                    alp_custom.alp_test_send_email.send_mail
('desarrollador@alpura.com',
                                'aworkflow@alpura.com',
                                'General exception JOB_IMPORTACION_VALIDA_STATUS',
                                SQLERRM,
                                '172.108.1.74',
                                25);
                    END;
                    '
,
  start_date    => TO_DATE('31-10-2014 13:00','DD-MM-YYYY HH24:MI'),
  repeat_interval => 'FREQ=MINUTELY; INTERVAL=2',
  enabled       => TRUE);
END;

```

El script crea el procedimiento calendarizado de tipo “PLSL_BLOCK” que consiste en ejecución de código PL/SQL. Ese código se encuentra en el parámetro “JOB_ACTION”, y simplemente realiza el llamado del procedimiento “VALIDA_ESTATUS_LINEAS” almacenado en un paquete. Si ocurre algún tipo de excepción se notifica por medio de correo electrónico al desarrollador. Al procedimiento calendarizado también se le especifica la fecha en que debe iniciar su funcionamiento y el intervalo de repetición.

El procedimiento “VALIDA_ESTATUS_LINEAS” se encarga de validar líneas y enviar notificaciones. Como se menciona con anterioridad es un procedimiento almacenado.

Se declaran dos cursores, uno para detectar los BATCHS_ID a procesar y otro para obtener el detalle de cada línea. Se declaran también variables auxiliares.

```

PROCEDURE valida_estatus_lineas
IS
  CURSOR batch_ids
  IS
    SELECT DISTINCT batch_id
    FROM   alp_tmp_requisitions_interface
    WHERE  transaccion = 'ORDERS'
    AND    tipo_documento = 'RECIBIDO'
    AND    status
           IN(
              'VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE'
            , 'ERROR EN VALORES DERIVADOS O CALCULADOS'
           );

  CURSOR batch_detail(p_batch_id NUMBER)
  IS
    SELECT *
    FROM   alp_tmp_requisitions_interface
    WHERE  batch_id = p_batch_id
    AND    transaccion = 'ORDERS'
    AND    tipo_documento = 'RECIBIDO'

    AND    status
           IN(
              'VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE'
            , 'ERROR EN VALORES DERIVADOS O CALCULADOS'
           );

  v_valida_batch   VARCHAR2(100);
  v_count          NUMBER;
  v_batch_id      NUMBER;
  v_estatus       VARCHAR2(100);
  v_sysdate_loop  DATE;
  v_wait counter  NUMBER :=0;

```


Por cada uno de los BATCH_ID encontrados para procesar se deben validar los estatus de las líneas. Como primer paso esperamos un minuto. Esto es porque nosotros estamos procesando por BATCH_ID, y debemos asegurarnos que se hayan insertado todos los registros correspondientes al mismo y hayan sido validadas por el disparador. Recorremos en otro ciclo interno todas las líneas del mismo BATCH_ID e identificamos si alguna presenta error. Si es el caso se guarda en la bandera auxiliar V_VALIDA_BATCH el valor de “FALSE”. Si el lote se compone de 20 líneas y una se encuentra en error se debe notificar que no se recibió de forma correcta.

```

BEGIN
--Validamos cada uno de los registros
FOR x IN batch_ids
LOOP
DBMS_LOCK.sleep (60);
v_batch_id := x.batch_id;
--Por cada uno de los registros debemos validar que tengan el status correcto
v_valida_batch := 'TRUE';
FOR y IN batch_detail(v_batch_id)
LOOP
IF y.status = 'ERROR EN VALORES DERIVADOS O CALCULADOS' THEN
v_valida_batch := 'FALSE';
END IF;
END LOOP;
END LOOP;

```

Si la bandera de validación tiene valor “FALSE” marcamos todas las líneas de ese lote con mensaje de error. También mandamos llamar el procedimiento para notificar líneas con error. Una vez que este proceso termina actualizamos nuevamente las líneas con estatus de que ya se mandó el correo de aviso.

```

--De acuerdo al status marcamos las lineas para ser procesadas o no
IF v_valida_batch = 'FALSE' THEN
UPDATE alp_tmp_requisitions_interface
SET status = 'REQUISITION ERROR - ERROR EN VALORES DERIVADOS O
CALCULADOS'
, tipo_documento = 'RECIBIDO / PROCESADA'
WHERE batch_id = x.batch_id
AND status IN ('ERROR EN VALORES DERIVADOS O CALCULADOS',
'VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE')
AND transaccion = 'ORDERS';
COMMIT;
--Aquí necesitamos enviar el error
errores_requisicion
(
x.batch_id
, NULL
, v_estatus
);
--Ahora actualizamos nuevamente el estatus
UPDATE alp_tmp_requisitions_interface
SET status = 'CORREO ENVIADO || REQUISITION ERROR - ERROR EN VALORES DERIVADOS
O CALCULADOS'
, tipo_documento = 'RECIBIDO / PROCESADA'
WHERE batch_id = x.batch_id
AND status IN ('REQUISITION ERROR - ERROR EN VALORES DERIVADOS O
CALCULADOS')
AND transaccion = 'ORDERS';
COMMIT;

```

Si todas las líneas fueron validadas por el disparador sin error se actualiza el estatus para ser importadas a la EBS. Si en alguna parte del proceso ocurre algún tipo de excepción el desarrollador es notificado por medio de correo electrónico.

```

ELSIF v_valida_batch = 'TRUE' THEN
  UPDATE alp_tmp_requisitions_interface
  SET     status      = 'IMPORT REQUISITION SOA'
        , tipo_documento = 'RECIBIDO / PROCESADA'
  WHERE  batch_id = x.batch_id
  AND    status      = 'VALORES DERIVADOS Y CALCULADOS INSERTADOS EXITOSAMENTE'
  AND    transaccion = 'ORDERS';
  COMMIT;
END IF;

END LOOP;

EXCEPTION WHEN OTHERS THEN
  alp_custom.alp_test_send_email.
  send_mail ('danlopez@alpura.com',
            'aworkflow@alpura.com',
            'Exception valida_estatus_lineas en ALP_TETRAPAK (Test 139)',
            SQLERRM,
            '172.108.1.74',
            25);
END;

```

6.1.5 Importación de requisiciones mediante BPEL SOA

El proceso BPEL SOA se encuentra corriendo programada cada 2 minutos. La forma en que funciona el compuesto es por medio de un poll, una opción de SOA para que cada determinado tiempo realice una consulta a alguna tabla (ALP_TMP_REQUISITIONS_INTERFACE) y consulte un campo; en este caso busca el campo ESTATUS, y busca todos aquellos que tengan como contenido “IMPORT REQUISITION SOA”. Si no hay coincidencia no realiza ningún proceso; si por el contrario detecta algún registro que cumpla con ello entra al procedimiento. La primera parte “ReceiveTMPReqInterface” simplemente almacena de forma temporal todos los registros leídos, es un constructor de BPEL tipo Receive. El siguiente paso es copiar todos los registros leídos a la tabla de interface de requisiciones (estándar de la aplicación), esto se está realizando por medio de la función “Transform1”, una extensión de Oracle que facilita dicha operación.

“InvokePOReqInterface” hace la invocación de la tabla de interface por medio de un Servicio BPEL (Adaptador de Base de Datos). El proceso “Transform1” entre los dos elementos requeridos se encarga del mapeo.

“Dehydrate1” es otra extensión de Oracle y simplemente es equivalente a un COMMIT de base de datos.

“InvokePOCount” realiza la invocación de un adaptador de base de datos, que también va a la tabla de interface, pero en esta ocasión para traer solamente los registros con estatus pendiente y que correspondan al proceso de Tetra Pak.

“Assign1” es un constructor de BPEL de tipo Actividad. Internamente realiza el conteo de registros a procesar (Ver Figura 5).

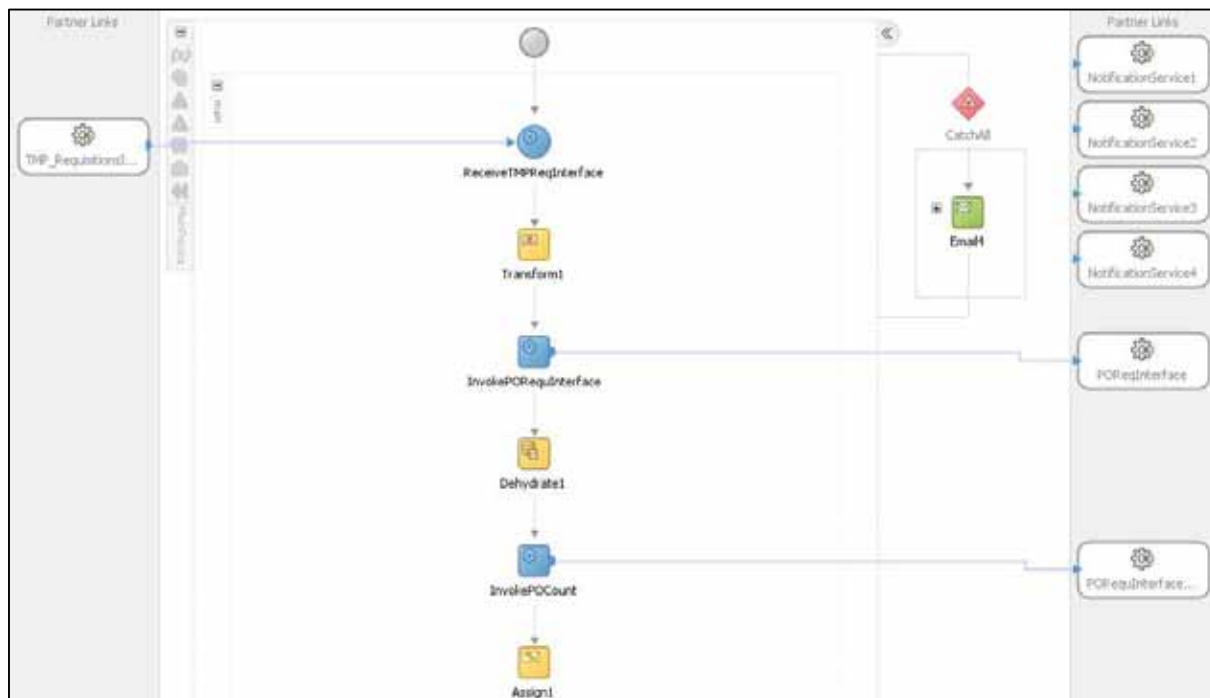


Figura 5. Primera sección del proceso de BPEL SOA

“Email1” es una extensión de Oracle, hace invocación al “NotificationService1”, y simplemente se encarga de enviar un correo electrónico con el número de registros a procesar. Esta opción es para debug.

Posteriormente es un ciclo, donde va a recorrer todos los registros. “InvokeInitialize” es un adaptador de base de datos que manda llamar un procedimiento almacenado en un paquete, que lo único que hace es colocar variables de entorno. Esto es necesario para más adelante poder invocar un concurrente de Oracle puesto para que se genere un entorno como si te encontraras en la EBS.

“InvokeREQUIMPORT” realiza el llamada de un adaptador de base de datos, en esta ocasión apuntando a un concurrente de Oracle. Este es estándar de la aplicación y se encarga de realizar el proceso natural de importación de requisiciones en base a parámetros. “Assign3” se encarga de asignar tales parámetros, entre los que se encuentra el BATCH_ID leído de la tabla de interface de requisiciones. “Email2” también es para debug. “Dehydrate3” realiza otro COMMIT (Ver Figura 6).

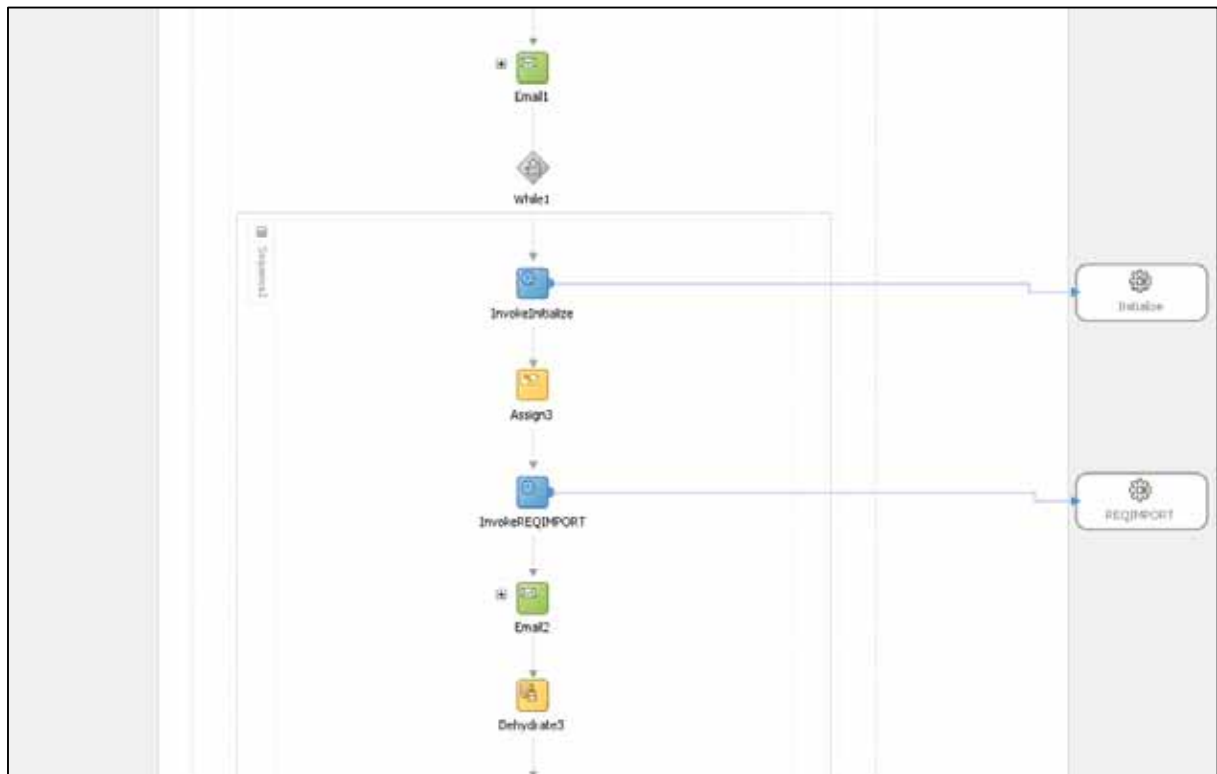


Figura 6. Segunda sección del proceso de BPEL SOA

“InvokeREQUIIMPORT” genera una salida, esta puede ser 0 o un Id. El Id es un número que genera en automático la EBS cuando un concurrente es ejecutado. Si regresa 0 indica que no se puede ejecutar el concurrente; por ejemplo, si no se hubiera mandado llamar el proceso de inicialización de variables de ambiente la salida sería 0.

Lo siguiente es una condición, equivalente al IF-ELSE, donde se pregunta si la salida es diferente de 0, y tiene dos caminos. Cuando es 0, se actualiza la tabla de interface (ALP_TMP_REQUISITIONS_INTERFACE) en todos los registros con el mismo BATCH_ID procesado, el cuál fue asignado en “Assign5”, con un estatus de error. Cuando es diferente de 0 (la EBS generó el concurrente) se realizan otras validaciones. “InvokeUpdateTMP_Req” actualiza también la tabla de interface (ALP_TMP_REQUISITIONS_INTERFACE) concatenando a lo que había ya en estatus con el Id generado. Después “InvokeStatusRequisition” manda llamar un procedimiento almacenado que valida las líneas procesadas de acuerdo al BATCH_ID y el Id del concurrente generado; estos dos valores se asignan en “Assign6”. “Email3” también es solamente para debug (Ver Figura 7).



Figura 7. Tercera sección del proceso de BPEL SOA

Después se realiza otro COMMIT (Dehydrate2) y se reinician variables para el siguiente elemento del ciclo (Assign2).

6.1.6 Obtención del estatus del concurrente de Importación de Requisiciones

Este procedimiento es llamado mediante el proceso de BPEL SOA por el adaptador “InvokeStatusRequisition”. Es otro procedimiento almacenado. Se detalla a continuación.

Una vez que las líneas son procesadas por el concurrente de importación de requisiciones hay dos opciones: que haya terminado en error por alguna validación o que haya sido exitoso y la requisición ya se encuentre disponible en la tabla correspondiente y por consiguiente en el módulo de la EBS indicado. Si es el primer caso, van a permanecer en la tabla de interface de Oracle (PO_REQUISITIONS_INTERFACE_ALL) con estatus “ERROR”. Para el segundo caso desaparecen de dicha tabla y se almacenan en la tabla de requisiciones de Oracle (PO_REQUISITION_HEADERS_ALL). Se deben considerar ambas opciones.

El procedimiento se manda a llamar con dos parámetros de entrada, V_REQUEST_ID y V_BATCH_ID. El resto son variables de salida. La primera corresponde al Id que genera el concurrente de importación de requisiciones de Oracle. El segundo es el identificador del lote procesado.

Se declaran variables auxiliares. Lo que se hace es entrar a un ciclo que busque en cada una de las dos tablas posibles hasta que exista una coincidencia. Busca el registro como error para almacenarlo en la variable V_TRANSACTION_ID, si no existe valor genera una excepción y a la variable se le asigna NULL. Busca el registro como exitoso, si encuentra valor pobla la variable

auxiliar V_REQUISITION_HEADER_ID, en caso contrario la vuelve NULL. Después de preguntar por ambos registros entra a un condicionante: si la primera variable tiene contenido, es decir encontró una requisición en error, marca el estatus con un mensaje de incidencia y manda llamar al procedimiento que notifica a los usuarios. En cambio si la segunda variable tiene un valor diferente de NULL se marca el estatus con un mensaje de éxito y se manda llamar el procedimiento que notifica que la requisición se generó con éxito y se encuentra disponible en el módulo correspondiente.

El ciclo tiene otra forma de detenerse y es cuando el contador llega a un número máximo, un número aleatorio. Esto se implementó así porque el concurrente que genera la requisición puede tardar de acuerdo a las solicitudes y peticiones que tengan la EBS o la base de datos y si existe algún tipo de error y las líneas no fueron procesadas se debe evitar que el programa se quede en un ciclo infinito que pudiera causar problemas de rendimiento. Si existe alguna incidencia se notifica al desarrollador.

```

PROCEDURE get_status_requisition
(
    v_request_id          IN  NUMBER
  , v_batch_id           IN  NUMBER
  , v_estatus            OUT VARCHAR2
  , v_error              OUT VARCHAR2
  , v_requisition_header_id OUT NUMBER
  , v_transaction_id     OUT NUMBER
)
IS
    v_value              NUMBER;
    v_flag               VARCHAR2(5000) := NULL;
    v_estatus_err        VARCHAR2(500);
    v_counter            NUMBER         := 0;
BEGIN
    --Inicializamos con error el status y error por si se agota el tiempo de espera
    v_estatus := 'ERROR';
    WHILE v_flag IS NULL
    LOOP
        --Buscamos ahora en la tabla de interface en caso de algún error
        BEGIN
            SELECT transaction_id
            INTO   v_transaction_id
            FROM   apps.po_requisitions_interface_all
            WHERE  request_id      = v_request_id
            AND    batch_id       = v_batch_id
            AND    process_flag    = 'ERROR';
        EXCEPTION WHEN NO_DATA_FOUND THEN
            NULL;
        WHEN OTHERS THEN
            v_flag := 'OK';
            v_estatus := 'ERROR';
            v_error := SQLERRM;
        END;
        --Buscamos en la tabla de requisiciones en caso contrario
        BEGIN
            SELECT requisition_header_id
            INTO   v_requisition_header_id
            FROM   po_requisition_headers_all
            WHERE  request_id = v_request_id;
        EXCEPTION WHEN NO_DATA_FOUND THEN
            NULL;
        WHEN OTHERS THEN
            v_flag := 'OK';
            v_estatus := 'ERROR';
            v_error := SQLERRM;
        END;
        --Buscamos si alguno de las variables ya tiene valor
        --En caso contrario volvería a entrar al WHILE
    END LOOP;
END;

```

```

--Actualizamos la tabla de interface
UPDATE alp_tmp_requisitions_interface
SET status = status || ' || ERRORES DE IMPORTACION - ' || v_request_id
WHERE batch_id = v_batch_id
AND status = 'REQUISITION PROCESSED';
COMMIT;
--Debemos notificar que no se generó la requisición y el concurrente
--de importación
errores_requisicion_imp
(
    v_transaction_id
    , v_batch_id
    , v_request_id
    , v_estatus_err
);
ELSIF v_requisition_header_id IS NOT NULL THEN
v_estatus := 'REQUISICION_ENCONTRADA';
v_flag := 'OK';
v_error := NULL;

UPDATE alp_tmp_requisitions_interface
SET status = status || ' || REQUISICION_ENCONTRADA - ' || v_request_id
WHERE batch_id = v_batch_id
AND status = 'REQUISITION PROCESSED';
COMMIT;

--Como se generó correctamente la requisición notificamos
exito_requisicion
(
    v_batch_id
    , v_request_id
    , v_estatus
);
END IF;
IF v_counter >= 100000000 THEN
v_flag := 'ERROR';
v_estatus := 'ERROR';
v_error := 'No se encontró información después del LOOP 100';
END IF;
v_counter := v_counter + 1;
END LOOP;

v_estatus := 'OK';
EXCEPTION WHEN OTHERS THEN
v_estatus := 'EXCEPTION: ' || SQLERRM;
v_error := SQLERRM;
END;

```

6.1.7 Notificación de mensajes por medio de correo electrónico

Cuando en el procedimiento de VALIDA_ESTATUS_LINEAS se debe notificar que ocurrió algún problema en la validación de los datos se manda llamar al procedimiento ERRORES_REQUISICION. Lo que realiza es el armado de la notificación que será enviada por medio de correo electrónico. El parámetro que recibe de entrada es el V_BATCH_ID, es decir, el lote de las líneas a procesar. Como el requerimiento funcional era que el contenido del mensaje incluyera una tabla, el cuerpo de la notificación se hizo por medio de HTML. Una cuestión a destacar aquí es que el cuerpo del mensaje se almacena en un tipo de variable de base de datos CLOB, puesto que el límite de un VARCHAR2 de tamaño de Oracle no era suficiente para algunos de los mensajes que contenían una gran cantidad de líneas. El código HTML se guarda en forma de variable y la tabla se arma con un ciclo, pues el tamaño de filas de la misma depende completamente del número de registros a procesar.

El primer paso es crear las variables auxiliares y se declara un cursor que va a leer todas las líneas con BATCH_ID necesario. Se tiene un cursor más que obtiene los correos electrónicos de los

usuarios involucrados en el proceso (almacenados en un juego de valores) para poder notificarles a todos y cada uno de ellos.

```

PROCEDURE errores_requisicion
(
    v_batch_id      IN NUMBER
    , v_estatus     OUT VARCHAR2
)
IS
v_user_preparer   VARCHAR2 (32766);
v_fecha          VARCHAR2 (32766);
v_web_reference   VARCHAR2 (32766);
v_origen         VARCHAR2 (32766);
v_str_header     VARCHAR2 (32766);
v_string_table   VARCHAR2 (32766);
v_str_foot       VARCHAR2 (32766);
v_string_complete VARCHAR2 (32766);
l_clob           CLOB;
CURSOR c_lines(p_batch_id NUMBER)
IS
    SELECT *
    FROM alp_tmp_requisitions_interface
    WHERE batch_id = p_batch_id
    AND status = 'REQUISITION ERROR - ERROR EN VALORES DERIVADOS O CALCULADOS'
    ORDER BY TO_NUMBER(header_attribute9) ASC;
--Almacena todos los correos
l_email_list     VARCHAR2 (500);
--Lista de envío
CURSOR c_correos
IS
    SELECT ffv.FLEX_VALUE
    FROM fnd_flex_value_sets ffvs, FND_FLEX_VALUES ffv
    WHERE 1 = 1
    AND ffvs.FLEX_VALUE_SET_ID = ffv.FLEX_VALUE_SET_ID
    AND ENABLED_FLAG = 'Y'
    AND ffvs.FLEX VALUE SET NAME LIKE 'ALP B2B CORREOS';

```

Se obtienen algunos datos que deben presentarse en el correo y se guarda detalle del código HTML que va a conformar el mensaje. Esto se almacena primeramente en una variable de tipo VARCHAR2, pero al terminar de guardarla se va escribiendo en la variable de tipo CLOB.

```

BEGIN
--obtenemos valores del primer header
SELECT user_preparer
, document_date
, header_attribute8
, header_attribute_category
INTO v_user_preparer
, v_fecha
, v_web_reference
, v_origen
FROM alp_tmp_requisitions_interface
WHERE batch_id = v_batch_id
AND status = 'REQUISITION ERROR - ERROR EN VALORES DERIVADOS O CALCULADOS'
AND rownum = 1;
v_str_header :=
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Untitled Document</title> <style type="text/css">
.celda { color: #09C;} .letra { color: #FFF;} .letra { font-style: italic; } .letra2 { color:
#000; } </style> </head>
<body>
<p>Se ha recibido una requisición B2B Tetra Pak / Alpura con error, por favor validar la
información:<br />
Usuario: <strong>'          || v_user_preparer ||'</strong><br />
Fecha: <strong>'          || v_fecha          ||'</strong><br />
Web Reference ID: <strong>' || v_web_reference ||'</strong><br />
Origen: <strong>'          || v_origen         ||'</strong></p>
<p>null; </p>
<table width="669" height="136" border="2">

```



```

<th bgcolor="#0099FF" scope="col">Artículo</th>
<th bgcolor="#0099FF" scope="col">UDM</th>
<th bgcolor="#0099FF" scope="col">Cantidad</th>
<th bgcolor="#0099FF" scope="col">Estado</th>
<th bgcolor="#0099FF" scope="col">Descripción</th>
</tr>
';

--Inivializamos el CLOB
DBMS_LOB.createtemporary(l_clob, FALSE);
DBMS_LOB.writeappend (l_clob, LENGTH(v_str_header), v_str_header);

```

Por cada una de las líneas se arma el detalle de la tabla, también se escribe en la variable CLOB. Una vez que se tiene listo el cuerpo de mensaje se obtiene cada una de las direcciones de correo electrónico a procesar y se manda llamar al procedimiento que hace el envío de correo. Se le pasan como parámetros el contenido del mensaje, la lista de correos y el título del correo. Si existe alguna excepción se avisa al desarrollador.

```

FOR x IN c_lines(v_batch_id)
LOOP
v_art_description := get_articulo_description(x.item_segment1);
v_string_table :=
'<tr>'
|| '<td>' || x.header_attribute9 || '</td>'
|| '<td>' || x.item_segment1 || '</td>'
|| '<td>' || v_art_description || '</td>'
|| '<td>' || x.uom_code || '</td>'
|| '<td>' || x.quantity || '</td>'
|| '<td>' || x.status || '</td>'
|| '<td>' || x.error_description || '</td>'
|| '</tr>' || chr(13);

DBMS_LOB.writeappend (l_clob, LENGTH(v_string_table), v_string_table);

v_string_table := NULL;
END LOOP;

v_str_foot :=
'
</table>
</body>
</html>
';
DBMS_LOB.writeappend (l_clob, LENGTH(v_str_foot), v_str_foot);
FOR r_correos IN c_correos
LOOP
l_email_list := l_email_list || r_correos.FLEX_VALUE || ',';
END LOOP;

send_email
(
v_string_complete
, l_email_list
, 'Error en recepción de requisición de compra B2B Tetra pak / Alpura'
, l_clob
, v_salida_email
);
v_estatus := 'OK';
DBMS_LOB.freetemporary(l_clob);
DBMS_OUTPUT.PUT_LINE('v_salida_email: ' || v_salida_email);
EXCEPTION WHEN OTHERS THEN
v_estatus := 'Exception errores_requisicion: ' || SQLERRM;
alp_custom.alp_test_send_email.
send_mail ('danlopez@alpura.com',
'aworkflow@alpura.com',
'Exception errores_requisicion en ALP_TETRAPAK (Test 139)',
SQLERRM,
'172.108.1.74',
25);
END;

```

El proceso de notificación de errores durante el proceso de importación y el de éxito de la misma son muy similares, solamente se agregan algunas columnas o se cambia el texto del título del correo.

Ejemplo de un mensaje de notificación de una requisición de orden de compra con error:

Error en recepcion de requisicion de compra B2B Tetra pak / Alpura

aworkflow@alpura.com
para

Se ha recibido una requisicion B2B Tetra Pak / Alpura con error, por favor validar la informacion:
 Usuario: **JLOZANO**
 Fecha: **16/09/2014**
 Web Reference ID: **15-123456789**
 Origen: **B2B Requisition Tetrapak**

No.	Codigo	Articulo	UDM	Cantidad	Estado	Descripcion
10	DIENTC0088	ENVASE TBA 1 LITRO CLASICA EXPORTACION	pz	10000	REQUISITION ERROR - ERROR EN VALORES DERIVADOS O CALCULADOS	EL ARTICULO NO EXISTE PARA LA ORGANIZACION LA UNIDAD DE MEDIDA NO CORRESPONDE CON EL ARTICULO
610	DIEMNT0005	CINTA PEP A872972801	pz	610000	REQUISITION ERROR - ERROR EN VALORES DERIVADOS O CALCULADOS	LA UNIDAD DE MEDIDA NO CORRESPONDE CON EL ARTICULO
620	DIENPK0004	POLIETILENO TETRA MOULDING DIMC-A	pz	620000	REQUISITION ERROR - ERROR EN VALORES DERIVADOS O CALCULADOS	LA UNIDAD DE MEDIDA NO CORRESPONDE CON EL ARTICULO

Figura 9. Ejemplo de email que arriba cuando existe un error en el proceso de recepción de requisición

Ejemplo de una notificación de una requisición generada con éxito:

Recepcion de requisicion de compra B2B Tetra pak / alpura no. 63017

aworkflow@alpura.com
para

Se ha recibido una requisicion B2B Tetra pak / alpura para su validacion y autorizacion.
 No. de Requisicion: **63017**
 Usuario: **JLOZANO**
 Fecha: **25/09/2014**
 Web Reference ID: **25-12296771**
 Origen: **B2B Requisition Tetrapak**

No.	Codigo	Articulo	UDM	Cantidad
10	DIENTC0049	ENVASE TBA 250 MILILITROS KIDS	pz	1000000
20	DIENTC0152	ENVASE TBA CHOCOLATE VENEZUELA 250 ml	pz	1100000
30	DIENTC0153	ENVASE TBA CREMA UHT EXPORTACION USA 250 ml	pz	1200000

Figura 10. Ejemplo de email que arriba cuando el proceso de recepción de requisición es exitoso

6.2 Envío de orden de compra

Una vez que la requisición se ha generado en la EBS de Oracle y el usuario que creará una orden de compra con dicha requisición fue notificado, realiza las validaciones internas necesarias y completa el proceso. Cuando la orden de compra es generada se procede a su aprobación, lo que genera su liberación. Cuando esto ocurre está lista para ser enviada nuevamente a Tetra Pak para notificarles que el proceso se completó y deben enviar el material. La forma en que esta operación es detectada es mediante un disparador sobre la tabla de liberación de órdenes de compra (propia de la aplicación). Se valida que la información esté completa de la orden de compra y se inserta en la tabla de interface; también se envía un correo electrónico a los involucrados de que se generó con éxito. Delta también tiene un servicio programado que detecta nuevos registros en la tabla de interface con tipo de transacción ORDERS, con un estatus PENDIENTE y un tipo de documento ENTREGA (mismos que se insertan para el envío de orden de compra). Llegado el momento en que detecta registros con estas características procede a generar un EDIFACT que será enviado a Tetra Pak como para que sean notificados y como siguiente paso envíen el material.

El proceso de envío de orden de compra se presenta en la Figura 11.

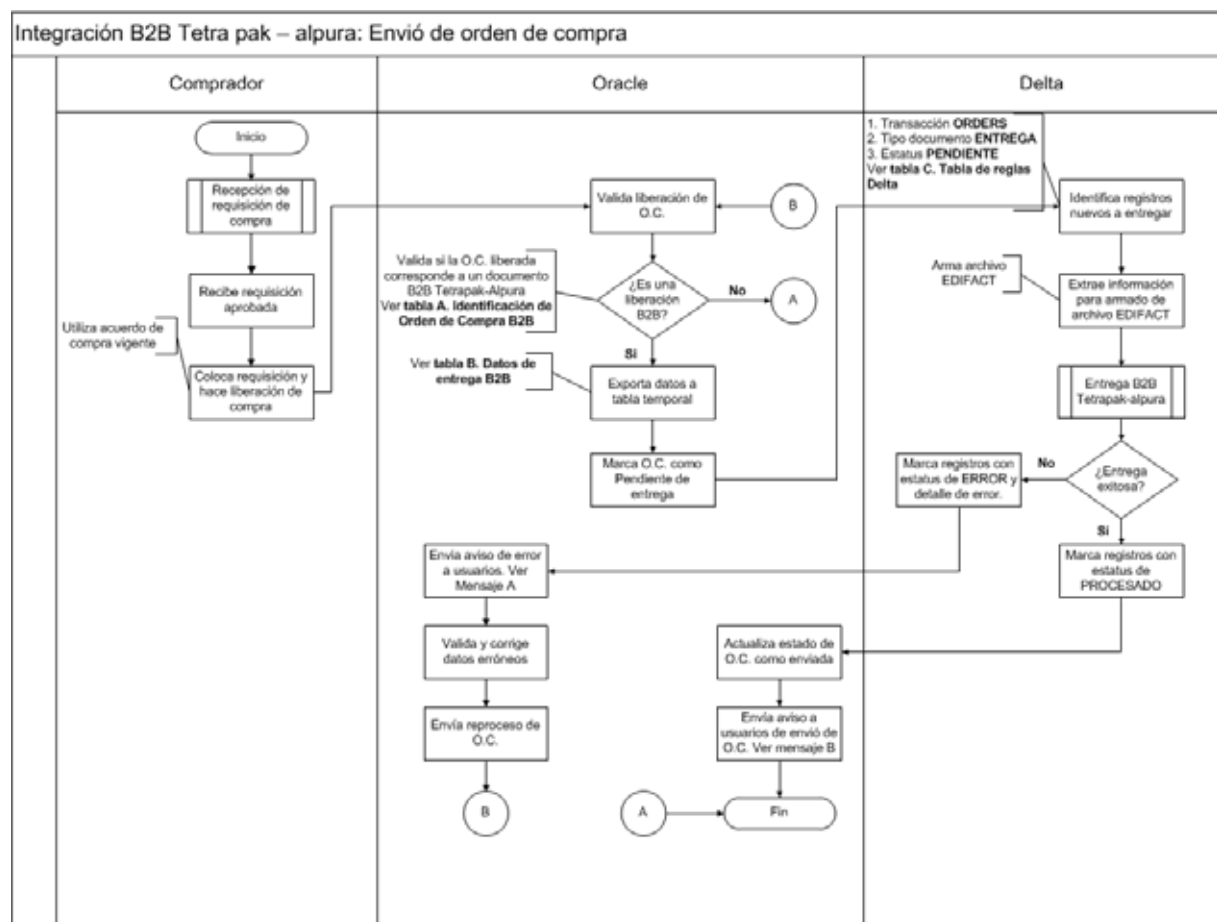


Figura 11. Proceso de envío de orden de compra.

6.3 Confirmación de orden de compra

Para corroborar que Tetra Pak recibió con éxito la orden de compra envía un nuevo EDIFACT que es mapeado e interpretado por Delta depositándolo en la tabla de interface con un tipo de transacción ORDRSP. Cuando esto ocurre existe otro trigger que se encarga de recibir este tipo de mensajes para su validación. Aquí lo que se corrobora es que la información que está recibiendo y generando Tetra Pak en cuanto a artículos, cantidades, etc., sea igual a la que se generó en la EBS y es la que Alpura espera.

Nuevamente, dependiendo si las validaciones fueron correctas o existe algún tipo de error se notifica por medio de correo electrónico a los involucrados.

La Figura 12 permite visualizar el proceso de Confirmación de Orden de Compra.

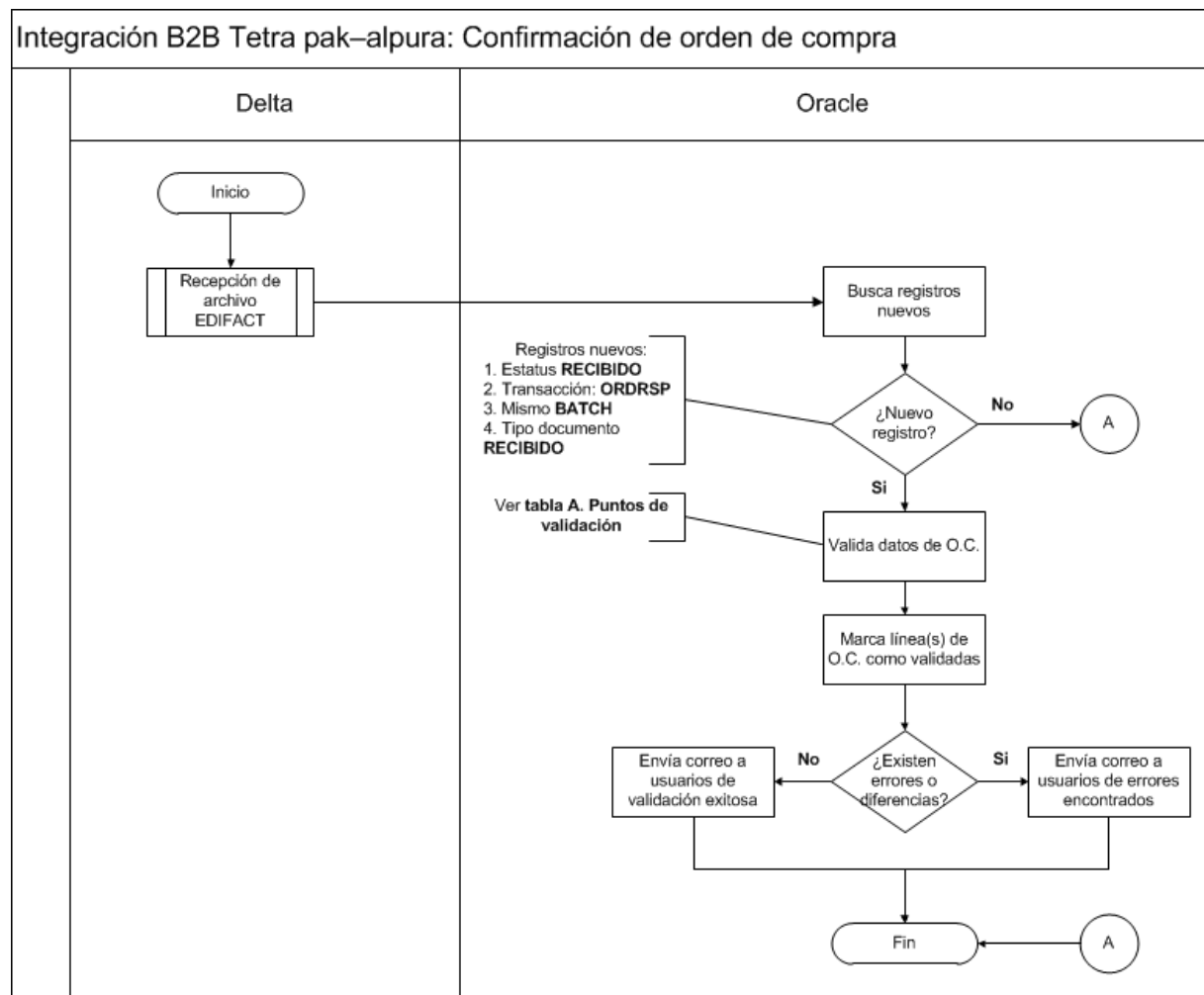


Figura 12. Proceso de envío de confirmación de orden de compra

6.4 Notificación de Entrega

Llegado el momento en que Tetra Pak ya tiene listo el material de la Orden de Compra envía primeramente un nuevo mensaje EDIFACT, en esta ocasión con un tipo de transacción “DESADV” y un número de entrega (DELIVERY_NUMBER) con el contenido de lo que posteriormente

enviará físicamente. Cuando esto ocurre nuevamente se envía un correo de notificación a los usuarios y los registros quedan en la tabla de interface listos para ser recibidos por medio de una pantalla de Oracle Forms creada específicamente con ese objetivo. Es muy importante mencionar que si Tetra Pak envía el material de forma física pero nunca lo notificó por medio del EDIFACT no se puede recibir puesto que no aparecerá como registro en la pantalla por estar ausente en la tabla de interface.

Ya que el material arribó físicamente a la planta el usuario encargado de la tarea de recepción utiliza la pantalla de recepción donde selecciona la orden de compra y el número de liberación correspondiente para poder consultar y realizar la recepción de material.

La Figura 13 muestra el proceso de notificación de entrega.

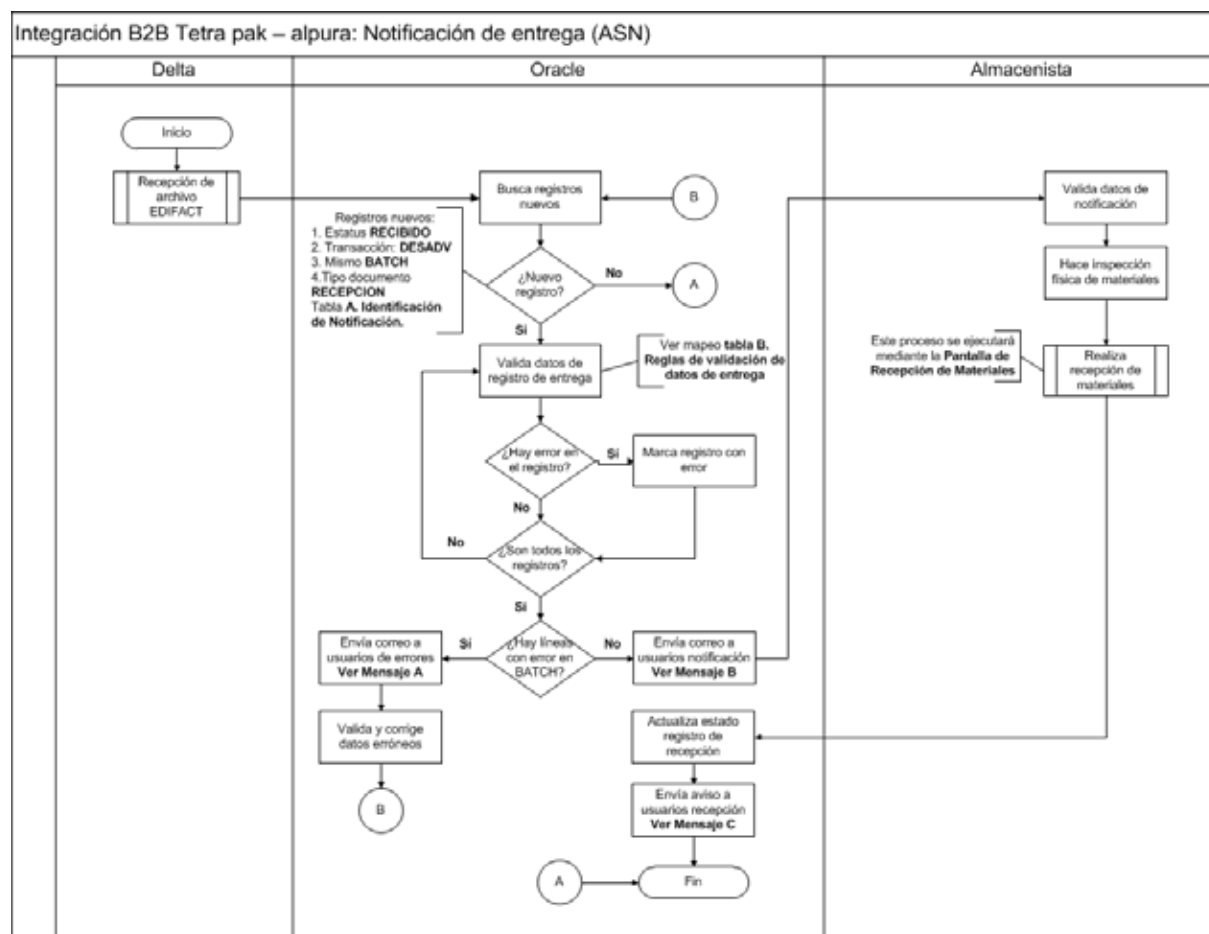


Figura 12. Proceso de envío de confirmación de orden de compra

Referencias

Business-to-business (n.d.). En Wikipedia. Recuperado el 5 de noviembre de 2014, de <http://es.wikipedia.org/wiki/Business-to-business>.

Intercambio electrónico de datos (n.d.). En Wikipedia. Recuperado el 5 de noviembre de 2014, de http://es.wikipedia.org/wiki/Intercambio_electr%C3%B3nico_de_datos.

EDIFACT (n.d.). En Wikipedia. Recuperado el 5 de noviembre de 2014, de <http://es.wikipedia.org/wiki/EDIFACT>.

Creando Programas Concurrentes en EBS R12, Aplicación Personalizada (n.d.). En Blogspot. Recuperado el 5 de noviembre de 2014, de <http://devbolivia.blogspot.mx/2013/06/creando-programas-concurrentes-en-ebs.html>.

Compartiendo Tips de Oracle (n.d.). En Blogspot. Recuperado el 5 de noviembre de 2014, de <http://profesionghh.blogspot.mx/2014/05/programas-concurrente.html>.

Date tricks on SQL (n.d.). En Blogspot. Recuperado el 5 de noviembre de 2014, de <http://mexico-desarrolla.blogspot.mx/>.

BPEL, Business Process Execution Language (n.d.). En Wikipedia. Recuperado el 5 de noviembre de 2014, de <http://searchsoa.techtarget.com/definition/BPEL>.

Juego de valores (n.d.). En Blogspot. Recuperado el 5 de noviembre de 2014, de <http://profesionghh.blogspot.mx/2014/05/juego-de-valores.html>.