



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Máster Universitario en Ingeniería y Tecnología de
Sistemas Software

Departamento de Sistemas Informáticos y Computación

**Análisis de deficiencias de NRT-DWH
(Near-Real-Time DataWarehouse)**

Autor: Luis Juan García Guerola

Tutor: Juan Carlos Casamayor Ródenas

Curso 2019-2020

TABLA DE CONTENIDOS

TABLA DE CONTENIDOS	2
1. INTRODUCCIÓN	4
1.1 OBJETIVO.....	4
1.2 METODOLOGIA DE TRABAJO	5
2. VISIÓN GENERAL DE NRT-DWH	7
2.1 COMPONENTE 1: APLICACION NRT-DWH	7
2.2 COMPONENTE 2: BASE DE DATOS NRT-DWH	8
3. EVALUACIÓN DE NRT-DWH	9
3.1 INGESTIÓN DE DATOS	9
3.1.1 Situación actual	9
3.1.2 Problemas detectados	12
3.1.3 Mejoras propuestas	14
3.2 FLUJOS DE DATOS	16
3.2.1 Situación actual	16
3.2.2 Problemas detectados	33
3.2.3 Mejoras propuestas	35
3.3 REPOSITORIO DE DATOS	36
3.3.1 Situación actual	36
3.3.2 Problemas detectados	40
3.3.3 Mejoras propuestas	42
3.3.4 Ejemplos de problemas.....	46
3.4 CALIDAD DE DATOS.....	53
3.4.1 Situación actual	53
3.4.2 Problemas detectados	54
3.4.3 Mejoras propuestas	55
3.4.4 Ejemplos de problemas.....	56
3.5 SEGURIDAD DE DATOS	61
3.5.1 Situación actual	61
3.5.2 Problemas detectados	62
3.5.3 Mejoras propuestas	63
3.5.4 Ejemplos de problemas.....	64
3.6 RENDIMIENTO.....	65
3.6.1 Situación actual	65
3.6.2 Problemas detectados	67
3.6.3 Mejoras propuestas	68
3.7 GESTIÓN DE CAMBIOS.....	69
3.7.1 Situación actual	70
3.7.2 Problemas detectados	71
3.8 GESTIÓN DE PRODUCCIÓN / MONITORIZACIÓN Y ESTABILIDAD	72
3.8.1 Situación actual	72
3.8.2 Problemas detectados	72
3.8.3 Mejoras propuestas	73
3.9 INFORMES DE DOWNSTREAM	76
3.9.1 Situación actual	76
3.9.2 Problemas detectados	76
3.9.3 Mejoras propuestas	76
4. VISIÓN DE MERCADO PARA LA ESTRATEGIA Y LA ARQUITECTURA DE DATOS	78

4.1 ARQUITECTURA DE REFERENCIA	78
4.2 MEJORES PRÁCTICAS PARA CADA ÁREA INVESTIGADA	79
5. PLAN DE ACCIÓN PROPUESTO.....	81
5.1 PROBLEMAS PRINCIPALES – PUNTOS MÁS DÉBILES.....	81
5.2 PLAN DE EVOLUCIÓN	83
5.3 DISEÑO ARQUITECTÓNICO PROPUESTO.....	86
5.3.1 Desacoplamiento de la ingestión de datos	86
5.3.2 Desacoplamiento de los Downstreams: capa de informes	92
5.3.3 Programa de calidad de datos	96
5.3.4 Mejora de la estabilidad de la producción.....	96
5.4 COSTE Y BENEFICIOS DE IMPLEMENTACIÓN.....	97
6. CONCLUSIONES.....	99
7. ANEXO I.....	100
7.1 BIBLIOGRAFIA	100

1. INTRODUCCIÓN

1.1 OBJETIVO

NRT-DWH (Near Real Time Data Warehouse) es una aplicación de referencia en lo que se refiere a valores financieros.

La aplicación se creó hace más de 15 años y tuvo un fuerte crecimiento debido al interés de los consumidores de la información almacenada y generada.

Este intenso y rápido crecimiento, obligó a tomar una serie de medidas tácticas que condujeron a un diseño poco óptimo de la aplicación. Ahora, el cliente solicita realizar un análisis objetivo e independiente al equipo actual de trabajo, que haga resaltar las deficiencias y proponga una serie de mejoras a medio y largo plazo.

En la sección "[Visión general de NRT-DWH](#)" se proporciona una descripción general de alto nivel del sistema NRT-DWH, pero el propósito clave de este documento es asesorar acerca de los siguientes resultados, para las áreas de interés identificadas en la aplicación de NRT-DWH:

1. Proporcionar una visión del estado actual, con cierto nivel de detalle que permita comprender los problemas.
2. Identificar los problemas / debilidades clave que existen en la aplicación.
3. Proporcionar propuestas para mejorar la aplicación y remediar los problemas detectados, proporcionando una visión de alto nivel del esfuerzo que podría ser necesario para solucionarlos (bajo / medio / alto)

La siguiente tabla proporciona el mapeo entre las áreas de interés definidas con el cliente:

Áreas de interés	Sección del documento
Proceso de ingestión de datos de NRT-DWH comparado con las buenas prácticas en la industria.	Ingestión de Datos
Flujos de Datos	Flujos de Datos
Repositorio de Datos <ul style="list-style-type: none">• Entidades que incluyen datos de referencia, datos transaccionales, datos del ciclo de vida de las transacciones, posiciones, acciones corporativas e información de saldo de efectivo.• Modelo Lógico de Datos (pruebas de diseño y validación de datos)• Modelo Físico de Datos	Repositorio de Datos
Facilidad para incorporar cambios al modelo de datos u otras áreas de la aplicación	Gestión del Cambio
Gestión y solución de errores con los sistemas origen de datos Informes / Acciones de Calidad de Datos	Calidad de Datos
Monitorización / Gestión de Producción	Gestión de Producción, Monitorización y Estabilidad
Rendimiento	Rendimiento
APIs para recuperar datos de NRT-DWH.	Informes Finales

1.2 METODOLOGIA DE TRABAJO

El objetivo previsto por el cliente, es que el análisis y asesoramiento sea realizado por un equipo completamente independiente del que se encarga actualmente de la aplicación. Con esta premisa, a continuación se describe la metodología de trabajo propuesta, con el fin de documentar el estado actual del sistema NRT-DWH, detección de problemas y propuestas de mejora.

Establecer una reunión inicial con las partes principales interesadas (Analistas de Negocio, Analistas Funcionales y Tecnicos Especialistas) para obtener una visión general del sistema, los repositorios de documentación y los contactos principales.

Configurar el entorno local: aplicaciones, repositorios, herramientas, etc. y familiarizarse con la interfaz de usuario [Arquitectura de la Interfaz de Usuario](#). Con este enfoque, podremos revisar el código fuente y realizar algunas pruebas para comprender las especificaciones funcionales y técnicas.

Revisar el código y comprender un flujo de datos genérico. End Of Day es el proceso de reconciliación que debe cubrirse. [Flujos de Datos](#)

Preparar algunas consultas para calcular los volúmenes por ubicación y comprender el ratio de procesamiento, el rendimiento y los datos más importantes. Los errores y los lotes de reprocesamiento se describirán como parte de la [Calidad de Datos](#) para comprender dónde se están causando las brechas.

Se agregará un nivel de detalle para explicar el estado actual de la siguiente manera. Esto se basa en el modelado de procesos, un estándar para crear documentos AS IS:

- 1) [Diagrama de procesos](#): solo los pasos principales para explicar cómo fluyen los datos. Entidades genéricas y específicas.
- 2) [Mapa de procesos](#): luego mapearemos los procesos y explicaremos cómo interactúan los datos con las partes interesadas.
- 3) [Modelo de procesos](#): tomando el mapa de procesos, aumentaremos el nivel de detalle y el modelo de datos del documento, procesos operativos y técnicos, posibles errores y reprocesamiento, lagunas de datos, etc...

Ejemplo de flujo de transacción (funcional y técnico)

1. Después de revisar el código, enviar los mensajes correspondientes para reproducir los casos de uso del Ciclo de vida de la transacción (pendiente, cancelado, liquidado, etc.). Los volúmenes por ubicación mostrarán los casos de uso más significativos. Luego hay que centrarse en ellos y describir desde la recepción del mensaje XML hasta la ingestión de datos. Ingestión de datos. [Ingestión de datos](#)
2. Se debe obtener las tablas afectadas en la base de datos durante la ingestión de datos. Esto incluirá la lógica de procesamiento en los componentes de Java y la lógica de los triggers^[1] de la base de datos (triggers con eventos anteriores y posteriores a la carga de las tablas afectadas). Este ejercicio mostrará la imagen completa de los datos para el flujo de Transacción. [Flujos de Datos](#)
3. Con Oracle SQL Data Modeler, se crea el modelo de datos físicos para el flujo actual. [Repositorio de Datos](#)
4. A continuación se extrae el modelo lógico. En caso de que las claves ajenas no relacionen las tablas, describiremos las claves lógicas y haremos una explicación coherente del modelo. [Repositorio de Datos](#)
5. Describir en un diagrama cómo fluyen los datos del gestor de colas MQ series (XML enviado) hasta que los datos se almacenen en la base de datos.
6. Explicar la facilidad de incorporar cambios en el modelo de datos de Transaction en caso de que un downstream solicite nuevos atributos. La lógica de procesamiento de Java también se verá afectada. [Gestión del Cambio](#)

Análisis de deficiencias – NRT-DWH

Ejemplo de transacción para revisar el rendimiento:

Podemos analizar, para un ciclo de vida representativo de la transacción, las diferentes fases de los mensajes XML que se reciben hasta que los datos se insertan en la base de datos. es decir: enviar 100k mensajes o reproducir un día típico de EOD (posiciones) para una ubicación / sucursal específica. El objetivo de estas pruebas es medir dónde puede mejorar NRT-DWH y detectar cuellos de botella (algunos de ellos podrían estar en procesos paralelos).

Ejemplo de fases medidas:

1. Leer el mensaje en MQ-Series (Sistema de mensajería asíncrono) ^[2]
2. Obtener un mensaje XML y validar contra DTD (*Definición de Tipo de Documento*)^[3]
3. Crear un objeto antes de realizar la lógica de procesamiento o los cálculos.
4. Ejecutar cálculos para obtener el objeto resultante.
5. Mapear el objeto final con las columnas de la base de datos
6. Registrar los datos del mapeo en la base de datos.

Los [Informes de Downstream](#) también son importantes para asegurarse de que los datos sean correctos. Se describen las APIs y cómo los sistemas de consumo de datos de NRT-DWH. Para lograr esta parte, se debe revisar las páginas de Confluence (Sistema de documentación Online) y los entornos actuales de NRT-DWH (Control-M, Unix box ...) para comprender la comunicación entre los sistemas.

2. VISIÓN GENERAL DE NRT-DWH

NRT-DWH es el datawarehouse global (casi en tiempo real) del cliente para el negocio de valores en su área financiera. El sistema en sí contiene dos componentes principales (descritos a continuación), pero a su vez se conecta a una serie de sistemas que actúan tanto de orígenes como de destinos de datos.

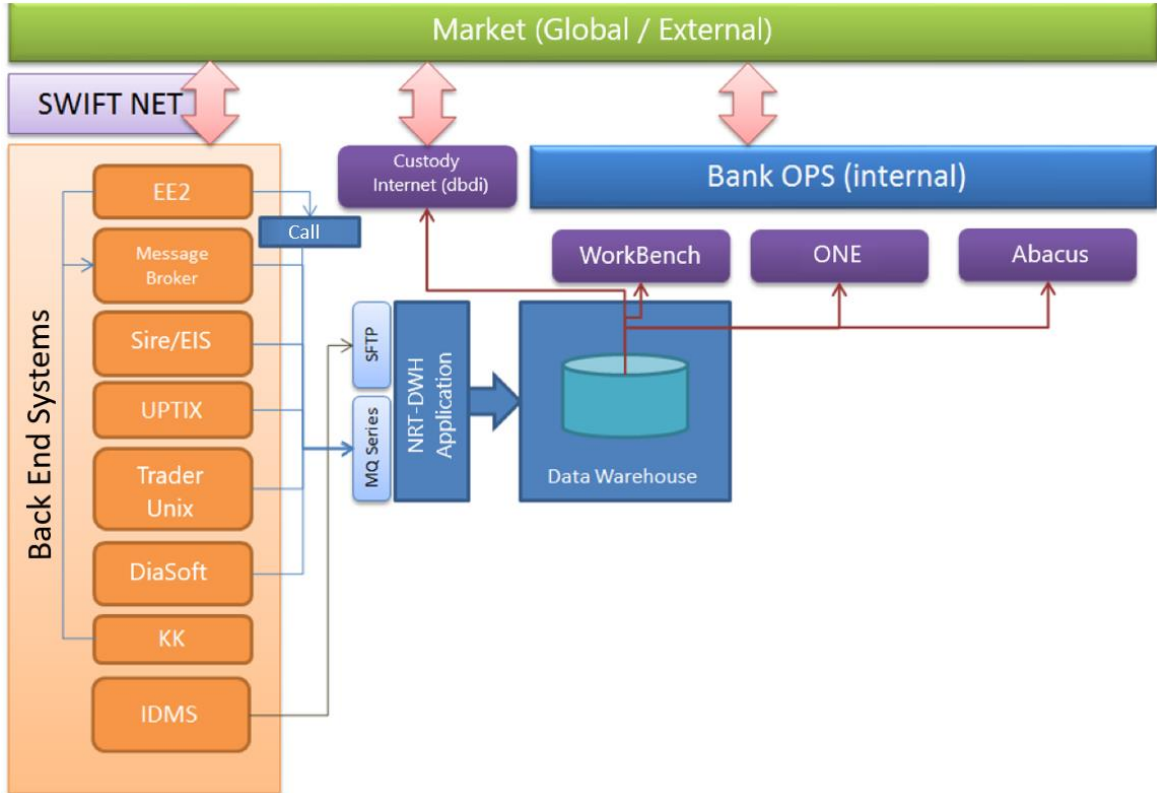


Figura 1.1. NRT-DWH – Vista de Contexto del Sistema

2.1 COMPONENTE 1: APLICACION NRT-DWH

La función principal de J2EE en la aplicación NRT-DWH es procesar los mensajes recibidos via MQ Series, parsear los mensajes y cargar el contenido en la base de datos aplicando diversas reglas de negocio. El lenguaje estandar de comunicación de los mensajes de NRT-DWH es XML.

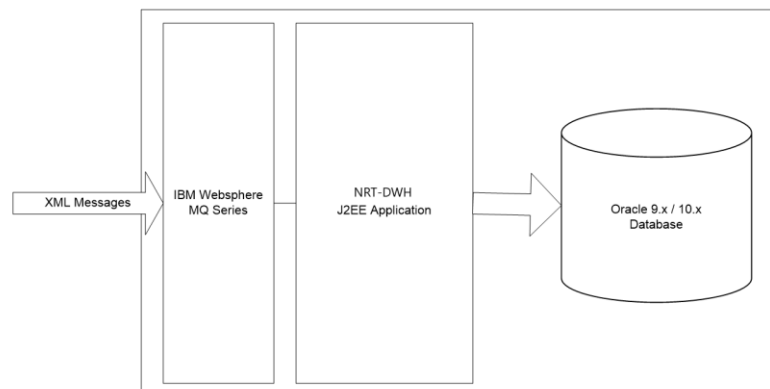


Figura 1.2. NRT-DWH – Componentes

3. EVALUACIÓN DE NRT-DWH

3.1 INGESTIÓN DE DATOS

3.1.1 SITUACIÓN ACTUAL

La ingestión de datos de NRT-DWH se especifica en el siguiente diagrama. En esta sección, se explicará un flujo de datos genérico para introducir los modelos de datos para las entidades funcionales específicas (casos de uso).

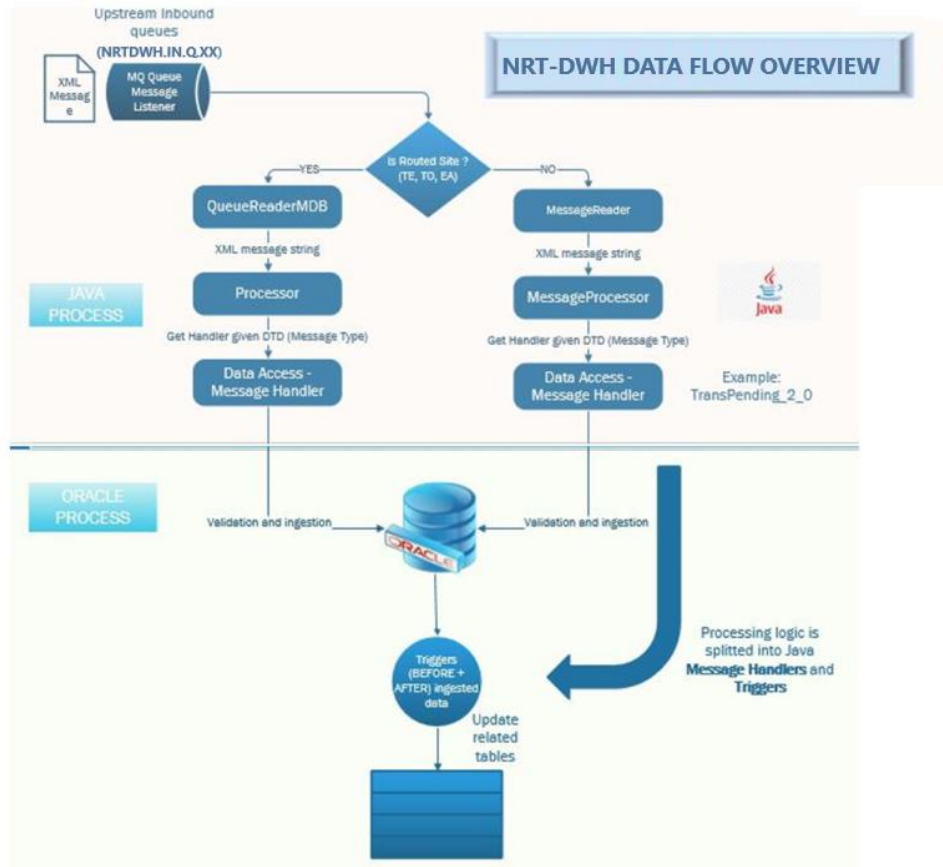


Figura 2. Estado Actual de Ingestión de Datos NRT-DWH

3.1.1.1 PROCESO JAVA

1) MQ Queue Message Listener

Un sistema Upstream (Trader, KK, Sire, etc...) manda un mensaje XML a la cola de entrada correspondiente de NRT-DWH. Cada cola corresponde a una ubicación específica, que es una subentidad de una aplicación Upstream. Hay algunas ubicaciones específicas que contienen subniveles de cola para procesar mensajes en paralelo y así mejorar el rendimiento.

Análisis de deficiencias – NRT-DWH

La siguiente tabla muestra los sistemas Upstream con las ubicaciones disponibles que son identificadas por cada cola:

Upstream	Sitios
Cuentas	Cd
Hielo	EA* , si
IDMS	I2, I8, IB, IC, IH, II, IJ, IK, IL, IM, IN, IP, IS, IT, IV, IX, IY
Kk	IE, IE
Uptix	DJ, UB, UI, UJ, REINO UNIDO, UL, UM, UT, UV, UY
Winbank	Wq
Internet	I1, I3, I4, I5, IA, IF, IG, IR, IU, IW, IZ
Comerciante	T1, T3, T4, T5, T8, T9, TB, TC, TD, TG, TH, TI, TJ, TK, TL, TM, TN, TO* , TP, TQ, TR, TS, TT, TU, TV, TX, TY, TZ
Trader MVS	AR, T2, TE*

***NOTA:** Ubicaciones enrutadas **en negrita**

InboundMsgListener es la clase Java que implementa el proceso de escucha y recepción de mensajes. Los mensajes recibidos pueden ser reenviados a un subnivel para una serie de ubicaciones: **TE, IE, TU, TD, EA, TQ, TO**

2) Decisión de enrutamiento de la ubicación.

Cuando se lee el mensaje XML, NRT-DWH toma la decisión de enviar el mensaje a un lector específico. Si la ubicación remitente es **TE, TO, EA**, el proceso pasará por QueueReaderMDB, de lo contrario MessageReader. Esta decisión enrutará el mensaje a un conjunto de controladores de mensajes que ingestan el contenido del mensaje XML en la base de datos.

3) Message Reader (QueueReaderMDB, MessageReader)

En esta fase, el mensaje se convierte en una cadena y se envía al procesador correcto.

1) Message Processor (Processor, MessageProcessor)

El procesador de mensajes será responsable de los siguientes pasos:

- 4.1) **Obtener la DTD especificada** en el contenido del mensaje. Se busca con el patrón ".dtd" dentro del mensaje. No se valida primero ningún esquema. Sólo se pone la cadena.
- 4.2) **Insertar una fila** en la tabla RCM, que es una especie de **tabla sin procesar** para realizar un seguimiento de los mensajes que fluyen a través de NRT-DWH. Esta fila se actualizará más adelante con el estado diferente para el mensaje. Indicará si el estado es Correcto, Reprocesar, Error o Rechazado.
- 4.3) **Asignar la DTD (análisis XML) con el " Data Access-Message Handler " correcto**, que extiende BaseHandler (Processor) o implementa INrtDwhContentHandler(MessageProcessor) en función de la "decisión de la ruta de la ubicación". Los controladores de mensajes se duplican para cada tipo de mensaje para ambas rutas de decisión.

- 4.4) **Analizar el mensaje XML con el Message Handler (SAX Parser basado en controladores de contenido).** Todos los controladores de mensajes se duplican para los sitios “enrutados” y “no enrutados”, implementando la misma lógica de negocio.
- 4.5) **Ingestar^(A) filas en la base de datos desde el Message Handler.** Los atributos XML se asignan con las columnas de base de datos y se convierten después de evaluar alguna lógica, es decir: los mensajes de transpending se asignan con TransPending_X_X.dtd y la clase TransPending_X_X controlará la ingestión.
- 4.6) **Procedimientos de Post-procesamiento para reconciliar otros datos. Ej:** Cálculo de la posición después del evento Transaction.

3.1.1.2 PROCESO DE ORACLE

- 2) **Datos ingeridos.** Los datos se ingestan con instrucciones de inserción o actualización, en función de los requisitos de cada entidad.
- 3) **Triggers:** las tablas ingestadas tienen triggers (disparadores) con puntos de disparo anteriores y posteriores a la inserción de cada fila. Los datos se distribuyen a las tablas relacionadas. En este punto, podemos vincular todas las tablas con el modelo de datos para cada entidad.

Evaluación de NRT-DWH

3.1.2 PROBLEMAS DETECTADOS

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DI01	Lector de mensajes	Lector de mensajes: decisión del enrutamiento de la ubicación (implementación de código duplicado)	Cuando el mensaje se lee desde la cola MQ, el código fuente se divide en dos flujos de trabajo diferentes con una lógica similar. El primer flujo (QueueReaderMDB) se aplica a TE, EA, TO (ubicaciones enrutadas), el segundo (Message Reader) se aplica al resto de los casos, pero no hay	Código duplicado. Difícil de mantener. Alta posibilidad de perder la funcionalidad y también un alto esfuerzo de prueba debido a los dos flujos utilizados.	Alto
DI02	Lector de mensajes	Análisis/modelado XML: DTD versionado (muchas versiones extendidas)	Los DTD se versionan para las diferentes entidades y se necesita una clase Java para implementar el controlador de mensajes para cada versión DTD. es decir: TransPending_1_8.dtd, TransPending_2_0.dtd, etc. En caso de que se necesite un cambio, el esfuerzo de desarrollo es alto para ser implementado.	Afecta a la capacidad de mantenimiento, ya que se necesitan demasiados cambios para una nueva funcionalidad. Sujeto a incompatibilidades entre versiones. Falta de estructura empresarial porque el modelo no está centralizado. Las entradas de validación de datos retrasan el control a capas más profundas.	Alto
DI03	Acceso a datos/Capa de negocio	El acceso a datos y las capas de lógica empresarial se mezclan en el procesador de mensajes.	La lógica de negocios y el acceso a datos se implementan en la misma clase Java para cada controlador de mensajes	Aporta confusión en términos de mantenimiento y escalabilidad del código fuente, porque la lógica de negocios y los procesos de acceso se mezclan en las mismas partes del código fuente.	Alto

Evaluación de NRT-DWH

DI04	Acceso a datos	Instrucciones de ingestión SQL sin instrucción preparada	La creación de instrucciones SQL dinámicamente desde los controladores de mensajes con cadenas evita el análisis de Oracle, ya que las variables de enlace no se utilizan y deben analizar cada vez que se ejecutan consultas con JDBC. Este es un problema para el rendimiento. En el código actual, las instrucciones INSERT y UPDATE se crean consultando el diccionario de datos y realizando sustituciones para cada valor sin parámetros.	La seguridad está comprometida. Fácil de producir errores de calidad de datos y aplicaciones. El rendimiento también es un problema, porque la instrucción SQL se genera dinámicamente.	Muy Alto
DI05	Acceso a datos	Operaciones innecesarias de acceso a bases de datos	A partir de los objetos de acceso a datos Java, algunos métodos como generateInsertUpdate podrían ser más eficientes. Ejecutan varias consultas para decidir si se necesita una operación INSERT o UPDATE para realizar la ingestión.	El rendimiento es menor.	Medio
DI06	Base	Triggers ejecutados durante la ingestión	Hay varios triggers con puntos de disparo anteriores y posteriores a la inserción de cada fila en la tabla de base de datos. Aparentemente, algunos triggers contienen lógica de negocios y podrían retrasar la transacción confirmada por Oracle.	Mantenimiento, porque es difícil seguir el flujo de datos. Las buenas prácticas recomiendan evitar varios triggers del mismo tipo	Medio

Evaluación de NRT-DWH

3.1.3 MEJORAS PROPUESTAS

Esta primera propuesta se basa en el estado actual y en cómo evolucionar NRT-DWH gradualmente para mejorar la escalabilidad, el mantenimiento y el rendimiento. Este enfoque supone que la aplicación se hospeda en un servidor Weblogic y que un framework controlará la aplicación. Los servicios para la ingestión son transaccionales.

Problema #	Resumen del Problema	Solución Propuesta	Coste
DI01	Lector de mensajes: Decisión del enrutamiento de la ubicación	Después de analizar el código fuente, aparentemente la decisión del lector de mensajes no es necesaria, porque la lógica implementada para ambos flujos de trabajo no tiene diferencias significativas. Si simplificamos el flujo, no se necesitan clases Java duplicadas, lo que mejora la capacidad de mantenimiento, el rendimiento y la facilidad de cambiar el código para obtener más requisitos empresariales. Este cambio solo se puede implementar si la capa Dete Message Reader, Capa de negocio y Acceso a datos se combinan en un flujo.	Muy Alto
DI02	Análisis XML: DTD versionado	Las versiones actuales de DTD se migrarían a un modelo jerárquico y se implementarían mediante un único esquema XSD. Las clases Java serán generadas por las bibliotecas JAXB. Esta jerarquía incluirá las diferentes entidades como Transacciones, Acciones Corporativas, Saldo de Efectivo, Datos Maestros. Los mensajes XML se validarán con el modelo y se tratarán en caso de excepciones de esquema.	Muy Alto
DI03	Se mezclan el acceso a datos y la lógica empresarial.	Crear una capa de negocio para las diferentes entidades y clases de acceso a datos que utilizará la capa empresarial. Solo las operaciones SQL se colocan en la capa de acceso a datos. La lógica de negocios debe separarse en entidades funcionales. Por lo tanto, se tiene un componente diferente de lógica de negocios para cada entidad, como Transacción, Acción Corporativa, etc.	Alto
DI04	Instrucciones de ingestión SQL sin instrucción preparada	Implementar una asignación desde el objeto resultante después de analizar el XML (clases Java –JAXB) al registro final que se ingirió en la base de datos. En caso de usar un marco ORM, es posible que se agreguen claves subrogadas a las tablas ingestadas.	Alto
DI05	Operaciones innecesarias de acceso a bases de datos	Evitar los métodos manuales de acceso a datos e intentar utilizar un marco ORM como Hibernate por ejemplo. La ingeniería inversa es necesaria para obtener los objetos que ingestan los datos, generando instrucciones de inserción o actualización.	Alto
DI06	Triggers ejecutados durante la ingestión	Determinar si tiene sentido ejecutar varios triggers para la misma tabla. Determinar si realmente es lógica de negocios o cálculo de tablas derivadas y mover sólo la lógica de negocios a la capa de aplicación.	Medio

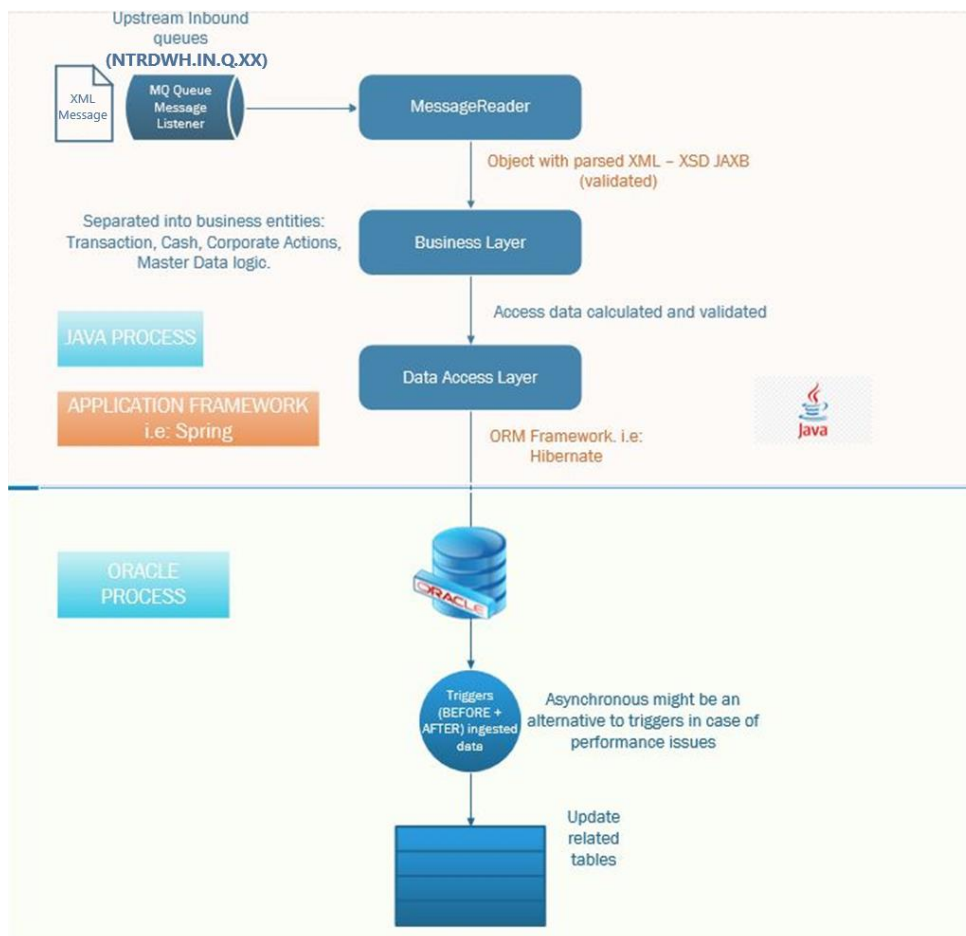


Figura 3. Ingestión de datos NRT-DWH – Estado futuro. Propuesta 1.

3.1.3.1 PROCESO JAVA

1) Escucha de mensajes de cola de MQ

El agente de escucha de mensajes se implementará con el marco de trabajo de aplicación seleccionado.

2) Lector de mensajes.

La decisión de ruta para los sitios TE, EA, TO está agregando una complejidad que no debería ser necesaria. Después de analizar el código en el estado actual, parece que ambos flujos están intentando realizar las mismas operaciones. En caso de que algo sea un poco diferente se puede personalizar, pero debe ser una razón para dividir en dos flujos diferentes.

El mensaje se analiza y se valida con una biblioteca de análisis XML como JAXB (basada en XSD). Se generarán las clases Java y se puede controlar la estructura jerárquica con el recurso XSD. Se mantienen muchas versiones en el estado actual, y es cierto que proporciona cierta flexibilidad para ingerir versiones diferentes, pero deben ser compatibles. De lo contrario NRT-DWH podría ingerir datos sin control.

3) Capa de negocio.

La capa de negocio se implementará con las diferentes entidades, organizando toda la lógica en entidades empresariales como Transacción, CashBalance, Acciones Corporativas, Datos Maestros, etc. La idea es que el objeto recibido después del análisis se transformará en el objeto que la base de datos puede comprender para ser ingerido.

Evaluación de NRT-DWH

4) Capa de acceso a datos.

Aprovechando un marco ORM (asignación relacional de objetos), NRT-DWH puede ingerir datos asignando los objetos con las tablas de base de datos. Se mejorará el rendimiento (las consultas actuales han sido modificadas por varios equipos en su ciclo de vida y en muchos casos contienen código poco optimizado) y la escalabilidad y la facilidad de implementar nuevos cambios en el modelo se controlará en un único punto, no codificando los tiempos de servidor de lógica de negocios.

3.1.3.2 PROCESO DE ORACLE

5) Triggers

Los triggers se deben simplificar y optimizar en relación a las necesidades actuales para mejorar el rendimiento, puesto que hay algunos que son obsoletos. Una alternativa a este paso podría ser también implementar esta parte con una solución asíncrona en caso de que el rendimiento se convierta en un problema.

3.2 FLUJOS DE DATOS

3.2.1 SITUACIÓN ACTUAL

Los flujos de datos descritos en las secciones siguientes se representan de la siguiente manera para los volúmenes de datos manejados por el proceso para el intervalo 4-septiembre-2018 al 4-octubre-2018:

Categoría	El éxito total	Total fallido	Reproceso total	Total por día
Posición de efectivo	27878.67	11.97	0	27890.63
Posición	60392.43	0.03	0	60392.47
Acción Corporativa	75453.47	13.43	0	75466.9
Transacción en efectivo	82713	11.47	0	82724.47
Transacción	418955.17	5131.4	46.13	424132.7
Datos de referencia	434346.07	153.83	1.1	434501
Total	1135542.77	5326.17	47.23	1140916.17

El problema más importante son los eventos de transacción con error. Precisión del 99,95% de los eventos sucedidos.

3.2.1.1 EVENTO DE TRANSACCIÓN

El objetivo en las secciones de transacciones es describir cómo funciona la aplicación para esta entidad y obtener una imagen completa de los principales puntos de evolución. El procedimiento para lograr esta descripción será el siguiente:

- 1) **Evento de Transacción genérica:** imagen completa de un evento de transacción en general.
- 2) **Eventos de Transacciones específicas:** descripción de los diferentes eventos de transacción y las características particulares.
- 3) **Ciclo de vida comercial:** posibles transiciones en el estado de la transacción y el marco de control
- 4) **Volúmenes de datos:** estadísticas de eventos de transacción, que proporcionarán evidencia de las lagunas.

3.2.1.1.1 EVENTO DE TRANSACCIÓN GENÉRICA

Después de analizar los eventos de transacción, podemos extraer una idea genérica basada en las acciones realizadas con respecto a los flujos de datos. Este es el estado actual de la transacción en NRT-DWH y nos permitirá evolucionar la aplicación en los flujos de datos.

Evaluación de NRT-DWH

En la siguiente tabla podemos ver la siguiente información:

Estadísticas de transacciones genéricas: promedios por día para cada estado del mensaje. Esto pondrá en contexto la calidad de los datos y cómo afectará al cálculo final. Los altos porcentajes de transacciones fallidas y rechazadas producirán incoherencias entre los datos proporcionados por los sistemas ascendentes y los datos que NRT-DWH almacena realmente. Cuanto mayor sea el número de mensajes que no se realicen correctamente, mayor error de cálculo producirá en las entidades de transacción y posición.

Fases de eventos de transacción:

- **Ingestión:** las tablas principales rellenas durante la ingestión de datos. En este caso, TRA y TRG son las tablas principales para almacenar el estado actual de la transacción, mientras que TEG y TEV representan la auditoría y diferentes mensajes recibidos para cada transacción.
- **Tablas actualizadas por triggers:** estas son las tablas "satélite" actualizadas por eventos de transacción. Podemos considerar estas tablas derivadas como resultado de los eventos Transaction.
- **Postprocesamiento:** después de la ingestión y los triggers calculados, se necesitan algunas operaciones para alinear la información de posición según la nueva transacción y normalmente se ejecutan con procedimientos almacenados. Las unidades negociadas/liquidadas para la transacción se añadirán/restarán de la posición global.

TRANSACCIÓN GENÉRICA					
Estadísticas genéricas	Día medio total	Porcentaje sobre el total de mensajes	Media de éxitos Diaria	Media de Rechazos Diaria	Media Fallos Diaria
	363282	31.84 %	358104 (98.57 %)	46 (0.013%)	5131 (1.41 %)
Descripción del negocio	Eventos que definen el ciclo de vida de una operación. Los eventos de transacción de seguridad tienen un efecto inmediato en la posición intradía.				
Ingestión	Inserte en TRA/TRG (dependiendo de GrossNetFlag) o actualice si la transacción ya existe. Inserte eventos de transacción en TEV/TEG para realizar un seguimiento de las diferentes operaciones en el ciclo de vida de la transacción.				
Triggers: tablas actualizadas	CASH_TRANSACTIONS (Transacciones en efectivo), PRI (Precios de seguridad). MSG_CACHE (solo en eventos TransPending y TransDBRef)				
Post-procesamiento	Si se selecciona la tabla TRA (GrossNetFlag diferente a G), actualiza POS/POB/PSI con NRT_DWH.PA_PSI_CALC. P_UPDATE_POS_POB para posiciones negociadas o liquidadas, dependiendo del tipo de evento				

Si una transacción no se procesa correctamente, afecta a la posición intradía. Los eventos recibidos durante el proceso de "Fin del día" reconciliarán las posiciones y calcularán si hay diferencias con el proceso intradía. RECON_POS almacena todas estas diferencias y diagnostica si la causa raíz fue originada por una brecha en NRT-DWH o el sistema ascendente no envió ninguna transacción relacionada a la posición. Al mejorar el proceso intradía tendremos mejores informes en línea para los sistemas posteriores.

Evaluación de NRT-DWH

3.2.1.1.2 DETALLES DE LA TRANSACCIÓN: EVENTOS ESPECÍFICOS

En la tabla siguiente se describen las características específicas de cada evento. El primer paso inserta/actualiza la tabla TRA, lo que significa que realmente siguen la ingestión de transaction. El segundo paso inserta/actualiza los eventos de transacción que no ingestan TRA/TEV, sino tablas adicionales para transacciones.

Estas estadísticas se han recuperado durante 1 mes (septiembre – octubre 2018) de producción.

Evento: Nombre del tipo de mensaje.

DTD/Message Handler: diferentes versiones utilizadas actualmente por los Upstreams. Solo unos pocos tipos de mensajes utilizan varias versiones simultáneamente.

Ubicaciones emisoras: hay varias posibilidades para este atributo:

- ALL: todas las ubicaciones envían mensajes para el tipo de mensaje especificado
- TE, EA, TO : ubicaciones enrutadas
- NOT TE, EA, TO: ubicaciones no enrutadas (el prefijo de tipo de mensaje UX indica este flujo)
- <ubicación específica>: sólo unas pocas ubicaciones están enviando los eventos
- NADA: el controlador de mensajes existe, pero no se recibió ningún evento durante el último mes.

Características específicas: esta columna describe las operaciones especiales del tipo de mensaje. Algunos de ellos están vacíos, porque siguen la estructura del evento de transacción genérico.

Evento	DTD/Controlador de mensajes	Ubicaciones de remitente	Características específicas
SWIFTRepair	SWIFTRepair_0_5, 0_6	TODO	También SWIFT_REPAIR ingesta. No hay evento grabado en TEV/TEG. Comprobar si se trata de un problema.
TransAutoSett	TransAutoSett_2_2	TE, EA, TO	
TransAutoSettUX	TransAutoSettUX_1_1	NO TE, EA, TO	updatePosPob no se ejecuta para el flujo de ubicaciones enrutadas (conveniente - TE, EA, TO). Esto podría ser un problema (el evento no se utiliza actualmente en PROD)
TransCancel	TransCancel_1_7	TE, EA, TO	updatePosPob es llamado varias veces dependiendo del estado de la transacción anterior a la cancellationMessageHandlers del flujo de sitios no enrutados (controladores XML - NOT TE, EA, TO) también actualiza las posiciones incluso con GrossNetFlag. Los enrutadores de las ubicaciones solo son llamados para TRA. Esto podría ser un problema. (el evento no se utiliza actualmente)
TransCancelUX	TransCancelUX_1_1	NO TE, EA, TO	
TransDBCRef	TransDBCRef_0_7	TE	Ingerir MSG_CACHE. No hay evento grabado en TEV/TEG. Comprobar si se trata de un problema.
TransDirectSettleFree	TransDirectSettleFree_2_1	TE , EO	Este evento puede ejecutar varias actualizaciones de posición y transacción en función de las etiquetas presentes en el XML (Detalles, Liquidación, Cancelación)
TransDirectSettlePay	TransDirectSettlePay_2_0	TE	Este evento puede ejecutar varias actualizaciones de posición y transacción dependiendo de las etiquetas presentes en el XML (Detalles, Pago, Liquidación, Cancelación)

Evaluación de NRT-DWH

TransDispoStatus	TransDispoStatus_0_8	TE	No hay evento grabado en TEV/TEG. Comprobar si se trata de un problema
TransExFlag	Nada	NADA	Sólo actualiza ExFlag ("CREX", "DBEX" y "SPEX") en TRA. Ningún evento grabado en TEV
TransFailed	TransFailed_1_3	TODO	
TransMatch	TransMatch_0_6	TE	Según la etiqueta XML SettlementLock, controla los registros de la tabla IHM. Para todos los casos, NRTDWH. El procedimiento almacenado IHMatching actualiza TRA. Esto significa que la ingestión de TRA se retrasa al postprocesamiento después de la ingestión de IHM. Para los sitios no enrutados, el flujo (controladores xml - NOT TE, EA, TO) TRA no se actualiza. Solo se ingesta IHM. Comprobar si se trata de un problema
TransPending	TransPending 2_2, 2_3, 2_5	TODO	Las ubicaciones enrutadas (TE, EA, TO) se actualizan MSG_CACHE. Este no es el caso de las ubicaciones no enrutadas. Comprobar si se trata de un problema
TransSettled	TransSettled 1.7 y 1.8	TE, TG, A, EA	
TransSettledUX	TransSettledUX_1_1	NO TE, EA, TO	
TransTradeConf	TransTradeConf_2_2	TE	Este evento puede ejecutar varias actualizaciones de posición y transacción dependiendo de las etiquetas presentes en el XML (Details, Payment, Settlement, Cancellation)
TransTransfer	TransTransfer_1_6	TE, PARA	
TransTransferUX	TransTransferUX_1_1	NO TE, EA, TO	
UpdatePending	Nada	TODO	Este evento es similar a TransPending. No hay evento grabado en TEV/TEG. Comprobar si se trata de un problema (el evento no se utiliza actualmente)

Eventos de transacción adicionales que no actualizan las tablas de transacción/posición

Evento	DTD/Controlador mensajes	de Ubicaciones remitente	de Características específicas
TransAddReferences	TransAddReferences_0_1	TE, TD	Ingesta TRR (referencias de transacción de mercado)
TransClientIndicators	TransClientIndicators_1_0	TE	Ingesta TAD (indicadores de cliente)
Transbeneficiary	TransBeneficiary_0_6	TO	Ingesta TRB (información del beneficiario de la transacción)
Translink	TransLink_0_5	NO TE, EA, TO	Ingesta TRL (Información de TradeCrossReference (archivo CIDF))
TransFX	TransFX_0_8	NO TE, EA, TO	Ingesta FXT (transacciones FX)

Evaluación de NRT-DWH

3.2.1.1.3 CICLO DE VIDA COMERCIAL

En la tabla siguiente se describe el estado de transición diferente en los eventos Transaction. Estos eventos actualizan **TRA_TRANS_STAT** con las siguientes opciones:

- **Pending (P)**: transacción iniciada pero no liquidada.
- **Partially Settled (PS)**: las unidades liquidadas son unidades inferiores que se negocian
- **Settled (S)**: transacción totalmente liquidada.
- **Cancelled (C)**: transacción cancelada por el cliente.
- **Failed (F)**: no ha llegado ninguna transacción de liquidación, pero se supera la fecha de liquidación.
- **Transferred (T)**: unidades movidas de una ubicación a otra.

Todos estos eventos se auditan en la tabla **TEV**. Por lo tanto, debe haber un registro diferente para cada transición especificada en TEV. TEV_STATUS y el número de secuencia de eventos único TEV. TEV_NO indicará el orden de los eventos de la transacción.

En la tabla, podemos ver las diferentes opciones que se pueden resumir como:

- **Verde**: posible transición y uso funcional. Varios tipos de mensajes pueden lograr la misma transición. Se enumeran en la cuadrícula.
- **RCM Status = Failed. ERROR**: NRT-DWH proporciona un ERROR y marca el mensaje como FAILED en la tabla RCM.
- **RCM Status = Reprocess N/A (functionally) Sent to reprocess**: esta transición no se aplica funcionalmente, pero NRT-DWH envía el mensaje que se va a volver a procesar (marcado como REPROCESS en la tabla RCM). Esto significa que se deben procesar otros eventos relacionados con la transacción antes de éste. Hay un reprocesador en línea para volver a intentar los eventos periódicamente (cada 10 segundos). Ejemplo: TransFailed llegó antes de la TransPending inicial.
- **N/A (functionally)**: esta transición no tiene sentido funcional. NRT-DWH anulará el estado de la transacción, pero debería ser el caso. Estas transiciones deben tener algún tipo de gestión de control.
- **N/A (same status)**: la transición es para el mismo status, se inserta un nuevo registro en la tabla TEV y los datos también se actualizan.

Nota: El eje X representa el estado final. El eje Y representa el estado antiguo de la transacción

Evaluación de NRT-DWH

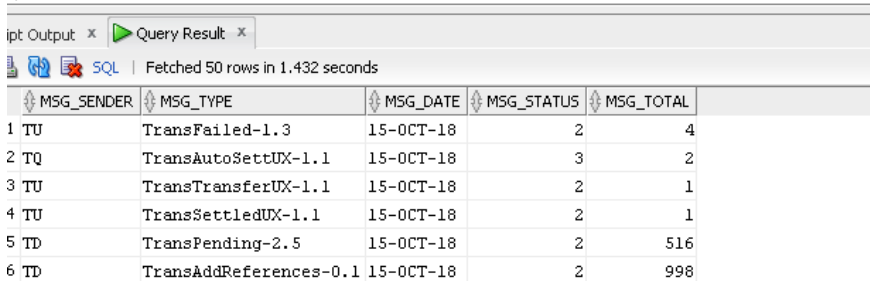
	PENDING	PARTIALLY SETTLED	SETTLED	FAILED	CANCELLED	TRANSFERRED
(Initial)	TransPending	RCM Status = Reprocess N/A (functionally) . Transaction does not exist . Sent to Reprocess	TransAutoSett TransTradeConf (Settlement) TransDirectSettleFree (Settlement) TransDirectSettlePay (Settlement)	RCM Status = Reprocess N/A (functionally). Transaction does not exist . Sent to Reprocess	RCM Status = Reprocess N/A (functionally). Transaction does not exist .Sent to Reprocess	TransTransfer
P	TransPending UpdatePending	TransDispoStatus	TransAutoSett TransSettled TransTradeConf (Settlement) TransDirectSettleFree (Settlement) TransDirectSettlePay (Settlement) TransDispoStatus	EOD (settlement Date exceeded) TransFailed	TransDirectSettleFree (Cancellation) TransDirectSettlePay (Cancellation) TransTradeConf (Cancellation) TransCancel	N/A (functionally)
PS	RCM Status = Failed. ERROR. Transaction is not in pending status	N/A (same status)	TransAutoSett TransSettled TransTradeConf (Settlement) TransDirectSettleFree (Settlement) TransDirectSettlePay (Settlement) TransDispoStatus	EOD (settlement Date exceeded) TransFailed	TransDirectSettleFree (Cancellation) TransDirectSettlePay (Cancellation) TransTradeConf (Cancellation) TransCancel	N/A (functionally)
S	RCM Status = Failed. ERROR Transaction is not in pending status	TransDispoStatus	N/A (same status)	RCM Status = Failed ERROR Failed update of a Settled/Cancelled transaction is not in Pending status	TransDirectSettleFree (Cancellation) TransDirectSettlePay (Cancellation) TransTradeConf (Cancellation) TransCancel	N/A (functionally)
F	TransPending UpdatePending	TransDispoStatus	TransAutoSett TransSettled TransTradeConf (Settlement) TransDirectSettleFree (Settlement) TransDirectSettlePay (Settlement) TransDispoStatus	N/A (same status)	TransDirectSettleFree (Cancellation) TransDirectSettlePay (Cancellation) TransTradeConf (Cancellation) TransCancel	N/A (functionally)
C	RCM Status = Failed. ERROR Transaction is not in pending status	RCM Status = Failed. ERROR Failed Settlement. The transaction is cancelled	RCM Status = Failed. ERROR Failed Settlement. The transaction is cancelled	RCM Status = Failed. ERROR. Failed update of a Settled/Cancelled transaction is not in Pending status	N/A (same status). RCM Status = Failed. ERROR. Transaction already canceled	N/A (functionally). Transaction is already cancelled
T	N/A (functionally)	N/A (functionally)	N/A (functionally)	N/A (functionally)	N/A (functionally)	N/A (same status)

3.2.1.1.4 VOLÚMENES DE DATOS DE TRANSACCIONES POR ESTADO

Los volúmenes de datos de transacción proporcionarán las cifras de eventos de transacción y el detalle sobre los problemas significativos

MSG_STATISTICS tabla contiene los volúmenes de datos por tipo de mensaje y estado. Aquí tenemos un extracto para saber cómo se ve.

```
select * from cris.msg_statistics where msg_type like 'Trans%' order by msg_date desc
```



MSG_SENDER	MSG_TYPE	MSG_DATE	MSG_STATUS	MSG_TOTAL
1 TU	TransFailed-1.3	15-OCT-18	2	4
2 TQ	TransAutoSettUX-1.1	15-OCT-18	3	2
3 TU	TransTransferUX-1.1	15-OCT-18	2	1
4 TU	TransSettledUX-1.1	15-OCT-18	2	1
5 TD	TransPending-2.5	15-OCT-18	2	516
6 TD	TransAddReferences-0.1	15-OCT-18	2	998

Las estadísticas de mensajes extraídas de PROD el 4 de octubre^{de} 2018 durante los últimos 30 días. Estos volúmenes donde se cargan en una tabla diferente para administrar solo los eventos Transaction y, finalmente, la consulta se construyó para recuperar todas las estadísticas.

Aquí podemos encontrar la consulta que recupera los volúmenes de Transacción y organizar la información que se mostrará en la cuadrícula de Excel a continuación.

```
select msg_type, "Average message type per day", nvl(round(100*"Average message type per day" / "Total Transactions per Day",2),0) "% over Total Transaction",  
"Total Success",round(100*"Total Success" / "Average message type per day",2) "% Success",  
nvl("Total Failed",0) "Total Failed",  
nvl(round(100*"Total Failed" / "Average message type per day",2),0) "% Failed",  
nvl("Total Reprocess",0) "Total Reprocess",  
nvl(round(100*"Total Reprocess" / "Average message type per day",2),0) "% Reprocess"  
from (  
select msg_type , decode(msg_status,2,'SUCCESS', 3, 'FAILED', 4, 'REPROCESS') msg_status,  
round(sum(total) over (partition by msg_type order by msg_type)/30) "Average message type per day" ,  
round(sum(total) over(),/30) "Total Transactions per Day",  
nvl(round(total/30),0) "Avg Status per Message Type"  
from stats_test_tran  
)  
pivot(  
sum("Avg Status per Message Type") for msg_status in ('SUCCESS' as "Total Success", 'FAILED' as "Total Failed", 'REPROCESS' as "Total Reprocess")  
)  
order by 2 desc;
```

La cuadrícula es editable con Excel, y podemos filtrar en consecuencia. La descripción de cada columna es la siguiente:

- **Tipo de mensaje:** tipo de mensaje de transacción.
- **Tipo de mensaje promedio por día: 424K transacciones** en total por día. Los volúmenes se calcularon durante 30 días, y esto es un promedio por día.
- **% sobre Transacciones totales:** tipo de mensaje promedio por día / transacción total por día
- **Total de éxito y % de éxito:** total de filas correctas por día y el porcentaje sobre los mensajes recibidos para este tipo de mensaje.

Evaluación de NRT-DWH

- **Total de errores y % de error:** total de filas con errores por día y el porcentaje sobre el total de mensajes recibidos para este tipo.
- **Reproceso total y % reprocesar:** número de filas que están en estado de reproceso y porcentaje sobre los mensajes recibidos para este tipo. Esto debe ser cercano a 0, porque estos mensajes no se volverán a procesar una vez que se cambie la fecha de negocio.

En rojo, podemos encontrar las cifras que evidencian las cuestiones que deben ser analizadas. En las secciones Problemas detectados y Mejoras propuestas encontraremos referencias a estos números.

Message Type	Average message		% over Total							
	type	per day	Transaction	Total Success	% Success	Total Failed	% Failed	Total Reprocess	% Reprocess	
TransFailed		113525	26,77	111762	98,45	1721	1,52	42	0,04	
TransPending		109333	25,78	108226	98,99	1106	1,01	0	0	
TransAddReferences		59556	14,04	59556	100	0	0	0	0	
TransSettled		54072	12,75	52333	96,78	1737	3,21	2	0	
TransTransferUX		24222	5,71	24221	100	0	0	0	0	
TransSettledUX		21797	5,14	21797	100	0	0	0	0	
TransClientIndicators		16590	3,91	16590	100	0	0	0	0	
TransDispoStatus		7179	1,69	7174	99,93	4	0,06	2	0,03	
TransMatch		7007	1,65	7007	100	0	0	0	0	
TransTradeConf		2721	0,64	2250	82,69	472	17,35	0	0	
SWIFTRepair		2087	0,49	2020	96,79	68	3,26	0	0	
TransCancel		1730	0,41	1729	99,94	1	0,06	0	0	
TransDBCRef		1274	0,3	1274	100	0	0	0	0	
TransCancelLUX		1041	0,25	1041	100	0	0	0	0	
TransBeneficiary		661	0,16	661	100	0	0	0	0	
TransLink		575	0,14	575	100	0	0	0	0	
TransAutoSettUX		289	0,07	289	100	0	0	0	0	
TransAutoSett		284	0,07	280	98,59	4	1,41	0	0	
TransDirectSettlePay		59	0,01	43	72,88	16	27,12	0	0	
TransFX		59	0,01	59	100	0	0	0	0	
TransDirectSettleFree		50	0,01	47	94	3	6	0	0	
TransTransfer		22	0,01	22	100	0	0	0	0	
Total		424133		418956		5132		46		

3.2.1.2 POSICIÓN

Los eventos de posiciones provienen de cada sitio durante el proceso de fin del día. Esta es una ventana de tiempo representada entre la llegada del mensaje "Fin del inicio del día" y el mensaje "Fin del día final". El sistema ascendente enviará todos los eventos Position (Position_0_8.dtd versión en producción) para reconciliar lo que NRT-DWH calculó durante el proceso intradía de transacción con las posiciones que la upstream espera tener para abrir la nueva fecha de negocio.

En caso de que NRT-DWH no tenga las unidades negociadas o liquidadas calculadas correctamente, la posición final serán los valores representados por el sistema ascendente. Todos los huecos detectados se registrarán en la RECON_POS tabla, como podemos ver en la siguiente tabla.

Evaluación de NRT-DWH

3.2.1.2.1 POSICIÓN GENÉRICA

POSICIÓN GENÉRICA					
Estadísticas genéricas	Día medio total	Porcentaje sobre el total de mensajes	Media de éxitos Diaria	Media de Rechazos Diaria	Media Fallos Diaria
	60392	5.29 %	60392 (100 %)	0 (0%)	0 (0 %)
Descripción del negocio	Eventos que definen la seguridad liquidada o las unidades negociadas por una cuenta identificada, seguridad, ubicación y a partir de la fecha.				
Ingestión Tablas temporales	Insertar en POS_TEMP (TE, TO) o insertar en TMP_POS (resto de sitios). La posición viene en fragmentos en los mensajes. Cada mensaje puede contener varias filas de posición.				
triggers	N/A				
Post-procesamiento	REGISTROS DE RECONCILIACIÓN	Insertar en RECON_POS todos los descansos borrados durante el proceso de fin del día. 2 tipos: src_type evento de posición 'XML' recibido: NRT-DWH calculado incorrectamente intraday src_type : 'EOD' Ningún evento de posición recibido de la conexión en sentido ascendente y POS.pos_units es diferente de POS.pos_cur_units			
	FIN DEL DÍA (por sitio)	TE, PARA	La posición se repara durante el proceso final de EOD: PA_PSI_CALC.p_update_pos_pob_eod_te_to: recalcula los registros POS, POB y PSI		
		Resto de Ubicaciones	La posición se repara en línea , cuando se procesa el mensaje de posición. EA (EIS Amsterdam) se procesa en línea debido a problemas de rendimiento, en lugar de esperar al proceso de finalización de EOD. PA_PSI_CALC.p_update_pos_pob_eod: recalcula los registros POS, POB y PSI para cada TMP_POS.		

La principal diferencia entre TE, TO con el resto de las ubicaciones es que TE, TO esperan hasta el proceso "Fin del día final" para reconciliar las posiciones, mientras que el resto de las ubicaciones insertan los fragmentos de datos (varias posiciones en un mensaje de posición) en una tabla temporal y vuelve a calcular los registros de posiciones en línea para cada mensaje. Una vez completado el mensaje, la tabla temporal se elimina con la transacción confirmada.

Evaluación de NRT-DWH

3.2.1.2.2 VOLÚMENES DE DATOS CON REGISTROS DE POSICIÓN RECONCILIADOS

En la tabla siguiente, podemos ver las **estadísticas para los registros reconciliados** en PDV (todos los registros de posición son correctos). Las filas reconciliadas sólo se encuentran en TE (Trader Alemania), TQ(Trader Istanbul). Tarifas bajas para TD (Trader España) y EA (EIS Amsterdam). La población total se reconcilió un **1,159%**, pero casi todos los desajustes pertenecen a TE (**85 %**)

Site	Position rows	Avg day Position rows	Position message	Avg day Position message	Reconciliati on Rows	Avg day reconciliati on rows	% Recon
TE	45409	1513.63	4549	151.63	16241	541.37	35.766
TQ	638	21.27	438	14.60	30	1.00	4.702
TD	35359	1178.63	7273	242.43	192	6.40	0.543
EA	1549939	51664.63	1549939	51664.63	2568	85.60	0.166
TL	152	5.07	65	2.17	0	0.00	0
TY	824	27.47	518	17.27	0	0.00	0
TH	403	13.43	114	3.80	0	0.00	0
TB	44	1.47	21	0.70	0	0.00	0
TI	2172	72.40	1593	53.10	0	0.00	0
TX	18	0.60	10	0.33	0	0.00	0
T9	351	11.70	194	6.47	0	0.00	0
TG	247	8.23	65	2.17	0	0.00	0
TT	130	4.33	63	2.10	0	0.00	0
TU	274	9.13	122	4.07	0	0.00	0
TK	762	25.40	355	11.83	0	0.00	0
TM	53	1.77	13	0.43	0	0.00	0
T5	380	12.67	211	7.03	0	0.00	0
TC	191	6.37	110	3.67	0	0.00	0
TJ	128	4.27	73	2.43	0	0.00	0
TS	4246	141.53	2013	67.10	0	0.00	0
Total		54724		52257.97		634.37	1.159

En la siguiente tabla, podemos ver los volúmenes por mes para las discordancias agrupadas por "Tipo" de causa raíz:

EOD: El proceso de fin de día reconcilió las filas sin el evento Position enviado por upstream, por lo que la discordancia es causada por el Upstream.

XML: Error de cálculo de NRT-DWH durante el proceso intradía.

Site	Type	Pos Rows	Messages	Recon	Total Rows	Total Recon	% Rec
EA	EOD	1085043	1085043.00	2491	1498351.00	2568	97.00
EA	XML	413308	413308.00	77	1498351.00	2568	3.00
TD	EOD	5075	1186.00	20	28770.00	192	10.42
TD	XML	23695	4843.00	172	28770.00	192	89.58
TE	EOD	19982	2003.00	15833	32263.00	16241	97.49
TE	XML	12281	1230.00	408	32263.00	16241	2.51
TQ	EOD	181	126.00	5	465.00	30	16.67
TQ	XML	284	197.00	25	465.00	30	83.33

Las discordancias de EA y TE provienen de Upstream con un total de **18324 filas al mes**(media por día 610)

Las discordancias de TD y TQ provienen del cálculo de NRT-DWH con un total de **197 filas al mes** (media por día 3,28)

Podemos concluir que los desajustes **provienen esencialmente de Downstreams**.

Evaluación de NRT-DWH

3.2.1.3 TRANSACCIÓN DE EFECTIVO

3.2.1.3.1 TRANSACCIÓN GENÉRICA DE EFECTIVO

En la tabla siguiente se muestran las características genéricas para un evento de transacción en efectivo:

TRANSACCIÓN DE EFECTIVO GENÉRICA						
Estadísticas genéricas		Día medio total	Porcentaje sobre el total de mensajes	Media de éxitos Diaria	Media de Rechazos Diaria	Media Fallos Diaria
		82724	7.25 %	82713(99.95 %)	0 (0%)	12 (0.05 %)
Descripción del negocio	Eventos que definen los movimientos en saldos de efectivo identificados por un número de transacción por fecha de valor o fecha contable					
Ingestión	Insertar en CTX					
desencadenantes	CASH_TRANSACTIONS (Transacciones en efectivo)					
Post-procesamiento	Llamar a NRTDWH.PA_UTILS. P_UPDATE_BAL_FCA actualizar BAL (Saldo de efectivo) y FCA (Pronóstico de efectivo)					

3.2.1.3.2 DETALLES DE LA TRANSACCIÓN DE EFECTIVO: EVENTOS ESPECÍFICOS

Los eventos específicos de la transacción en efectivo se detallan de la siguiente manera:

Evento	DTD/Controlador de mensajes	de	Ubicaciones de remitente	de	Características específicas
CashTransAutoSett	CashTransAutoSett_0_5		NO TE, EA, PARA		
CashTransaction	CashTransaction_1_0 y 1_3		NO TE, EA, PARA		
SwiftCashAutoSettle	SwiftCashAutoSettle_0_5		NO TE, EA, PARA		No hay actualización en BAL/FCA. Sólo la ingestión de CTX
SwiftIntradayTransaction	SwiftIntradayTransaction_0_5 y 0_6		NO TE, EA, PARA		Ingesta CTI(Cash Transaction Intraday) en lugar de CTX. Actualiza BAL.bal_current, pero no FCA

Evaluación de NRT-DWH

3.2.1.3.3 VOLÚMENES DE DATOS DE TRANSACCIONES EN EFECTIVO

Los volúmenes de producción muestran que la precisión es cercana al 100% de los eventos correctos para las transacciones en efectivo.

Message Type	Average		% over		Total		Cash		Transacti	
	message	type per	Total	% Succ	Total Fa	% Faile	Total R	% Repr		
	day	on	Total St	% Succ	Total Fa	% Faile	Total R	% Repr		
CashTransAutoSett	41688	50.39	41688	100	0	0	0	0		
CashTransaction	26196	31.67	26196	100	0	0	0	0		
SwiftCashAutoSettle	9878	11.94	9875	99.97	3	0.03	0	0		
SwiftIntradayTransaction	4962	6	4954	99.84	9	0.18	0	0		
Total	82724		82713	99.9525	12	0.0525				

3.2.1.4 POSICIÓN DE EFECTIVO

3.2.1.4.1 POSICIÓN DE EFECTIVO GENÉRICA

Durante el proceso de fin de día, los eventos de posición de efectivo se envían desde el Upstream de la siguiente manera:

POSICIÓN DE EFECTIVO GENÉRICA							
Estadísticas genéricas		Día total	medio	Porcentaje sobre el total de mensajes	Media de éxitos Diaria	Media de Rechazos Diaria	Media Fallos Diaria
		27891		2.44 %	27878(99.95 %)	0 (0%)	12 (0.05 %)
Descripción del negocio	Eventos que definen el Saldo de efectivo y pronóstico de caja al final del día.						
Ingestión	Inserte o actualice BAL. La actualización se realiza si existe el identificador de cuenta de efectivo proporcionado y la fecha para el saldo. Actualiza el saldo de cierre con la adición de saldo abierto y el importe de previsión.						
desencadenantes	N/A						
Post-procesamiento	REGISTROS DE RECONCILIACIÓN	Las discrepancias de equilibrio se almacenan en la tabla ERR con una descripción específica *					
	FIN DEL DÍA	PA_UTILS. P_SET_BAL_CURRENT_TO_EOD : Actualizaciones BAL_CURRENT con BAL_CLOSE					

*NOTA: Construir un reconciliador como RECON_POS para la posición de efectivo, en lugar de utilizar sólo la tabla ERR.

Evaluación de NRT-DWH

3.2.1.4.2 DETALLES DE LA POSICIÓN DE EFECTIVO: EVENTOS ESPECÍFICOS

Los eventos específicos de Posición de efectivo se describen a continuación para cada tipo de mensaje:

Evento	DTD/Controlador mensajes PROD	de	Ubicaciones remitente en PROD	de en	Características específicas
CashBalanceForecast	CashBalanceForecast_0_9		NO TE, EA, PARA		
SwiftBalance	SwiftBalance_0_5		NO TE, EA, PARA		
CashInterest	CashInterest_0_7		NO TE, EA, PARA		Solo actualiza BAL

3.2.1.4.3 VOLÚMENES DE DATOS DE POSICIÓN DE EFECTIVO

Los volúmenes de datos de producción de posición de efectivo muestran que el porcentaje de eventos correctos es cercano a 100, pero la buena medida de precisión sería un informe de conciliación análogo que ya está implementado para las posiciones de seguridad, para mostrar el número de filas conciliadas por el proceso de fin de día.

Message Type	Average message type per day	% over Total Cash Positio	Total Succes	% Succes	Total Failed	% Faile	Total Reproces	% Repr
CashBalanceForecast	25154	90.19	25150	99.98	3	0.01	0	0
SwiftBalance	1752	6.28	1743	99.49	9	0.51	0	0
CashInterest	985	3.53	985	100	0	0	0	0
Total	27891		27878	99.82333				0

Evaluación de NRT-DWH

3.2.1.5 ACCIÓN CORPORATIVA

3.2.1.5.1 ACCIÓN CORPORATIVA GENÉRICA

En la tabla siguiente, se definen los eventos de acción corporativa en general para las diferentes fases y estadísticas globales:

ACCIÓN CORPORATIVA GENÉRICA						
Estadísticas genéricas		Día medio total	Porcentaje sobre el total de mensajes	El éxito promedio del día	Día medio rechazado	Día medio fallido
		75468	6.61 %	75455(99.98 %)	0 (0%)	13 (0.016 %)
Descripción del negocio	Eventos que definen las acciones corporativas					
Ingestión	Insertar o actualizar en tablas de acciones corporativas. Cada tipo de mensaje de acción corporativa tiene una tabla diferente.					
desencadenantes	N/A					
Post-procesamiento	N/A					

3.2.1.5.2 DETALLES DE LA ACCIÓN CORPORATIVA: EVENTOS ESPECÍFICOS

Los eventos de acción corporativa específicos alimentan tablas diferentes para cada tipo de mensaje. Estas tablas forman parte del tablespace CA_DATA01 en la mayoría de los casos:

Evento	DTD/Controlador de mensajes	Ubicaciones de remitente	Características específicas
CaTakeoverExchange	CaTakeoverExchange_1_0 y 1_6	TODO	Ingesta CTK, CAN, CAP, CRC
CaDividendInterest	CaDividendInterest_1_1, 1_5 y 1_7	TODO	Ingesta CAO, CDO, CAI, CDV, CAF, CAP, CDN, CRC
CaRights	CaRights_1_0, 1_9	TODO	Ingesta CAO, CDO, CAI, CSR, CAS, CAP, CRC. Tenga en cuenta que NO TE, EA, TO flow también ingesta SWT
CaRedemption	CaRedemption_0_9, 1_3	TODO	Ingesta CRD, CAF, CAP, CRC, CAI
CaMeet	CaMeet_1_3	TODO	Ingesta PRX, PRT. Tenga en cuenta que NO TE, EA, TO flow también ingesta CAP
ProxyVoting	ProxyVoting_0_9	TE	Ingesta PRX, PRT. No es que TE, EA, TO flow también ingesta PRX_CAOPT
CaAdHoc	CaAdHoc_0_7	TODO	Ingesta CAH, CRC. TE, EA, TO flow también ingesta CAP
CaEvent	CaEvent_0_7	TE,EA	Ingesta CEV
CaInstrMappingOptions	CaInstrMappingOptions_1_0	TE	Ingesta CIO
CaNewSecurity	CaNewSecurity_0_6	TODO	Ingesta CNS
CAOptionAttributes	CaOptionAttributes_1_0	TE	Ingesta COA
CAMaster	CAMaster_0_5	TODO	Ingesta CAM

Evaluación de NRT-DWH

CPEvent	Nada	NADA	Ingesta CPE, CAM
DividenMaster	DividendMaster_0_5	TODO	Ingesta DVM
CAInstrucción	CAInstruction_0_5	TODO	Ingesta CIN y activa inserciones en CIN_SCI.
CASWIFTDates	CASWIFTDates_1_0	TE	Ingesta CDA
CASWIFTText	CASWIFTText_1_0	TE	Ingesta PCX
Cambs	Cambs	NADA	Ingesta CMB, CMF, CAP, CRC
CaTax	Nada	NADA	Ingesta CAT

3.2.1.5.3 VOLÚMENES DE DATOS DE ACCIÓN CORPORATIVA

Los volúmenes de datos de acción corporativa muestran que los eventos correctos están cerca del 100% en Producción:

Message Type	Average message type per day	% over Total Corporate Action	Total Suc	% Success	Total Fai	% Failed	Total Rep	% Reproc
CaEvent	59482	78.82	59482	100	0	0	0	0
CaDividendInterest	7174	9.51	7169	99.93	6	0.08	0	0
CaMeet	2687	3.56	2687	100	0	0	0	0
CAMaster	2220	2.94	2220	100	0	0	0	0
DividendMaster	1674	2.22	1674	100	0	0	0	0
CaTakeoverExchange	664	0.88	662	99.7	2	0.3	0	0
CANewSecurity	415	0.55	411	99.04	4	0.96	0	0
CAInstruction	361	0.48	361	100	0	0	0	0
CaRedemption	271	0.36	270	99.63	1	0.37	0	0
CaAdHoc	199	0.26	199	100	0	0	0	0
CaRights	110	0.15	110	100	0	0	0	0
CASWIFTDates	79	0.1	79	100	0	0	0	0
ProxyVoting	49	0.06	49	100	0	0	0	0
CASWIFTText	42	0.06	41	97.62	0	0	0	0
CAInstrMappingOptions	34	0.05	34	100	0	0	0	0
CAOptionAttributes	7	0.01	7	100	0	0	0	0
Total	75468		75455					0

Evaluación de NRT-DWH

3.2.1.6 DATOS DE REFERENCIA

3.2.1.6.1 DATOS DE REFERENCIA GENÉRICOS

DATOS DE REFERENCIA GENÉRICOS					
Estadísticas genéricas	Día medio total	Porcentaje sobre el total de mensajes	El éxito promedio del día	Día medio rechazado	Día medio fallido
	434501	38.08 %	434346(99.99 %)	1 (0.01%)	154 (2.81%)
Descripción del negocio	Eventos que definen los datos de referencia/maestro utilizados por la aplicación NRT-DWH				
Ingestión	Insertar o actualizar la tabla de datos de referencia.				
desencadenantes	N/A				
Post-procesamiento	N/A				

3.2.1.6.2 DATOS DE REFERENCIA: DETALLES ESPECÍFICOS

En la siguiente hoja de cálculo se describen los eventos de datos de referencia con tipo de mensaje, tabla ingestada y descripción empresarial (recuperados de dba_tab_comments).



Datos_Referencia.xlsx

Evaluación de NRT-DWH

3.2.1.6.3 VOLÚMENES DE DATOS DE REFERENCIA

Los eventos de acción corporativa específicos alimentan tablas diferentes para cada tipo de mensaje para volúmenes de producción. Estas tablas forman parte del espacio de tablas REFDAT en la mayoría de los casos. Cliente (100%) e Identificador(6,82%) eventos deben revisarse, porque están causando demasiados mensajes con errores.

Message Type	Average message type per day	% over Total Reference Data	Total Success	% Success	Total Failed	% Failed	Total Reproces	% Reproces
Security	215341.3	49.56	215341.17	99.9999	0.13	0	0	0
Price	101438.77	23.35	101438.23	99.9995	0.53	0	0	0
ContractNote	24037.33	5.53	24037.33	100	0	0	0	0
STPInfo	22513.43	5.18	22512.7	99.9967	0.73	0	0	0
Counterparty	19528.8	4.49	19528.43	99.9981	0.37	0	0	0
Fee	9752.87	2.24	9751.77	99.9887	0	0	1.1	0.0113
AccruedInterest	6288.97	1.45	6288.97	100	0	0	0	0
GroupMaster	5661.7	1.3	5661.7	100	0	0	0	0
UPTIXSecurityTransCanc	5209.23	1.2	5209.17	99.9987	0.07	0	0	0
UPTIXSecurityTrans	5209.23	1.2	5209.17	99.9987	0.07	0	0	0
UPTIXSecurityTransAdd	5209.23	1.2	5209.17	99.9987	0.07	0	0	0
UPTIXValuation	4530.9	1.04	4529.73	99.9743	1.17	0.03	0	0
ABErrorInfo	2464.53	0.57	2464.53	100	0	0	0	0
SecurityPoolFactor	2226.4	0.51	2226.4	100	0	0	0	0
FxRate	1251.83	0.29	1251.83	100	0	0	0	0
UPTIXTransMaster	1055.13	0.24	1055.13	100	0	0	0	0
Account	619.9	0.14	619.7	99.9677	0.2	0.03	0	0
ABReservationInfo	379.27	0.09	379.27	100	0	0	0	0
CashAccount	336.17	0.08	336.17	100	0	0	0	0
Relationship	283.8	0.07	283.8	100	0	0	0	0
UPTIXAccountMaster	257.8	0.06	257.8	100	0	0	0	0
UPTIXSecurityMaster	242.3	0.06	242.3	100	0	0	0	0
UPTIXUnitsInIssue	187.3	0.04	187.3	100	0	0	0	0
Client	149.7	0.03		0	149.7	100	0	0
UsTax	135.63	0.03	135.63	100	0	0	0	0
StaticMetaData	57.6	0.01	57.6	100	0	0	0	0
CashAccPrime	46.13	0.01	46.13	100	0	0	0	0
Shareholder	42.93	0.01	42.93	100	0	0	0	0
Identifier	11.73	0	10.93	93.1818	0.8	6.82	0	0
Location	11.6	0	11.6	100	0	0	0	0
ABAccountStatic	8.9	0	8.9	100	0	0	0	0
LocationMaster	5.93	0	5.93	100	0	0	0	0
UPTIXTransMatrix	3.8	0	3.8	100	0	0	0	0
Currency	0.3	0	0.3	100	0	0	0	0
CashBlockFunds	0.27	0	0.27	100	0	0	0	0
Calendar	0.13	0	0.13	100	0	0	0	0
CashUnblockFunds	0.07	0	0.07	100	0	0	0	0
ClientTaxID	0.07	0	0.07	100	0	0	0	0
Total	434500.98		434346.06	97.186916		2.8126316		0.0113

Evaluación de NRT-DWH

3.2.2 PROBLEMAS DETECTADOS

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DF01	Transacción genérica	Ingestión y postprocesamiento en diferentes órdenes	Relacionado con DI01. La ingestión y el procesamiento posterior se mezclan en los controladores de mensajes. A veces se cambia el orden de ejecución.	Datos incoherentes. La transacción se puede ingerir parcialmente	Alto
DF02	Específico Transacción	Registros TEV/TEG no insertados	Los eventos de transacciones no se insertan en TEV/TEG para varios tipos de transacción. es decir: TransDBCRef, UpdatePending, SwiftRepair, TransDispoStatus	La trazabilidad no se completa en la tabla Transaction events. Datos incoherentes	Medio
DF03	Transacción específica	Tipos de mensajes versionados	Solo 3 de 24 eventos de tipo de mensaje de transacción usan versiones diferentes de controladores de mensajes al mismo tiempo en producción. Estos eventos son SwiftRepair, TranSettled yTransPending.	Difícil de mantener diferentes versiones para DTD y controladores de mensajes en ambos flujos (Ubicaciones enrutadas y no enrutadas)	Muy Alto
DF04	Transacción específica	Diferentes operaciones para el mismo tipo de mensaje	Para el flujo de ubicación enrutada (TE, TO EA) están ejecutando diferentes operaciones que las del flujo de ubicación no enrutada. Ligeras diferencias. es decir: TransAutoSettUX actualiza la posición en ubicaciones no enrutadas, pero no en ubicaciones enrutadas. Ejemplo 2: TransMatch es diferente en la ingestión de IHM para los flujos y la misma versión.	Datos incoherentes. Los mismos mensajes derivarían en diferentes comportamientos	Alto

Evaluación de NRT-DWH

DF05	Volúmenes de datos	Reprocesar transacciones	Comprobar volúmenes de datos: el 90 % de las transacciones REPROCESS (42 mensajes por día) en la tabla RCM proceden de eventos TransFailed. Es un evento que cambia el estado de las transacciones a error. La fecha de liquidación ha expirado, pero no ha llegado ninguna transacción de liquidación. Db-Trader Germany (TE) envió 189 eventos TransFailed que se enviaron para volver a procesar en NRT-DWH, pero Trader MMA (T5) están enviando eventos que están en Reproceso de forma permanente. Revise con Trader MMA.	El error de cálculo de la posición y el estado de la transacción no han cambiado correctamente. EOD actualizará el estado, pero los datos de posición y equilibrio serán incorrectos en el proceso intradía.	Medio
DF06	Volúmenes de datos	Transacciones fallidas	Hay una gran población de eventos de transacción fallidos. Alrededor de 5K por día. Los eventos más afectados son TransSettled, TransFailed, TransPending y TransTradeConf. Los eventos deben ser revisados con los sistemas Upstream.	Error de cálculo de posición intradía y estado de transacción incorrecto.	Alto
DF07	Ciclo de vida comercial	Marco de control de transiciones N/A	El estado de transición que no se aplique funcionalmente debe restringirse. es decir: el evento TransTransfer no debe cambiar a ningún otro estado de transacción	Falta de mecanismo de control.	Bajo
DF08	Posición de efectivo – Post Procesamiento	Informes de posición de caja de conciliación	Durante el fin del día, las posiciones de efectivo se restablecen en NRT-DWH en caso de desajustes. Hay un informe de conciliación para Transacción de seguridad, pero falta para Posiciones de efectivo. Sería una buena idea implementar el informe de conciliación en efectivo. Estas entradas se almacenan actualmente en ERR, pero es una tabla para procesar errores.	Estadísticas que faltan para posiciones de efectivo conciliadas	Bajo
DF09	Genérico	Auditoría de registros actualizados en tablas principales	Cuando se actualiza una transacción, hay una trazabilidad en las tablas de eventos de transacción, pero este mecanismo no está presente para la posición, la posición de efectivo o la transacción.	Trazabilidad. No se pueden recuperar datos históricos.	Alto

Evaluación de NRT-DWH

3.2.3 MEJORAS PROPUESTAS

Problema #	Resumen del Problema	Solución Propuesta	Coste
DF01	Ingestión y postprocesamiento en diferentes órdenes	Alinear el orden de "Ingestión – Postprocesamiento" para los diferentes controladores de mensajes para lograr la ejecución correcta.	Alto
DF02	Registros TEV/TEG no insertados	Insertar los registros TEV/TEG para los eventos que faltan	Bajo
DF03	Tipos de mensajes versionados	Compilar el XSD correspondiente que fuerza la estructura correcta para un mensaje. DtD versionados y controladores de mensajes añaden complejidad al código y son difíciles de mantener	Muy Alto
DF04	Diferentes operaciones para el mismo tipo de mensaje	Combinar ambos flujos de mensajes en uno. Las diferencias son leves e implementan un cambio para los ubicaciones enrutadas y no enrutadas aumenta el error de desarrollo y afecta a la estabilidad de la producción	Muy Alto
DF05	Reprocesar transacciones	Analizar las transacciones con Trader para encontrar las causas principales que están causando estas brechas en la producción	Medio
DF06	Transacciones fallidas	Analizar las transacciones con Trader para encontrar las causas principales que están causando estas brechas en la producción	Medio
DF07	Marco de control de transiciones N/A	Implementar nuevas excepciones para evitar transiciones imposibles durante el ciclo de vida del comercio	Bajo
DF08	Informes de posición de caja de conciliación	Implementar un informe de reconciliación para posición de efectivo	Medio
DF09	Auditoría de registros actualizados en tablas principales	Revisar si se deben auditar columnas de transacciones adicionales y el mismo mecanismo para la posición, posición de efectivo, transacción, acción corporativa y datos de referencia	Muy Alto

3.3 REPOSITORIO DE DATOS

3.3.1 SITUACIÓN ACTUAL

En las secciones siguientes, se describirá el modelo de datos NRT-DWH para las diferentes entidades empresariales, con el fin de proporcionar una evaluación en términos de repositorio de datos.

Modelos de datos :

- Transacción
- Posición
- Transacción de efectivo y saldo

Para el modelo de datos anterior se puede encontrar una descripción detallada en las secciones siguientes. Las acciones corporativas y los datos de referencia se ingestan directamente y los modelos de datos correspondientes contienen varias tablas sin relaciones entre ellas. Puede encontrar más información sobre el contenido de estos modelos de datos a través de los siguientes enlaces:

[Acciones corporativas](#)

[Datos de referencia](#) (abra el documento de Excel que se encuentra en el enlace)

Evaluación de NRT-DWH

3.3.1.1 TRANSACCIÓN

La imagen que se encuentra a continuación muestra el modelo de base de datos lógica actual para Transacciones obtenido del modelo físico en SQL Developer Data Modeler mediante ingeniería inversa.

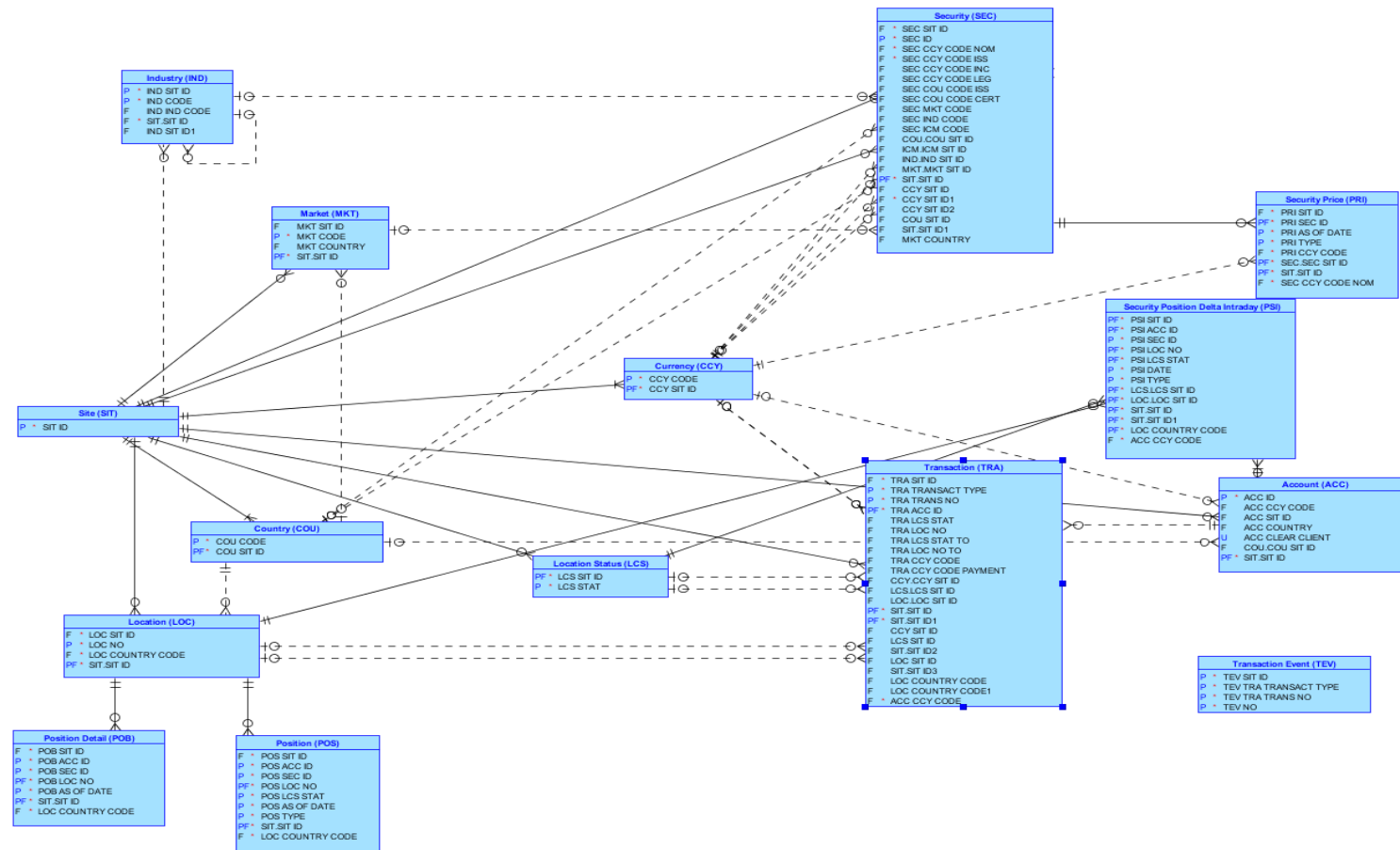


Figura 4.1 Modelo de datos lógico de transacciones de NRT-DWH: Estado actual.

Evaluación de NRT-DWH

3.3.1.2 POSICIÓN

La imagen que se encuentra a continuación muestra el modelo de base de datos lógico actual para Posiciones obtenido del modelo físico en SQL Developer Data Modeler mediante ingeniería inversa.

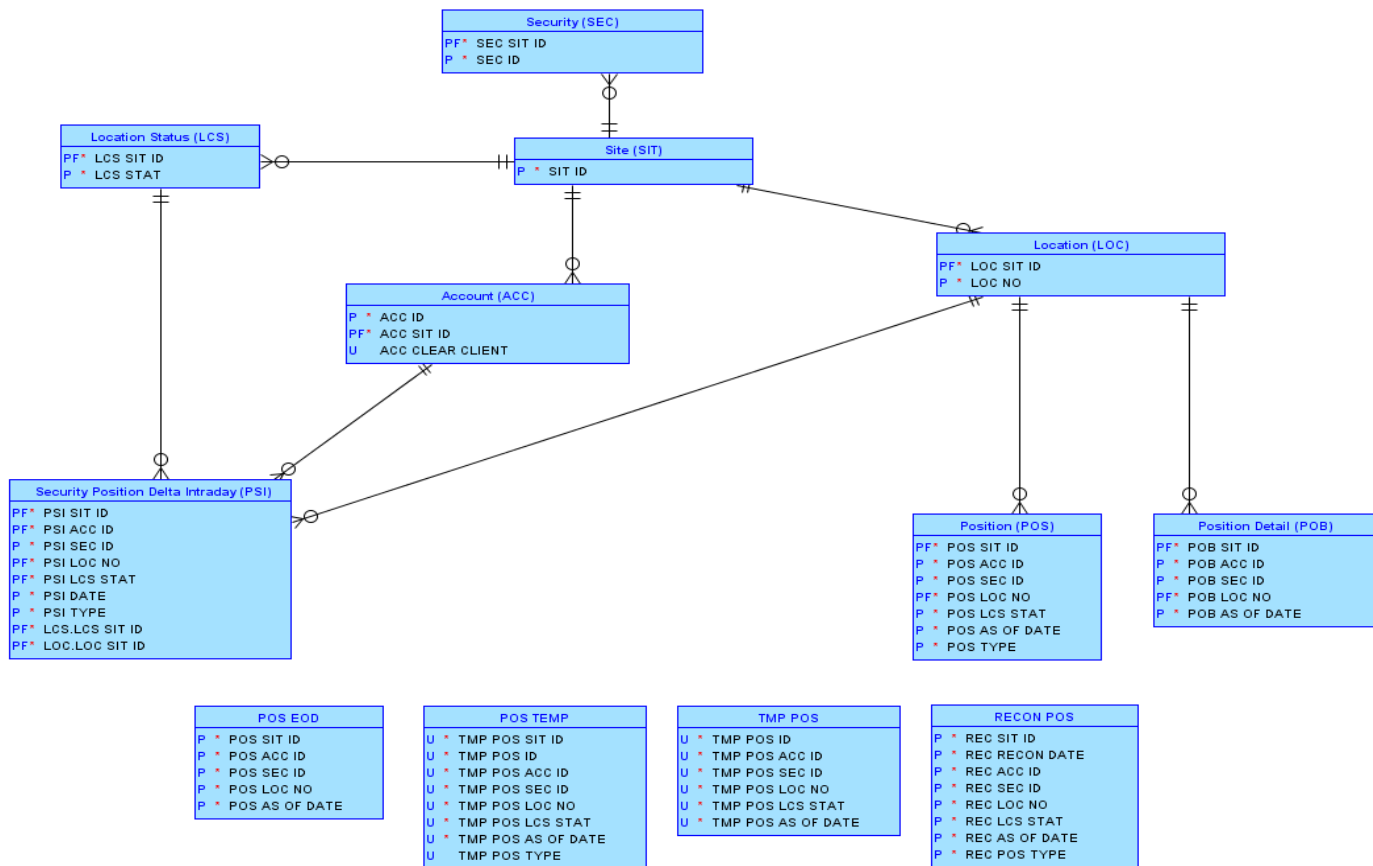


Figura 4.2 Estado actual del modelo de datos lógico de Posiciones de NRT-DWH.

Evaluación de NRT-DWH

3.3.1.3 TRANSACCIÓN EN EFECTIVO Y POSICIÓN DE EFECTIVO

La imagen que se encuentra a continuación muestra el modelo actual de base de datos lógica para la transacción de efectivo y la posición de efectivo obtenido del modelo físico en SQL Developer Data Modeler mediante ingeniería inversa.

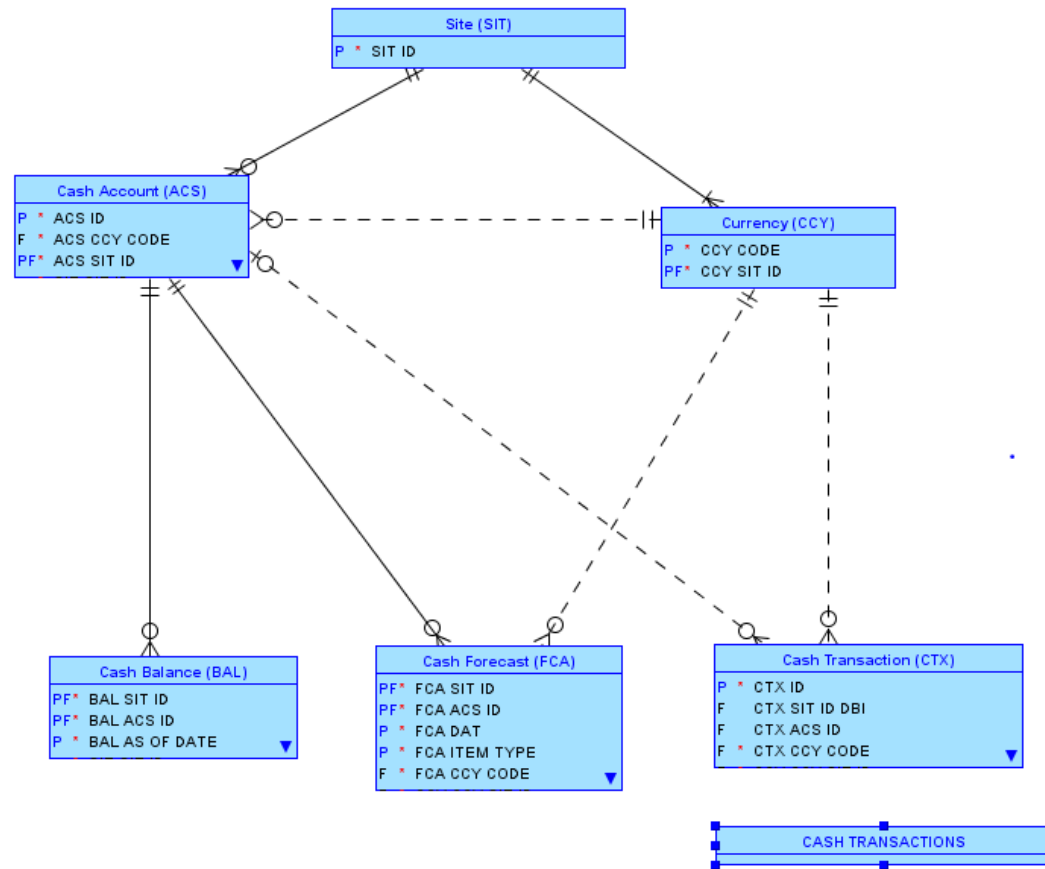


Figura 4.3 Estado actual del modelo de datos lógico de Posiciones de NRT-DWH.

Evaluación de NRT-DWH

3.3.2 PROBLEMAS DETECTADOS

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DM01	Físico	Faltan FK's	Algunos FK parecen faltantes dentro del modelo de base de datos. Es decir: Las tablas SEC y TRA no tienen ninguna relación.	Impactos en la calidad de los datos. Datos incoherentes entre entidades.	Medio
DM02	Físico	PK Subrogada vs Compuesta	En la mayoría de los casos, los PK compuestos se utilizan en el modelo de base de datos. Esta opción utiliza más recursos físicos para crear índices que subrogadas y también las combinaciones son más complejas.	Datos de combinación de bajo rendimiento entre tablas. Las buenas prácticas ^[5] recomiendan evitar PK con campos de cadena para mejorar el rendimiento.	Medio
DM03	Físico	Alineación de tipos de datos PK dentro del modelo de base de datos	Hay PK que las columnas de tipo de datos no coinciden con las columnas de tipo de datos FK que deberían asociarse. De esta manera no se podría definir tal relación	Errores incoherentes de datos y calidad de datos. Las buenas prácticas ^[6] obligan a alinear los tipos de datos en cada relación dentro del modelo de datos.	Alto
DM04	Lógico	Revisión de tipos de datos para nombres de columna similares	Hay columnas con el mismo contenido que tienen diferentes tipos de datos (precisión, longitud, escala, etc...). Es decir: XXX_SIT_ID, XXX_SEC_ID.	Impactos en la calidad de los datos. Puede producir errores de código debido a instrucciones SQL no válidas (unir campos no coincidentes). Las buenas prácticas recomiendan alinear los tipos de datos dentro de todos los campos del modelo de base de datos.	Alto
DM05	Lógico	Restricciones de comprobación constante frente a tabla de referencia	Algunas restricciones se crean con validaciones codificadas de forma rígida. Es decir: TRA_CHK_TRANS_STAT restricción en la tabla TRA tiene una condición manual TRA_TRANS_STAT IN ('C', 'F', 'P', 'S', 'T', 'PS')	Dificultad para mantener el código en caso de nuevos valores.	Bajo
DM06	Físico	Múltiples tipos de objetos grandes en una sola tabla	Algunas tablas tienen varios CLOB, XML y cualquier otro objeto grande, y esto podría generar algunos problemas de rendimiento	Impactos en el rendimiento y el almacenamiento. Las buenas prácticas recomiendan tener solo los campos obligatorios y, en algunos casos, insertar un vínculo externo al objeto en el campo correspondiente.	Medio

Evaluación de NRT-DWH

DM07	Físico	Objetos del espacio de tablas correspondiente	Hay objetos de base de datos asignados fuera de su espacio de tablas correspondiente. Es decir: En el entorno DEV hay 54 índices en el espacio de tablas REFDAT y 32 índices en el espacio de tablas USERS.	Algunas tareas limpias de administración de bases de datos pueden afectar a esos objetos. Las buenas prácticas recomiendan crear un conjunto de espacios de tablas para cada aplicación (DATA, HUGE DATA e INDEXES).	Bajo
DM08	físico	Columnas de ID concatenadas automáticamente	Hay columnas con contenido concatenado (separados por comas). Es decir: RCM_ID en la tabla TRA. (000186536558,000186536559). Esta columna se define con un VARCHAR2(4000) en la mayoría de los casos. En caso de que se concatenaron demasiados identificadores, todas las operaciones DML que contienen este atributo generarán una excepción SQL. El código fuente evita este problema en la mayoría de los casos mediante una función de Oracle para truncar el contenido con 4000 caracteres (F_FORMAT_RCM_ID)	Pérdida del negocio y problemas de calidad de datos debido a excepciones SQL producidas por errores de aplicación (contenido de campo desbordado)	Alto
DM09	Lógico	Tablas potenciales con datos duplicados	Parece que hay algunas tablas que estructuran y los datos de contenido son muy similares a otras tablas en el modelo de base de datos. Un ejemplo a investigar puede ser TRA vs TRG, donde también FK y PK son los mismos.	Lógica de negocios compleja, dificultad en el mantenimiento del código y demasiado almacenamiento necesario.	Bajo
DM10	Lógico	Revisión de PK	Verificación PK en términos de negocio. SIT_ID se utiliza como parte de PK en la mayoría de las tablas de referencia. es decir: Países o monedas duplicados por ubicación. La tabla CTX (Cash Transaction) tiene una clave subrogada (CTX_ID) pero no un índice único, por lo que no hay ninguna clave natural ni integridad. Revise esta tabla.	Redundancia de datos, claves primarias complejas e impacto en el rendimiento.	Bajo

3.3.3 MEJORAS PROPUESTAS

La imagen que se encuentra a continuación contiene una propuesta con algunas nuevas relaciones entre las tablas principales.

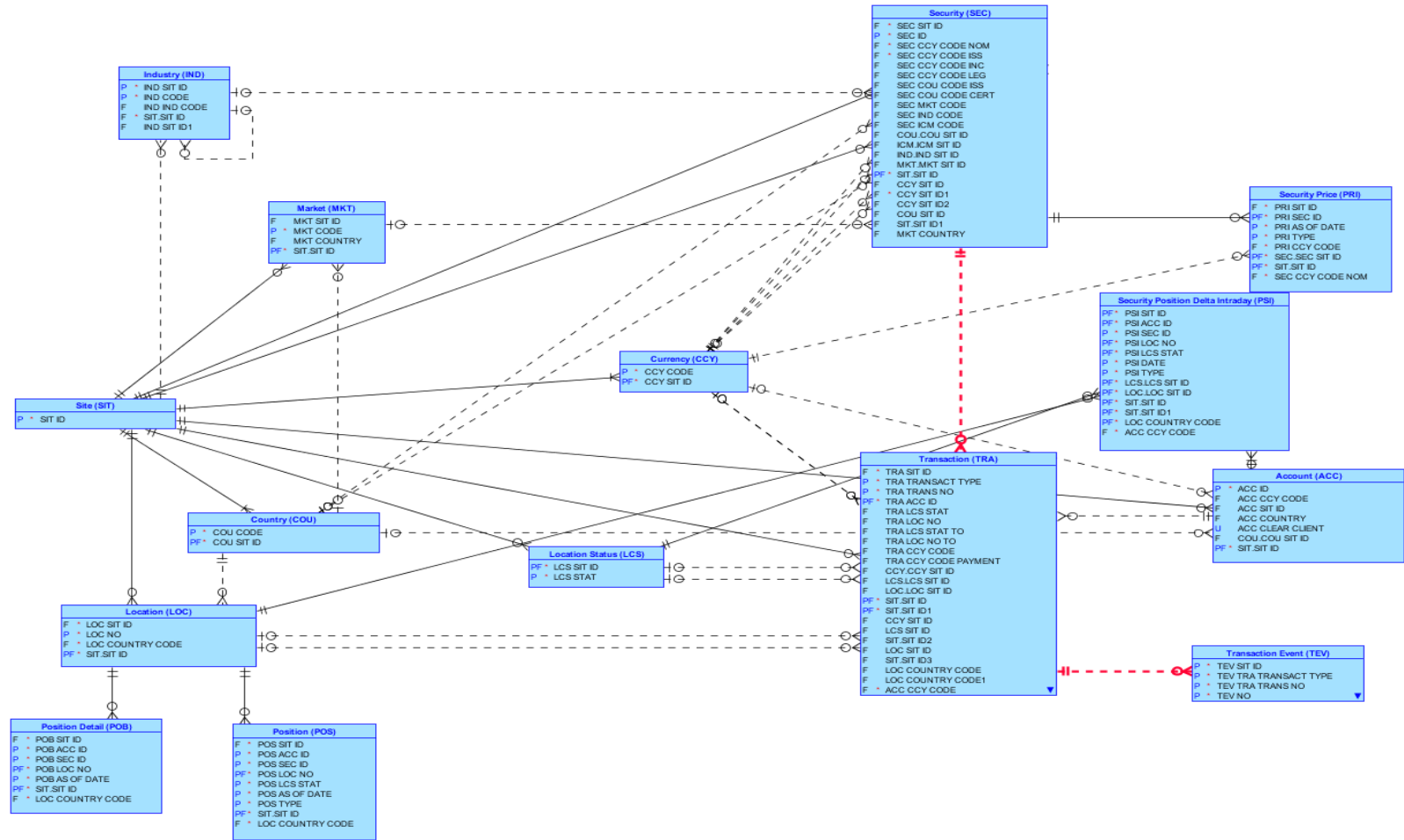


Figura 5.1 Estado futuro del modelo de datos lógico de transacciones de NRT-DWH.

Evaluación de dbCRIS

3.3.3.1 POSICIÓN

La propuesta de modelo de datos de posición con algunas relaciones nuevas entre las tablas principales se muestra en la imagen siguiente.

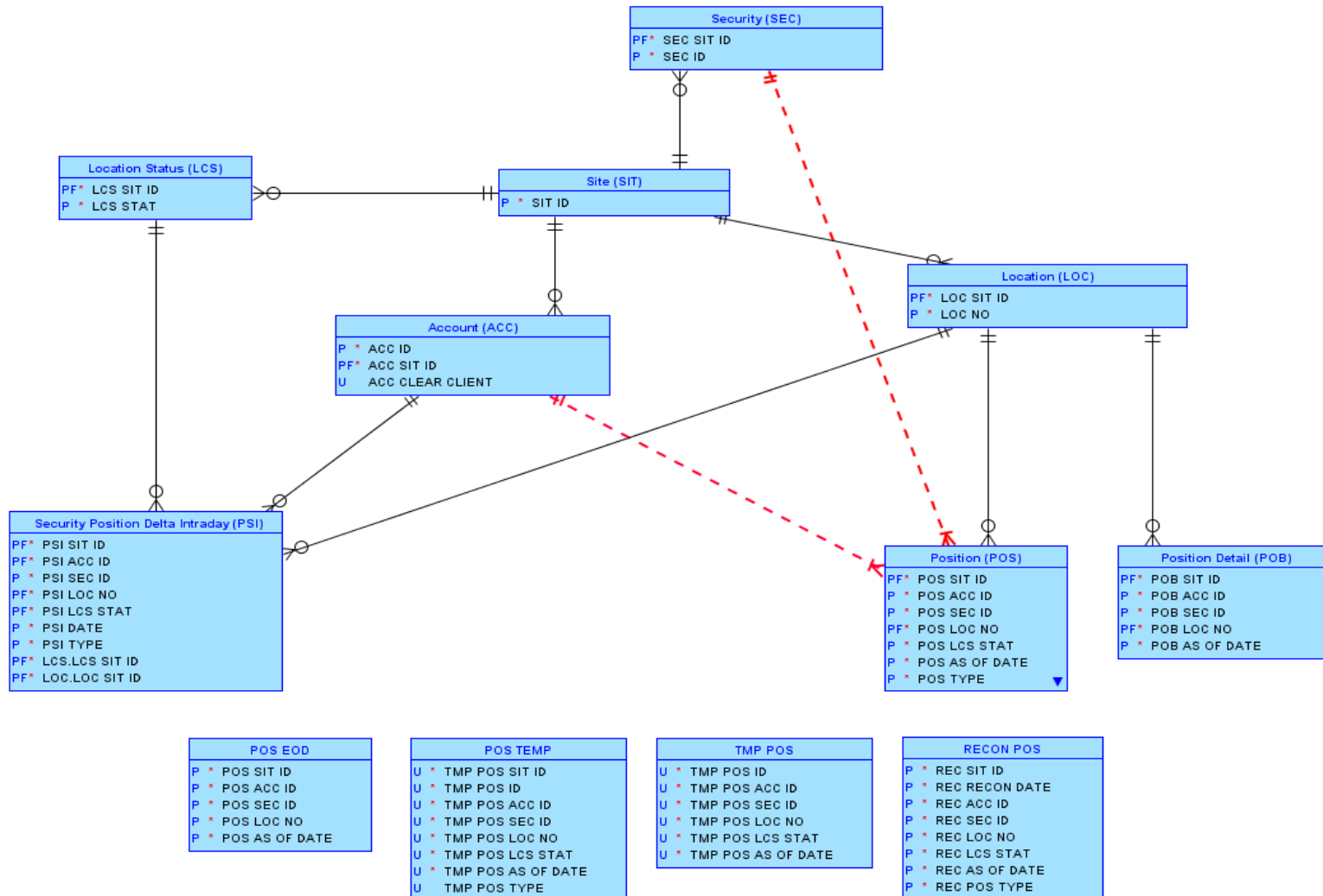


Figura 5.2 Estado futuro del modelo de datos lógico de Posiciones de NRT-DWH.

3.3.3.2 TRANSACCIÓN DE EFECTIVO Y SALDO

La siguiente imagen contiene una propuesta con algunas nuevas relaciones para el modelo de datos de Transacción de efectivo y Saldo

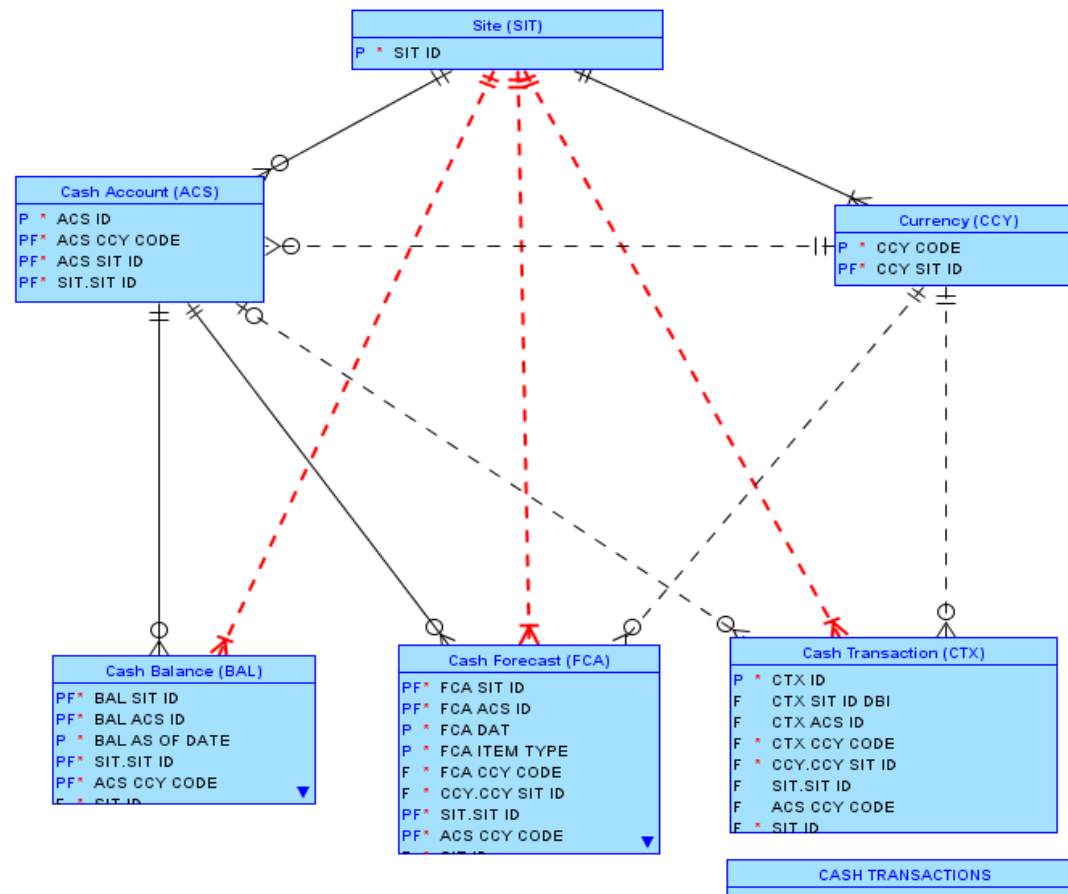


Figura 5.3 NRT-DWH Transacción en efectivo y Balance Modelo de datos lógicos – Estado futuro.

Evaluación de NRT-DWH

Problema #	Resumen del Problema	Solución Propuesta	Coste
DM01	FK's faltantes	Revisar el modelo de base de datos y evalúe todos los FK posibles que se puedan considerar que faltan.	Medio
DM02	PK Subrogada vs Compuesta	En algunos casos puede ser interesante evitar PKs compuestas mediante el uso de PKs subrogadas. La antigua PK compuesta se convertiría en una clave simple. Tenemos que evaluar si es una ventaja en términos de rendimiento.	Muy Alto
DM03	Alineación de tipos de datos PK dentro del modelo de base de datos	Alinear las especificaciones de PK frente a FK y revisar todas las columnas potenciales para alinear los tipos de datos entre todas las tablas de base de datos.	Bajo
DM04	Revisión de tipos de datos para nombres de columna similares	Todas las columnas con el mismo contenido deben tener el mismo tipo de datos. Es decir: XXX_SIT_ID, XXX_SEC_ID.	Bajo
DM05	Restricciones de comprobación constante frente a tabla de referencia	Evaluar evitar restricciones de comprobación constante y utilizar tablas de referencia con el FK correspondiente.	Medio
DM06	Múltiples tipos de objetos grandes en una sola tabla	Revisar todas las tablas con varios CLOB, XML y cualquier otro objeto grande. Como esto podría crear un problema de rendimiento	Medio
DM07	Objetos del espacio de tablas correspondiente	Mover todos los objetos detectados al espacio de tablas correcto	Bajo
DM08	Columnas de ID concatenadas automáticamente	Cambiar los valores concatenados automáticamente por un nuevo ID con una tabla de referencia correspondiente.	Alto
DM09	Tablas potenciales con datos duplicados	Parece que hay algunas tablas cuyos datos de contenido pueden ser muy similares a otras tablas en el modelo de base de datos. Algunos ejemplos a investigar son TRA vs TRG y TEV vs TEG	Alto
DM10	Revisión de PK	Analizar si los PK se representan con precisión y evitar valores duplicados dentro de las entidades.	Alto

Evaluación de NRT-DWH

3.3.4 EJEMPLOS DE PROBLEMAS

3.3.4.1 DM01 – FKS FALTANTES

La tabla SEC (Securities) debe tener una relación con la tabla TRA (Transacciones).

3.3.4.2 DM02 – PK SUBROGADA VS COMPUESTA

Si el Downstream necesita los campos compuestos, es posible tenerlos como atributo en la tabla, pero sería interesante tener la clave subrogada, es decir, un campo numérico adicional mediante el que se implementaría la relación y evitaría tener que comparar a través de la clave simple que está compuestas por varios campos de texto.

Security Transaction (TRA)		Security (SEC)	
P *	TRA TRANSACT TYPE	VARCHAR (1)	F SEC SIT ID
P *	TRA TRANS NO	VARCHAR (16)	P * SEC ID
F *	TRA ACC ID	VARCHAR (15)	F * SEC CCY CODE NOM
F	TRA LCS STAT	VARCHAR (4)	F * SEC CCY CODE ISS
F	TRA LOC NO	VARCHAR (12)	F SEC CCY CODE INC
F	TRA LCS STAT TO	VARCHAR (4)	F SEC CCY CODE LEG
F	TRA LOC NO TO	VARCHAR (12)	F SEC COU CODE ISS
F	TRA CCY CODE	VARCHAR (3)	F SEC COU CODE CERT
F	TRA CCY CODE PAYMENT	VARCHAR (3)	F SEC MKT CODE
F	TRA SIT ID1	VARCHAR (2)	F SEC IND CODE
PF *	TRA SIT ID3	VARCHAR (2)	F SEC ICM CODE
F	LCS SIT ID	VARCHAR (2)	F SEC SIT ID1
F	CCY.CCY SIT ID	VARCHAR (2)	F SEC SIT ID2
F	CCY SIT ID	VARCHAR (2)	F COU SIT ID
			F * CCY.CCY SIT ID
			F CCY SIT ID
			F * CCY SIT ID1
			F CCY SIT ID2

Figura 6. Ejemplo de PK Subrogada vs Compuesta.

3.3.4.3 DM03 – ALINEACIÓN DE TIPOS DE DATOS DE PK EN EL MODELO DE BASE DE DATOS

Las buenas prácticas recomiendan alinear los tipos de datos de columna especialmente cuando forman parte de una clave.

El modelo contiene diferencias y se muestra un ejemplo a continuación.

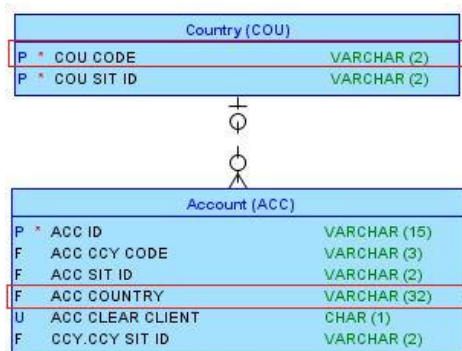


Figura 7. PK y FK relacionado con diferentes tipos de datos ejemplo.

Evaluación de NRT-DWH

La siguiente consulta obtiene todos los valores diferentes.

select

```
ucc.constraint_name, ucc.table_name, ucc.column_name , utc.data_type, utc.data_length, utc.data_precision,
utc.data_scale, utc.nullable, uc.r_constraint_name RELATED_KEY_NAME , u_rel.table_name
related_table_name, u_rel.column_name related_column_name, utc_rel.data_type related_data_type,
utc_rel.data_length related_data_length, utc_rel.data_precision related_data_precision, utc_rel.data_scale
related_data_scale, utc_rel.nullable related_nullable
```

from

```
dba_cons_columns ucc , dba_constraints uc, dba_cons_columns u_rel, dba_tab_columns utc, dba_tab_columns
utc_rel
```

where

```
ucc.table_name in ('TRA', 'TRG', 'TEV', 'POS', 'POB', 'TEG', 'PSI', 'PRI', 'COU', 'IND', 'MKT', 'SEC', 'ICM', 'CCY',
'MST_TEMP', 'MST', 'LOC', 'ACC', 'LCS', 'CASH_TRANSACTIONS', 'SIT')
and uc.constraint_type = 'R' and ucc.constraint_name = uc.constraint_name
and uc.r_constraint_name = u_rel.constraint_name and ucc.position = u_rel.position
and ucc.table_name = utc.table_name and u_rel.table_name = utc_rel.table_name
and ucc.column_name = utc.column_name and u_rel.column_name = utc_rel.column_name
-- Conditions for differences
and (utc.data_type != utc_rel.data_type or utc.data_length != utc_rel.data_length or utc.data_precision !=
utc_rel.data_precision or utc.data_scale != utc_rel.data_scale)
```

3.3.4.4 DM04 – TIPOS DE DATOS PARA LA REVISIÓN DE NOMBRES DE COLUMNA SIMILARES

Además, Las buenas prácticas recomiendan tener el mismo tipo de datos para cada columna del modelo que puede contener los mismos datos. La siguiente imagen muestra algunos campos potenciales para aplicar estos cambios.

ORIG_TABLE	ORIG_COL_NAME_TR	ORIG_COL_NAME	ORIG_DATATYPE	ORIG_LEN	ORIG_NULL	COMP_TABLE_NAME	COMP_COL_NAME_TR	COMP_COL_NAME	COMP_DATATYPE	COMP_LEN	COMP_NULL
TRA	ID	RCM_ID	VARCHAR2	4000	Y	CID	ID	RCM_ID	VARCHAR2	50	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	CIN_SCI	ID	RCM_ID	VARCHAR2	64	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	MK_BAL	ID	RCM_ID	VARCHAR2	1000	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	MSG_CACHE	ID	RCM_ID	VARCHAR2	50	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	MSG_REPROCESSING_LOG	ID	RCM_ID	VARCHAR2	30	N
TRA	ID	RCM_ID	VARCHAR2	4000	Y	POS2	ID	RCM_ID	VARCHAR2	1000	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM	ID	RCM_ID	VARCHAR2	50	N
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM_HIST	ID	RCM_ID	VARCHAR2	50	N
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM_RECON_ERROR	ID	RCM_ID	VARCHAR2	30	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM_RECON_ERROR_LIST	ID	RCM_ID	VARCHAR2	30	Y
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM_REPROCESS	ID	RCM_ID	VARCHAR2	50	N
TRA	ID	RCM_ID	VARCHAR2	4000	Y	RCM_TEMP	ID	RCM_ID	VARCHAR2	50	N
TRA	ID	RCM_ID	VARCHAR2	4000	Y	TRL1	ID	RCM_ID	VARCHAR2	2	Y

Figura 8. Posibles columnas similares con diferentes tipos de datos ejemplo.

Evaluación de NRT-DWH

La consulta para obtener todos los campos posibles también con valores de precisión y escala es la siguiente:

```
select
    orig_table, orig_col_name_tr, orig_col_name, orig_datatype, orig_len, orig_prec, orig_scale, orig_null
    , comp_table_name, comp_col_name_tr, comp_col_name, comp_datatype, comp_len, comp_prec, comp_scale, comp_null
from
    (select
        distinct ca.owner||'. '||ca.table_name orig_table, substr(ca.column_name,5,length(ca.column_name)-4)
        orig_col_name_tr, ca.column_name orig_col_name, ca.data_type orig_datatype, ca.data_length orig_len,
        ca.data_precision orig_prec, ca.data_scale orig_scale, ca.nullable orig_null
        , cb.owner||'. '||cb.table_name comp_table_name, substr(cb.column_name,5,length(cb.column_name)-4)
        comp_col_name_tr, cb.column_name comp_col_name, cb.data_type comp_datatype, cb.data_length comp_len,
        cb.data_precision comp_prec, cb.data_scale comp_scale, cb.nullable comp_null
    from dba_tab_columns ca, dba_tab_columns cb
    where
        ca.table_name in ('TRA', 'TRG', 'TEV', 'POS', 'POB', 'TEG', 'PSI', 'PRI', 'COU', 'IND', 'MKT', 'SEC', 'ICM', 'CCY',
        'MST_TEMP', 'MST','LOC', 'ACC', 'LCS', 'CASH_TRANSACTIONS', 'SIT')
        and substr(ca.column_name,5,length(ca.column_name)-4) = substr(cb.column_name,5,length(cb.column_name)-
        4)
        and length(ca.column_name) > 11
        and ca.table_name <> cb.table_name
        and (ca.data_type <> cb.data_type or (ca.data_type = cb.data_type and ca.DATA_LENGTH <> cb.DATA_LENGTH))
    UNION
    select
        distinct ca.owner||'. '||ca.table_name orig_table, substr(ca.column_name,5,length(ca.column_name)-4)
        orig_col_name_tr, ca.column_name orig_col_name,
        ca.data_type orig_datatype, ca.data_length orig_len, ca.data_precision orig_prec, ca.data_scale orig_scale,
        ca.nullable orig_null
        , cb.owner||'. '||cb.table_name comp_table_name, substr(cb.column_name,5,length(cb.column_name)-4)
        comp_col_name_tr, cb.column_name comp_col_name,
        cb.data_type comp_datatype, cb.data_length comp_len, cb.data_precision comp_prec, cb.data_scale comp_scale,
        cb.nullable comp_null
    from dba_tab_columns ca, dba_tab_columns cb
    where
        ca.table_name in ('TRA', 'TRG', 'TEV', 'POS', 'POB', 'TEG', 'PSI', 'PRI', 'COU', 'IND', 'MKT', 'SEC', 'ICM', 'CCY',
        'MST_TEMP', 'MST','LOC', 'ACC', 'LCS', 'CASH_TRANSACTIONS', 'SIT')
        and ca.column_name = cb.column_name
        and length(ca.column_name) < 11
        and ca.table_name <> cb.table_name
        and (ca.data_type <> cb.data_type or (ca.data_type = cb.data_type and ca.DATA_LENGTH <> cb.DATA_LENGTH))
    ) type_comp
--where orig_col_name = 'RCM_ID' and orig_table = 'TRA'
order by 1,3,9,11
```


Evaluación de NRT-DWH

3.3.4.5 DM05 – RESTRICCIONES DE COMPROBACIÓN CONSTANTE FRENTE A TABLA DE REFERENCIA

Algunas restricciones validan el contenido con un conjunto de valores insertados manualmente. Es interesante aplicar una tabla de referencia y de esta manera el filtro aplicado no depende del código de aplicación desplegado, sino de una actualización mediante DML del contenido de la tabla generada.

La siguiente consulta obtiene todas las restricciones detectadas.

```
select
  owner, constraint_name, constraint_type, table_name, search_condition, status, last_change
from
  dba_constraints
where
  constraint_type = 'C'
  and table_name in ('TRA', 'TRG', 'TEV', 'POS', 'POB', 'TEG', 'PSI', 'PRI', 'COU', 'IND', 'MKT', 'SEC', 'ICM', 'CCY',
  'MST_TEMP', 'MST', 'LOC', 'ACC', 'LCS', 'CASH_TRANSACTIONS', 'SIT')
  and constraint_name like '%CHK%' or constraint_name like 'SIT_CC0%'
```

OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME	SEARCH_CONDITION	STATUS	LAST_CHANGE
CRIS	SIT_CC01	C	SIT	SIT_TKT_REF_FLAG IN ('Y','N')	ENABLED	20-JAN-10
CRIS	SIT_CC02	C	SIT	SIT_BENCHMARK_FLAG IN ('Y','N')	ENABLED	20-JAN-10
CRIS	TRG_CHK_TRANS_STAT	C	TRG	trg_trans_stat IN ('C', 'F', 'P', 'S', 'I', 'PS')	ENABLED	27-NOV-17
CRIS	TRA_CHK_TRANS_STAT	C	TRA	TRA_TRANS_STAT IN ('C', 'F', 'P', 'S', 'I', 'PS')	ENABLED	23-MAY-17

Figura 9. Restricciones con condiciones de búsqueda manual.

3.3.4.6 DM06 – MÚLTIPLES TIPOS DE OBJETOS GRANDES EN UNA SOLA TABLA

Algunas tablas del modelo de base de datos contienen un conjunto de columnas con tipos de datos grandes. A continuación se muestra un ejemplo con algunas columnas de la tabla PRX que tiene 14 columnas con datos grandes.

PRX COUNTER MOTION DEU	CLOB (4000)
PRX TEXT DEU	CLOB (4000)
PRX DEPOSITORY	CLOB (4000)
PRX TEXT GENERAL	CLOB (4000)
PRX AGM REPEAT DATE	Date (7)
PRX APPLICATOIN TERM	CLOB (4000)
PRX VOTING RIGHT RESTRIC TYPE	CLOB (4000)
PRX DIFFER LHT	VARCHAR (1)
PRX DIFFER LUT	VARCHAR (1)
PRX WKNISIN CHILD	CLOB (4000)
PRX AGENDA DEU	CLOB (4000)
PRX AGENDA RESULT DEU	CLOB (4000)
PRX ADDITION DOC	CLOB (4000)
PRX DIFFERENT TIME LIMIT	CLOB (4000)
PRX AGENDA ENGL	CLOB (4000)
PRX AGENDA RESULT ENGL	CLOB (4000)
PRX LAST CHANGE DATE	Date (7)

Figura 10. Ejemplo de tabla con varias columnas de tipo de datos grandes.

Evaluación de NRT-DWH

La siguiente consulta obtiene todas las tablas que almacenan tipos de datos grandes que pueden afectar al rendimiento de la aplicación.

```

select
    owner,table_name, count(1) as big_datatypes_columns
from dba_tab_columns
where
    data_type in ('CLOB','XMLTYPE','RAW','LONG','BLOB','LONG RAW')
    and owner in ('NRTDWH','DCSW_OWNER')
having count(1) > 1
group by owner,table_name
order by 1, 2
    
```

54 Tablas (DEV)			11 Tablas (UAT)		
OWNER	TABLE_NAME	NUM_LARGE_COLUMNS	OWNER	TABLE_NAME	NUM_LARGE_COLUMNS
NRTDWH	ETOSIT_DSSP	2	NRTDWH	PRX	14
NRTDWH	ETSECT	2	DCSW_OWNER	INC_PLSQL_ERR	2
NRTDWH	MGW_MESSAGE_TAB	2	DCSW_OWNER	JASPER_REPORT_MASTER	2
NRTDWH	MONITORING_QUERY	2	DCSW_OWNER	JR_REPORT_FORMAT	2
NRTDWH	PMT_REC	2	DCSW_OWNER	MG_FEED_OUT_MSG	2
NRTDWH	PRX	14	DCSW_OWNER	PMT_APPROVED	3
NRTDWH	REPAIR_XMLMSG_QTAB	3	DCSW_OWNER	PMT_CLIENT	3
NRTDWH	TMP_PRX	13	DCSW_OWNER	PMT_REC	2
NRTDWH	TOAD_PLAN_TABLE	2	DCSW_OWNER	SDM_GENERIC_TABLE	2
NRTDWH	TS_PARAMETERS_QTAB	2	DCSW_OWNER	V_ONE_IT_GMFFM	24
DCSW_OWNER	BO_EE2_OUT_FEED	2	DCSW_OWNER	V_PROXY_VOTING	13
DCSW_OWNER	FM_CONTENT_TABLE_02	2			

Figura 11. Conjunto de tablas con varias columnas de tipo de datos grandes.

3.3.4.7 DM07 - OBJETOS DEL TABLESPACE CORRESPONDIENTE

Las buenas prácticas recomiendan distribuir objetos utilizando su tablespace correspondiente, incluso en la medida de lo posible en sistemas de ficheros distintos. Básicamente, la recomendación es tener un tablespace para los índices, uno para los datos y el último para datos grandes (objetos que utilizan campos LOB).

La siguiente consulta obtiene los objetos asignados fuera de su tablespace recomendado

SEGMENT_TYPE	TYPE	TABLESPACE_NAME	NUM_OBJECTS
INDEX PARTITION	ERROR	DCSW_DATA02	5
INDEX	ERROR	POSDAT01	1
INDEX	ERROR	REFDAT	2
INDEX	ERROR	DCSW_DATA02	80
LOBSEGMENT	ERROR	DCSW_DATA02	16
INDEX SUBPARTITION	ERROR	SEC_NU01	2
INDEX SUBPARTITION	ERROR	CA_LOB01	43
TABLE PARTITION	ERROR	CRIS_PART	66
INDEX PARTITION	ERROR	CRIS_PART	66
TABLE SUBPARTITION	ERROR	SECACT	1
INDEX SUBPARTITION	ERROR	SEC_NU02	2
INDEX SUBPARTITION	ERROR	SEC_NU05	2
INDEX SUBPARTITION	ERROR	CA_DATA01	1
TABLE	ERROR	CRIS_PART	2
LOBINDEX	ERROR	DCSW_DATA02	16
TABLE SUBPARTITION	ERROR	SECCLOSE	1
INDEX SUBPARTITION	ERROR	SEC_NU06	2
INDEX SUBPARTITION	ERROR	SEC_NU04	2

Figura 12. Conjunto de objetos asignados fuera de su tablespace recomendado.

La consulta utilizada para obtener la tabla anterior es:

```
select * from
(select segment_type,
(case when lpad(segment_type,5,0)='TABLE' and instr(tablespace_name,'DAT')>0 then 'TABLE'
when lpad(segment_type,5,0)='INDEX' and (instr(tablespace_name,'IND')>0 or
instr(tablespace_name,'IDX')>0) then 'INDEX'
when lpad(segment_type,3,0)='LOB' and instr(tablespace_name,'LOB')>0 then 'LOB'
else 'ERROR'
end) type,
tablespace_name, count(1) num_objects from dba_segments
where
owner in ('NRTDWH','DCSW_OWNER')
group by segment_type, tablespace_name) obj
where obj.type = 'ERROR'
```

Evaluación de NRT-DWH

3.3.4.8 DM08: COLUMNAS DE ID CONCATENADAS AUTOMÁTICAMENTE

Algunas columnas de la aplicación se rellenan concatenando un conjunto de valores separados por comas. Puede producir errores de desbordamiento si la cantidad de valores es alta. Un ejemplo es la columna RCM_ID en la tabla TRA como se muestra a continuación.

TRA_SIT_ID	TRA_TRANSACT_TYPE	TRA_TRANS_NO	RCM_ID
T2	Y	00000000004	000000722891,000000746236,000000746620,000000746769
T2		00000000004	000000746895,000000747060
TL	Y	00000000005	000000701327
T2	Y	00000000005	000000723036,000000724441,000000746242,000000746627,000000746794
T5	Y	00000000005	000022442932
T4	Y	00000000006	000001447870
TL	Y	00000000006	000000701329
T2	Y	00000000006	000000723038,000000743092,000000746244,000000746629,000000746796
T2	Y	00000000007	000000723040,000000744709,000000746246,000000746818,000000746799
TL	Y	00000000007	000000701668

Figura 13. Ejemplo de conjunto de campos rellenos con valores concatenados automáticamente.

3.3.4.9 DM09 – TABLAS POTENCIALES CON DATOS DUPLICADOS

Hay algunas tablas que pueden tener una fecha potencialmente duplicada. Algunos ejemplos de este problema pueden ser TRA / TRG y también tablas TEV / TEG.

3.3.4.10 DM10 – REVISIÓN DE PK

Algunas tablas del modelo de datos usan la columna de ubicación como parte de su clave principal. Sería interesante si es necesario en tablas como País o Moneda, por ejemplo.

COU_CODE	COU_SHORT	COU_SIT_ID
MM	MYANMAR	T4
MM	MYANMAR	IG
NA	NAMIBIA	TG
NA	NAMIBIA	TL
NA	NAMIBIA	T3
NA	NAMIBIA	I2
NA	NAMIBIA	T2
NA	NAMIBIA	I4
NA	NAMIBIA	T4
NA	NAMIBIA	I3
NA	NAMIBIA	TE
NA	NAMIBIA	IG
NA	NAMIBIA	IE
NA	NAMIBIA	IL

Figura 14. Ejemplo de conjunto de revisión de PK.

3.4 CALIDAD DE DATOS

La calidad de los datos^[7] permite evaluar los datos para garantizar que los atributos se proporcionen como integridad (se proporcionan atributos esperados), formatos de datos (metadatos) y corrección (los datos describen correctamente la entidad), coherencia (los datos en toda la empresa deben sincronizarse entre sí), la puntualidad (datos correctos para la persona correcta en el momento adecuado) y algunos otros.

3.4.1 SITUACIÓN ACTUAL

El sistema actual contiene algunos procesos para aplicar reglas de calidad de datos. Algunos de ellos requieren volver a procesar los mensajes de entrada para actualizar los registros de base de datos con los valores correctos.

Evaluación de NRT-DWH

3.4.2 PROBLEMAS DETECTADOS

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DQ01	Lógico	Falta de procesos de calidad de datos	NRT-DWH no contiene procesos de calidad de datos para mejorar la fiabilidad de los datos. Generación de perfiles de datos para garantizar que faltan formatos de datos o integridad de datos.	Dificultad para alinear los datos con las prácticas recomendadas de calidad de datos. Baja fiabilidad de los datos. Difícil entender cómo se pueden rellenar las entidades, las relaciones y el contenido de los datos y cómo se relacionan.	Muy Alto
DQ02	Lógico	Columnas con todos los valores NULL	En la mayoría de los casos hay columnas completamente llenas de valores NULL.	Impactos en el almacenamiento. Las buenas prácticas recomiendan evitar campos no requeridos dentro del modelo.	Medio
DQ03	Lógico	Columnas clave potenciales	Una columna puede ser un PK potencial si su índice de unicidad es cercano al 100%	El rendimiento para acceder a los datos pudo ser bajo.	Medio
DQ04	Lógico	Formato de valores de datos	Algunos datos se pueden cargar con un formato diferente al esperado. Es decir: las fechas como tipo de datos varchar y valores decimales con baja precisión pueden producir malos cálculos finales.	Impactos en la calidad de los datos y, a veces, en el almacenamiento y el rendimiento. Las fechas almacenadas como cadena requieren más recursos cuando se usan como filtro.	Alto
DQ05	Lógico	Problemas de datos	Faltan datos necesarios o diferentes de lo esperado. En algunos casos se requiere una actualización manual para ajustar el valor.	Impactos en la calidad de los datos y directamente en la confianza descendente en el sistema	Medio
DQ06	Lógico	Columnas no correspondibles dentro de una entidad	Cada entidad dentro del modelo debe tener sólo un conjunto de columnas necesarias	Impactos en el almacenamiento. Las buenas prácticas recomiendan tener solo los campos obligatorios para cada entidad.	Alto
DQ07	Lógico	Faltan datos del instrumento	Los datos del instrumento para los procesos de transacción y posición provienen de los Upstreams, pero no se ingestan.	Los datos del instrumento no pueden ser utilizados o consumidos por los niveles inferiores porque no existen en la base de datos.	Alto

Evaluación de NRT-DWH

3.4.3 MEJORAS PROPUESTAS

Problema #	Resumen del Problema	Solución Propuesta	Complejidad
DQ01	Falta de procesos de calidad de datos	Crear un procedimiento estratégico que proporcione un conjunto de acciones para mejorar la calidad de los datos. Crear un glosario para definir los metadatos correctos para comprender las entidades, las relaciones y el contenido de los datos.	Alto
DQ02	Columnas con todos los valores NULL	Revisar todas las columnas completamente cargadas con valores NULL y elimínelas del sistema si no es necesario para mejorar el rendimiento.	Bajo
DQ03	Columnas clave potenciales	Utilizar las columnas seleccionadas como PK en la tabla correspondiente	Medio
DQ04	Formato de valores de datos	Cambiar el tipo de datos en las posibles columnas detectadas	Alto
DQ05	Problemas de datos	Analizar la causa raíz por la que falta el valor o es diferente de lo esperado e intente resolverlo mediante una nueva carga o recálculo.	Alto
DQ06	Columnas no correspondibles dentro de una entidad	Eliminar todos los campos no requeridos en todas las tablas del modelo de datos.	Alto
DQ07	Faltan datos del instrumento	Revisar el modelo de datos para incorporar los datos si son requeridos por los niveles inferiores.	Alto

3.4.4 EJEMPLOS DE PROBLEMAS

3.4.4.1 DQ02 – COLUMNAS CON TODOS LOS VALORES NULL

Hay columnas en las tablas del modelo que están completamente vacías y Las buenas prácticas recomiendan evitarlo. A continuación se muestra un ejemplo.

TRA_SIT_ID	TRA_TRANSACTION_TYPE	TRA_TRANS_NO	TRA_LINK_REF	TRA_INACTIVE_DATE	TRA_CLEARING_SERVICE
TD		00033955410	(null)	(null)	(null)
TD		00033955416	(null)	(null)	(null)
TD		00033955418	(null)	(null)	(null)
TD		00033955420	(null)	(null)	(null)
TD		00033955422	(null)	(null)	(null)
TD		00033955424	(null)	(null)	(null)
TD		00033955426	(null)	(null)	(null)
TD		00033955428	(null)	(null)	(null)
TD		00033955430	(null)	(null)	(null)
TD		00033955432	(null)	(null)	(null)
TD		00033955434	(null)	(null)	(null)
TD		00033955436	(null)	(null)	(null)
TD		00033955438	(null)	(null)	(null)

Figura 15. Columnas NULL de la tabla de transacciones.

3.4.4.2 DQ03 – COLUMNAS CLAVE POTENCIALES

Las columnas de clave potenciales son aquellas que todos los valores de su contenido son diferentes. La siguiente consulta obtiene las columnas de clave potenciales dentro del modelo de datos de NRT-DWH.

```

select
    tcs.owner, tcs.table_name, tcs.column_name, tcs.num_distinct, tts.num_rows, case when
    tts.num_rows = 0 then 0 else round(tcs.num_distinct/tts.num_rows*100,2) end pct_uni
from dba_tab_col_statistics tcs, dba_tab_statistics tts
where
    tcs.table_name=tts.table_name
    and tcs.owner=tts.owner
    and tcs.owner in ('NRTDWH','DCSW_OWNER')
    and tcs.table_name in ('TRA', 'TRG', 'TEV', 'POS', 'POB', 'TEG', 'PSI', 'PRI', 'COU', 'IND', 'MKT', 'SEC', 'ICM',
    'CCY', 'MST_TEMP', 'MST',
    'LOC', 'ACC', 'LCS', 'CASH_TRANSACTIONS', 'SIT')
    and tts.partition_name is null
order by 6 desc
    
```

OWNER	TABLE_NAME	COLUMN_NAME	NUM_DISTINCT	NUM_ROWS	PCT_UNI
CRIS	SIT	SIT_DESC	76	76	100
CRIS	SIT	SIT_ID	76	76	100
DCSW_OWNER	CASH_TRANSACTIONS	CASH_UUID	155193	155193	100
CRIS	TEV	RCM_ID	7084173	7084173	100
DCSW_OWNER	CASH_TRANSACTIONS	CASH_TNX_ID	155193	155193	100
CRIS	TEG	RCM_ID	1850035	1850035	100
CRIS	TRG	TRG_TRANS_NO	2571264	2581847	99.59
DCSW_OWNER	CASH_TRANSACTIONS	CASH_TNX_NO	153584	155193	98.96
CRIS	ACC	ACC_ID	217328	219756	98.9
CRIS	SEC	RCM_ID	24637440	24951612	98.74
CRIS	ACC	ACC_ID_ORG	216544	219756	98.54
CRIS	TRA	TRA_TRANS_NO	2977024	3025189	98.41
CRIS	SEC	SEC_ID	24502272	24951612	98.2
DCSW_OWNER	CASH_TRANSACTIONS	CASH_DCS_REF	151936	155193	97.9
CRIS	SEC	SEC_WPK	24141824	24951612	96.75

Figura 16. Columnas clave potenciales.

Evaluación de NRT-DWH

3.4.4.3 DQ04 – FORMATO DE VALORES DE DATOS

Algunos datos se pueden almacenar con un formato diferente al necesario. Las buenas prácticas recomiendan alinear los tipos de datos de columna con el contenido esperado. Los números con más precisión de la necesaria o las fechas almacenadas como una cadena, por ejemplo, podrían producir problemas de rendimiento y de inconsistencia de datos.

3.4.4.3.1 LONGITUD DEL CAMPO DE CADENA SUPERIOR AL CONTENIDO

En esta sección se muestra un ejemplo en el que la longitud del campo nunca se rellena por completo. Se recomienda ajustar los tipos de datos al contenido del valor máximo.

TABLE_NAME	COLUMN_NAME	MAX_LENGTH	DATA_TYPE	DATA_LENGTH
ACS	ACS_ID_NID	7	VARCHAR2	12

Figura 17. Longitud del campo de cadena superior al contenido.

La siguiente consulta se utilizó para obtener el ejemplo:

```
Select
    col.table_name as table_name, col.column_name as column_name,
    max(length(acs_id_nid)) as max_length, col.data_type as data_type,
    col.data_length as data_length
from NRTDWH.acs ca
join
(select
    table_name, column_name, data_type, data_length
from dba_tab_columns
where
    owner = 'NRTDWH'
    and table_name = 'ACS'
    and column_name = upper('acs_id_nid')) col
on col.table_name='ACS'
group by col.table_name, col.column_name, col.data_type, col.data_length
```

3.4.4.3.2 LONGITUD DE CAMPO SUPERIOR AL CONTENIDO Y NO ALINEADA CON UN POSIBLE FK

A continuación se muestran los resultados del ejemplo donde la precisión del número de campo es diferente de lo esperado. Las buenas prácticas recomiendan ajustar los tipos de datos y la precisión al contenido del valor máximo.

TABLE_NAME	COLUMN_NAME	MAX_LENGTH	DATA_TYPE	DATA_PRECISION	DATA_SCALE	ORIG_TABLE_NAME	ORIG_COLUMN_NAME	ORIG_DATA_TYPE	ORIG_DATA_PRECISION	ORIG_DATA_SCALE
ACC	ACC_INDUSTRY_CODE	10	NUMBER	22	0	IND_CODE	IND_CODE	NUMBER	10	0

Figura 18. Longitud de campo superior al contenido y no alineada con una posible FK.

Evaluación de NRT-DWH

Y esta es la consulta utilizada para obtener el ejemplo:

```
select
col.table_name as table_name, col.column_name as column_name,
max(length(ca.ACC_INDUSTRY_CODE)) as max_length, col.data_type as data_type,
col.data_precision as data_precision, col.data_scale, col2.table_name as orig_table_name,
col2.column_name as orig_column_name, col2.data_type as orig_data_type,
col2.data_precision as orig_data_precision, col2.data_scale as orig_data_scale
from
NRTDWH.acc ca
join
(select
table_name, column_name, data_type, data_precision, data_scale
from dba_tab_columns
where
owner ='NRTDWH'
and table_name = 'ACC'
and column_name = upper('ACC_INDUSTRY_CODE')) col
on col.table_name='ACC'
join
(select
table_name, column_name, data_type, data_precision, data_scale
from dba_tab_columns
where
owner ='NRTDWH'
and table_name = 'IND'
and column_name = upper('IND_CODE')) col2
on col2.table_name = 'IND'
group by
col.table_name, col.column_name, col.data_type, col.data_precision, col.data_scale,
col2.table_name, col2.column_name, col2.data_type, col2.data_precision, col2.data_scale
```

3.4.4.3.3 FECHAS POTENCIALES EN FORMATO DE CADENA

En esta sección hay algunos ejemplos donde el campo contiene fechas potenciales en formato de cadena.

KEY	TABLE_NAME	COLUMN_NAME	MAXLENGTH	DATA_TYPE	DATA_LENGTH
STP_ORIG_TRADE_DAT	STP	STP_ORIG_TRADE_DAT	8	VARCHAR2	50
STP_ORIG_SETTL_DAT	STP	STP_ORIG_SETTL_DAT	8	VARCHAR2	50
STP_ORIG_COUPON_DAT	STP	STP_ORIG_COUPON_DAT	8	VARCHAR2	50

Figura 18.1. Fechas potenciales en formato de cadena (estructura).

STP_ORIG_TRADE_DAT	STP_ORIG_SETTL_DAT	STP_ORIG_COUPON_DAT
20180720	20180724	20171016
20180720	20180724	(null)
20180720	20180725	(null)

Figura 18.2. Fechas potenciales en formato de cadena (datos).

Evaluación de NRT-DWH

Y la consulta correspondiente para comparar la estructura.

```
select mx.key,
col.table_name as table_name, col.column_name as column_name, mx.maxlength,
col.data_type as data_type, col.data_length as data_length
from
(select max(length(STP_ORIG_TRADE_DAT)) as STP_ORIG_TRADE_DAT, max(length(STP_ORIG_SETTL_DAT)) as
STP_ORIG_SETTL_DAT, max(length(STP_ORIG_COUPON_DAT)) as STP_ORIG_COUPON_DAT from NRTDWH.stp)
unpivot ( maxlength for key in(
STP_ORIG_TRADE_DAT,
STP_ORIG_SETTL_DAT,
STP_ORIG_COUPON_DAT)) mx
join
(select table_name, column_name,data_type, data_length
from
dba_tab_columns
where
owner ='NRTDWH'
and table_name = 'STP'
and column_name in ('STP_ORIG_TRADE_DAT', 'STP_ORIG_SETTL_DAT', 'STP_ORIG_COUPON_DAT' )) col
on mx.key=col.column_name;
```

3.4.4.4 DQ05 – PROBLEMAS DE DATOS

En cuanto a los problemas de datos, hay algunos ejemplos que investigar. Uno de ellos puede ser los datos que faltan que se pueden resolver mediante una ejecución de reproceso.

Por otro lado, algunas tablas de referencia con datos que se van a evaluar. Las buenas prácticas recomiendan evitar los datos duplicados y los valores no relacionados con la entidad a la que pertenecen.

- Datos duplicados dentro de las tablas de referencia. El código de país debería ser el mismo sea cual sea la ubicación de la que proviene el mensaje.

COU_CODE	COU_SHORT	COU_SIT_ID	RCM_ID
MM	MYANMAR	T4	000001398270,000001408963,000001417596
MM	MYANMAR	IG	,000000204315
NA	NAMIBIA	TG	,000000204329
NA	NAMIBIA	TL	000000687261
NA	NAMIBIA	T3	000000781245,000000789626,000000797944,000000806071,000000814450,000000822611,000000830782,000000923305
NA	NAMIBIA	I2	000000706153
NA	NAMIBIA	I2	000000706153
NA	NAMIBIA	I4	000001398284,000001408977,000001417610
NA	NAMIBIA	T4	000001398284,000001408977,000001417610
NA	NAMIBIA	I3	000000781245,000000789626,000000797944,000000806071,000000814450,000000822611,000000830782,000000923305
NA	NAMIBIA	TE	,0000C2D5A7B4331A00019984714
NA	NAMIBIA	IG	,000000204329
NA	NAMIBIA	IE	,0000C2D5A7B4331A00019984714
NA	NAMIBIA	IL	000000687261

Figura 19. Ejemplo de datos duplicados.

Evaluación de NRT-DWH

- Datos inesperados relacionados con el contenido de la entidad. Hay datos dentro de la dimensión de país que no corresponden realmente a un país.

COU_CODE	COU_SHORT	COU_SIT_ID	RCM_ID
NQ	NASDAQ	IL	000000867833,000000943432
NQ	NASDAQ	TL	000000867833,000000943432
71	NATO	IO	000000000000000088906173916
71	NATO	TE	0000C2D5A7B479E100019984507
71	NATO	IE	0000C2D5A7B479E100019984507

Figura 20. Datos inesperados relacionados con el contenido de una entidad.

3.4.4.5 DQ06 – COLUMNAS NO INVESTIGADAS DENTRO DE UNA ENTIDAD

RCM_ID columna se rellena en algunas tablas con un conjunto de valores concatenados por coma. Si no se requiere el uso de este campo, elimínalo puede mejorar el almacenamiento y el rendimiento. Es decir: Las tablas País (COU) y Moneda (CCY) contienen este campo RCM_ID.

COU_CODE	COU_SHORT	COU_SIT_ID	RCM_ID	COU_COUNTRY_CODE_ISO3166_1
82	AFDB	TE	,0000C2D5A7B47E1C00019984515	(null)
82	AFDB	IO	000000000000000088986173924	(null)
AF	AFGHANISTAN	IR	,00000000636453	(null)
AF	AFGHANISTAN	TR	,00000000636453	(null)
AF	AFGHANISTAN	IE	,0000C2D5A7B380E600019984549	(null)
AF	AFGHANISTAN	T2	000000705998	(null)
AF	AFGHANISTAN	I2	000000705998	(null)
AF	AFGHANISTAN	IG	,000000204174	(null)
AF	AFGHANISTAN	TL	000000687106	(null)
AF	AFGHANISTAN	IL	000000687106	(null)
AF	AFGHANISTAN	TG	,000000204174	(null)
AF	AFGHANISTAN	I4	000001398129,000001408822,000001417455	(null)
AF	AFGHANISTAN	T4	000001398129,000001408822,000001417455	(null)
AF	AFGHANISTAN	TE	,0000C2D5A7B380E600019984549	(null)
AF	AFGHANISTAN	T3	000000781090,000000789471,000000797789,000000805916,000000814295,000000822456,000000830627,000000923149	(null)
AF	AFGHANISTAN	I3	000000781090,000000789471,000000797789,000000805916,000000814295,000000822456,000000830627,000000923149	(null)
AL	ALBANIA	T3	000000781093,000000789474,000000797792,000000805919,000000814298,000000822459,000000830630,000000923152	(null)

Figura 21. Posibles columnas no investigadas.

3.4.4.6 DQ07 – FALTAN DATOS DE INSTRUMENTOS

Hay algunos datos **del instrumento** que vienen de los Upstreams pero no se ingestan en la base de datos NRT-DWH esta es la razón por la cual las downsatreams no pueden utilizar esta información.

Los campos proporcionados por los Upstreams son:

- InstrumentID
- InstrumentIDType
- InstructingBranchBIC

Los eventos que utilizan esta información como origen, son los siguientes y todos ellos son la última versión de cada tipo de mensaje excepto TransPending-2.3.

- Position-0.8.dtd
- TransAutoSett-2.2.dtd
- TransAutoSettUX-1.1.dtd
- TransDirectSettleFree-2.0.dtd
- TransDirectSettlePay-2.1.dtd
- TransDispoStatus-0.8.dtd
- TransMatch-0.6.dtd
- TransPending-2.3.dtd**

Evaluación de NRT-DWH

- **TransPending-2.5.dtd**
- TransTradeConf-2.2.dtd
- TransTransfer-1.6.dtd
- TransTransferUX-1.1.dtd

Es curioso que el tipo de mensaje TransPending utilice la información con la versión 2.3, se la pierda con la versión 2.4 y finalmente los datos se utilicen de nuevo en la última versión 2.5.

3.5 SEGURIDAD DE DATOS

La seguridad de los datos hace referencia a las acciones de usuario permitidas y no permitidas en la base de datos y los objetos que contiene.

3.5.1 SITUACIÓN ACTUAL

Después de revisar la documentación de Confluence^[8] con respecto a las directrices de desarrollo de bases de datos, algunos temas deben tratarse en esta sección.

Las páginas de confluence revisadas son:

- <https://confluence.intranet.nrtldwh.com/display/WBC/Coding+standards>
- <https://confluence.intranet.nrtldwh.com/display/WBC/SQL+Script+creation+standards>

Los temas que se tratarán en los problemas detectados se agrupan por:

- Grants
- Sinónimos
- Tablas de auditoría

Evaluación de NRT-DWH

3.5.2 PROBLEMAS DETECTADOS

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DS01	Grants	Grants asignados directamente a los usuarios	Una gran cantidad de Grants se asignan al usuario directamente, en lugar del rol asociado.	Difícil crear nuevos usuarios asignados a roles y obtener todos los permisos organizados correctamente. Los objetos de base de datos pueden haber perdido permisos o extra.	Medio
DS02	Grants	Política de Grants para el nuevo desarrollo	No se especifica ninguna directiva de concesión cuando un desarrollador necesita implementar cambios que impliquen nuevos objetos de base de datos. Solo la cláusula que las concesiones deben asignarse a roles en lugar de usuarios.	Se necesita comunicación adicional para pensar en las concesiones correctas para los roles requeridos. Cada vez que se crea un nuevo objeto hay un ejercicio de proporcionar las concesiones correctas.	Medio
DS03	Sinónimos	Sinónimos públicos vs sinónimos privados	La mayoría de los sinónimos de la base de datos NRT-DWH son privados. Solo algunos de ellos son sinónimos públicos, por lo que se puede usar el mismo alias para todos los esquemas. Los sinónimos públicos solo se crean una vez para cada objeto, pero podrían causar problemas de espacio de nombres en los diferentes esquemas	Es un asunto de seguridad de datos pensar en los sinónimos (público vs , que podrían mostrar un conflicto en diferentes esquemas en caso de sinónimos públicos. La base de datos NRT-DWH se utiliza para varias aplicaciones y podría generar problemas con sinónimos públicos si los espacios de nombres no son administrados por la misma persona.	Bajo
DS04	Sinónimos	Política de sinónimos para el nuevo desarrollo	No se especifica ninguna directiva de sinónimos cuando un desarrollador necesita implementar cambios. Qué sinónimos son necesarios para la nueva implementación, con una política clara	Se necesita comunicación adicional para definir los sinónimos cada vez que se crea un nuevo objeto. Sinónimos perdidos para los objetos existentes.	Medio
DS05	Auditoría	Definir una directiva de auditoría para tablas importantes	Las tablas de auditoría proporcionan la capacidad de investigar la causa raíz de los problemas de calidad de los datos. Algunos datos de NRT-DWH tienen tablas de auditoría, pero esta población podría aumentarse.	Registros de producción actualizados y sin trazabilidad de los cambios. Falta de claves para investigar problemas de datos.	Alto

Evaluación de NRT-DWH

3.5.3 MEJORAS PROPUESTAS

Problema #	Resumen del Problema	Solución Propuesta	Coste
DS01	Grants asignados directamente a los usuarios	Investigar todos los casos con Grants asignados directamente a los usuarios y proponer los roles correctos para proporcionar los privilegios	Alto
DS02	Política de Grants para el nuevo desarrollo	Definir una directiva "Grants" en objetos de base de datos y documento en Confluence . Si las conclusiones se pueden automatizar, podría haber una herramienta de "Grants automáticos", con un criterio para conceder y revocar en consecuencia. En ese caso, el desarrollador no definiría las concesiones directamente, sino sólo los parámetros correctos a la herramienta, con el fin de "auto-asignar" o revocar las concesiones	Medio
DS03	Sinónimos públicos vs sinónimos privados	Obtener los sinónimos públicos de VCTL_OWNER y AUD_OWNER objetos y determine qué usuarios deben tener sinónimos privados.	Bajo
DS04	Política de sinónimos para el nuevo desarrollo	Definir una directiva de "sinónimos" en objetos de base de datos y documentar en Confluence. Si la directiva es lo suficientemente específica, implemente una herramienta de "Auto-synonyms", para proporcionar los sinónimos correctos con un script genérico, en lugar de crear los sinónimos privados en el script desarrollado (implementación de SQL).	Medio
DS05	Definir una directiva de auditoría para tablas importantes	Revisar las tablas importantes de NRT-DWH y defina tablas de auditoría, rellenas por triggers, para realizar un seguimiento de la información clave. Las políticas de "auditoría de grano fino" de Oracle pueden ayudar a implementar mecanismos genéricos para la auditoría.	Alto

3.5.4 EJEMPLOS DE PROBLEMAS

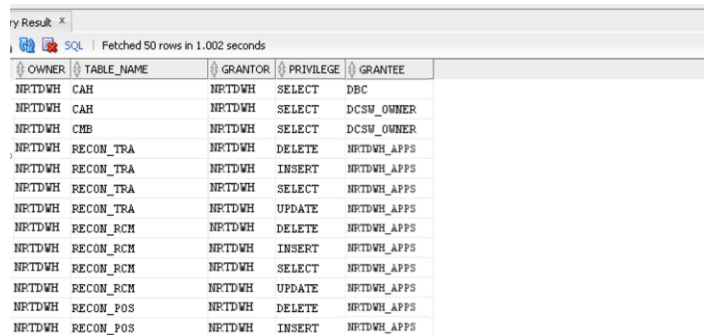
3.5.4.1 DS01 – GRANTS ASIGNADOS DIRECTAMENTE A LOS USUARIOS

La siguiente consulta muestra los permisos directos asignados a un usuario para permitir acciones para cada tabla.

```
select owner, table_name, grantor, privilege, grantee  
from DBA_TAB_PRIVS  
where owner in ('NRTDWH', 'NRTDWH_ALL') and grantee in (select username from dba_users)
```

es decir: Asignar roles a NRTDWH_UPDATE_ROLE, NRTDWH_READONLY_ROLE en lugar de NRTDWH_APPS (usuario de la aplicación)

Resultado:



OWNER	TABLE_NAME	GRANTOR	PRIVILEGE	GRANTEE
NRTDWH	CAH	NRTDWH	SELECT	DBC
NRTDWH	CAH	NRTDWH	SELECT	DCSM_OWNER
NRTDWH	CHB	NRTDWH	SELECT	DCSM_OWNER
NRTDWH	RECON_TRA	NRTDWH	DELETE	NRTDWH_APPS
NRTDWH	RECON_TRA	NRTDWH	INSERT	NRTDWH_APPS
NRTDWH	RECON_TRA	NRTDWH	SELECT	NRTDWH_APPS
NRTDWH	RECON_TRA	NRTDWH	UPDATE	NRTDWH_APPS
NRTDWH	RECON_RCM	NRTDWH	DELETE	NRTDWH_APPS
NRTDWH	RECON_RCM	NRTDWH	INSERT	NRTDWH_APPS
NRTDWH	RECON_RCM	NRTDWH	SELECT	NRTDWH_APPS
NRTDWH	RECON_RCM	NRTDWH	UPDATE	NRTDWH_APPS
NRTDWH	RECON_POS	NRTDWH	DELETE	NRTDWH_APPS
NRTDWH	RECON_POS	NRTDWH	INSERT	NRTDWH_APPS

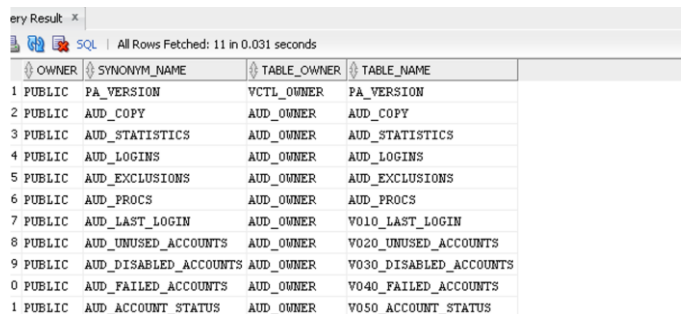
Figura 22. Permisos asignados directamente a los usuarios.

3.5.4.2 DS02 – SINÓNIMOS PÚBLICOS

La siguiente consulta recupera los sinónimos públicos para los propietarios de tablas de aplicación.

```
select owner, synonym_name, table_owner, table_name  
from dba_synonyms  
where owner = 'PUBLIC' and table_owner not like '%SYS%' and table_owner not in ('ORACLE')
```

Resultado:



OWNER	SYNONYM_NAME	TABLE_OWNER	TABLE_NAME
PUBLIC	PA_VERSION	VCTL_OWNER	PA_VERSION
PUBLIC	AUD_COPY	AUD_OWNER	AUD_COPY
PUBLIC	AUD_STATISTICS	AUD_OWNER	AUD_STATISTICS
PUBLIC	AUD_LOGINS	AUD_OWNER	AUD_LOGINS
PUBLIC	AUD_EXCLUSIONS	AUD_OWNER	AUD_EXCLUSIONS
PUBLIC	AUD_PROCS	AUD_OWNER	AUD_PROCS
PUBLIC	AUD_LAST_LOGIN	AUD_OWNER	V010_LAST_LOGIN
PUBLIC	AUD_UNUSED_ACCOUNTS	AUD_OWNER	V020_UNUSED_ACCOUNTS
PUBLIC	AUD_DISABLED_ACCOUNTS	AUD_OWNER	V030_DISABLED_ACCOUNTS
PUBLIC	AUD_FAILED_ACCOUNTS	AUD_OWNER	V040_FAILED_ACCOUNTS
PUBLIC	AUD_ACCOUNT_STATUS	AUD_OWNER	V050_ACCOUNT_STATUS

Figura 23. Sinónimos públicos.

3.6 RENDIMIENTO

Esta sección pretende ser un análisis de algunos puntos de la aplicación NRT-DWH en los que el rendimiento es una cuestión de debate. En general, el rendimiento es aceptable en el programa y no hay grandes problemas en términos de desarrollo. Hay consultas específicas que se pueden mejorar, aunque es un área que ya está cubierta en el equipo actual de NRT-DWH.

Aquí es útil proporcionar algunos comentarios sobre grandes procesos donde se pueden desarrollar mejoras en caso de que el rendimiento se convierta en un problema en el futuro. Estamos hablando de escalabilidad horizontal, en comparación con el enfoque actual.

Revisión del rendimiento:

- **Procesamiento paralelo simultáneo:** los eventos XML se escuchan desde las colas de entrada ascendentes, pero los mensajes se redirigen a las colas internas NRT-DWH para el procesamiento simultáneo. La decisión se toma sobre la base de reglas para distribuir los mensajes al destino interno correspondiente.
- **Ingestión de datos de extremo a extremo:** en el enfoque actual, se reciben los eventos XML y el siguiente evento para la misma cola no se puede procesar hasta que finalice el actual. Un gran evento XML enviado desde sistemas ascendentes que afecta a una gran cantidad de filas podría producir un cuello de botella, ya que debe esperar hasta que finalicen las fases de ingestión, triggers y postprocesamiento.

3.6.1 SITUACIÓN ACTUAL

Esta sección describe la situación actual para algunos procesos en NRT-DWH que son una cuestión de discusión en términos de rendimiento:

3.6.1.1 PROCESAMIENTO PARALELO SIMULTÁNEO

La siguiente imagen muestra cómo se procesan los eventos simultáneamente durante el proceso de ingestión:

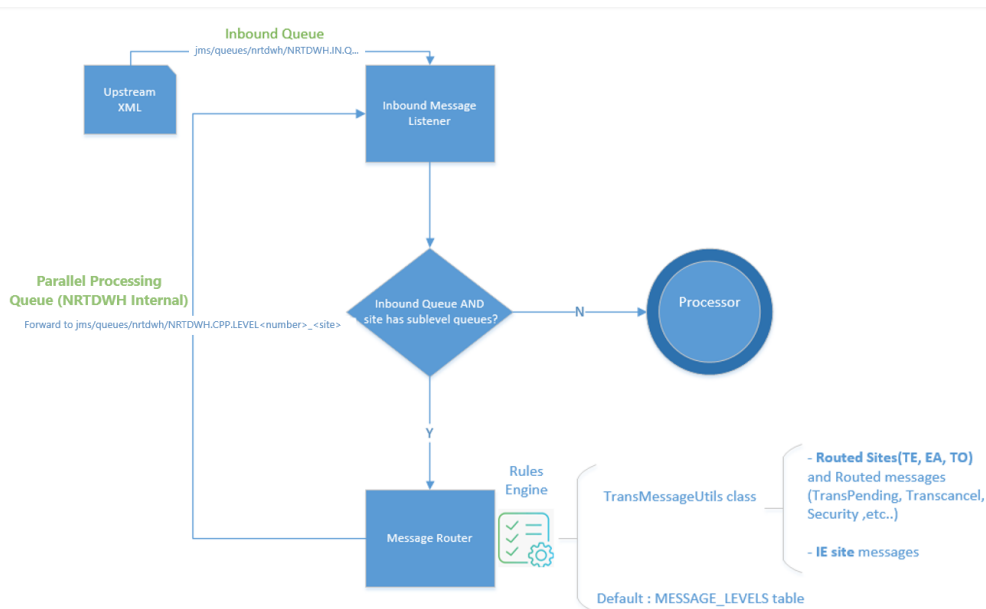


Figura 24. Estado actual "Procesamiento paralelo simultáneo"

- **Escucha de Mensajes de entrada:** la clase `InboundMsgListener` lee los mensajes de las colas de entrada, que comienzan con `.jms/queues/nrtldwh/NRTDWH.IN.Q`. Sólo en caso de que el remitente (Ubicación) tenga colas de subnivel para el procesamiento paralelo (Ubicaciones TE, EA y TQ) el mensaje será reorientado, de lo contrario será procesado directamente. Si la primera condición se corresponde con, después el router de mensajes tendrá la responsabilidad de reenviar el mensaje.
- **MessageRouter** : dado el nombre DTD y el remitente la decisión de la cola interna para procesar el mensaje se toma dentro de `MessageRouter`. En caso de que sea un mensaje `EndOfDay`, se detendrán todas las colas de subniveles.

En este punto está la complejidad. El **motor de reglas** para obtener el destino MQ correcto se toma de varios puntos: reglas codificadas en `MessageRouter`, clase `Java TransMessageUtils` y el contenido de `MESSAGE_LEVELS` tabla para su distribución.

Este código se puede encontrar en `MessageRouter.getSublevelQueueJNDI`:

- 1) Si el mensaje proviene de ubicaciones enrutadas (TE, EA, TO) y es un mensaje enrutado (`TransCancel`, `Transfailed`, `TransPending`, `Security`, etc...) `TransMessageUtils` será el responsable de asignar la cola correcta.
- 2) Si el mensaje procede de la ubicación IE, **`TransMessageUtils`** también será la clase que asignará la cola.
- 3) De lo contrario, las reglas se ejecutan en **`MESSAGE_LEVELS`** tabla. Esto se hace desde `NrtldwhMDBDAO.getRoutedSublevelQueue`. Básicamente asigna la cola correcta mediante la recuperación del patrón `"sharedLevelQueue-<sender><level>:<sender><level>"`

A continuación, el mensaje se redirigirá a la cola específica (`"jms/queues/nrtldwh/NRTDWH.Cpp.LEVEL<number>_<site>"`).

- IE : NIVEL 0
- TO : NIVEL 0
- EA : NIVEL 0 a 6
- TE : NIVEL 0 a 16
- TQ : NIVEL 0 a 5.

En resumen, el proceso de ingestión simultánea se obtiene **redireccionando los mensajes a colas INTERNAS de MQ** y leyendo todas ellas. El objetivo era obtener las dependencias entre los mensajes y procesarlos en grupos, algún tipo de "Message Groups". Este enfoque puede tener algunos puntos calientes en términos de escalabilidad. Supongamos que los volúmenes se incrementan en gran medida para una ubicación o en general, y las reglas serán difíciles de administrar. Veremos en la siguiente sección, cómo centralizar el motor de reglas para encontrar una forma de controlar el procesamiento paralelo, sin crear colas MQ adicionales.

Evaluación de NRT-DWH

3.6.2 PROBLEMAS DETECTADOS

Problema #	Resumen del Problema	Descripción del problema	Consecuencias	Impacto
DP01	Procesamiento paralelo: el motor de reglas no está centralizado y envía cada mensaje a las colas internas.	Como las reglas para el procesamiento paralelo se distribuyen en varias clases Java y tablas de base de datos, es difícil mantener las reglas y evitar errores. El enfoque actual intenta simular un equilibrio de carga, pero se basa en volúmenes de datos para un único punto en el tiempo. Esto puede cambiar en el futuro. Hay un tiempo que consume para que cada mensaje sea redirigido de nuevo a NRT-DWH..	Difícil mantener el conjunto actual de reglas. La escalabilidad horizontal tiene un problema, porque es necesario crear una nueva infraestructura (colas MQ) para dar cabida al aumento de los volúmenes de datos. Fácil de tener errores al aplicar las reglas con el enfoque actual.	Medio
DP02	Proceso de ingestión: De extremo a extremo.	La ingestión de datos es un proceso de extremo a extremo en el enfoque actual. Esto significa que la conexión en sentido ascendente envía mensajes XML y se procesan hasta el final en un solo paso. Sólo hay una fase.	El proceso de ingestión no se divide en varias partes, por lo que los eventos de la misma cola deben esperar hasta que finalice el evento anterior. En caso de que se envíe un XML grande y se vean afectadas muchas filas, no establecería el procesador libre hasta que el evento esté completamente terminado, pero otros procesos podrían estar ejecutándose en ese momento	Bajo
DP03	Paquete de pruebas de rendimiento que falta	No hay ningún mecanismo para ejecutar la prueba de esfuerzo para los componentes clave en la aplicación NRT-DWH. No se registra ninguna medida sobre el rendimiento del servidor de aplicaciones y de la base de datos para su posterior análisis.	Difícil de predecir problemas de rendimiento en producción cuando hay grandes cargas de datos procedentes de la conexión en sentido ascendente.	Medio

Evaluación de NRT-DWH

3.6.3 MEJORAS PROPUESTAS

Problema #	Resumen del Problema	Solución Propuesta	Complejidad
DP01	Procesamiento paralelo: el motor de reglas no está centralizado y envía cada mensaje a las colas internas.	Motor de reglas centralizado en la base de datos y manejable con lenguaje natural para tener control total de las reglas y dependencias. En lugar de redirigir los mensajes, ejecute las reglas de una tabla de mensajes en línea para recuperar los identificadores que están listos para ser procesados (paralelos) y mantenga los mensajes que necesitan esperar a la siguiente iteración (serie)	Alto
DP02	Proceso de ingestión: De extremo a extremo.	Dividir el proceso de ingestión en varias fases para paralelizar (si es posible) los diferentes procesos. Esto es muy común en las soluciones ETL <ol style="list-style-type: none"> 1) Leer mensaje XML y rellenar tablas de ensayo y tabla RCM (mensaje sin formato) 2) Aplique reglas e ingesta tablas finales. Ingerir también tablas derivadas en esta fase (actualmente relacionadas con triggers). 3) Aplicar cálculo post-procesamiento (posición, saldo, etc...) 	Alto
DP03	Paquete de pruebas de rendimiento	Implementar un paquete de pruebas de rendimiento para la ingestión de datos y simule cargas pesadas para el proceso de fin de día. Monitorizar la aplicación DAP en el servidor Weblogic y analizar las métricas para predecir los posibles problemas de producción.	Medio

3.6.3.1 DP01 - PROCESAMIENTO PARALELO SIMULTÁNEO

La siguiente imagen describe el procesamiento paralelo propuesto:

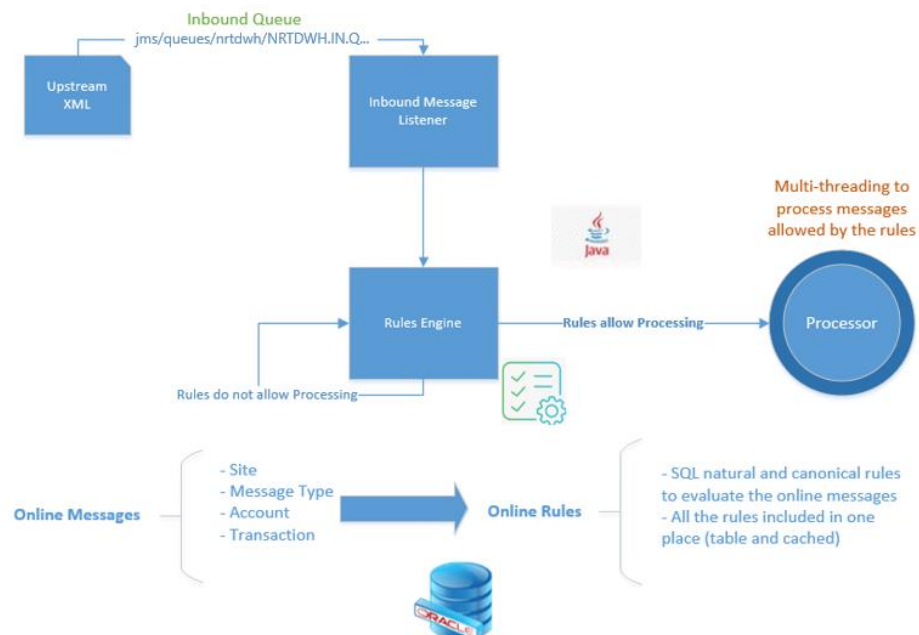


Figura 25. Mejora propuesta "Procesamiento paralelo simultáneo"

- **Escucha de Mensajes de entrada:** la clase InboundMsgListener lee los mensajes de las colas de entrada, que comienzan con `jms/queues/nrtldwh/NRTDWH.IN.Q`.
- **Motor de reglas:** ejecuta las reglas en la tabla `online_messages` y recupera la matriz de id que se permite procesar. El contenido de la matriz se procesa en paralelo abriendo varios subprocesos desde el contexto EJB/Spring para proporcionar simultaneidad.

Ejemplo (sólo una simulación, no el caso real):

- ONLINE_MESSAGE tabla:

Id: 1 ; Msg_type - TransPending ; Ubicación: TE; Account_id - Acc1 ; Trans_number - Trans1

- ONLINE_RULES tabla:

Msg_type = TransPending; Site in TE; Rule text = msg_type in ('Account','Currency', 'Location', 'LocStat')

Este mensaje se validaría con el motor de reglas, por lo que evaluará si hay un mensaje adicional pendiente de procesar antes del actual, que msg_type es Account, Currency, Location or LocStat. No hay ningún mensaje, a continuación, se seleccionará en la matriz resultante de mensajes candidatos que se procesarán.

El motor de reglas evaluará permanentemente el contenido de la tabla ONLINE_MESSAGE para distribuir los eventos que se pueden procesar. Cuando se seleccionan los identificadores, se eliminarán de la tabla `online_message`. Hay que tener en cuenta que la tabla de ONLINE_RULES debe almacenarse en caché para un rendimiento adecuado.

- **Procesador:** dentro del contexto EJB/Spring (si ya se ha migrado) ejecutar los eventos resultantes de la matriz del `messageid` para proporcionar paralelismo. Debe ser lo suficientemente inteligente como para abrir solo los subprocesos necesarios, como una diferencia con el enfoque actual de las colas internas de procesamiento paralelo, que debe crearse ya de antemano. La escalabilidad horizontal mejorará drásticamente con este enfoque.

3.7 GESTIÓN DE CAMBIOS

En esta sección, se describirá la facilidad de incorporar solicitudes de cambio en NRT-DWH. Se asumirá que un cambio estándar afectará al proceso de extremo a extremo desde el mensaje XML enviado por los Upstreams hasta el informe de front-end en el Downstream.

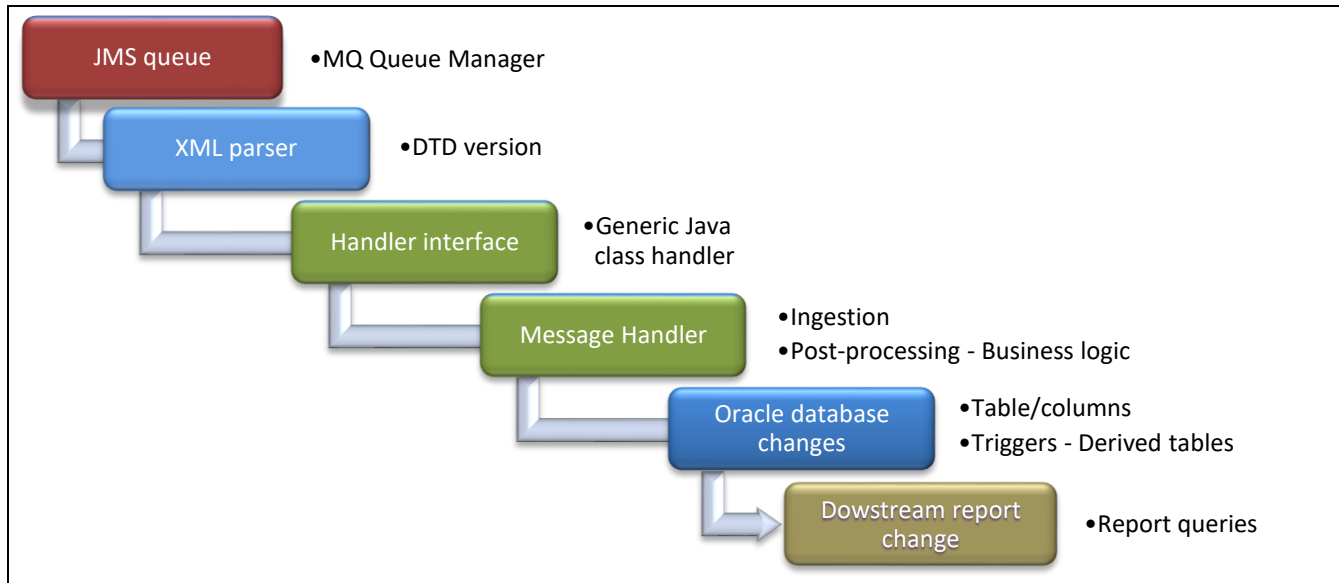
En términos de desarrollo, las solicitudes de cambio están fuertemente relacionadas con las secciones Ingestión de datos, Flujos de datos, Repositorio de datos e Informes posteriores especificados en estos documentos. En otras palabras, las piezas que pertenecen al "ciclo de vida del desarrollo".

Dado este enfoque, la situación actual se describirá en las líneas siguientes para los diferentes componentes, pero los problemas son solo los mismos que se especifican en otras secciones. La estimación y el daño de las "Solicitudes de cambio" se cubren con el resto de los problemas.

Evaluación de NRT-DWH

3.7.1 SITUACIÓN ACTUAL

La siguiente imagen describe los componentes técnicos que se actualizarán en las solicitudes de cambio estándar:



Directrices generales:

1. Los sistemas Downstream solicitan un nuevo atributo/informe.
2. Los expertos de NRT-DWH se en contacto con el Upstream para averiguar si se puede proporcionar el nuevo atributo.
3. En caso de que los Upstreams correspondientes puedan entregar los datos, entonces se incluirán en un nuevo atributo dentro del XML para un tipo de mensaje determinado, que se ingirió finalmente en la base de datos
4. El Downstream modificará o creará el informe correcto para mostrar los nuevos atributos.

3.7.1.1 AGREGUE UNA NUEVA COLA MQ SI SE REQUIERE UNA NUEVA UBICACIÓN

En caso de que se agregue una nueva ubicación a NRT-DWH, se deben realizar las siguientes acciones:

- Crear el MQ en el Gestor de colas necesario.
- Regenerar el archivo de enlace con la nueva cola MQ incluida.
- Desplegar en DAP Weblogic el nuevo destino extranjero.
- Probar con los eventos XML ascendentes.

3.7.1.2 ACTUALIZAR LA VERSIÓN DTD

Dependiendo de las dimensiones del cambio, la nueva versión DTD extenderá la versión anterior o se agregará una nueva.

Nuevo tipo de mensaje: la versión DTD debe seguir el formato <MessageType>-1.0.dtd.

Versión anterior extendida (pequeños cambios): incremente la versión secundaria del último tipo de mensaje.

Crear una nueva versión principal (grandes cambios): incremente la versión principal del último tipo de mensaje.

Todos los DTD se mantienen en " NRT-DWH-DCS\src\main\application\APP-INF\classes\com\dcs\resources\nrtldwh\DTD "

Evaluación de NRT-DWH

3.7.1.3 MODIFICAR OBJETOS DE CONTROLADOR PARA AMBOS FLUJOS

- **TE, TO, flujo de EA (ubicaciones enrutadas):** agregar un nuevo tipo de mensaje en HANDLER_OBJECTS clase para encontrar el controlador de mensajes adecuado, dada la versión dtd del mensaje bajo el paquete [com.dcs.utils.nrtldwh.expedient.utils](#)
- **No TE, TO, flujo de EA (ubicaciones no enrutadas):** No es necesario agregar el nuevo tipo de mensaje en ninguna clase de controlador genérico. Analizará directamente con el analizador SAX con el controlador de mensajes proporcionado, que proporciona la interfaz INrtldwhContentHandler.

3.7.1.4 IMPLEMENTAR CONTROLADOR DE MENSAJES PARA AMBOS FLUJOS

- **TE, TO, flujo de EA (ubicaciones enrutadas):** agregar la versión del controlador de mensajes, que coincidiría con el tipo de mensaje DTD y la entrada en HANDLER_OBJECTS. La clase Java seleccionada se ubicará en el paquete [com.dcs.utils.nrtldwh.expedient.utils](#). Toda la lógica para el proceso de ingestión está contenido en esta clase.
- **No TE, TO, flujo de EA (ubicaciones no enrutadas):** el controlador de mensajes Java seleccionado debe encontrarse en [com.dcs.utils.nrtldwh.xml.msghandlers](#), donde se ejecuta el proceso de ingestión y la lógica empresarial.

Los controladores de mensajes tienen el siguiente formato para ambos flujos:
<MessageType>_<majorversion>_<minorversion>.java bajo proyecto wb-nrtldwh-shared-jar

Implementar la asignación del atributo XML a la columna de base de datos correspondiente.

3.7.1.5 ACTUALIZAR LA IMPLEMENTACIÓN DE LA BASE DE DATOS

- **Modificar tabla:** agregar atributos con el tipo de datos acordado con upstream y downstream.
- **Nueva tabla:** se trata de un nuevo proceso de ingestión, por lo que se considera que este tipo de cambio incluirá nuevos controladores, dtd y tipos de mensajes en Java. Agregue sinónimos, grants y considerar si los triggers son necesarios para las tablas derivadas.

3.7.1.6 MODIFICAR LAS CONSULTAS DE INFORMES POSTERIORES

Después de un acuerdo con los sistemas Downstream, los informes se implementarán utilizando las nuevas tablas/atributos como datos de origen.

3.7.2 PROBLEMAS DETECTADOS

Problemas detectados especificados en otras secciones implicadas en el ciclo de desarrollo:

[3.1.2 Ingestión de datos](#)

[3.2.2 Flujos de datos](#)

[3.3.2 Posición de datos](#)

[3.9.2 Informes Downstream](#)

Evaluación de NRT-DWH

3.8 GESTIÓN DE PRODUCCIÓN / MONITORIZACIÓN Y ESTABILIDAD

3.8.1 SITUACIÓN ACTUAL

El sistema de supervisión actual solo se aplica a los procesos de ingestión y a la carga de datos.

3.8.1.1 MANEJO DE ERRORES

NRT-DWH controla varios tipos de errores, que se pueden encontrar en la clase NrtDwhError. Se pueden organizar en las siguientes categorías:

- **Errores técnicos:** la base de datos no está disponible, la excepción de comunicación de MQ Series, la excepción de acceso a ficheros, etc...
- **Errores funcionales:** son excepciones provocadas por NRT-DWH para controlar errores funcionales. Normalmente se representan en la tabla RCM (mensajes RAW) y se detallan en la tabla ERR. Las excepciones controladas durante el proceso de ingestión de datos dan como resultado el estado Error y Reproceso en la tabla RCM.
- **Validación XML:** NRT-DWH analiza los mensajes XML entrantes con la biblioteca SAX en los DTD. En caso de que se genere cualquier error SAX durante el análisis se marcará como un error XML. La validación DTD, como los campos obligatorios, no está activada actualmente en NRT-DWH. Por lo tanto, este tipo de validación se retrasa hasta el primer error funcional generado por el proceso de ingestión. es decir: un campo obligatorio en La ingestión de transacciones podría dar lugar a una excepción controlada "Reproceso", porque la transacción no existe en el sistema.

3.8.2 PROBLEMAS DETECTADOS

Podría ser interesante anticipar los posibles errores y tomar medidas que lo impidan mediante el uso de aplicaciones de supervisión de infraestructura. A continuación se enumeran algunas posibles acciones de supervisión.

Problema #	Resumen del Problema	Descripción del problema	Impacto
DMON01	Error y reprocesar los mecanismos de retroalimentación de mensajes	Ningún mecanismo para analizar los problemas de producción relacionados con la funcionalidad NRT-DWH	Muy Alto
DMON02	Consultas de larga ejecución	Las posibles consultas que podrían utilizar demasiado tiempo para ejecutarse no se supervisan continuamente para evitar grandes problemas de rendimiento	Medio
DMON03	Instancias saludables de DAP	Las instancias del servidor de aplicaciones se supervisan desde el portal DAP, pero faltan algunas alertas del sistema, específicamente para las alertas de registro.	Medio
DMON04	Instancias en buen estado de la base de datos	Las instancias de base de datos solo se supervisan desde el lado DBA. Falta de mecanismos adicionales para detectar problemas.	Medio
DMON05	Informes intermedios	Los problemas de rendimiento posteriores no se supervisan diariamente.	Medio
DMON06	Falta de control de excepciones durante la validación dtd	La validación DTD no está activada, por lo que transfiere la responsabilidad a los controladores de mensajes para generar el error. es decir: los campos obligatorios no se detectan durante el análisis XML y los controladores de mensajes generan la excepción finalmente, porque el atributo es necesario para el cálculo.	Alto

Evaluación de NRT-DWH

3.8.3 MEJORAS PROPUESTAS

Existen algunos productos comercializados para aplicar en estos casos.

Problema #	Resumen del Problema	Solución Propuesta	Coste
DMON01	Error y reprocesar los mecanismos de retroalimentación de mensajes	Dos acciones: 1) Investigar Error y vuelva a procesar mensajes con sistemas ascendentes y corrija los problemas para mejorar la confiabilidad de los sistemas de nivel inferior. 2) Supervisar el error y vuelva a procesar los mensajes con algunos de los sistemas de alerta propuestos (Geneos/AppDynamics) recuperando el estado del evento en la tabla RCM, con el fin de actuar rápidamente en los problemas de producción	Muy Alto
DMON02	Consultas de larga ejecución	Implementar alertas (si no se especifica de forma predeterminada en la plataforma) para clasificar las consultas de larga ejecución para un análisis de rendimiento adicional	Medio
DMON03	Instancias saludables de DAP	Detectar los componentes clave en la instancia DAP, como el espacio del montón, las colas MQ, la CPU, la memoria, etc... e implementar alertas basadas en el umbral.	Medio
DMON04	Instancias en buen estado de la base de datos	Supervisar la base de datos Oracle en busca de bloqueos, procesos, estadísticas y otros parámetros útiles, con el fin de detectar problemas de base de datos	Medio
DMON05	Informes intermedios	Revisar los posibles cuellos de botella y mejorar el rendimiento dentro de los informes utilizados por los usuarios finales	Alto
DMON06	Falta de control de excepciones durante la validación dtD	Opción 1) Analizar todos los tipos de mensaje y comprabarque la validación DTD se puede activar sin riesgo. Opción 2) En caso de que se implemente el modelado XSD, esto debe hacerse desde cero	Medio

Evaluación de NRT-DWH

3.8.3.1 GENEOS (GRUPO ITRS):

Esta aplicación recoge información de diferentes puntos de la aplicación y analiza esta información detectando y alertando de posibles problemas. Los cuadros de mando permiten tomar decisiones rápidas actuando inmediatamente si fuera necesario y proporcionando una alta confiabilidad al sistema. Muchos proyectos utilizan Geneos^[9] para supervisar sus aplicaciones.

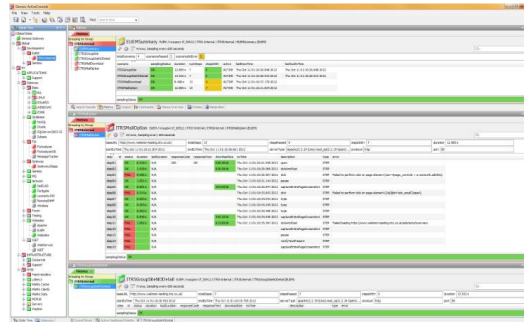


Figura 26. Consola de monitorización Geneos.

3.8.3.2 APPDYNAMICS (CISCO):

Appdynamics^[10] está creciendo actualmente porque la aplicación está en la parte superior del cuadrante de Gartner cuando se trata de monitorización de aplicaciones.



Figura 27. Monitorización de aplicaciones del cuadrante de Gartner.

La aplicación supervisa cada línea de código e inmediatamente proporciona la información relevante que nuestros clientes necesitan para resolver rápidamente los problemas, realizar mejoras en la experiencia del usuario y garantizar que las aplicaciones siempre cumplan con las expectativas de rendimiento de los empleados o clientes. Las características son similares a Geneos, pero hay algunas diferencias entre ellas que se pueden encontrar a continuación en una tabla comparativa específica.

Evaluación de NRT-DWH

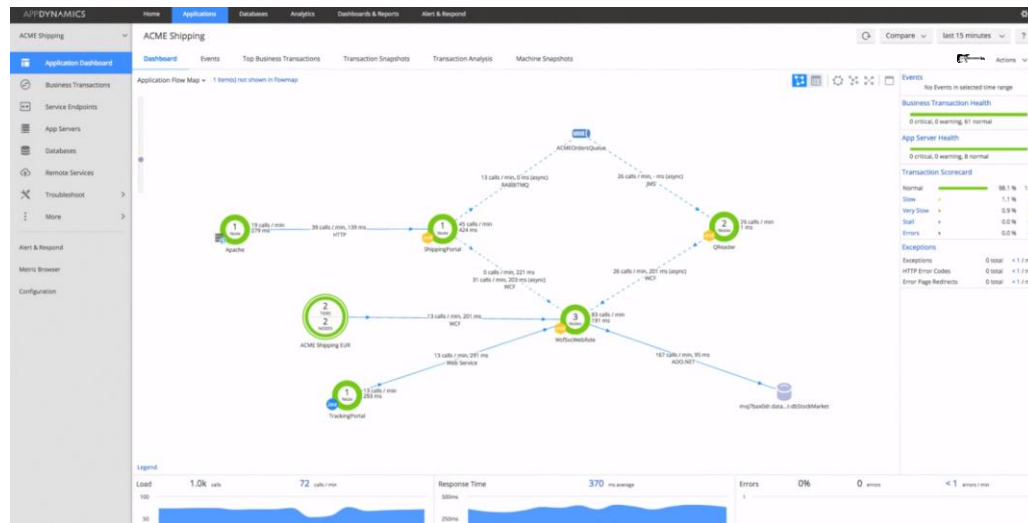


Figura 28. Consola de monitorización de la aplicación.

3.8.3.3 TABLA COMPARATIVA GENEOS VS APPDYNAMICS

La siguiente tabla es un subconjunto de tipos de monitorización relevantes, destacando qué herramienta se puede utilizar para cada tipo de Monitorización. Algunas superposiciones son inevitables y, con suerte, la siguiente información puede ayudar.

Tipo de monitorización Geneos	Tipo de monitorización Geneos	Tipo de monitorización Geneos
Monitorización de infraestructura Incluyendo métricas del sistema operativo, CPU, disco, Memoria, Red Etc.	Sí, para la supervisión de aplicaciones Propósitos Hay una superposición conocida con Nimsoft que es utilizado por el equipos de infraestructura. Nimsoft métricas también se consumen y reutilizado por otras herramientas	Sí AppDynamics tiene una 'máquina agente' para la monitorizacion de la infraestructura que tenemos disponible
Supervisión de bases de datos	Sí, para la supervisión de aplicaciones Propósitos OEM es la base de datos estratégica de DB herramienta de monitorización, pero el Geneos bases de datos y complementos de sql-toolkit son mucho más populares entre los equipos de aplicación	No AppDynamics tiene un agente de base de datos que es muy útil para análisis de rendimiento y profundo solución de problemas de buceo de consultas y SP. No está en nuestro alcance actual para introducirlo, sin embargo, parece extremadamente útil y se ejecuta de forma remota
Monitorización de procesos	Sí	No
Supervisión de archivos de registro	Sí	No
Monitorización de Llegada de archivos	Sí	No
Supervisión de colas	Sí	No
Control-M - Supervisión de fallos de trabajo	Sí	No
Supervisión del estado de la instancia DAP	Sí - uso de una API	Sí

Evaluación de NRT-DWH

Supervisión de transacciones Java	No	Sí
Supervisión de memoria Java	No	Sí
Análisis de transacciones Java	No	Sí
Análisis de memoria Java	No	Sí
Detección de fugas de Java	No	Sí
Excepciones Java (DAP)	No	Sí
Excepciones java (no DAP)	Sí	Sí
Errores de Log4J (DAP)	Sí - usando el apéndice log4j	Sí
Errores de Log4J (no DAP)	Sí	Sí
Análisis del rendimiento de la prueba de carga	No	Sí
Supervisión de la experiencia del usuario final - Web	No	Sí También vale la pena mencionar que DB's herramienta estratégica para el fin sintético La Monitorización de la experiencia del usuario es dbSAM
Supervisión de la experiencia del usuario final - Móvil	No	Sí

3.9 INFORMES DE DOWNSTREAM

En este momento cualquier cambio requerido e implementado en la aplicación NRT-DWH puede afectar directamente a los informes de Downstream. Los Downstreams tienen acceso directamente al modelo de base de datos física para implementar sus consultas.

3.9.1 SITUACIÓN ACTUAL

NRT-DWH tiene un conjunto de Downstreams que acceden al sistema para generar sus propios informes.

3.9.2 PROBLEMAS DETECTADOS

Los posibles problemas detectados son los siguientes:

Problema #	Resumen del Problema	Descripción del problema	Impacto
DW01	Varios sistemas posteriores acceden directamente a la base de datos NRT-DWH	Los Downstreams acceden directamente a la base de datos física y dificulta realizar cambios en el modelo sin afectar a los clientes.	Muy Alto
DW02	Los informes se proporcionan al usuario por correo electrónico sin mostrar el contenido en línea en el portal de Workbench	Los usuarios no ven el contenido de su selección hasta que se recibe el correo electrónico. Si el conjunto de filtros era incorrecto, deben esperar el error cuando llegue el correo electrónico, cambiar su selección y ejecutar de nuevo.	Alto

3.9.3 MEJORAS PROPUESTAS

Evaluación de NRT-DWH

Problema #	Resumen del Problema	Solución Propuesta	Coste
DW01	Varios sistemas Downstream acceden directamente a la base de datos NRT-DWH	Véase la propuesta de desacoplamiento Downstream en la sección 3.9.3.1	Muy Alto
DW02	Los informes se proporcionan al usuario por correo electrónico sin mostrar el contenido en línea en el portal de Workbench	Véase la solución propuesta en el ejemplo siguiente: sección 3.9.3.2	Alto

3.9.3.1 DW01 - VARIOS SISTEMAS DOWNSTREAM ACCEDEN DIRECTAMENTE A LA BASE DE DATOS NRT-DWH

- Implementar un modelo lógico y preparar un modelo conceptual para permitir que el nivel inferior acceda solo a los datos necesarios. Esta opción permite aislar el modelo de datos físicos NRT-DWH para realizar los cambios adecuados.
-

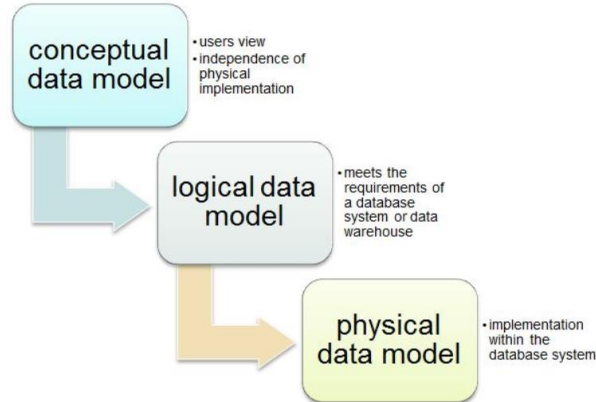


Figura 29. Modelos de datos orientados al desacoplamiento de bases de datos de usuario.

Evaluación de NRT-DWH

- Datos precalculados para mejorar el tiempo de respuesta de los informes.

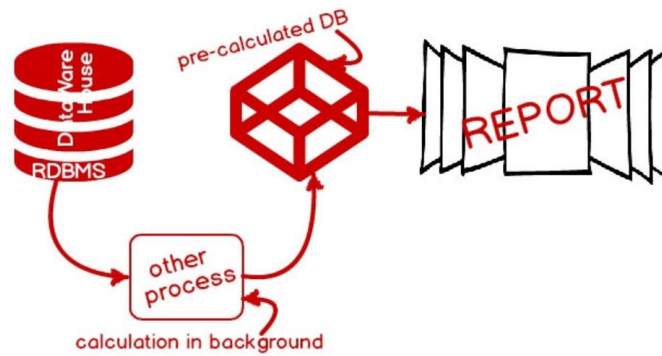


Figura 30. Estructura de datos precalculada.

Esta propuesta se detalla en la siguiente [sección 5.3.2. Desacoplamiento Downstream](#).

3.9.3.2 DW02 - LOS INFORMES SE PROPORCIONAN AL USUARIO POR CORREO ELECTRÓNICO SIN MOSTRAR EL CONTENIDO

Hay dos maneras de resolver este problema:

Proporcionar visualización de Excel/PDF a los informes front-end: para mostrar los informes en la pantalla, se debe utilizar una biblioteca Java para crear una hoja de cálculo de Excel o PDF. Esto se puede hacer con la biblioteca **Apache POI^[12]** para guardar archivos XLS/XLSX o **Jasper Reports^[12]** en caso de que se requiera algo más evolucionado con Excel o PDF.

Herramienta de informes: Esta opción permite crear informes en línea (cualquier usuario) y también permite crear informes predefinidos (desarrolladores). Todos los informes necesarios pueden ser implementados (por los desarrolladores o por los usuarios finales) y compartidos con otros usuarios.

4. VISIÓN DE MERCADO PARA LA ESTRATEGIA Y LA ARQUITECTURA DE DATOS

Las normas de mercado deben impulsar nuestras decisiones y sugerencias en este análisis. La inclusión de normas de datos nos dará la confianza para evaluar NRT-DWH desde un punto de vista subjetivo con el fin de resaltar posibles puntos débiles y mejoras basadas en desalineamientos del mercado.

4.1 ARQUITECTURA DE REFERENCIA

Como referencia, utilizaremos una Arquitectura de Referencia: "Una Arquitectura de Referencia es una fuente autorizada de información sobre un tema específico o un área de misión que guía y restringe las instancias de múltiples arquitecturas y soluciones".

Evaluación de NRT-DWH

4.2 MEJORES PRÁCTICAS PARA CADA ÁREA INVESTIGADA

Esta sección muestra los estándares de la industria, Las buenas prácticas genéricas y los escollos para cada área de investigación definida en este documento de evaluación.

Zonas a investigar	Mejores prácticas / Estándares de la industria	Trampas
Proceso de ingestión de datos & Flujos de datos <i>Cómo se ingestan los datos en la aplicación y se gestionan los diferentes flujos de datos (ciclo de vida de los eventos)</i>	<ul style="list-style-type: none"> Comprensión clara de las fuentes de datos, la frecuencia y el historial de los datos adquiridos: modelar las interfaces y el abastecimiento de datos dentro del propio sistema. Establecer un único contrato con los sistemas Upstream: una comprensión clara de las clases de datos que se ingestan La ingestión carga datos puramente y no se aplica ninguna lógica de negocios en esta etapa Los datos deben ser rastreables en cualquier punto del flujo de trabajo 	<ul style="list-style-type: none"> Dificultades para rastrear los problemas de calidad de los datos a fuentes autorizadas Múltiples formatos de datos es una sobrecarga, aunque a veces es necesario, siempre es un peligro que los datos recibidos no se entiendan bien – las fuentes ya pueden haber realizado filtrado/ transformación, etc. Mezclar la lógica con la ingestión/asignación introduce la opacidad y, de nuevo, los problemas son difíciles de resolver (y a menudo nunca se detectan)
Repositorio de datos (modelos de datos lógicos y físicos) <i>Modelos de datos lógicos y físicos. Cobertura de entidades para incluir datos transaccionales, posiciones, datos de referencia, etc.</i>	<ul style="list-style-type: none"> Los dominios lógicos y los repositorios físicos de datos han sido identificados, documentados e inventariados El objetivo después de la ingestión debe ser un modelo canónico sin manipulación de datos hasta el momento Se debe admitir el control de versiones de datos ingeridos (inmutabilidad de los datos de origen) Bucle de comentarios para obtener problemas de validez e integridad de los datos, por ejemplo, padres inexistentes para claves ajenas 	<ul style="list-style-type: none"> Ofuscación de los problemas de datos de origen – necesita una investigación local primero que utiliza el tiempo y el esfuerzo Ningún control de versiones significa que es imposible para los consumidores intermedios entender los datos históricos y la evidencia de por qué una salida es la forma en que era anteriormente Las fuentes nunca abordan los problemas y la "lógica de negocio" termina siendo implementada en la capa de ingestión. Esto es difícil de mantener y el conocimiento "por qué hicimos eso" se pierde en los repositorios de código y JIRA.
Informes de Downstream <i>APIS para recuperar datos de la aplicación NRT-DWH para el consumo</i>	<ul style="list-style-type: none"> Establecer un contrato con los datos expuestos por el sistema a sus consumidores Desacoplamiento técnico entre los sistemas con el fin de facilitar la gestión del cambio en diferentes sistemas 	<ul style="list-style-type: none"> Sin una definición clara entre una aplicación y sus consumidores, no habrá control de cómo se utilizan los datos de la aplicación. Tener acceso directo de los consumidores a la aplicación es una fuente de problemas potenciales y cambios difíciles.

Evaluación de NRT-DWH

<p>Calidad de los datos <i>Gestión y corrección de problemas de datos con sistemas ascendentes. Informes y acciones de calidad de datos.</i></p>	<ul style="list-style-type: none"> • Se establece un programa de calidad de datos, se han identificado partes responsables y se han asignado y comunicado funciones y responsabilidades. • Se identifica y evalúa la calidad de los almacenes de datos existentes (los datos se perfilan, analizan y clasifican) • La calidad de los nuevos datos se supervisa analizada y notificada (los puntos de control de calidad de los datos están en su lugar, las métricas se capturan y se notifican, los procesos de calidad son auditables) 	<ul style="list-style-type: none"> • No hay "propiedad" para los problemas de calidad de los datos y la corrección se vuelve imposible. • Se requiere el estado actual para establecer un plan inicial sobre cómo mejorar • Si no hay un análisis continuo de los datos que se están ingiriendo, el sistema puede llegar al punto de partida
<p>Gestión de cambios <i>Qué fácil es incorporar cambios en las diferentes áreas de la aplicación.</i></p>	<ul style="list-style-type: none"> • La aplicación se desarrolla en capas aisladas que permitirán implementar cambios para un área específica sin afectar a otras. • Los procesos automatizados de DevOps están en su lugar con el fin de permitir un tiempo rápido de comercialización para los cambios. 	<ul style="list-style-type: none"> • Las aplicaciones monolíticas tienden a amplificar y permitir el acoplamiento, lo que hace que la aplicación sea más difícil de mantener. • Con los procesos manuales de SDLC, las aplicaciones reducen la capacidad de entregar con mayor frecuencia y rapidez.
<p>Gestión de producción <i>Revisión de flujos de trabajo operativos. Monitorización y gestión del entorno productivo. Seguro de estabilidad de aplicaciones.</i></p>	<ul style="list-style-type: none"> • Los mecanismos adecuados se incluyen en el código para alertar al equipo de soporte de producción, con información significativa, de los problemas detectados en la aplicación. • Se establece el proceso con el fin de priorizar y asignar los incidentes detectados durante la ejecución de la aplicación. • Hay procesos adecuados en su lugar para monitorizar que los SLA de la aplicación se cumplen y alerta si es necesario. • La información indirecta (por ejemplo, registros) se mantiene y se analiza 	<ul style="list-style-type: none"> • Sin procesos de operaciones claros y establecidos habrá falta de servicio a las empresas. • Romper los SLA puede causar pérdida de confianza o incluso multas.
<p>Rendimiento <i>Detectar posibles cuellos de botella de la aplicación.</i></p>	<ul style="list-style-type: none"> • La aplicación está diseñada de una manera que permite el crecimiento de los volúmenes ingeridos sin afectar a los SLA establecidos. • La revisión técnica del código implementado se realiza periódicamente para garantizar que no se degrada a lo largo del tiempo. • Se deben realizar pruebas adecuadas para evitar problemas. 	<ul style="list-style-type: none"> • Sin tener previsto los volúmenes esperados y las pruebas adecuadas, es probable que la falta de servicio a las empresas ocurra a medio plazo.

5. PLAN DE ACCIÓN PROPUESTO

5.1 PROBLEMAS PRINCIPALES – PUNTOS MÁS DÉBILES

En esta sección los problemas principales serán descritos para evolucionar la aplicación NRT-DWH en un estado más estable. Las siguientes cuestiones se clasifican por importancia o prioridad en términos del impacto o beneficio.

Se pueden clasificar en 3 categorías:

- **Ingestión de datos.**
- **Sistemas Downstream.**
- **Programa de calidad de datos.**
- **Estabilidad de Producción:** error y volver a procesar los mensajes.
 - Flujos de datos: problemas de producción
 - Monitorización: mecanismos de retroalimentación

IMPORTANTE: Por favor, lea este comentario antes de mirar la tabla "Problemas Principales" a continuación.

Hay problemas detectados en este documento con el modelo de datos que afectan a la calidad de los datos y al rendimiento de los datos, pero NRT-DWH no puede implementar cambios agresivos del modelo de datos en este momento, porque afecta directamente a los sistemas Downstream, y al código Java NRT-DWH, que también está muy acoplado con el modelo de datos.

La propuesta es proporcionar primero la interfaz para sistemas de nivel inferior con un modelo lógico, desacoplar el código Java con el modelo de datos, diseñar las entidades correctamente en el esquema XML y luego NRT-DWH estará en una buena posición para evolucionar el modelo de datos. Estos pasos son necesarios para seguir una migración gradual y combinar el enfoque actual de NRT-DWH.

Problema #	Tipo de problema	Resumen del Problema	Descripción del problema	Impacto
DI04	Ingestión de datos	Ingestión SQL: instrucciones DML	Instrucción SQL sin instrucción preparada e ingestión concatenando nombres de columna en una cadena para ejecutar la instrucción DML. Esto es un problema en términos de rendimiento (análisis cada vez que se recibe un evento) y seguridad (inyección SQL). Los controladores de mensajes se ven afectados por este problema.	Muy Alto
DF05-DF06	Flujos de datos	Investigar mensajes de producción con errores y reprocesar (perenne)	En la sección de flujos de datos, se documenta que hay mensajes con errores significativos y de reproceso que se deben investigar con la conexión en sentido ascendente, para encontrar las causas raíz para mejorar el cálculo de la seguridad y de la posición de efectivo.	Muy Alto
DW01	Downstream	Consultas directas del Downstream al modelo de datos físico NRT-DWH	Difícil realizar cambios en el modelo de datos en NRT-DWH porque hay una dependencia directa con otros sistemas. Proporcione una interfaz (API) a los niveles inferiores con el modelo de datos lógicos disponible para los niveles inferiores.	Muy Alto

Evaluación de NRT-DWH

DQ01	Calidad de los datos	Falta de procesos de calidad de datos	NRT-DWH no contiene procesos de calidad de datos para mejorar la confiabilidad de los datos. Generación de perfiles de datos para garantizar que faltan formatos de datos o integridad de datos.	Muy Alto
DMON01	Producción Mgmt / Monitorización / Estabilidad	Error y volver a procesar los comentarios de los mensajes	Mejorar los mecanismos de retroalimentación para actuar rápidamente en los problemas de producción. Esto es básicamente para la estabilidad de la producción. Se necesitan herramientas para notificar los mensajes con errores y volver a procesar poniéndose en contacto con los sistemas ascendentes. Mejorar la precisión en el cálculo intradía proporcionará informes de buena calidad para los sistemas Downstream.	Muy Alto
DI01	Ingestión de datos	Decisión del enrutado de ubicación antes del procesamiento. Código duplicado en dos flujos	Código duplicado para DTD y controladores de mensajes en dos flujos comprobando la ubicación del remitente (rama ascendente). Difícil de implementar nuevos cambios y posibles funcionalidades que faltan en uno de los flujos. Afecta a la capacidad de mantenimiento y escalabilidad del código.	Alto
DI02	Ingestión de datos	Modelado XML con versiones extendidas con DTD	Difícil extraer el esquema actual para cada tipo de mensaje. Si las entidades de negocio se representan en un esquema preciso, es más fácil mantener el modelado XML e investigar problemas. La propuesta es XSD (implementada con JAXB).	Alto
DI03	Ingestión de datos	El acceso a datos y la lógica empresarial se mezclan	NRT-DWH es una aplicación para procesar mensajes, pero el código fuente no está organizado para un propósito empresarial. El código se mezcla en muchas clases Java durante la ingestión. Es difícil realizar nuevos desarrollos y apoyar las funcionalidades actuales. Este es un tema general	Alto

5.2 PLAN DE EVOLUCIÓN

En las siguientes imágenes, la visión general de NRT-DWH se representa para comprender la ruta de evolución.

La primera imagen describe el **estado actual** con el código fuente acoplado al modelo de datos físicos. Hay dificultades para realizar cambios en el diseño del modelo de datos porque pueden afectar fuertemente al proceso de ingestión de datos NRT-DWH y a los informes de Downstream. Como podemos ver a continuación, un cambio en el modelo de datos podría afectar a una gran cantidad de puntos de código fuente y hacer que el cambio sea inaccesible.

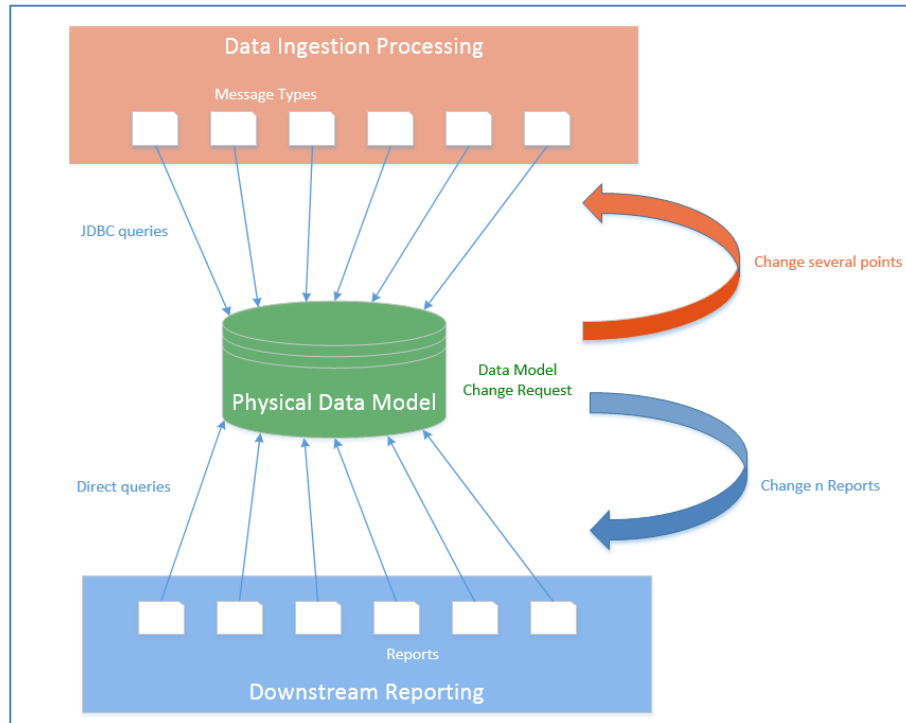


Figura 31. Estado actual del Modelo acoplado.

Evaluación de NRT-DWH

Un **plan de evolución** para llevar la aplicación NRT-DWH a un estado normalizado tendría el siguiente aspecto:

Fase 1: Desacoplar las capas externas (Ingestión e Informes) del modelo de datos físicos.

Fase 2: Evolucionar el diseño del modelo de datos y tomar acciones de rendimiento y tuning si es necesario.

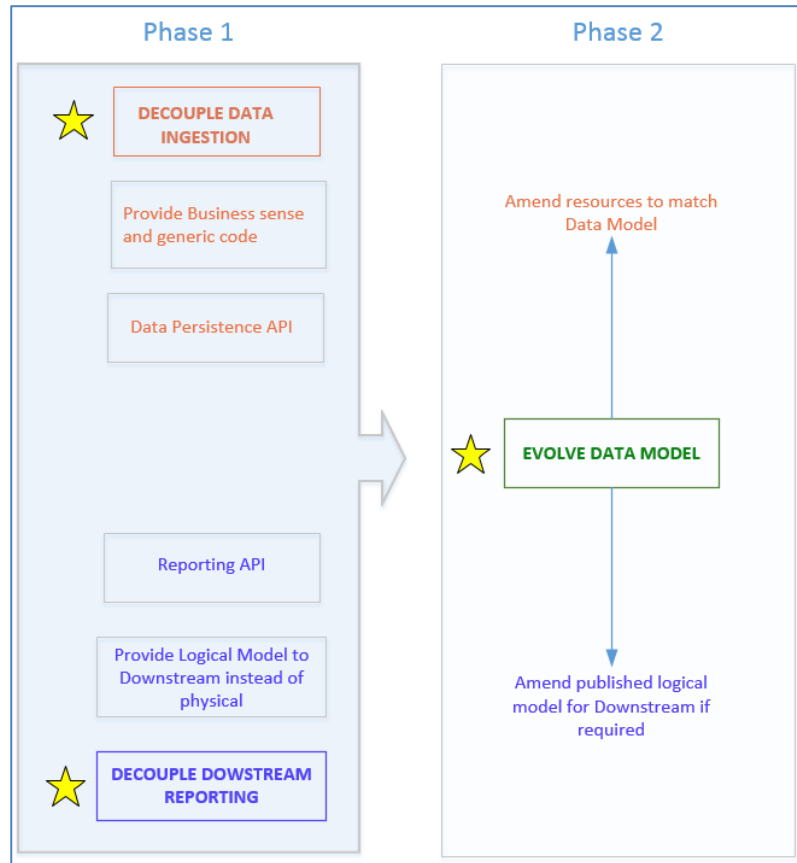


Figura 32. Fases del plan de evolución.

Evaluación de NRT-DWH

La imagen completa con el **estado final** se describe a continuación:

El código de ingestión de datos se organizará en entidades empresariales. Por lo tanto, probablemente un cambio común a varias entidades de negocio podría hacerse de una sola vez. Los cambios en el modelo de datos producirán cambios de configuración en la API de persistencia de datos, pero o asignaciones de datos, pero solo se realizarán una vez.

Los informes de Downstreams se presentarán en una sola capa con un modelo publicado para los sistemas Downstream en lugar de consultas directas. De esta manera, NRT-DWH puede controlar los datos que se proporcionan a los Downstream y realizar cambios en el modelo de datos.

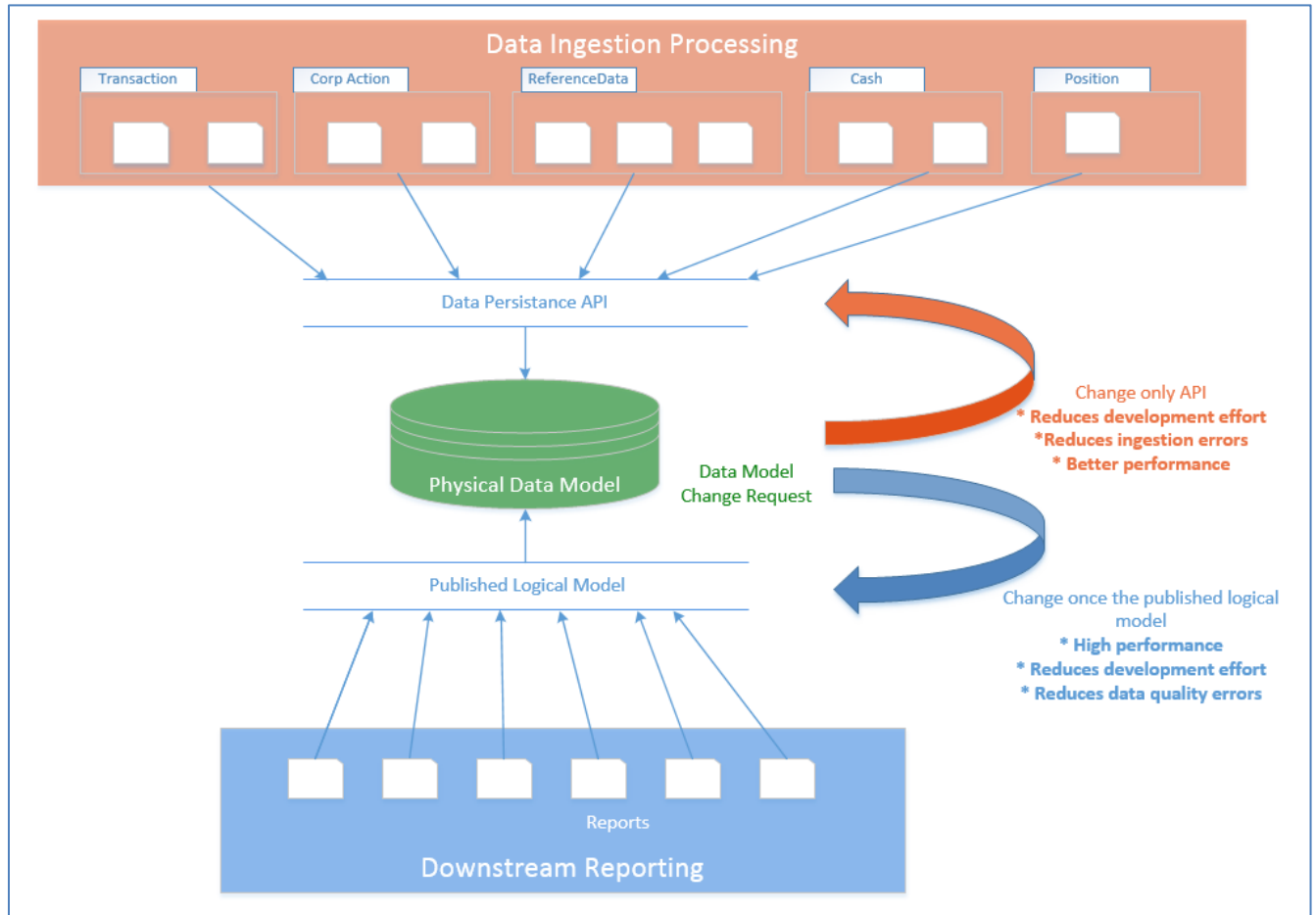


Figura 33. Estado futuro del modelo desacoplado.

5.3 DISEÑO ARQUITECTÓNICO PROPUESTO

El diseño arquitectónico propuesto se centra en evolucionar la aplicación NRT-DWH tal y como se describe en la sección anterior [5.2](#).

Son dos opciones para cada "problema de desacoplamiento". La primera propuesta pretende ser una migración con la misma infraestructura que se ejecuta hoy en día en NRT-DWH y tecnologías similares.

Por otro lado, la segunda propuesta trata de dar consejos sobre cómo construir NRT-DWH en las herramientas ETL y Reporting, lo que implica una solución disruptiva en comparación con lo que tenemos en este momento. Sin embargo, esta opción proporciona trazabilidad en el lado de la ingestión de datos y flexibilidad en los informes posteriores, para obtener informes diseñados por el usuario.

En las siguientes secciones, podemos ver la imagen completa:

[5.3.1. Desacoplamiento de la ingestión de datos](#)

- Migración gradual: enfoque de Java y Oracle
- Nuevo enfoque: Capa ETL.

[5.3.2. Desacoplamiento de los Downstreams](#)

- Migración gradual: vistas materializadas de Oracle.
- Nuevo enfoque: Herramienta de generación de informes (Cognos, Microstrategy, Oracle BI, etc...)

5.3.1 DESACOPLAMIENTO DE LA INGESTIÓN DE DATOS

En las secciones siguientes, se proponen dos arquitecturas diferentes para resolver paso a paso los "problemas principales" con respecto a la ingestión de datos descrita anteriormente, excepto los problemas de estabilidad de producción (DMON01 y DF05-06) y el desacoplamiento de los Downstreams, que se explicará en la sección [5.3.2](#).

5.3.1.1) **Enfoque Java y Oracle:** esta propuesta aprovecha la infraestructura actual y evoluciona gradualmente la aplicación NRT-DWH. Se detalla una prueba de concepto para los mensajes de seguridad.

5.3.1.2) **Enfoque ETL:** Desarrollo ETL alternativo con Informatica Powercenter y base de datos Oracle. Esta opción implica solicitar instancias en un entorno, que es nuevo en NRT-DWH, pero un proceso conocido por el Cliente.

Evaluación de NRT-DWH

5.3.1.1 ENFOQUE DE JAVA Y ORACLE: RUTA DE MIGRACIÓN GRADUAL Y REALISTA

En esta sección, se detallará una prueba de concepto para ver cómo NRT-DWH puede evolucionar a un estado futuro resolviendo algunos de los "problemas principales" detallados en la sección anterior para la ingestión de datos. Como resultado de esto, el código Java se desacoplará con el modelo de datos para el tipo de mensaje Seguridad.

La siguiente imagen describe la arquitectura que la prueba de concepto utilizará para lograr el "Desacoplamiento" en la ingestión de datos.

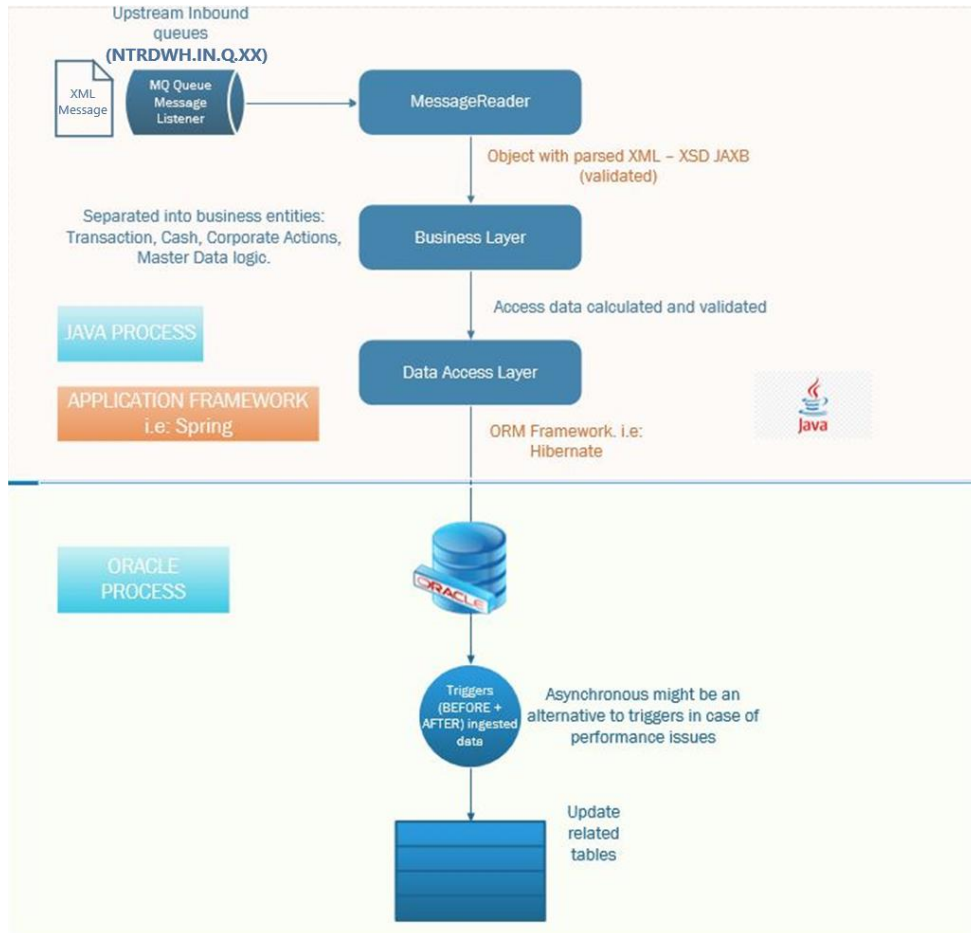


Figura 34. Estado futuro de la Ingestión de datos.

Esta prueba de concepto tomará el tipo de mensaje "Security", que representa más del 30% del volumen total actualmente en Producción, y describirá el proceso de migración para lograr paso a paso los objetivos principales. Este tipo de mensaje solo realiza la ingestión y es bastante sencillo comparar el nuevo enfoque con el actual. La infraestructura será la misma con el servidor de aplicaciones DAP Weblogic y la base de datos Oracle.

Evaluación de NRT-DWH

Principales objetivos:

Incorporar (Frameworks (Spring y Hibernate))

Cambiar el mecanismo ON/OFF de la funcionalidad de Valores con el código NRT-DWH actual, para tener la capacidad de restaurar el flujo antiguo.

Modelar la estructura de esquema XML para eventos NRT-DWH.

Aplicar pasos iterativos del ciclo de migración de entidades

- **Seleccionar** Tipo de mensaje que se va a migrar
- **Definir** entidad de tipo de mensaje a XSD
- **Implementar**
 - capas de acceso a datos y negocios
 - Objeto de acceso a datos
 - asignación XML -> Objeto de acceso a datos
- **Probar** el Diseño
- **Desplegar** en DAP Weblogic alternativo
- **Validar** los resultados y comparar el rendimiento

5.3.1.1.1 PASO 1: INCORPORAR MARCOS

El marco de Spring se agregará a NRT-DWH, con las dependencias de Maven adecuadas.

Agregar dependencias de Hibernate a la aplicación NRT-DWH.

Agregar un marco de asignación relacional de objetos como Hibernate para implementar objetos de acceso a datos

Implementar un DAO abstracto de Hibernate, por lo que la ingestión será genérica en NRT-DWH. La siguiente referencia es útil: <https://www.baeldung.com/simplifying-the-data-access-layer-with-spring-and-java-generics>

5.3.1.1.2 PASO 2: ACTIVAR / DESACTIVAR EL MECANISMO PARA EL CÓDIGO ANTIGUO Y NUEVO

En este paso se implementará un nuevo método para determinar si el evento Security pasará por el código antiguo o el nuevo código. Se llamará a este método desde el agente de escucha de mensajes entrantes (InboundMsgListener.onMessage) y recuperará un valor en la base de datos para decidir el código elegido.

- 1) Nueva tabla MIGRATED_EVENTS : messageType + Y/N (código nuevo o antiguo)
- 2) Consulta para comprobar si el mensaje actual es para messageType en eventos migrados y determinar si pasa por el código antiguo o nuevo. En caso de que la entrada no esté presente, se debe ejecutar el código antiguo.
- 3) Para la comparación, enviaremos el mismo conjunto de eventos dos veces con valores propietarios Y y N.

5.3.1.1.3 PASO 3: MODELO XML – XSD JAXB^[13] (SOLUCIÓN PROPUESTA DE DI02 ISSUE)

Crear el nuevo esquema XSD con la capa genérica y los eventos de Valores. El siguiente fragmento de código es solo un ejemplo para reflejar lo que se debe hacer, no el caso real en sí:

Evaluación de NRT-DWH

Los eventos de Valores son eventos de datos de referencia (abstractos) y el último es un evento NRT-DWH. Por lo tanto, hay una jerarquía y el modelo tiene un propósito empresarial. A continuación, SecurityEvent fuerza ese atributo SecId es obligatorio. Todos los eventos se modelarían en el esquema **NrtldwhModel.xsd**, pero solo los eventos de seguridad de este PoC.

Hay que tener en cuenta que el modelo de eventos de Valores es simple, pero para crear un esquema coherente, un **analista de negocios** debe estar en contacto con los desarrolladores para encontrar la representación adecuada del negocio NRT-DWH.

El esquema tendría el siguiente aspecto, solo representaba la clase Security. Esto es sólo un ejemplo, no la representación real en sí.

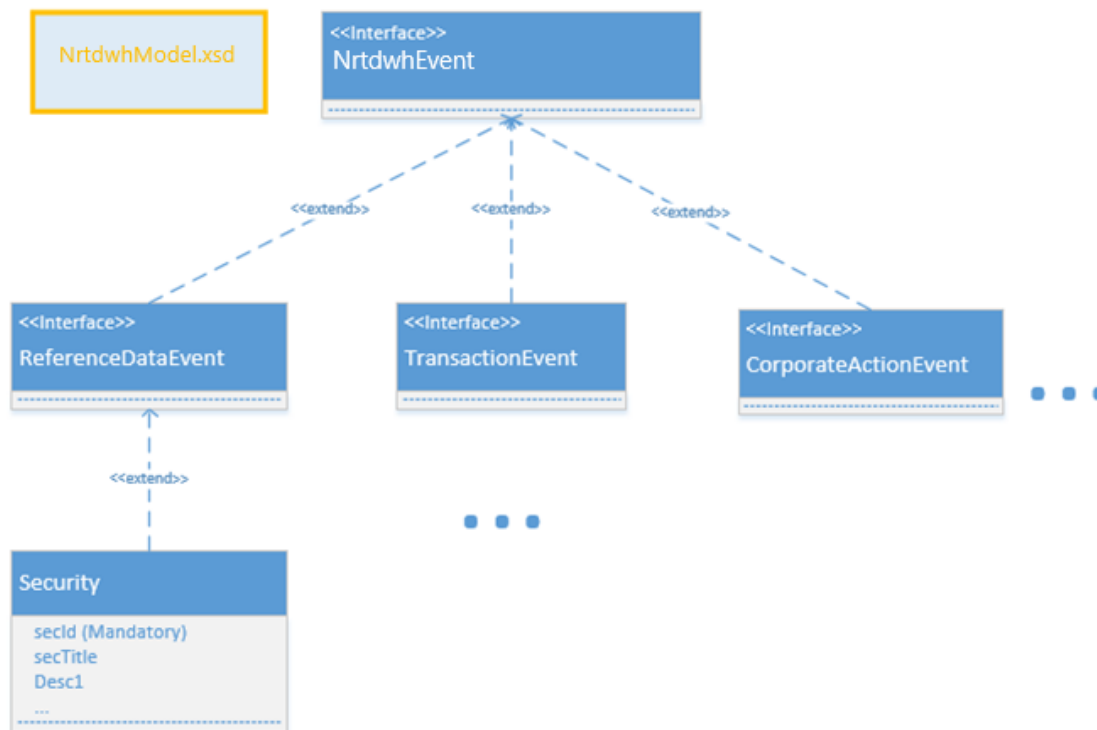


Figura 35. Ejemplo de modelo XSD.

5.3.1.1.4 PASO 4: SELECCIONE EL TIPO DE MENSAJE QUE DESEA MIGRAR (SOLUCIÓN PROPUESTA DE PROBLEMA DI04)

Elegir la entidad correspondiente que se va a implementar (el evento Security podría ser la primera opción).

5.3.1.1.5 PASO 5: DEFINIR ENTIDAD DE TIPO DE MENSAJE EN XSD (SOLUCIÓN PROPUESTA DE PROBLEMA DI04)

Definir el tipo de mensaje relacionado con la entidad seleccionada y agregue la estructura al XSD.

Evaluación de NRT-DWH

5.3.1.1.6 PASO 6: IMPLEMENTAR (SOLUCIÓN PROPUESTA DE PROBLEMA DI04)

La siguiente imagen muestra una aplicación típica, que se puede utilizar para implementar la solución NRT-DWH:

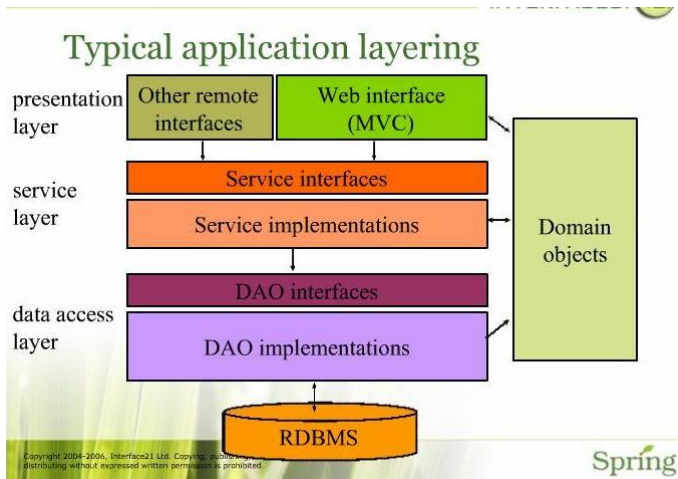


Figura 36. Capas de aplicaciones.

- Aplicar la ingeniería inversa de Hibernate a la tabla SEC (Security) para obtener la entidad de Valores.
- Capas de acceso empresarial y de datos: implemente la capa "ReferenceDataService" con una interfaz y las implementaciones adecuadas. Esta capa de servicio inyectará las interfaces de objeto de acceso a la base de datos (solo la seguridad en este PoC). Los servicios se anotarán como transaccionales.
- Implementar interfaces e implementaciones DAO^[14] para ReferenceData y Security
- Implementar el DTO^[15] de Security para asignar el objeto resultante XML (JAXB) a la entidad de acceso a datos de Valores. Esta es la alternativa a los " message handlers" para el proceso de ingestión actual.

5.3.1.1.7 PASO 5: DISEÑO DE PRUEBA

Construir un conjunto adecuado de pruebas para ingerir una cantidad considerable de valores (300K por día en producción)

5.3.1.1.8 PASO 6: DESPLIEGUE EN EL WEBLOGIC DAP

Preparar un entorno DAP de prueba y ejecuta las pruebas con el código antiguo y el nuevo.

5.3.1.1.9 PASO 7: VALIDAR LOS RESULTADOS

Comparar los resultados (utilizando volúmenes de producción) y medir el rendimiento en Weblogic y Oracle. Evaluar las mejoras y decidir si el proceso de migración debe continuar con esta ruta de evolución.

5.3.1.1.10 RESUMEN DE POC

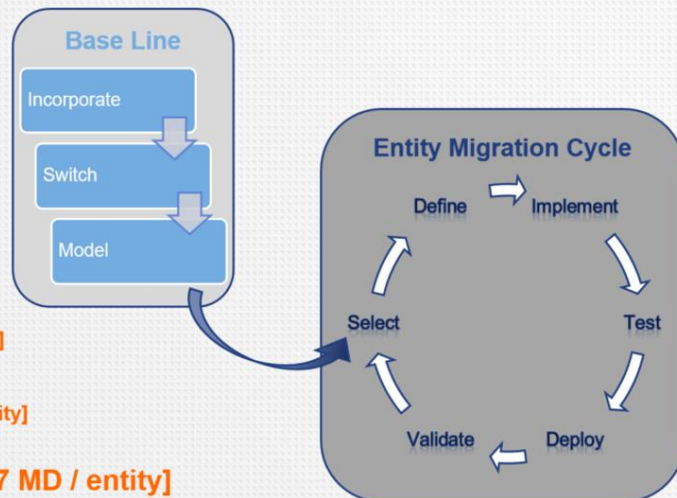
La diapositiva siguiente explica los pasos necesarios para aplicar la solución propuesta, agregando un coste estimado de implementación.

PoC : Java and Oracle gradual migration

NRT-DWH - Data Ingestion Decoupling

- PoC for the mid-level entity **security message type** (19% of the total messages) implemented with the proposed solution

1. **Incorporate** Frameworks
Spring [6 MD]
Hibernate for Data Persistence [4 MD]
2. **Switch** Old/New architecture [3 MD]
3. **Model** XML schema with XSD-JAXB [5 MD]
 1. **Select** Message Type to migrate
 2. **Define** message type entity to XSD [3 MD / entity]
 3. **Implement**
business and data access layers [3 MD / entity]
Data Access Object [3 MD / entity]
mapping XML -> Data access object [2 MD / entity]
 4. **Test** design [7 MD / entity]
 5. **Deploy** into alternative DAP Weblogic [5 MD / entity]
 6. **Validate** results and compare performance [4 MD / entity]



- **Total approx. Estimation : [18 MD] + [27 MD / entity]**

Figura 37. PoC de Migración gradual de Java y Oracle.

5.3.1.2 ENFOQUE ETL

Este punto proporciona una propuesta para aplicar un modelo ETL (ExtracciónTransformación-Carga) a la aplicación NRT-DWH. El enfoque tendrá dos procesos separados (Load Staging Area y Load Datawarehouse).

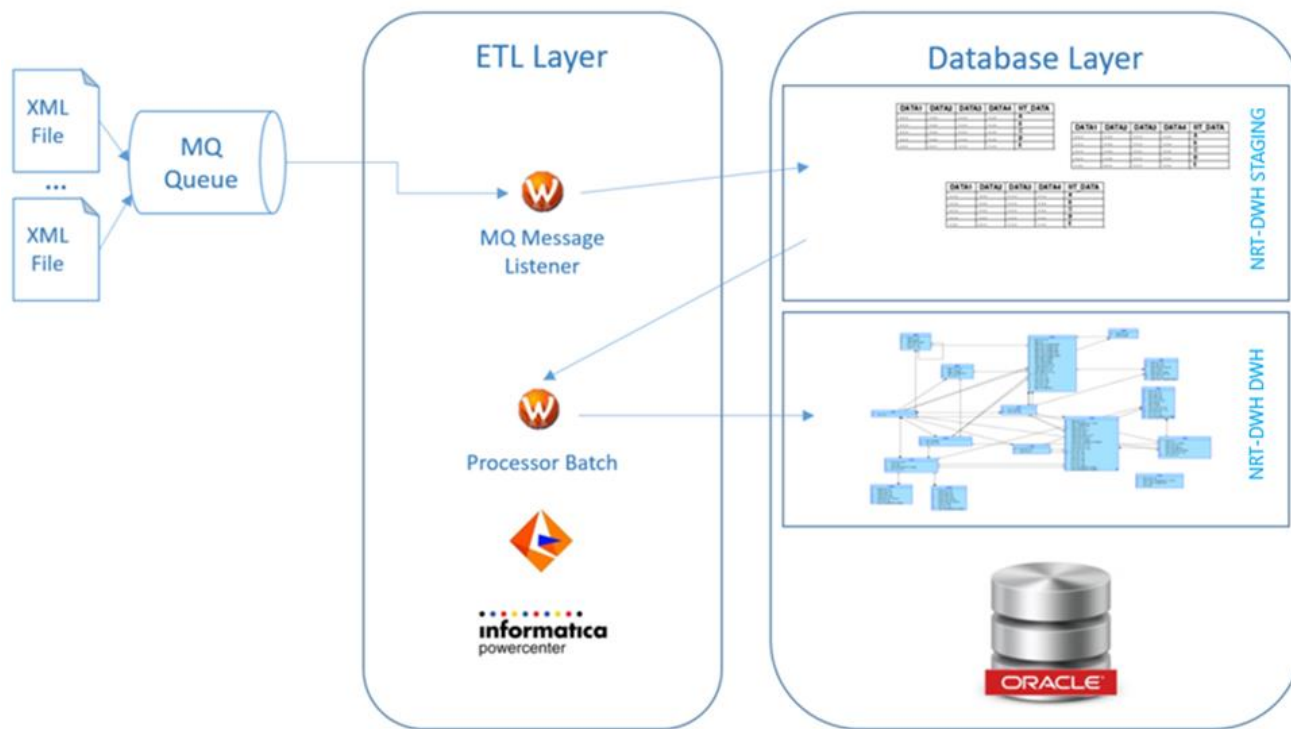


Figura 38. ETL enfoque vista de alto nivel.

Evaluación de NRT-DWH

El primero leerá (en tiempo real) los mensajes proporcionados por los Upstreams y cargará los datos sin procesar (datos de tabla sin procesar). El mismo proceso cargará también la información en formato de tabla para cada mensaje en su tabla de ensayo correspondiente distribuyéndolos a través de la lógica de negocios (Datos de referencia, Transacciones, Posiciones, etc...).

El segundo estará programado para leer la información del área de ensayo y calcular la información necesaria cargándola en las tablas finales del datawarehouse.

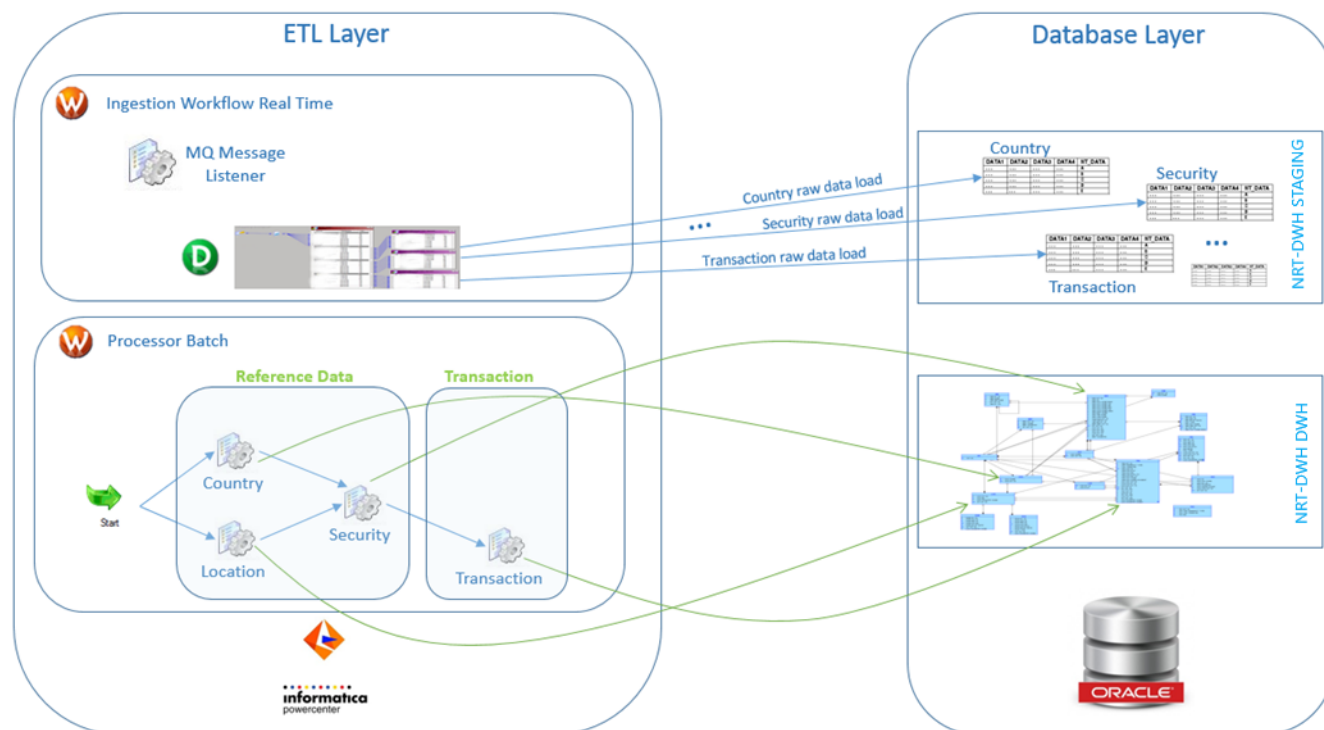


Figura 39. Vista detallada del enfoque ETL.

5.3.2 DESACOPLAMIENTO DE LOS DOWNSTREAMS: CAPA DE INFORMES

En las secciones siguientes, se proponen dos arquitecturas diferentes para proporcionar mejoras en "Desacoplamiento Downstream" (problema DW01)

5.3.2.1) **Vistas Materializadas de Oracle:** se trata de una solución gradual para migrar informes que se ejecutan en la misma infraestructura que la actual. Las vistas materializadas serán un único punto de acceso a los datos de notificación. El concepto cambiará a "datos publicados" en lugar de que los Downstreams soliciten información en el modelo de datos físicos NRT-DWH.

5.3.2.2) **Herramienta de informes:** Estas herramientas proporcionan más flexibilidad para generar informes ad hoc y también permiten tener informes predefinidos. El nivel inferior solo puede acceder a los datos a través del modelo de datos publicado evitando las nomenclaturas físicas.

5.3.2.1 VISTAS MATERIALIZADAS DE ORACLE: RUTA DE MIGRACIÓN GRADUAL Y REALISTA

Una arquitectura de vista materializada típica de Oracle funciona de la siguiente manera:

- Una vista materializada de Oracle es como una tabla que se actualiza periódicamente en función de una consulta con la información incremental para el intervalo de tiempo definido aunque también se podría sincronizar esta actualización con algún otro proceso del aplicativo.
- **MView (imagen siguiente)** se define con la consulta adecuada de una o varias tablas y el contenido incremental de esas tablas se almacena sincrónicamente en un registro **MView**. La directiva de actualización define la frecuencia de actualizaciones en el MView.
 - o **Al confirmar la actualización:** si hay una confirmación en una de las tablas dependientes, se actualiza MView.
 - o **Actualización a petición:** MView se actualiza con el registro MView.

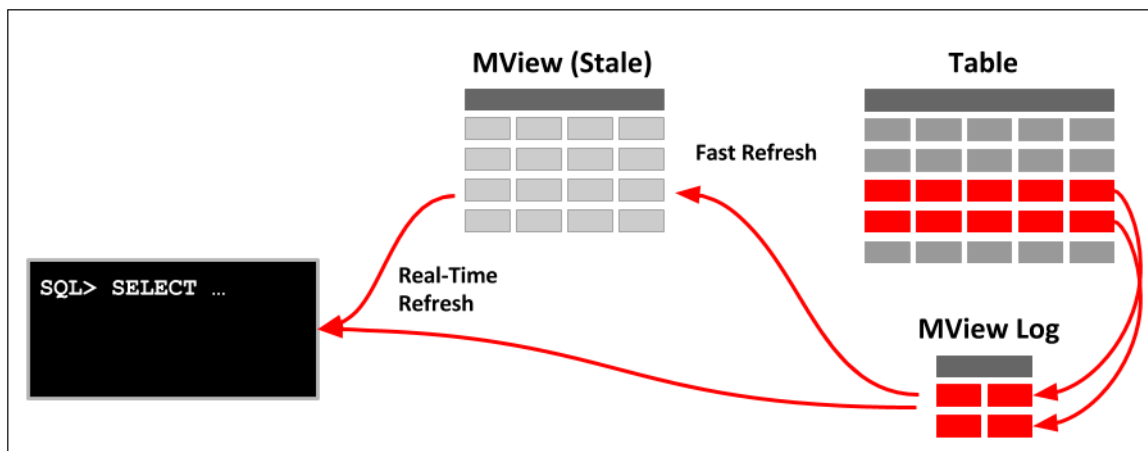


Figura 40. Vista materializada propuesta por Oracle: <https://oracle-base.com/articles/misc/materialized-views>

Con las vistas materializadas de Oracle podemos lograr dos objetivos en NRT-DWH en el lado de la generación de informes:

- 1) **Desacoplar los informes de los Downstreams del modelo de datos físicos NRT-DWH**, proporcionando solo la información necesaria. Así que la aplicación NRT-DWH puede controlar los datos publicados. Si se aplica una solicitud de cambio al modelo de datos NRT-DWH, solo necesitamos probar los datos proporcionados.
- 2) **Resolver problemas de rendimiento:** las vistas materializadas se actualizan en una frecuencia de tiempo determinada. De este modo, el sistema de nivel inferior no necesita ejecutar consultas pesadas para volver a calcular todo el informe si los datos agregados y transformados se calculan de antemano. El tiempo y los recursos consumen uso para ser un dolor en los entornos de informes.

La migración sería gradual y no habrá ningún impacto en la infraestructura de los Downstream actual, solo las consultas se verían afectadas.

5.3.2.1.1 PASO 1: ANALIZAR LAS CONSULTAS ACTUALES DE LOS SISTEMAS DOWNSTREAM.

Comprender correctamente el propósito de los informes y agruparlos por entidades lógicas. Por lo tanto, se podrá saber cómo deben ser las vistas materializadas, con los niveles correspondientes. Un **analista de negocios** debe proporcionar entradas para las entidades lógicas.

Evaluación de NRT-DWH

5.3.2.1.2 PASO 2: EVALUAR PROBLEMAS DE RENDIMIENTO

Si hay problemas de rendimiento, entonces debemos proporcionar datos precalculados en vistas materializadas optimizadas, como datos agregados comunes. Las vistas materializadas se actualizarán en una frecuencia casi en tiempo real en función de los requisitos empresariales.

5.3.2.1.3 PASO 3: IMPLEMENTAR LAS ENTIDADES PRINCIPALES PARA INICIAR EL PROCESO DE MIGRACIÓN

- Obtener las consultas originales con las tablas combinadas y crear la vista materializada.
- Generar un acuerdo sobre la frecuencia de refresco para la vista.

5.3.2.1.4 PASO 4: PRUEBA CORRECTAMENTE

Comparar los resultados entre las consultas actuales y las vistas materializadas optimizadas, para asegurarse de que proporcionan el mismo resultado en casos reales.

5.3.2.1.5 PASO 5: REVISAR EL RENDIMIENTO

Aplicar técnicas de rendimiento y tuning si todavía son necesarias.

5.3.2.1.6 PASO 6: CAMBIO DE CONSULTAS DE LOS SISTEMAS DOWNSTREAM

Los sistemas Downstream deben migrar la funcionalidad para leer datos de las vistas materializadas.

5.3.2.1.7 RESUMEN DE POC

En esta sección se explican los pasos necesarios para aplicar la solución propuesta, añadiendo un coste estimado de implementación.

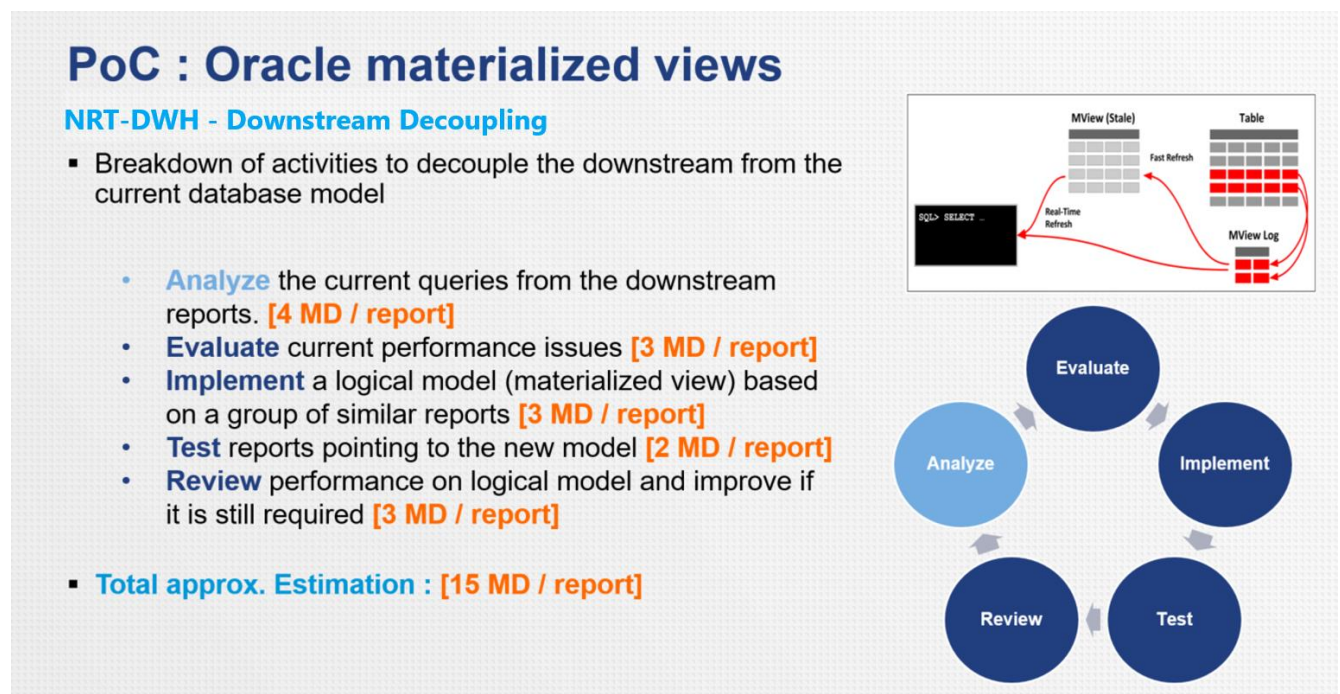


Figura 41. PoC de Vistas materializadas de Oracle.

5.3.2.2 HERRAMIENTA OLAP

Utilizando una herramienta OLAP específica (Cognos, Microstrategy, Oracle BI, etc.) es posible separar la actividad del modelo de datos intermedio y físico. Esta opción permite implementar cambios dentro del modelo de datos sin impacto en los informes de nivel inferior.

La herramienta OLAP permite separar el modelo de base de datos mediante tres etapas. El primero contiene todas las tablas físicas requeridas por el modelo. La segunda capa crea una vista lógica orientada a los usuarios finales aplicando una nomenclatura fácil y algunos cálculos si son necesarios.

Por último, la última capa (Presentación) muestra a los usuarios finales un modelo fácil que se puede utilizar para crear sus propios informes arrastrando y soltando. También algunos informes o paneles predefinidos se pueden compartir con los usuarios correspondientes. Se accede a esta capa de presentación a través de una URL y se puede personalizar la apariencia.

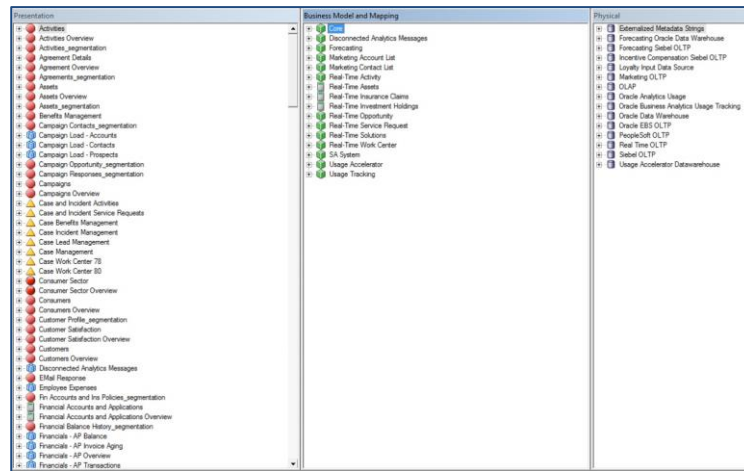


Figura 42. Vista de desarrollador del modelo de datos de Oracle.

Esta arquitectura también permite definir un conjunto de informes de supervisión orientados a los controladores de aplicaciones y a los Upstreams o Downstreams responsables del control de errores, mensajes reprocesados, datos reconciliados, posiciones actualizadas con el proceso EOD, etc.

El middleware proporciona una gran cantidad de tipos de gráficos para trazar datos en los cuadros de mando personalizados.

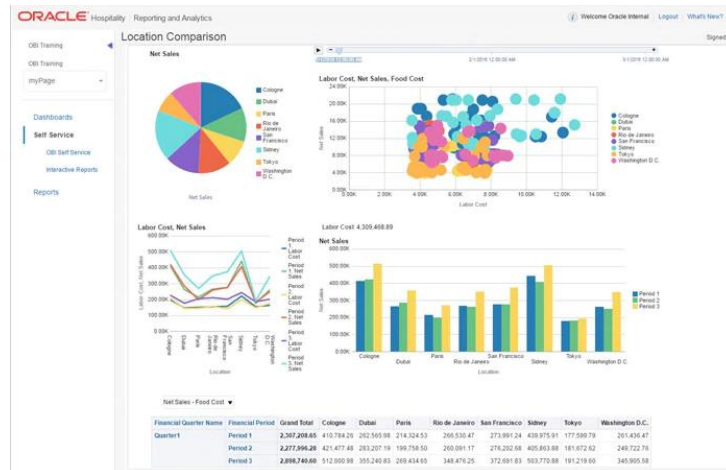


Figura 43. Ejemplo de informes y Cuadros de Mando de Oracle.

5.3.3 PROGRAMA DE CALIDAD DE DATOS

La solución propuesta se explica paso a paso en la siguiente diapositiva.

PoC : Data Quality Program

NRT-DWH - Data Quality

- Recommended activities to improve data quality.
- **Identify** & assess the degree of poor data quality within business objectives
- **Define** measures, performance targets and business rules related to data quality
- **Design** quality improvement processes to remediate process gaps.
- **Implement** the data quality design applying it to production environments
- **Monitor**, inspect and remediate not acceptable data quality issues

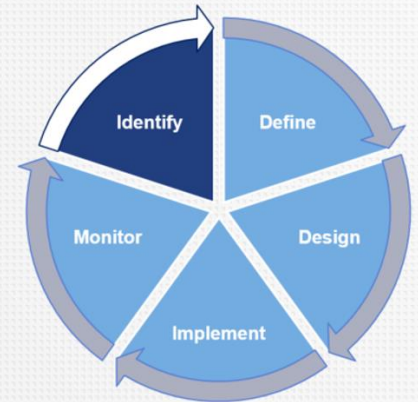


Figura 44. Programa de calidad de datos.

5.3.4 MEJORA DE LA ESTABILIDAD DE LA PRODUCCIÓN

La solución propuesta se explica paso a paso en la siguiente diapositiva, añadiendo un coste estimado de implementación.

PoC : Failed and Reprocess messages investigation

NRT-DWH - Production Stability

- Breakdown of activities to investigate failed and reprocess messages.

Classify Failed and reprocess messages [3 MD / error type]

- **Select error type** that affects the biggest set of data.
- **Contact stakeholders**. Upstream expert (Trader, EIS, IDMS, etc...) and **NRT-DWH** expert to track the specific Production issue [5 MD / error type]
- **Analyze** the root cause with development teams for the failed/reprocess message [9 MD / error type]
- **Implement** the fix [5 MD / error type]
- **Test** [3 MD / error type]

- **Total approx. Estimation : [25 MD / error type]**

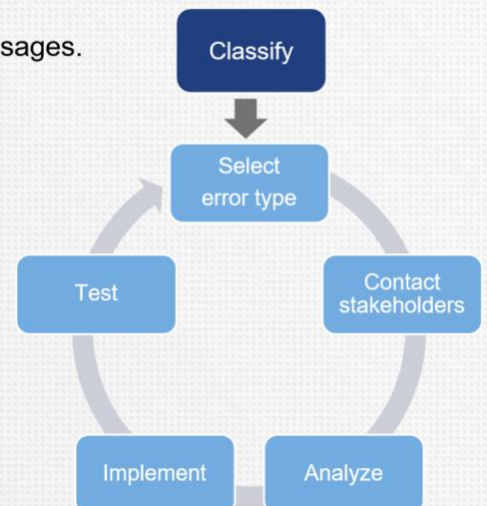


Figura 45. PoC de investigación de mensajes Error y Reprocesados.

Evaluación de NRT-DWH

5.4 COSTE Y BENEFICIOS DE IMPLEMENTACIÓN

La siguiente tabla proporciona una lista de beneficios que se pueden obtener a través de la implementación y los costes estimados asociados a las mejoras logradas.

Problema #	Resumen del Problema	Beneficios de implementación	Mejora de costes
DI04	SQL generado manualmente sin instrucciones preparadas	<ul style="list-style-type: none"> Mejorar el rendimiento mediante declaraciones preparadas Fácil comprensión del código Soluciona posibles vulnerabilidades en la ingestión de datos desde los Upstreams Aumentar la fiabilidad en el sistema Downstream (calidad de datos) 	<ul style="list-style-type: none"> Reducir el tiempo de ejecución aprox. 60% (variables de enlace) Reducir las correcciones manuales de datos (2 incidentes/mes)
DI01	Implementación de código duplicado	<ul style="list-style-type: none"> Reducir el código que falta debido a un único flujo de desarrollo Facilitar la codificación de mantenimiento 	<ul style="list-style-type: none"> Reducir el esfuerzo de desarrollo y pruebas aprox. 50%
DI02	Modelado XML con muchas versiones DTD extendidas	<ul style="list-style-type: none"> Requisitos empresariales centralizados en un único recurso (de 200 DTD a 1 XSD) Fácil de detectar problemas de validación y comprensión del código. Mejor escalabilidad, ya que el modelo XML se puede ampliar más fácilmente Mejore la robustez a medida que XSD valida los tipos de datos de entrada Los analistas de negocios pueden asesorar más fácilmente, ya que el modelo XSD es fácil de aprender 	<ul style="list-style-type: none"> Reducir el tiempo de análisis y modelado aprox. 40%
DI03	Acceso a datos y capas de lógica empresarial mezcladas en el procesador de mensajes	<ul style="list-style-type: none"> Reducir el tiempo y los problemas de desarrollo Aplicación más orientada al negocio Mejor diseño de aplicaciones (separación de capas de acceso empresarial y de datos) 	<ul style="list-style-type: none"> Reducir el tiempo de desarrollo aprox. 50% Reducir el tiempo de investigación de la causa raíz aprox. 30% Reducir las correcciones de desarrollo

Evaluación de NRT-DWH

DW01	Varios sistemas Down streams acceden directamente a la base de datos NRT-DWH	<ul style="list-style-type: none"> • El aislamiento del modelo de datos permite implementar cambios o nuevos informes más fácil • Acceso de los Downstreams centralizado • Mejorar la robustez y el rendimiento debido a que el acceso desde los Downstreams está controlado • Los datos precalificados podrían mejorar el rendimiento de los informes 	<ul style="list-style-type: none"> • Reducir el tiempo de desarrollo de los informes aprox. 30% • El tiempo de notificación podría mejorarse aproximadamente un 40% (dependiendo de la consulta)
DQ01	Falta de procesos de calidad de datos	<ul style="list-style-type: none"> • El programa de calidad de datos proporcionará medidas, reglas de negocio y una mejor comprensión del propósito de la aplicación. • Mejorará la fiabilidad de los datos y los informes posteriores. • Estos procesos asesorarán sobre las brechas de datos. 	<ul style="list-style-type: none"> • Mejorar la fiabilidad de los datos aprox. 50%. • Supervisar los problemas de datos para realizar un seguimiento de las posibles gaps.
DF05-DF06	Investigar la producción fallida y reprocesar mensajes (perennes) en el entorno de producción	<ul style="list-style-type: none"> • Datos intradía precisos y fiables que proporcionan • Mejora el cumplimiento con el cliente y los informes reglamentarios • El proceso de fin del día tendrá que conciliar menos datos 	<ul style="list-style-type: none"> • Mejorar la idoneidad de los datos aprox. 1% (valores de posición y balance) • Reducir las correcciones de datos manuales (al menos 100 incidentes agrupados por sitio y tipo de mensaje)
DMON01	Error y reprocesar los mecanismos de retroalimentación de mensajes	<ul style="list-style-type: none"> • Mejor problemas de comunicación con los Upstreams • Reducir los problemas reportados por los Downstreams 	<ul style="list-style-type: none"> • Reducir el análisis de causa raíz (5000 filas ingestadas por error y reprocesar filas ingestadas por día aprox.) • Reducir el tiempo de respuesta de los datos aprox. 30%

6. CONCLUSIONES

En definitiva, las conclusiones sacadas de lo que al proyecto se refiere, se citan a continuación.

Se trata de una propuesta real para proveer al sistema actual de unas funcionalidades que se adapten a las ventajas que ofrecen las nuevas tecnologías, puesto que todas las implementaciones realizadas hasta la fecha, se han basado en mantener el funcionamiento sin tener en cuenta las posibles deficiencias que esto puede provocar a la hora de desarrollar nuevas mejoras o incluso la repercusión que puede tener en lo que a rendimiento se refiere.

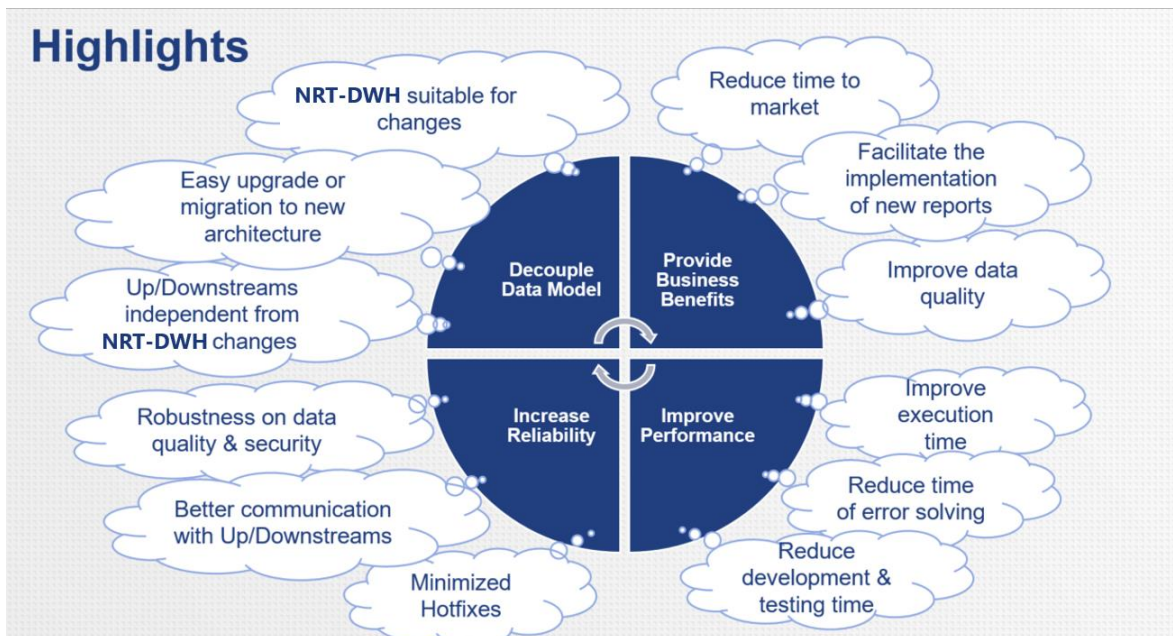
La propuesta contiene una primera opción más conservadora que consiste en actualización del sistema actual manteniendo la base de tecnología utilizada, visto que el cliente siempre ha optado por una vía en la que no se haga un gran esfuerzo para cambiar.

Por otra parte, la segunda opción proporciona un punto de vista más moderno y más orientado a las técnicas actuales aplicadas a los Datawarehouses, y que a su vez consiste en utilizar herramientas dedicadas en lo que a extracción de datos y análisis OLAP se refiere, y que facilitan tanto el desarrollo como la monitorización de procesos.

La forma de detallar estas dos opciones se decidió mediante las sesiones de seguimiento realizadas, en las que se iba proporcionando al cliente detalles de cada una de las secciones. Viendo que el cliente mostraba poco interés por un cambio radical en lo que a arquitectura se refiere, las aportaciones principales se fueron centrando en la primera opción, razón por la cual, el detalle de ésta es mucho más amplio.

En varias ocasiones se intentó hacer entender al cliente que el mayor esfuerzo a corto plazo de la segunda opción le proporcionaría un mayor beneficio a largo plazo en todos los sentidos, ya además este cambio implicaría también solucionar la gran mayoría de las deficiencias detectadas.

La siguiente imagen es un resumen de las ideas propuestas explicadas para cada área de interés en el documento. Esta es una forma holística y gráfica de ver todo el panorama.



Finalmente el cliente volvió a actuar de forma conservadora y decidió demorar el inicio del proyecto eligiendo como opción principal la primera de las ofrecidas.

7. ANEXO I

7.1 BIBLIOGRAFIA

- [1] ALCALDE, ALEJANDRO (2017). *PL/SQL. Disparadores o Triggers*. <https://elbauldelprogramador.com/plsql-disparadores-o-triggers> [Consulta: 16 de Septiembre de 2020].
- [2] IBM (2020). *IBM MQ*. <https://www.ibm.com/es-es/products/mq> [Consulta: 16 de Septiembre de 2020].
- [3] SINTES MARCO, BARTOLOMÉ (2015). *XML : Lenguaje de marcas extensible*. <https://www.mclibre.org/consultar/xml/lecciones/xml-dtd.html> [Consulta: 20 de Septiembre de 2020].
- [4] NAEEM TEHREEM (2020). *Ingestión de datos: definición, desafíos y mejores prácticas*. <https://www.astera.com/es/type/blog/data-ingestion> [Consulta: 15 de Noviembre de 2020].
- [5] University of Nebraska-Lincoln. *Database Design*. <https://its.unl.edu/bestpractices/database-design> [Consulta: 17 de Septiembre de 2020].
- [6] BASARANER CAGDAS (2012). *20 Database Design Best Practices*. <https://www.javacodegeeks.com/2012/02/20-database-design-best-practices.html> [Consulta: 21 de Septiembre de 2020].
- [7] Powerdata (2014). *Introducción a la Calidad de Datos: Definición, Control y Beneficios*. <https://blog.powerdata.es/el-valor-de-la-gestion-de-datos/bid/368784/introducci-n-a-la-calidad-de-datos-definici-n-control-y-beneficios> [Consulta: 16 de Septiembre de 2020].
- [8] Wikipedia (2020). *Confluence (software)*. [https://es.wikipedia.org/wiki/Confluence_\(software\)](https://es.wikipedia.org/wiki/Confluence_(software)) [Consulta: 24 de Julio de 2020].
- [9] ITRS Geneos (2018). *Geneos Product Overview – User Guide*. https://docs.itrsgroup.com/docs/geneos/4.3.0/Gateway_Reference_Guide/geneos_prod_overview_ug.html [Consulta: 27 de Julio de 2020].
- [10] APPDynamics art of Cisco (2009-2020). *AI-powered application performance management for the modern enterprise*. <https://www.appdynamics.com/product> [Consulta: 31 de Julio de 2020].
- [11] The APACHE Software Foundation (2020). *Apache POI - the Java API for Microsoft Documents*. <https://poi.apache.org/> [Consulta: 28 de Abril de 2020].
- [12] Software Selección (2020). *Jasper Reports*. <https://www.softwareseleccion.com/jasper+reports-p-2158> [Consulta: 2 de Mayo de 2020].
- [13] Arquitecturajava (2017). *Introducción a Java JAXB y el manejo de XML*. <https://www.arquitecturajava.com/introduccion-java-jaxb/> [Consulta: 7 de Mayo de 2020].
- [14] Baeldung (2020). *Simplify the DAO with Spring and Java Generics*. <https://www.baeldung.com/simplifying-the-data-access-layer-with-spring-and-java-generics> [Consulta: 8 de Mayo de 2020].
- [15] BLANCARTE, OSCAR (2018). *Data Transfer Object (DTO) – Patrón de diseño*. <https://www.oscarblancarteblog.com/2018/11/30/data-transfer-object-dto-patron-diseno/> [Consulta: 15 de Mayo de 2020].