



Herramientas de extracción, transformación y carga



I. Introducción

II. Objetivos

III. Qué es el proceso de extracción, transformación y carga (ETL)

IV. Proceso ETL en un proyecto de inteligencia de negocio

V. Tipos de cargas

VI. Gobierno del dato y orquestación

VII. Buenas prácticas

VIII. Herramientas ETL: Pentaho data integration

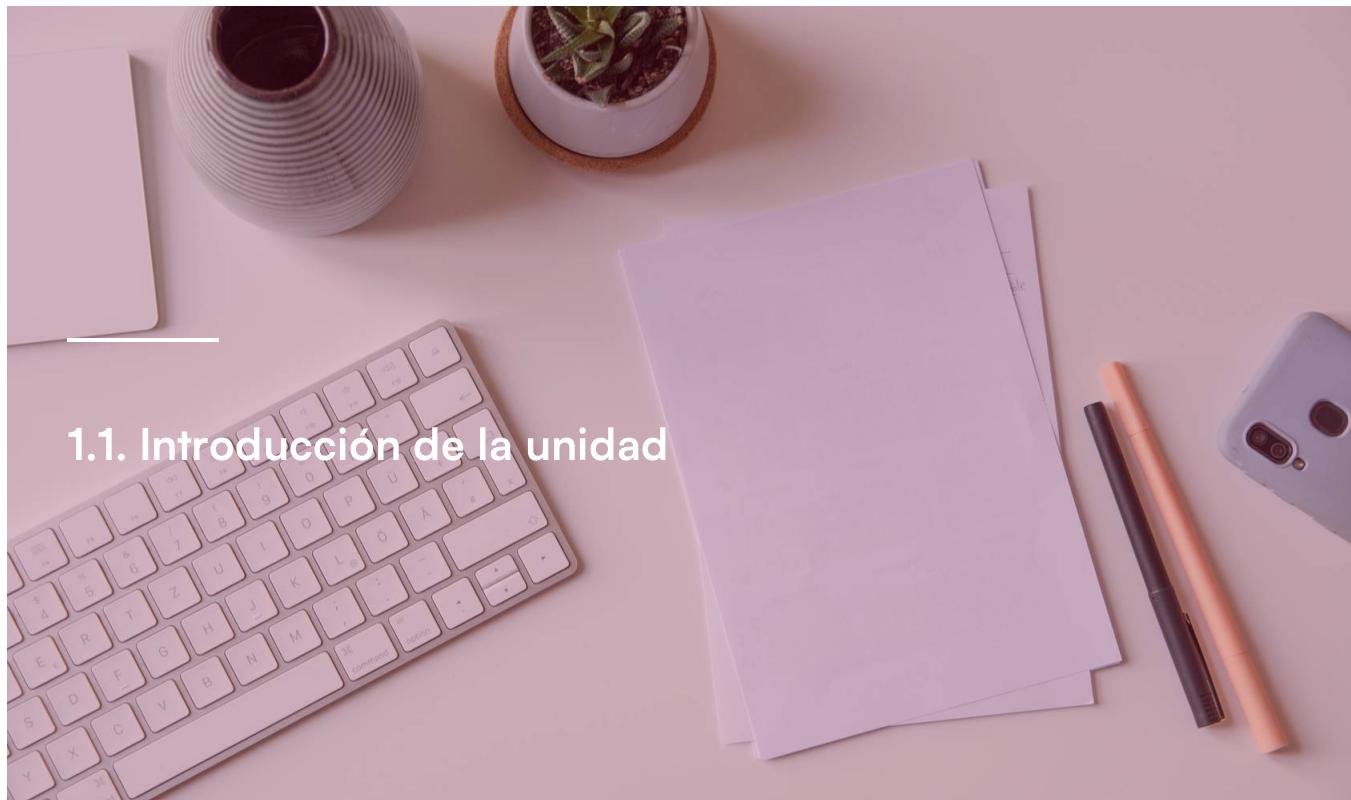
IX. Resumen

X. Caso práctico con solución

XI. Glosario

XII. Anexos

I. Introducción



1.1. Introducción de la unidad

Actualmente, tanto las empresas como las organizaciones invierten en proyectos de inteligencia de negocio con el objetivo de optimizar sus procesos y su actividad, y ser capaces de medir su desempeño. Su objetivo es mejorar el proceso de toma de decisiones generando un repositorio único de todas las fuentes que ofrezca una visión unificada y veracidad del dato.

Hasta este punto, en este módulo se han explicado la inteligencia de negocio y las bases de datos donde se crean *data warehouses* y *data marts*. Todo esto servirá como base para entender qué es un sistema de extracción, transformación y carga de datos. Estos procesos son más conocidos por sus siglas en inglés

ETL (*extract, transform, load*) y componen una de las piezas clave para el éxito de un sistema de inteligencia de negocio, y es en lo que profundizará esta unidad.

Como se escucha últimamente, los datos son el nuevo petróleo, pero estos **datos** hay que proveerlos a los usuarios en el **momento correcto** y con una **calidad** suficiente para poder tener un sistema de inteligencia de negocio exitoso.

En esta unidad se pone el foco en el proceso de **extracción** de la información desde su origen, y en cómo se **transforma** para **cargarla** y adaptarla al *data warehouse* empresarial. Se verán diversas técnicas para realizar estos procesos de carga, tanto para tablas de hechos como para las tablas dimensionales. Estas cargas pueden ser incrementales o cargas completas de datos, dependiendo de las necesidades, volumetrías y tiempos de reacción de los sistemas.

Los procesos ETL no son estándar, cada caso de uso y sistema tiene una casuística distinta. Esta unidad presentará las distintas estrategias de ETL que existen, para poder llevar a cabo un proceso de ingestión de datos apropiado para los sistemas de almacenamiento de datos tipo *data warehouse* en distintos casos de uso.

También se desarrollará un caso práctico en el que, usando una herramienta de ETL *open source*, se integrará la información de un fichero plano en un *data mart*. La unidad incluye un *webinar* en el que se mostrará el uso estándar de la herramienta ETL que se usará en el caso práctico.

II. Objetivos



2.1. Objetivos de la unidad

- 1 Conocer el proceso estándar de integración de datos en inteligencia de negocio: extracción, transformación y carga.
- 2 Entender qué procesos se implementan en cada una de las fases.
- 3 Entender las funciones de cada una de las áreas de almacenamiento de un sistema de BI.
- 4 Identificar diferentes tipos de cargas.

5

Conocer las bases de cada uno de los procesos de ETL.

6

Conocer una serie de buenas prácticas para este tipo de procesos.

7

Conocer e instalar Pentaho Data Integration Open Source.

8

Implementar un proceso de *data quality* con Pentaho Data Integration.

III. Qué es el proceso de extracción, transformación y carga (ETL)

3.1. Definición de ETL

En las unidades anteriores se vio cómo diseñar un *data warehouse* o un *data mart* en modelo estrella y copo de nieve. Así como los beneficios que traen para las organizaciones que integran este tipo de sistemas. Los procesos que alimentan de datos estas bases de datos son los de extracción, transformación y carga.

La extracción, transformación y carga de datos, más conocida por sus siglas en inglés, **ETL**, es el proceso que transporta los datos desde sus orígenes, los modifica convenientemente y los carga en los data warehouses y data marts de destino.

El proceso de ETL, en definitiva, es responsable de la extracción de datos y de su limpieza, conformación y localización en el almacén de datos.

Las herramientas de ETL recopilan datos de varias fuentes (tablas de bases de datos, archivos planos, ERP, internet, etc.) y les aplican transformaciones complejas.

Los procesos ETL son procesos cuyos flujos de datos están bien definidos, en los que se extraen datos de los sistemas de origen, como bases de datos relacionales, operacionales, ficheros, web services, etc. En siguientes pasos los datos extraídos se transforman, limpian, consolidan y se tratan para enriquecer tanto

dimensiones como las tablas de hechos con sus métricas. Este proceso de transformación siempre busca cumplir los requisitos solicitados por los usuarios de negocio. En el último paso se procede a la carga de la información formateada a un *data warehouse* o a un *data mart*.

En términos generales, tal y como señala Pentaho, las herramientas ETL deben estar orientadas a proporcionar lo siguiente:

Herramienta ETL 1

"Automatización de los procesos de extracción de datos de las diferentes fuentes identificadas, así como un mayor gobierno y control del proceso, con el fin de agilizar este procedimiento de descubrimiento de información y reducir el tiempo y el margen de error para cada nueva fase, así como añadir mayor flexibilidad para incorporar nuevas fuentes.

Herramienta ETL 2

Capacidad para acceder a diferentes tecnologías de origen de datos de forma eficiente.

Herramienta ETL 3

Proporcionar la gestión integrada del *Data Warehouse* y los *Data Marts* existentes en las fases de extracción, transformación y carga, para la implementación del *Data Warehouse* empresarial y de los *Data Marts*.

Herramienta ETL 4

Empleo de capa de metadatos que permita definir los objetos de negocio y las reglas de consolidación de una forma ágil e intuitiva.

Herramienta ETL 5

Acceso a una gran variedad de fuentes de datos diferentes.

Herramienta ETL 6

Control y manejo de excepciones".¹

¹SHitachi. Pentaho 9.2. [En línea] URL [disponible aquí](#).

Aunque conceptualmente es bastante sencillo, el proceso ETL es un componente crítico en una solución de inteligencia de negocio *end-to-end*. De hecho, es común reconocer que la construcción de estos procesos ETL, durante el proyecto de inteligencia de negocio, supone un gran esfuerzo en términos de tiempo y coste económico.

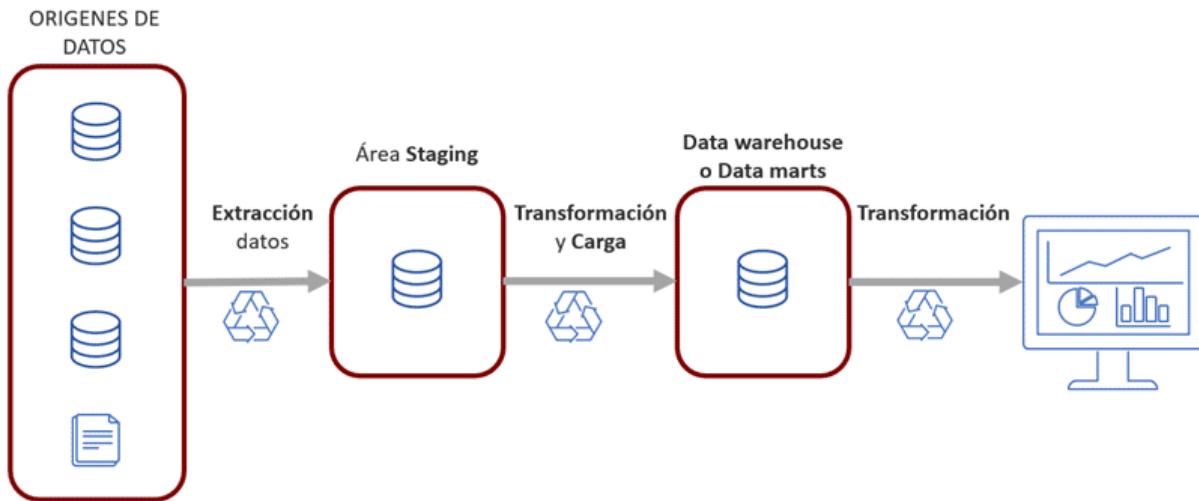


Figura 1. Proceso ETL.

Fuente: elaboración propia.

En los últimos años, la variedad de sistemas de datos origen y de formatos ha crecido exponencialmente. Existen muchos métodos de integración de datos en los sistemas de BI.

i Los procesos ETL vienen asociados a los sistemas de inteligencia de negocio. Hoy en día, con la diversidad de fuentes de datos, tanto en formato como en tipos de sistemas con los que interaccionar, las herramientas se han actualizado, adoptando las últimas tendencias de mercado. Esto incluye también el uso de la nube y la integración de datos en tiempo real. Por ello, las herramientas ETL siguen estando muy presentes en las organizaciones actualmente.

Las herramientas ETL aportan los siguientes valores a las empresas:

Sistemas centrales y organizados

para el movimiento, transformación y consolidación de información.

Entornos de trabajo

que promueven el desarrollo y la reutilización de código.

Integración de datos históricos

en tiempo diferido y datos en tiempo real o casi real.

Integración asíncrona de información en el *data warehouse*.

Esto permite cargar los datos independientemente de su origen.

Integración de diversos sistemas

y fuentes de datos en un repositorio único de datos que permite su análisis conjunto.

CONTINUAR

3.2. Fases y características de un proceso ETL

Tal y como sus siglas indican, el proceso ETL está compuesto por las fases de extracción, transformación y carga.

Extracción

El objetivo de esta fase es la extracción de los datos de cada una de las fuentes identificadas, realizando una copia exacta (sin procesamiento) de los datos de origen.

En esta fase se realiza el acceso a los datos de origen de una forma eficiente, periódica y automatizada, definiendo la conexión/integración con las fuentes de datos internas o externas que se hayan identificado.

Es importante destacar que en muchos casos se realiza extracción de fuentes operacionales, siendo crítica la carga que se realiza sobre estos sistemas. Es por ello por lo que hay que remarcar la importancia que tiene identificar y diseñar en esta fase las ventanas óptimas de carga sobre los orígenes, para generar la menor carga e impacto posibles sobre las fuentes.

En esta fase se deben **analizar los datos obtenidos** para verificar que cumplen las especificaciones deseadas. Algunas de estas validaciones son las siguientes:

- **Contrastar** con los datos de origen. Por ejemplo, el número de registros insertados y los totales de las métricas.
- **Cargar** solo datos que se van a usar en las siguientes fases.
- **Eliminar** datos duplicados.
- **Verificación** de tipo de datos.

Esta fase permite **integrar en un formato común** datos que provienen de sistemas totalmente dispares en cuanto a sistema operativo, protocolos de comunicación, API y estructura de datos. Así que es una primera fase preparatoria de datos, al estar todos en un mismo sistema denominado generalmente *staging*.

La extracción de datos tiene varios métodos para realizarse:

- **Extracción total de los datos de origen.**
- **Extracciones parciales** en las que se tienen en cuenta las **actualizaciones**.
- **Extracciones parciales** en las que **no** se tienen en cuenta las **actualizaciones**.

La extracción de datos debe **causar el mínimo impacto posible en el sistema origen** del que se extraen. La volumetría de los datos exportados puede provocar fallos de funcionamiento de los sistemas origen. Por esta razón los procesos ETL se planifican en ventanas de tiempo que no tengan un impacto directo en los orígenes.

Transformación

Es el segundo de los procesos y consiste, principalmente, en limpiar y conformar la información extraída de todas las fuentes. Este paso es el más laborioso y en el que ETL agrega más valor.

La **transformación** se basa en los datos almacenados en el área de **staging** en la fase de extracción. Estos datos extraídos desde la fuente origen están sin tratar, y no están pensados para la explotación, ya que su estructura no fue considerada para el análisis. Esta fase **procesará los datos basándose en transformaciones previamente acordadas** con el negocio o los usuarios finales.

En esta fase se realizan los procesos de limpieza de datos para aportarles calidad, eliminando, para ello, los datos erróneos, y, complementando con otras fuentes, se enriquecen los datos para que estos sean fiables.

En esta fase también se realiza el conformado y la unificación de fuentes y datos, lo que aporta unicidad a la información y veracidad y genera maestros. En esta fase se da la propia transformación de la información, en la que se aplican las reglas de negocio, comprobaciones, datos referenciales, etc. Este paso limpia y conforma los datos que se cargarán en el sistema final, que deben ser **precisos, completos, correctos y coherentes**.

Las transformaciones más comunes son las siguientes:

- **Limpiar** los campos que se van a cargar en la siguiente fase.
- **Traducir y mapear códigos entre maestros**, lo que permitirá tener una única tabla de maestros por cada dimensión que se vincule a diferentes tablas de hechos de diferentes orígenes. Por ejemplo, si la nacionalidad de los clientes en una fuente viene en número entero y en otra aparece el ISO2 del país, se debe decidir cuál será la codificación consolidada de ese maestro, y todos los datos cargados se tienen que transformar a esa codificación.
- **Consolidar y enriquecer dimensiones**; si se tienen tablas de búsqueda de diversas fuentes, se deben consolidar y enriquecer estas dimensiones.
- **Generar tablas de dimensiones** de conceptos que no tenían maestros, además de definir una codificación.
- **Dividir columnas**; por ejemplo, si la dirección completa de los clientes viene en un solo campo, hay que dividirlo en país, región, ciudad, etc., o fusionar columnas.
- **Consolidar tipos de datos y unidades de medida**, como monedas, fechas, peso, etc.

- **Validar** umbrales y formatos de los datos, como el formato del código postal.
- **Evitar** campos con **valores nulos**, cambiando su valor por valores por defecto.
- **Calcular métricas derivadas** usando las métricas originales como base.
- **Aplicación de reglas de calidad de datos**, y los procesos que realizar cuando no se cumplan los patrones definidos.

Carga —

Es el paso final del proceso y su objetivo es realizar de forma eficiente la carga de datos en el almacén de datos, y, más concretamente, en cada uno de los modelos multidimensionales definidos. Por tanto, su diseño debe prestar especial atención a minimizar los tiempos de carga y tener en cuenta que se trata de procesos intensivos de inserción —y actualización, si fuera necesario— de registros en tablas de elevada volumetría.

Existen varias **modalidades de carga de datos**, en algunos *data warehouses* se sobrescribe la información antigua con los nuevos datos, otros actualizan la información existente e incluso otros cargan fotos de datos.

Tras la carga se deben realizar **procesos de verificación** lo más automatizados posible, como los siguientes:

- Los **campos clave** por los que se vinculan las tablas no deben ser nulos.
- **Comprobar** que los **cálculos** de medidas derivadas funcionaron correctamente.
- **Comprobar** tablas de **dimensiones** y de **hechos**.
- **Verificar** los **informes** en el sistema de *reporting* de inteligencia de negocio.

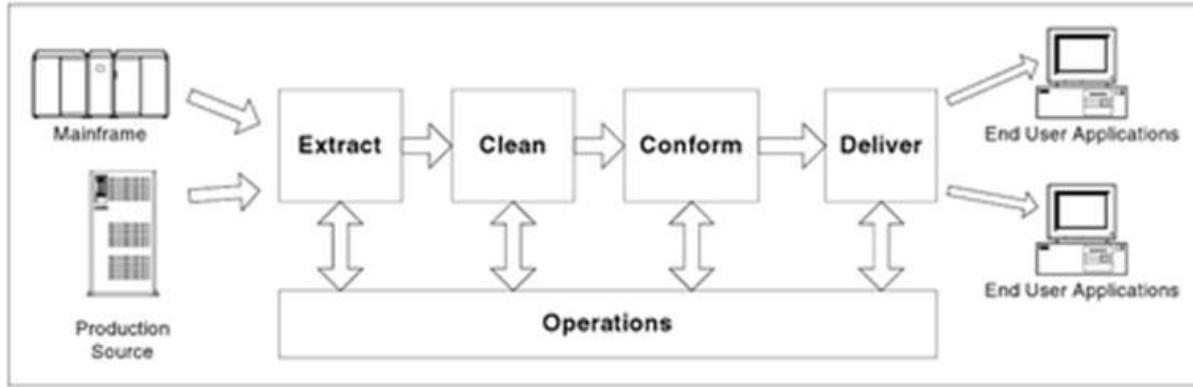


Figura 2. Flujo de datos.

Fuente: Kimball, R.; Caserta, J. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley & Sons; 2011.

Las principales características de un procesamiento ETL son las siguientes:

- **Procesamiento:** los datos se almacenan inicialmente en un repositorio común denominado área de *staging*, luego se transforman y transfieren al *data warehouse* o *data mart*.
- **Tiempos de transformación:** el sistema ETL tiene que esperar que acabe la extracción para empezar la transformación. A mayor volumen de datos, más tiempo de extracción y procesado necesita el proceso.
- Está especialmente pensado para *data warehouses* y *data marts*.
- El proceso ETL carga solo datos que se solicitaron por los usuarios para cada caso de uso.

Los procesos ETL no suelen ser sencillos, el inicio del desarrollo es clave, estos procesos mal planteados inicialmente acarrean muchos problemas en el mantenimiento y la escalabilidad del sistema.

Los problemas pueden venir por varios frentes:

1	2	3	4	
---	---	---	---	--

Una **dificultad** operativa excesiva.

1	2	3	4	
---	---	---	---	--

Un **mal diseño** de las soluciones.

1	2	3	4	
---	---	---	---	--

Calidad de los datos de origen; se recomienda analizar la calidad de los datos de origen durante el análisis.

1	2	3	4	
---	---	---	---	--

Falta de planificación y **análisis de la volumetría** de los datos de origen.



1	2	3	4
---	---	---	---

Carencia de estudio de ventanas de acceso a sistemas de origen.

1	2	3	4	
---	---	---	---	--

No planificar correctamente la escalabilidad de los procesos. Y de la solución ETL en sí.

IV. Proceso ETL en un proyecto de inteligencia de negocio

4.1. Fase de implementación en un proyecto ETL estándar

Para comprender mejor cuál es la función específica de cada una de las fases de un proceso ETL, se va a analizar la fase de implementación de un proyecto ETL estándar, dando por asumido que ya se ha realizado la fase de identificación de requisitos y diseño de arquitectura.

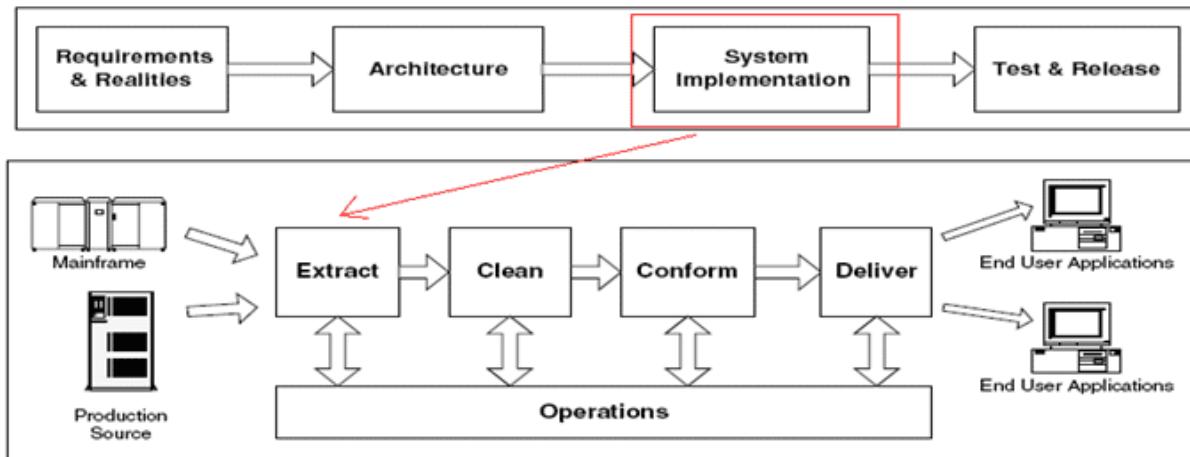


Figura 3. Flujo del proyecto: implementación.

Fuente: adaptado de Kimball y Caserta (2011).

A continuación, dentro de la implementación, se examinará cada uno de los apartados del flujo de datos relacionados con el proceso ETL.

4.1.1. Flujo de datos. Extracción

A continuación, dentro de la implementación, se examinará cada uno de los apartados del flujo de datos relacionados con el proceso ETL.

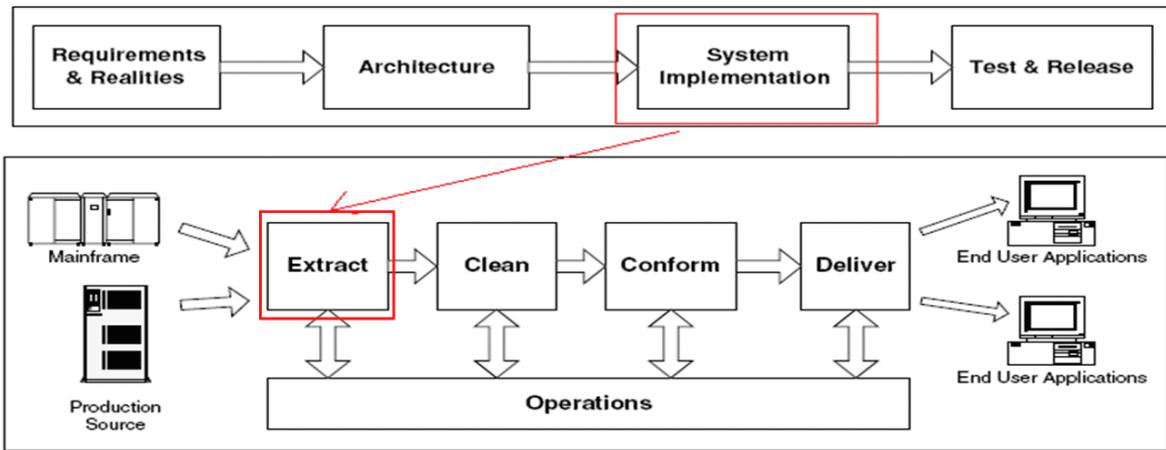


Figura 4. Flujo de datos: extracción.

Fuente: adaptado de Kimball y Caserta (2011).

Durante esta fase de extracción, las tareas que realizar e implementar serían las siguientes:

Analizar cómo han sido diseñadas las fuentes

- Identificar claves primarias y foráneas.
- Comprobar los tipos de datos y su formato.
- Comprobar relaciones existentes entre los datos.

Analizar cuál es el estado de las fuentes a nivel de datos

- Formatos de campos, fechas.
- *Missing values*, nulos.
- Volumen de datos.
- Forma de actualizar la información, borra o guarda histórico. Uso de campos *timestamp* para identificar el momento en que se modificó, creó o borró cada registro.

Asimilar las reglas (técnicas) de negocio que han sido aplicadas a los datos

Por ejemplo, diferencias en cómo se generan los códigos de cliente entre distintas fuentes; los productos solo vienen de una fuente...

Identificación de fuentes

Tablas de datos, Excel, ficheros planos, etc.

4.1.2. Flujo de datos: limpieza y conformado

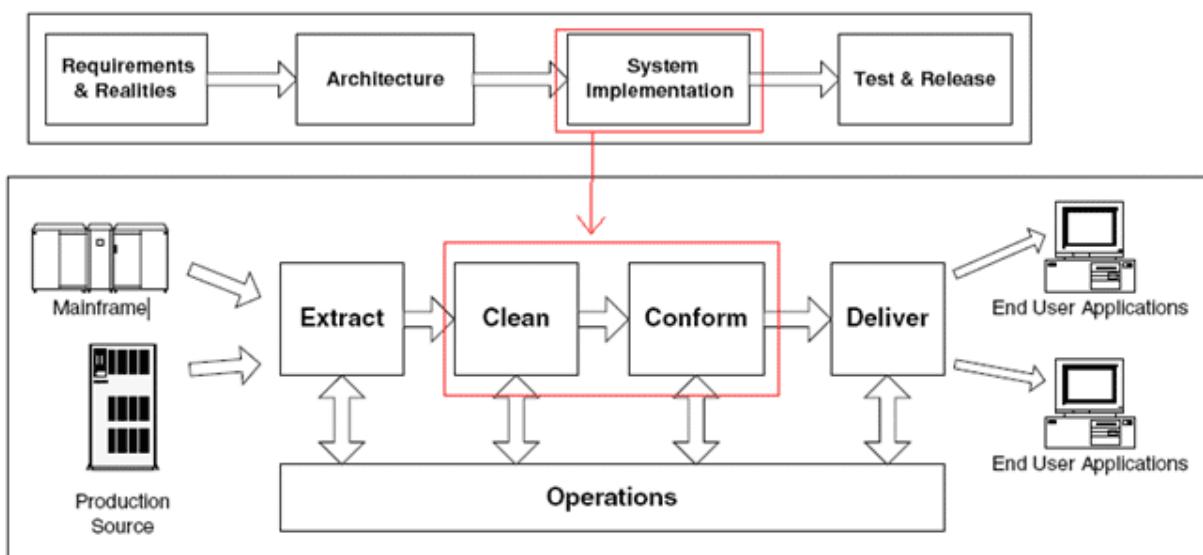


Figura 5. Flujo de datos: limpieza y conformado.

Fuente: adaptado de Kimball y Caserta (2011).

CONTINUAR

Son los pasos que más valor aportan al dato, por lo que es necesario un trabajo previo intenso de diseño para definir las reglas. A continuación, se describen las funciones que se realizan en esta fase:

La limpieza, por norma general, hace referencia a lo siguiente

- A nivel de columnas:
 - Verifica que no haya nulos donde no debe haberlos.
 - *Outlayers*.
 - Verifica la definición de tamaño de columna.
- A nivel de tablas:
 - Verifica las correctas relaciones definidas entre tablas.
- Aplicación de reglas de dato y valor:
 - La función depende de la herramienta o lenguaje que se emplee. Por ejemplo: si un producto es *premium*, su coste asociado ha de ser XXXX.
 - Formatos de campos. Por ejemplo, los códigos postales han de tener cinco dígitos.

Conformado o consolidación de conceptos

El conformado consiste en la unificación del concepto en toda la organización. Debe significar y expresarse de la misma manera, en toda la organización, independientemente del departamento y persona que lo utilicen. Un ejemplo en donde sería necesaria la aplicación del conformado podría ser el de un escenario bastante habitual en el que un departamento llama al sexo de las personas (H, M), otro, (1, 2), etc. En este caso, convendría realizar una unificación en toda la compañía de cómo se expresa dicho concepto.

i Es importante subrayar que, en todos los modelos de estrella, una dimensión tiene el mismo significado y atributos y viene de las mismas fuentes. El objetivo de ETL es implementar los procesos, no definir la dimensión. Pero esta dimensión debe ser válida para todas las tablas de hechos con las que se vincule, sea cual sea su origen.

También hay que tener presente que un hecho está conformado cuando significa lo mismo para todos los usuarios, departamentos y modelos, y se calcula igual en todas las estrellas que interviene y puede actuar directamente en comparaciones y cálculos.

De igual forma, se debe aplicar el conformado a los hechos y a lo que cuantifica o quiere medir cada hecho. Por ejemplo, son hechos habitualmente conformados los ingresos, las ventas directas por mes, las suscripciones al año, etc. Y todos ellos deben significar lo mismo sea cual sea el departamento o el usuario que los utilice. Además, es evidente, pero necesario, remarcar que el hecho de expresar el mismo concepto implica que deben ser calculados de la misma manera.

Por último, en ocasiones, puede encontrarse una situación en donde pueda existir (y se requiera) un mismo hecho con dos denominaciones. Aunque es una situación que es mejor evitar, es importante que este hecho también sea conformado. Pero ¿cómo se comparan? De la misma forma, las expresiones que los componen deber ser idénticas y deben obtener idénticos resultados.

4.1.3. Flujo de datos: carga o entrega

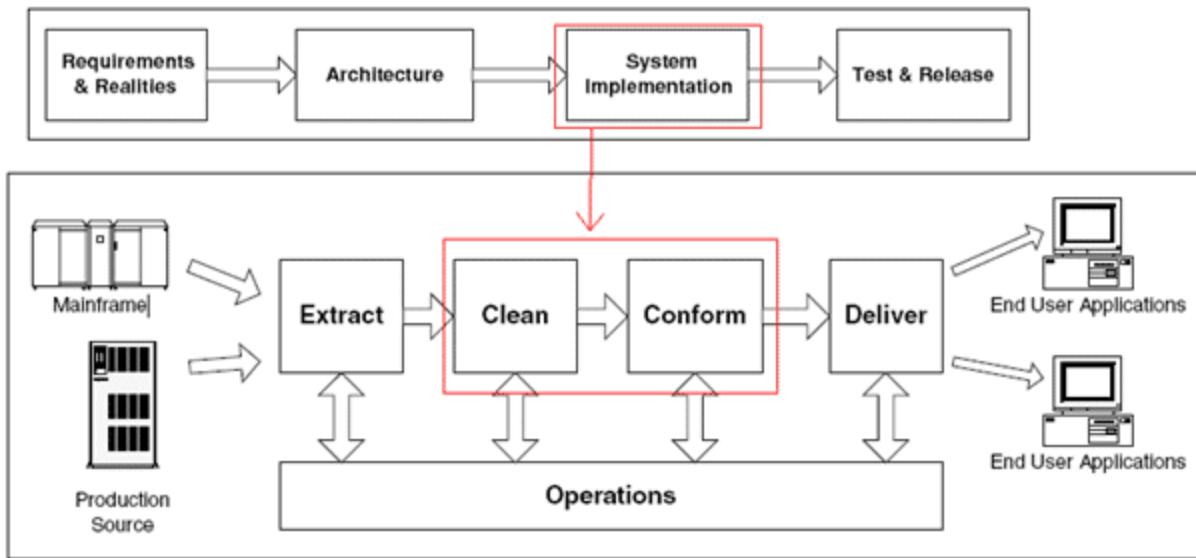


Figura 6. Flujo de datos: entrega.

Fuente: adaptado de Kimball y Caserta (2011).

DIMENSIONES

HECHOS

Como ya se ha ido introduciendo en este módulo, las dimensiones representan los ejes de análisis por los que se va a querer analizar la información. Independientemente del tipo de información que pueda almacenar la dimensión (tiempo, cliente, canal, producto, geografía, etc.), es posible hacer una primera gran categorización de las dimensiones atendiendo a dos grandes criterios:

- Tipo de dimensión en cuanto a distribución. Hace referencia a la distribución física de la dimensión. Como se está viendo, podemos encontrar implementaciones físicas de una dimensión completamente desnormalizadas en un modelo en estrella, en donde toda la información de la dimensión está contenida en una única tabla, o bien dimensiones normalizadas en un modelo copo de nieve, o **snowflake** en inglés, en donde la jerarquía contenida en la dimensión es reflejada físicamente en la dependencia de varias tablas.
- Tipo de dimensión en cuanto a cambio. Las dimensiones pueden contener información que no varía en el tiempo o, por el contrario, información que es susceptible de cambiar en el tiempo a mayor o menor velocidad. Son las denominadas dimensiones lentamente cambiantes o SCD (*slow changing dimensions*), que se verán más adelante en esta unidad.

A la hora de definir e implementar dimensiones, estas son algunas recomendaciones básicas que se deben seguir (las cuales afectan al rendimiento y mantenimiento de estas):

- Generar las claves subrogadas con secuencias. Esto significa crear una clave nueva, que sea un número entero autoincremental. Luego, las tablas de hechos harán referencia a la dimensión por esta clave.
- Utilizar estrategias *insert/update*.
 - Seleccionar qué campos hacen diferente a un elemento de dimensión de otro.
- Para las SCD, si la herramienta lo permite, dejar en sus manos el control de versiones.

DIMENSIONES

HECHOS

Del mismo modo, los hechos, como se ha dicho, representan y contienen aquello que se quiere medir o cuantificar. Independientemente de la temática del hecho que se esté definiendo, es importante definir y/o identificar para cada uno de ellos:

- Hechos:
 - Qué tipo de hechos son: aditivos, semiaditivos, no aditivos (*junk dimensions*).
 - Para cada uno de ellos, el tipo de función de agregación que se debe usar.
 - Identificar y diferenciar si es un hecho calculado en la ETL o al vuelo.
- A la hora de definir e implementar hechos, estas son algunas recomendaciones básicas que seguir (las cuales afectan al rendimiento y mantenimiento de estos):
 - Tipo de tabla de hechos que implementar.
 - Mantener la integridad referencial en el proceso ETL (*lookup*).
 - Tabla *lookup* vs. dimensión *lookup* (soluciones en memoria).
 - O con apoyo a una tabla del *data stage* mediante *joins*.

V. Tipos de cargas

Los tipos de carga se pueden categorizar de una forma sencilla de la siguiente manera:

Carga inicial

Rellenar todas las tablas de *data warehouse* por primera vez; en principio, solo se debe realizar una única vez tras construir el sistema.

Carga incremental

Este tipo de carga añade información al *data warehouse*, incluyendo nueva información sin borrar el contenido de las tablas.

2 of 4

Actualización completa

A veces es más rápido borrar el contenido de una o más tablas y volverlas a cargar de nuevo.

3 of 4

Carga de fotos

O *snapshots*, por su traducción al inglés. Se aplica, principalmente, a tablas de hechos. Se inserta la foto actual de los datos de origen en el momento de la carga, identificando todos los registros cargados en dicha iteración con una fecha de foto común. En este tipo de

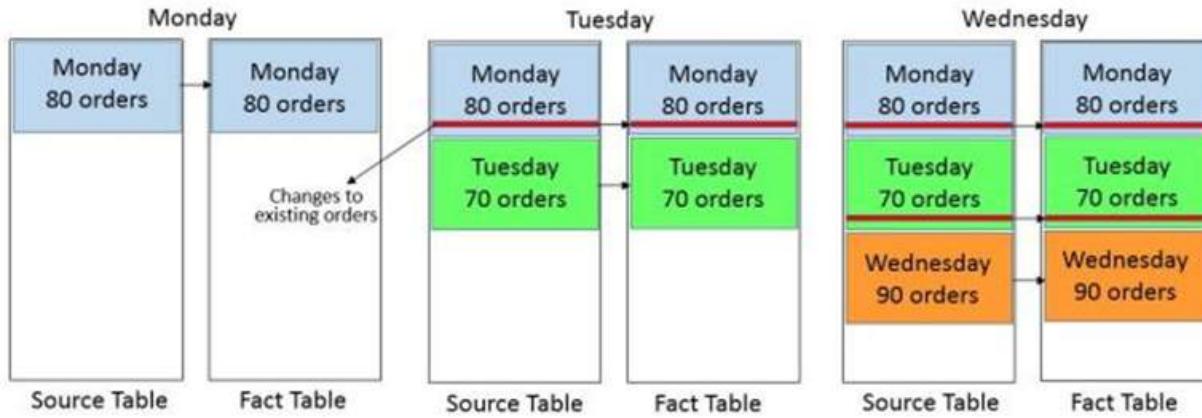


Figura 7. Snapshot fact table.

Fuente: <https://dwbi1.wordpress.com/2019/02/20/transactional-fact-tables/>

Otro concepto para tener en cuenta son las dimensiones lentamente cambiantes o *slowly changing dimension (SCD)*. Estos métodos se aplican en dimensiones que cambian de manera ocasional, aunque son técnicas que se pueden usar para todas las dimensiones. Ralph Kimball propuso tres tipos de procedimientos, a los que se añadieron dos más con el paso del tiempo. Este tipo de dimensiones deben tener una clave subrogada, es decir, **incremental autoasignada**.

SCD tipo 1: sobrescribir la fila

No guarda históricos, no añade filas a no ser que existan registros nuevos. Si el registro cambia, solo actualiza los cambios producidos.

SCD tipo 2: añadir fila

Si el registro que se va a insertar ya existía, pero ha cambiado, se inserta una nueva fila. Este método guarda el histórico de los datos. La tabla necesita tener varias columnas para registrar los cambios, como fecha inicio y fin del registro, versión y versión actual, que indica si es el último registro introducido.

El ejemplo presenta la situación de un cliente que cambia de ciudad. Si se sobrescribe el registro en la tabla dimensión cliente, todas las ventas de ese cliente cambiarían de ciudad. Teniendo un histórico de los cambios de clientes estas ventas se asignarán correctamente.

Figura 8. SCD 2.

Fuente: elaboración propia.

Clientes

ID	CUST_ID	First name	Last name	City
1	ASDF	Pedro	Romero	Madrid
2	GFDR	Juan	Sanches	Lima
ID CUST_ID First name Last name City				
1	ASDF	Pedro	Romero	Madrid
2	GFDR	Juan	Sanches	Lima
3	ASDF	Pedro	Romero	Barcelona

Pedidos

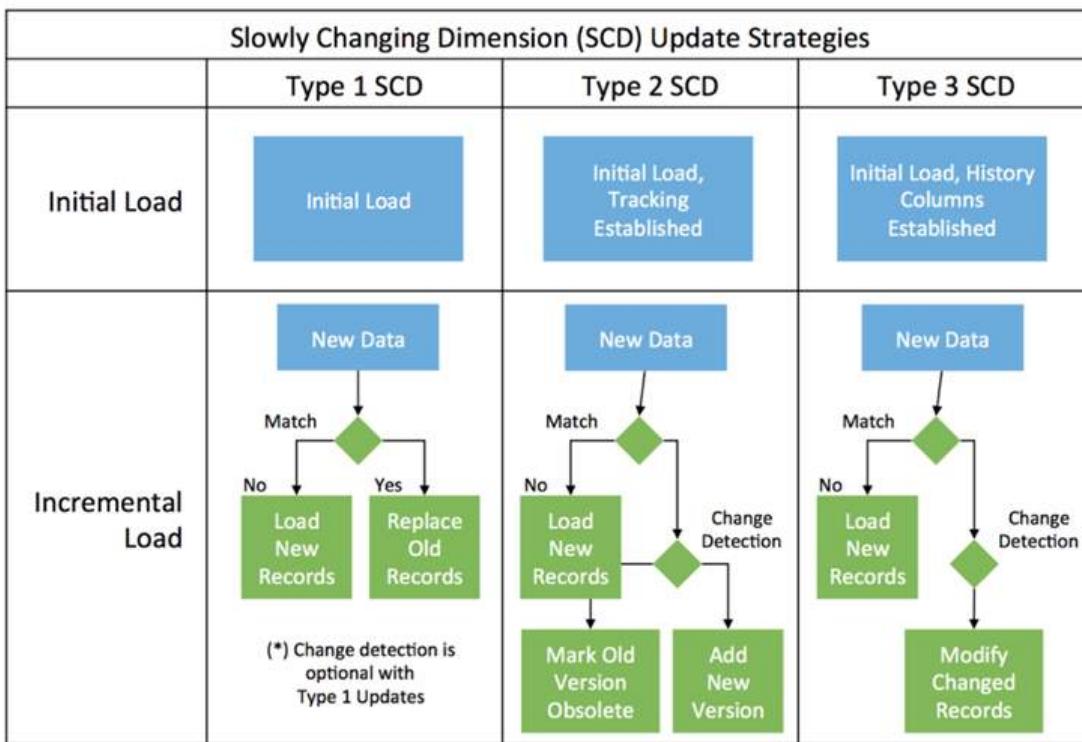
Order Id	Quantity	Customer Id
1	100	2
2	123	3
3	35	1
4	467	4

SCD tipo 3: añadir columna

Por eso hay que especificar sobre qué campos se quiere controlar el histórico. Para cada columna que se va a controlar los cambios tendrán dos campos, actual y anterior; cuando viene un valor que hay que actualizar, se asignan los datos de la columna actual a la anterior. Tiene la limitación a la hora de guardar valores históricos, ya que solo guarda el último cambio.

Figura 9. SCD 1, 2 y 3.

Fuente: Shanklin, C. “[Managing Slowly Changing Dimensions](#)”.



SCD tipo 4: tabla de historia separada

Se almacenan en una tabla separada los detalles de los cambios históricos producidos en la tabla de maestros; esta tabla sería similar a las de tipo 2. Y luego cuenta con una tabla de maestros con los valores actuales y sin histórico que será más ágil para consultas.

SCD tipo 6: híbrido

Realmente existen tipos 5 e incluso 7, que son versiones híbridas de los anteriores tipos. El tipo 6 es el más destacable. Combina los tipos 1, 2 y 3. Se denomina tipo 6 porque es la suma de los tipos que combina 1 + 2 + 3.

Este tipo se basa principalmente en el tipo SCD 2, pero incorpora los atributos actuales en las filas históricas para poder filtrar la tabla de hechos. En el ejemplo se puede ver cómo se añaden filas siguiendo la tipología 2, pero se crea un campo de valor actual que determina el valor actual de esos registros anteriores.

Figura 10. SCD 6.

Fuente: [Kimball Group](#).

Original row in Product dimension:							
Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Education	2012-01-01	9999-12-31	Current

Rows in Product dimension following first department reassignment:							
Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Strategy	2012-01-01	2012-12-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	Strategy	2013-01-01	9999-12-31	Current

Rows in Product dimension following second department reassignment:							
Product Key	SKU (NK)	Product Description	Historic Department Name	Current Department Name	Row Effective Date	Row Expiration Date	Current Row Indicator
12345	ABC922-Z	IntelliKidz	Education	Critical Thinking	2012-01-01	2012-12-31	Expired
25984	ABC922-Z	IntelliKidz	Strategy	Critical Thinking	2013-01-01	2013-02-03	Expired
31726	ABC922-Z	IntelliKidz	Critical Thinking	Critical Thinking	2013-02-04	9999-12-31	Current

Dependerá de la casuística de la dimensión el elegir una u otra opción. A veces se pueden combinar varias de ellas.

VI. Gobierno del dato y orquestación

Todas las herramientas tecnológicas hoy en día sirven para cubrir la mayoría de los requisitos empresariales. El **gobierno del dato** se ha convertido en un factor esencial para poder cumplir los requisitos establecidos por las organizaciones.

La **gestión del dato** ha evolucionado en los últimos años, junto con la inteligencia de negocio. Ya no se ve como algo disruptivo o innovador, sino como una necesidad básica que las empresas deben incorporar en sus sistemas.

La aplicación de buenas prácticas en la adopción del gobierno del dato facilitará obtener casos de uso exitosos. También es importante culturizar en los principios del gobierno del dato a los usuarios clave.

- (i) La orquestación de los procesos ETL permite la ejecución de los procesos ETL de forma sincronizada. La orquestación es necesaria, ya que normalmente los procesos de los sistemas de inteligencia de negocios y *big data* dependen de diversos orígenes de datos y de la finalización correcta de procesos previos.

Las tareas de orquestación incluyen configurar, coordinar y gestionar automatizaciones de los sistemas de integración de datos. La orquestación de procesos automatizados persigue la integración de modelos más grandes de datos.

Se entiende la **orquestación de datos** como la automatización de tareas basadas en datos de principio a fin. Procesos que involucran diversos sistemas, departamentos y fuentes de datos.

Ejemplo

Por ejemplo, si hay que cargar datos de un CRM de un uso intensivo se buscará una ventana en que no se afecte a su funcionamiento; estas cargas suelen realizarse de forma diaria por la noche.

Si, por otro lado, se cuenta con una aplicación de terceros a la que no se tiene acceso directo, como una web, habrá que realizar una descarga de ficheros planos para integrar dicha información. Si esta carga hay que realizarla tres veces al día, y depende de que la carga nocturna del CRM fuera correcta, esto solo se puede alcanzar con la automatización de los procesos. Automatizar todo esto se realiza con las herramientas de orquestación.

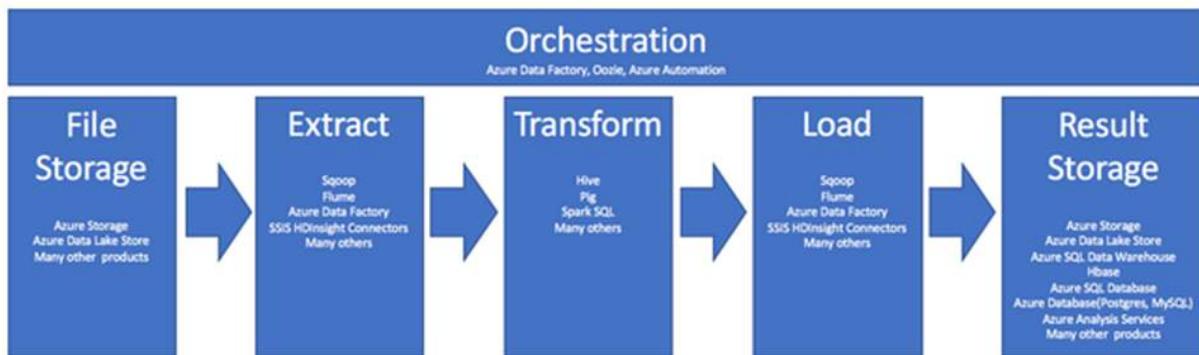


Figura 11. Orquestación ETL HDInsight.

Fuente: <https://docs.microsoft.com/es-es/azure/hdinsight/hadoop/apache-hadoop-etl-at-scale>.

La orquestación afecta a todos los pasos de un proceso de extracción, transformación y carga de datos. Normalmente, la orquestación tiene que sincronizar el funcionamiento de los distintos componentes de un sistema de integración de datos, que tienen que procesarse en el momento adecuado y con todas sus dependencias ejecutadas.

Por lo tanto, estos sistemas sirven, principalmente, para planificar tareas de procesado de datos. Establecer dependencias entre estos procesados y controlar el estado de las tareas.

VII. Buenas prácticas

Uno de los mayores problemas a la hora de diseñar y desarrollar ETL es el incremento del volumen de datos que cargar. Cuando estos volúmenes son altos, los tiempos de procesado crecen y pueden ser inmanejables.

Los típicos problemas de ETL son los tiempos de procesado, que retrasan la generación de informes automatizados. Estos problemas se pueden deber a una mala configuración de la base de datos en lo referente a indexación, otras veces el código ETL no es eficiente ni escalable.

La puesta en práctica de unos buenos principios en el desarrollo de ETL permitirá tener una solución fiable, flexible, con un buen rendimiento y fácil de escalar y mantener.

A continuación, se aportan recomendaciones para desarrollar y gestionar ETL. No siempre son de obligado cumplimiento, ya que ciertas restricciones pueden obligar a no cumplir alguna de estas prácticas:

CONTINUAR

- Las cargas deben realizar **validaciones**, como recuentos de filas o comprobaciones de métricas.
- **Linaje de datos.** Hay que conocer el origen, el momento y las transformaciones que originaron un dato.

- **Modularidad.** Se debe desarrollar el proceso ETL pensando en reutilizar el mayor volumen de código posible.
- **Gestión de errores.** Listado de acciones que llevar a cabo si el proceso ETL falla.
- **Gestión de datos incorrectos.** Protocolo que seguir si se encuentran datos incoherentes.
- Las cargas deben **move el menor volumen de datos**, solo el estrictamente necesario desde el sistema de origen. Añadir campos que identifiquen el momento en que se actualizó o se creó esa fila en el origen ayudará mucho a una posterior extracción.
- **No realizar cruces con tablas maestras muy grandes** durante la ETL; esto reducirá la velocidad de procesado.
- Se debe intentar realizar cargas por bloques, no fila a fila.
- Conviene **delegar** las tareas de consultas, uniones y búsquedas dentro de lo posible **a las bases de datos**, que lo realizarán más eficientemente que una herramienta ETL.
- Buscar el mayor **parallelismo** en el procesado de las tareas, usando tantos hilos paralelos como sea posible.
- Reconstruir los **índices** de las tablas una vez acabe el proceso ETL.
- Activar las **estadísticas** en la base de datos de destino. Esto optimizará las consultas que se realicen sobre ella.
- **LIMITAR consultas complejas** con varias uniones y cruces de tablas, ya que será menos óptimo. Se puede evaluar **generar tablas intermedias**.

VIII. Herramientas ETL: Pentaho data integration

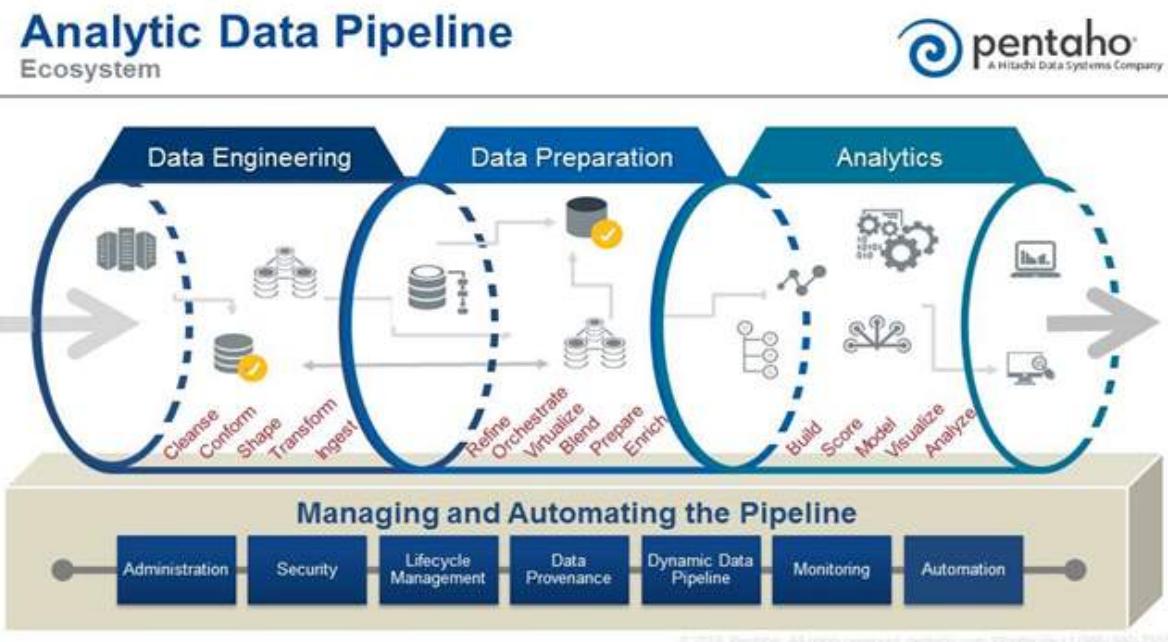


Figura 12. ETL pipeline.
Fuente: documentación oficial de Pentaho.

8.1. Qué es Pentaho Data Integration

Pentaho Data Integration (PDI), propiedad de Hitachi Vantara, y también conocido como Kettle, es un set de herramientas que permite diseñar ETL mediante transformaciones y trabajos que pueden ser ejecutados por las herramientas **Spoon**, **Pan** y **Kitchen**. Antes se conocía con el nombre de **Kettle**.

Es una aplicación totalmente orientada a las transformaciones y es sencilla a la hora de gestionar tipos de datos diferentes.

PDI está formado por un conjunto de subprogramas:

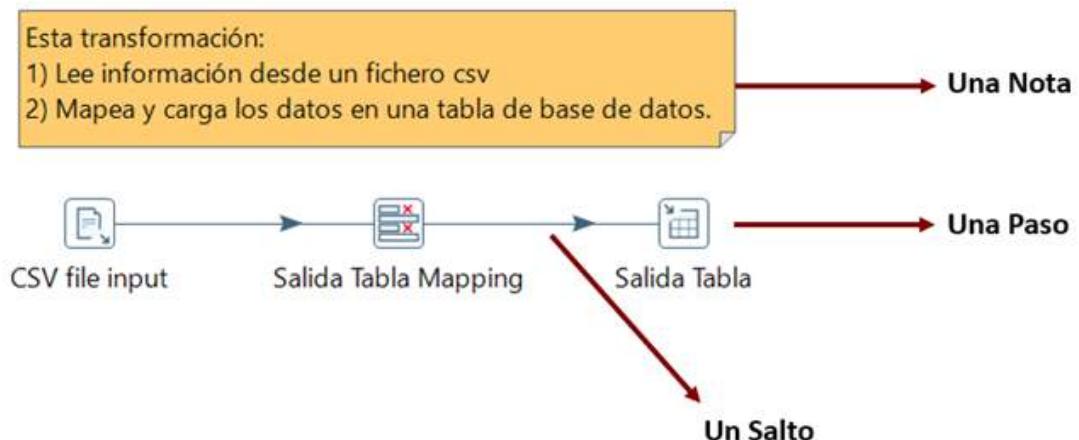
Spoon

Es la interfaz gráfica para diseño de trasformaciones y trabajos ETL. Es la pieza fundamental para la generación de procesos ETL con PDI. En esta herramientas es donde se realizan los desarrollos de los pasos de la ETL y permite probar los objetos que se van integrando en los nuevos desarrollos.

Desde **Spoon** se generan **pasos** para realizar las extracciones, transformaciones y cargas necesarias. Estas se sincronizan mediante **saltos** que organizan el orden de los pasos. Estos pasos se agrupan en **transformaciones**, que son la unidad mínima ejecutable.

Figura 13. Ejemplo transformación PDI.

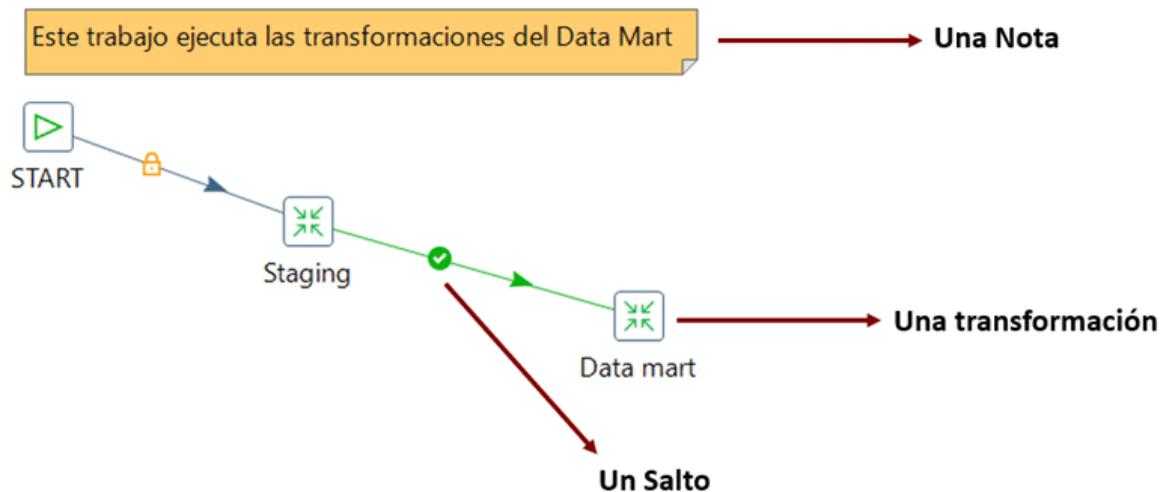
Fuente: elaboración propia.



Las tareas se agrupan, organizan y sincronizan en trabajos o *jobs*.

Figura 14. Ejemplo de tarea o *job* PDI.

Fuente: elaboración propia.



Pan

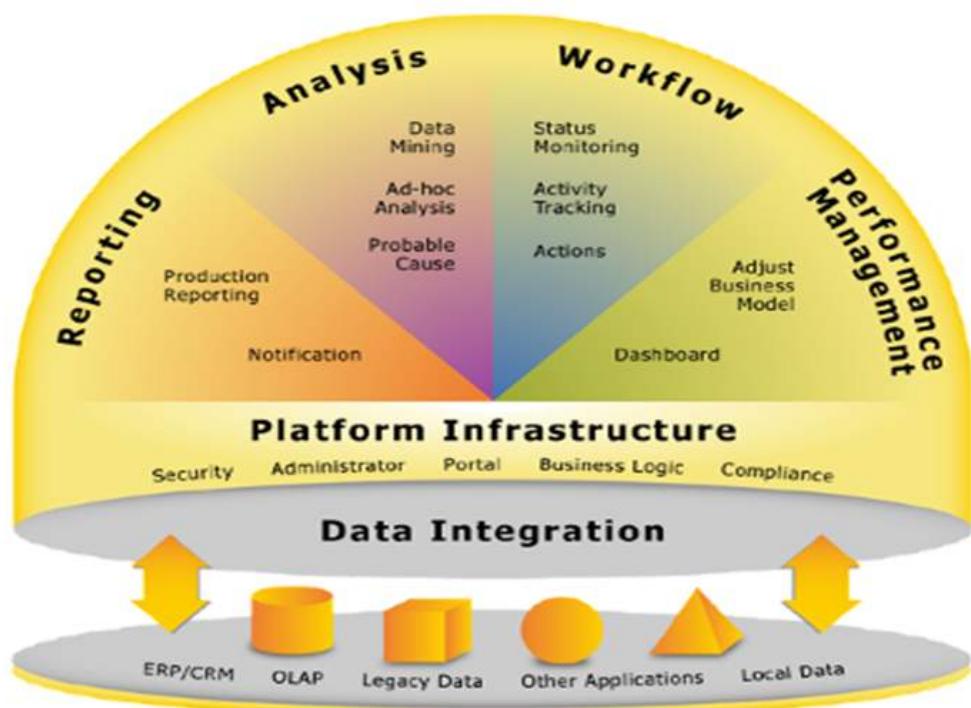
Es la solución que permite procesar las **tareas** de ETL desarrolladas con Spoon desde un fichero o desde un repositorio. Lanza los procesos de carga mediante *scripts*. Esta herramienta se lanza desde línea de comando.

Kitchen

Es un programa que ejecuta los trabajos diseñados por Spoon. Normalmente, estos trabajos son planificados en modo *batch* para ejecutar automáticamente en períodos regulares (*crontab -e*). Es parecido a la anterior solución pero ejecuta tareas. Esta herramienta se lanza desde línea de comando.

Figura 15. Estructura y capas identificadas en Pentaho Business Intelligence.

Fuente: documentación oficial de Pentaho.



CONTINUAR

8.2. Características y beneficios

- Permite trabajar con un repositorio en base de datos o en ficheros.
- Su interfaz gráfica permite crear transformaciones y trabajos de manera intuitiva mediante pasos modulares ya creados, conexiones con múltiples fuentes, etc.
- Distribución y combinación de diferentes fuentes, en diferentes *hosts*.
- Interfaz SQL y generador de código automático.
- Crea cálculos de una manera muy sencilla.
- Define qué se quiere hacer, no cómo hacerlo.
- Genera código XML y Java.
- Se debe intentar realizar cargas por bloques, no fila a fila.
- Instalación sencilla –solo extraer los ficheros, aplicación Java–.
- Fácil de mantener, con alto rendimiento y escalabilidad.
- Es posible parametrizar bastantes configuraciones (directorios, conexiones, correo electrónico).
- Posee una arquitectura de *plugin* que permite expandir sus funcionalidades.

8.3. Instalación de Pentaho Data Integration

Todos los pasos necesarios para su instalación se encuentran en el anexo 1 de esta unidad. Aunque ya se encontrará la solución instalada en la máquina virtual que se adjunta con el módulo.

8.4. Spoon: consola gráfica de diseño

A continuación, se muestra cómo iniciar Spoon en Pentaho Data Integration (PDI).

Para iniciar el cliente de escritorio de Pentaho Data Integration (PDI), hay que seguir las siguientes indicaciones.

Si se instaló Pentaho utilizando el método manual, se puede iniciar el cliente PDI desde el directorio Pentaho.

Hay que navegar a la carpeta donde se ha instalado PDI. Por ejemplo:

...\\pentaho\\design-tools\\data-integration.

Es necesario iniciar el cliente de PDI, según el sistema operativo que se utilice:

- Para Windows: hay que hacer doble clic en "Spoon.bat".
- Para Linux: hay que hacer doble clic en "spoon.sh" o abrir un terminal en el directorio que contiene el ejecutable "spoon.sh" y ejecutar: "./spoon.sh".
- Para Macintosh: hay que hacer doble clic en "spoon.sh" o abrir un terminal en el directorio que contiene el ejecutable "spoon.sh" y ejecutar "./spoon.sh"-

8.4.1. Perspectivas de trabajo

Pentaho Data Integration (PDI) permite utilizar herramientas que incluyen ETL y la programación de estas mismas en un entorno unificado: la interfaz del cliente PDI (Spoon).

Además, este entorno integrado posibilita una estrecha colaboración entre distintos usuarios, utilizando un repositorio con control de versiones, para crear soluciones de inteligencia empresarial de manera más rápida y eficiente.

Cuando se trabaja en el cliente de PDI, se pueden cambiar las perspectivas para modificar fácilmente lo siguiente:

1. Diseño de trabajos ETL y transformaciones.
2. Programación de trabajos y transformaciones.

Desde el cliente de PDI, se pueden cambiar las perspectivas, utilizando el ícono de Perspectiva, en la barra de herramientas.

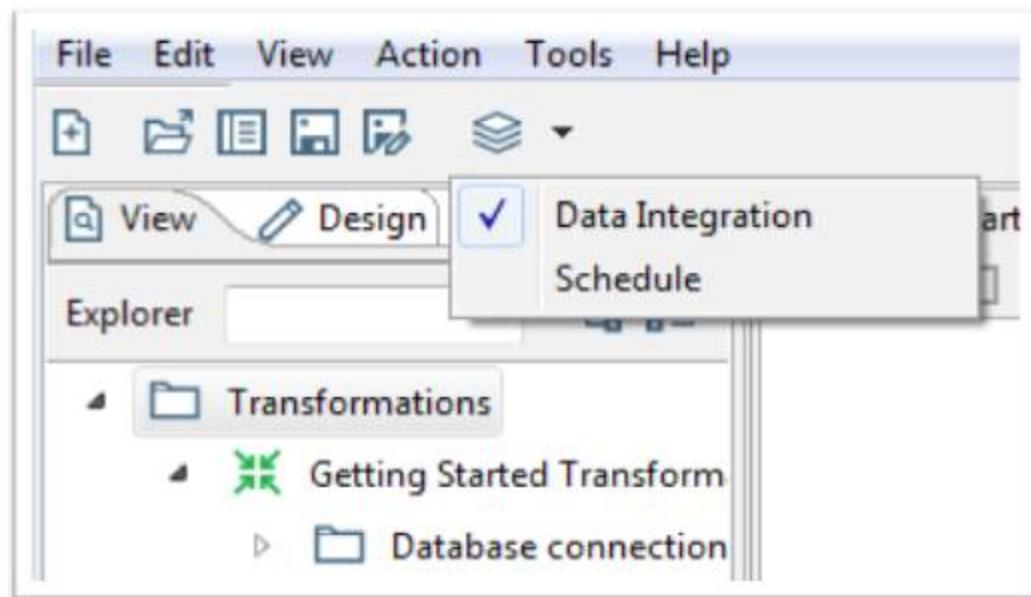


Figura 16. Opciones de perspectiva.
Fuente: documentación oficial de Pentaho.

CONTINUAR

8.4.2. Perspectivas de diseño *data integration*

La perspectiva de integración de datos permite crear transformaciones, trabajos e inspeccionar los datos, posibilitando actualizaciones iterativas a medida que se trabaja.

Consta de los siguientes elementos:

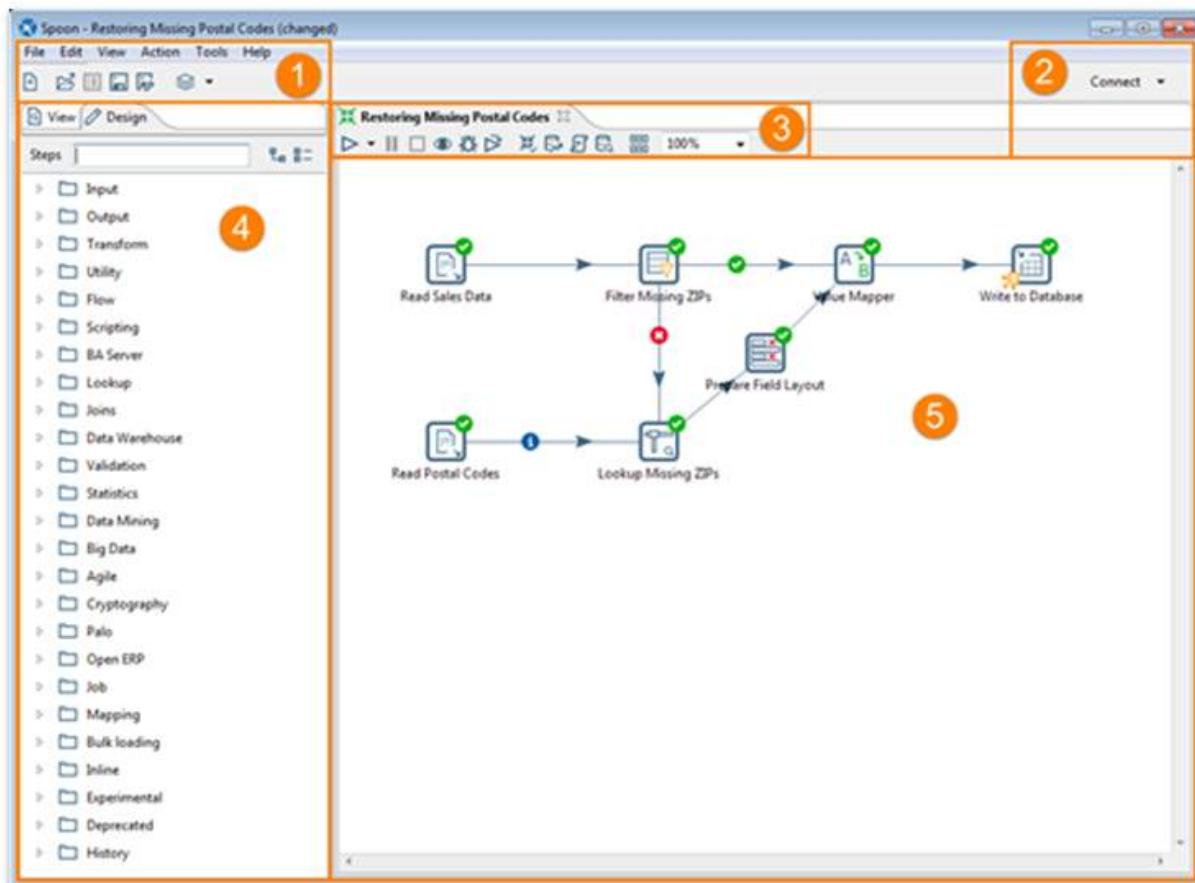


Figura 17. Componentes e iconos de la perspectiva de diseño data integration.

Fuente: adaptado de documentación oficial de Pentaho.

Componente	Nombre	Descripción
1	Barra de herramientas	Acceso con un solo clic a acciones comunes, como crear un nuevo archivo,

		abrir documentos existentes, guardar y guardar como.
2	Menú de conexiones	Crear y conectar a repositorios para almacenar centralmente trabajos y transformaciones de ETL.
3	Barra de herramientas secundaria	Proporciona accesos directos para acceder rápidamente a acciones comunes específicas de la transformación o el trabajo, como ejecutar, vista previa y depurar.
4	Pestañas de Diseño y Vista	<p>La pestaña Diseño proporciona una lista organizada de pasos de transformación o entradas de trabajo usadas para crear transformaciones y trabajos. Las transformaciones se crean simplemente arrastrando pasos de transformación de la pestaña Diseño al lienzo y conectándolos con saltos para describir el flujo de datos.</p> <p>La pestaña Vista muestra información para cada trabajo o transformación. Esto incluye información como las conexiones de bases de datos disponibles y qué pasos y saltos se</p>

		utilizan.
		En la imagen, está seleccionada la pestaña <i>Diseño</i> .
5	Espacio de trabajo	Área de diseño principal para generar transformaciones y trabajos que describen las actividades ETL que se desee realizar.

Tabla 1. Descripción de los componentes de la perspectiva de diseño *data integration*.

Fuente: elaboración propia a partir de documentación oficial de Pentaho.

Icono	Descripción
	Crear una nueva transformación o trabajo.
	Abrir transformación/trabajo del archivo, si no está conectado a un repositorio o desde el repositorio, si está conectado a uno.
	Explorar el repositorio.
	Guardar la transformación/trabajo en un archivo o en el repositorio.
	Guardar la transformación/trabajo con un nombre o nombre de archivo diferente (guardar como).
	Cambiar entre las diferentes perspectivas: <ul style="list-style-type: none"> Integración de datos — Crear transformaciones ETL y trabajos. Programar — Administrar actividades programadas de ETL en el servidor Pentaho.
	Ejecutar transformación/trabajo y establecer opciones de ejecución; ejecuta la transformación actual desde el archivo o repositorio XML.
	Pausar la transformación.
	Detener la transformación.
	Vista previa de la transformación: ejecuta la transformación actual de la memoria. Se puede obtener una vista previa de las filas producidas por los pasos seleccionados.
	Ejecutar la transformación en modo de depuración; permite solucionar los errores de ejecución.
	Reproducir el procesamiento de una transformación.
	Verificar transformación.
	Ejecutar un análisis de impacto en la base de datos.
	Generar el SQL que se necesita para ejecutar la transformación cargada.
	Ejecutar Database Explorer, que permite obtener una vista previa de los datos, ejecutar consultas SQL, generar DDL y más.
	Mostrar el panel de resultados de la ejecución.
	Bloquear la transformación.

Tabla 2. Descripción de los iconos de la perspectiva de diseño *data integration*.

Fuente: elaboración propia a partir de documentación oficial de Pentaho.

CONTINUAR

8.5. Conceptos básicos en PDI: transformaciones, trabajos y saltos

Pentaho Data Integration utiliza una representación gráfica de *workflow* o flujo de trabajo a modo de bloques de construcción, para crear procesos de transformación para los datos y otras tareas.

Los flujos de trabajo se crean mediante pasos o entradas a medida que se crean transformaciones y/o trabajos. Cada paso o entrada se une a un **salto** que pasa el flujo de datos de un elemento al siguiente.

8.5.1. Transformaciones

Una transformación es una red o conjunto de tareas lógicas llamadas **pasos**. Las transformaciones son esencialmente flujos de datos.

En el siguiente ejemplo, el desarrollador de la base de datos ha creado una transformación que lee un archivo plano, lo filtra, lo ordena y lo carga en una tabla de base de datos relacional.

Supóngase que el desarrollador de la base de datos detecta una condición de error y, en lugar de enviar los datos a un paso *dummy* —que no hace nada—, los datos se vuelven a registrar en una tabla.

La transformación es, en esencia, un gráfico dirigido de un conjunto lógico de configuraciones de transformación de datos. Los nombres de archivo de transformación tienen una extensión .KTR.

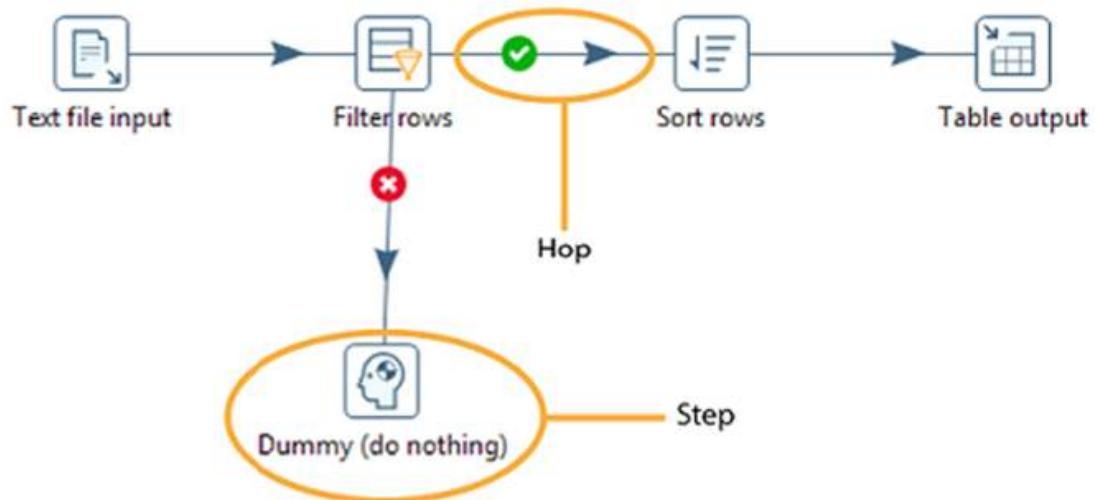


Figura 18. Creación de una transformación: pasos y saltos.

Fuente: documentación oficial de Pentaho.

Los dos componentes principales asociados a las transformaciones son pasos y saltos:

Pasos

Los pasos son los componentes básicos de una transformación, por ejemplo, una entrada de archivo de texto o una salida de tabla.

Hay más de 140 pasos disponibles en PDI de Pentaho y se agrupan según la función; por ejemplo, entrada, salida, secuencias de comandos, etc.

Cada paso de una transformación está diseñado para realizar una tarea específica, como leer datos de un archivo plano, filtrar filas e iniciar sesión en una base de datos, como se muestra en la figura 18. Los pasos se pueden configurar para realizar las tareas que se necesitan.

Todos los pasos en una transformación se inician y se ejecutan en paralelo, por lo que la secuencia de inicialización no es predecible. Por eso no se puede, por ejemplo, establecer una variable en un primer paso e intentar usar esa variable en un paso posterior.

Saltos

Los saltos son vías de datos que conectan los pasos entre sí y permiten que los metadatos del esquema transiten de un paso a otro.

En la figura 18 parece que está teniendo lugar una ejecución secuencial, sin embargo, no es así. Los saltos determinan el flujo de datos a través de los pasos, no necesariamente la secuencia en la que se ejecutan.

Cuando se ejecuta una transformación, cada paso se inicia en su propio hilo y empuja y pasa datos.

Se pueden conectar pasos juntos, editar pasos y abrir el menú contextual paso, haciendo clic para editar un paso:

Figura19. Submenú de acciones sobre el paso.

Fuente: documentación oficial de Pentaho.



Un paso puede tener muchas conexiones: algunos se unen a otros pasos, otros sirven como entrada o salida para otro paso. La secuencia de datos fluye a través de los pasos a los diversos pasos en una transformación.

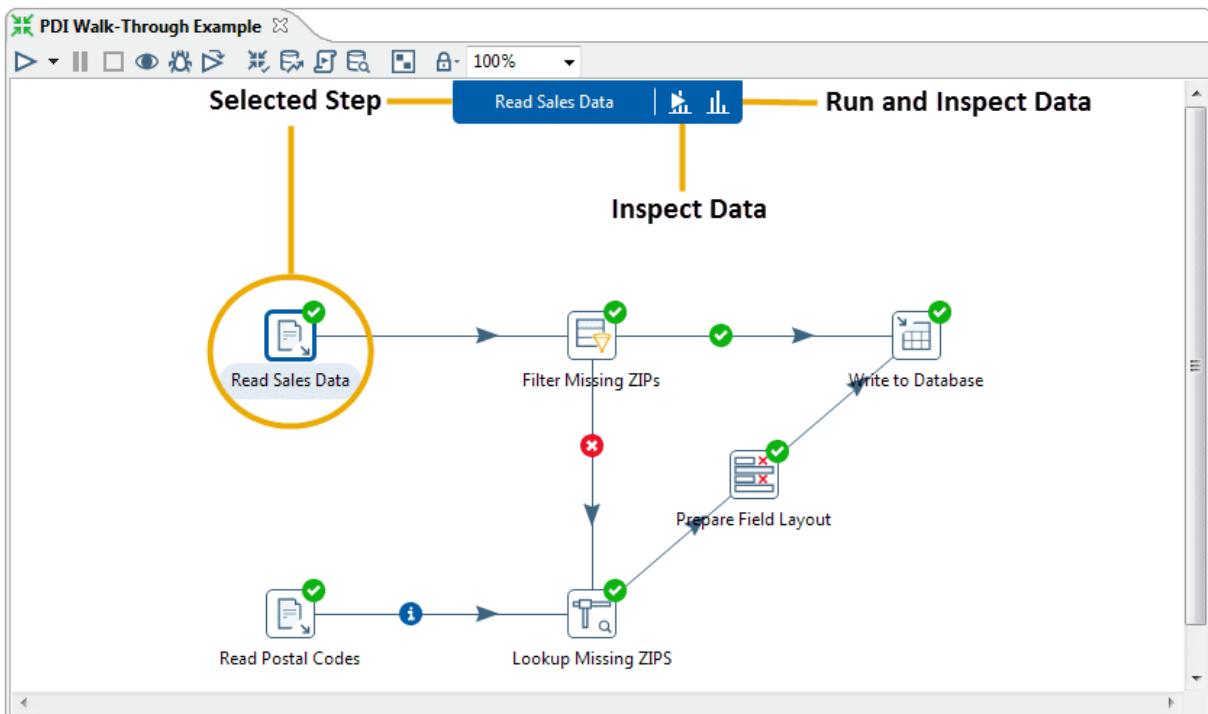
Por su parte, los saltos se representan en Spoon como flechas. Los saltos permiten que los datos transiten de un paso a otro y también determinan la dirección y el flujo de datos a través de los pasos. Si un paso envía salidas a más de un paso, los datos pueden copiarse en cada paso o distribuirse entre ellos.

Inspección de datos

Se pueden inspeccionar los datos para un paso a través de la barra de inspección desplegable. La barra aparece al hacer clic en el paso, como se muestra en la figura 20:

Figura 20. Paso: barra de inspección desplegable.

Fuente: documentación oficial de Pentaho.



Se pueden explorar los datos en cualquier momento, utilizando la barra de inspección, a través de las siguientes opciones:

INSPECCIONAR DATOS

EJECUTAR E INSPECCIONAR DATOS

Permite inspeccionar el flujo de datos de un paso, una vez que se ha ejecutado la transformación.

Esta opción no está disponible hasta que se ejecute la transformación.

INSPECCIONAR DATOS

EJECUTAR E INSPECCIONAR DATOS

Ejecuta la transformación y luego permite inspeccionar los datos de un paso.

8.5.2. Trabajos

Los trabajos son modelos similares al flujo de trabajo para coordinar los recursos, la ejecución y las dependencias de las actividades de ETL. Los trabajos agregan piezas individuales de funcionalidad para implementar un proceso completo.

Los ejemplos de tareas comunes realizadas en un trabajo incluyen obtener archivos FTP, verificar condiciones como la existencia de una tabla de base de datos de destino necesaria, ejecutar una transformación que llene esa tabla y enviar por correo electrónico un registro de error, si falla una transformación. El resultado final del trabajo podría ser una actualización del *data warehouse* durante la noche, por ejemplo.

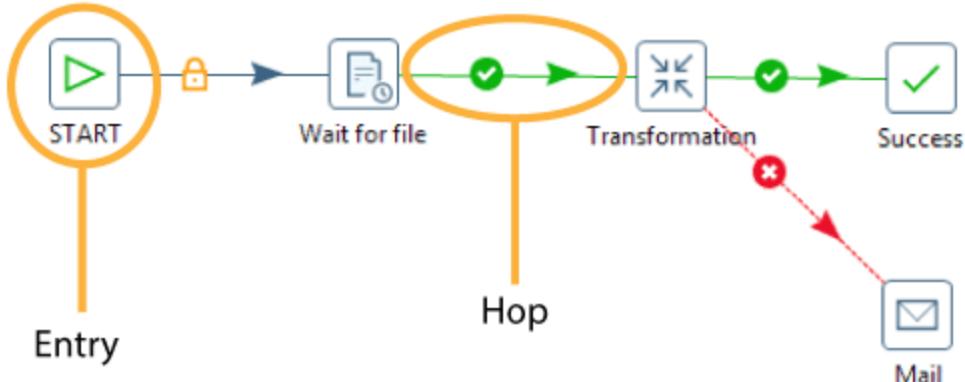


Figura 21. Ejemplos de trabajos..

Fuente: documentación oficial de Pentaho.

Los trabajos se componen de saltos entre trabajos, otros trabajos, entradas y configuraciones de trabajo.

Entradas

Las entradas de trabajo son las piezas configuradas individualmente, tal y como se muestra en la figura 21; son los principales bloques de construcción de un trabajo. En las transformaciones de datos, estas piezas individuales se denominan pasos.

Las entradas de trabajo pueden proporcionar una amplia gama de funciones que van desde la ejecución de transformaciones hasta la obtención de archivos desde un servidor web.

Una sola entrada de trabajo se puede colocar varias veces en el lienzo; por ejemplo, se puede tomar una sola entrada de trabajo, como una ejecución de transformación, y colocarla en el lienzo varias veces utilizando diferentes configuraciones.

Las configuraciones de trabajo son las opciones que controlan el comportamiento de un trabajo y el método de registrar las acciones de un trabajo. Los nombres de archivo de trabajo tienen una extensión

Saltos

Además del orden de ejecución, un salto también especifica la condición en la que se ejecutará la siguiente entrada de trabajo. Se puede especificar el modo evaluación haciendo clic derecho en el salto de trabajo.

Un salto de trabajo es solo un flujo de control. Los saltos se vinculan a las entradas de trabajo y, en función de los resultados de la entrada de trabajo anterior, determinan qué sucederá a continuación.

Opción	Descripción
Incondicional	Especifica que la siguiente entrada de trabajo se ejecutará independientemente del resultado de la entrada de trabajo de origen.
Seguir cuando el resultado es verdadero	Especifica que la siguiente entrada de trabajo se ejecutará solo cuando el resultado de la entrada de trabajo de origen sea verdadero; esto significa una ejecución exitosa, como archivo encontrado, tabla encontrada, sin error, etc.
Seguir cuando el resultado es falso	Especifica que la siguiente entrada de trabajo solo se ejecutará cuando el

resultado de la entrada de trabajo de origen sea falso, lo que significa que la ejecución no fue exitosa, el archivo no se encontró, la tabla no se encontró, se produjo un error, etc.

Tabla 3. Opciones de ejecución de entrada de trabajo.

Fuente: adaptado de documentación oficial de Pentaho.

Los saltos se comportan de manera diferente si son usados en un trabajo o si se usan en una transformación.

Trabajando con saltos

Un salto conecta un paso de transformación o entrada de trabajo con otro. La dirección del flujo de datos se indica mediante una flecha. Para crear el salto, hay que hacer clic en el paso de origen, luego presionar la tecla <MAYÚS> y trazar una línea hacia el paso de destino. Alternativamente, se pueden dibujar o trazar saltos pasando el ratón sobre un paso hasta que aparezca el menú emergente. Hay que arrastrar el icono del inicio de salto desde el paso de origen al paso de destino.

Figura 22. Opción de salir, dentro del submenú de acciones sobre el paso.

Fuente: documentación oficial de Pentaho.



Los métodos adicionales para crear saltos incluyen lo siguiente:

- Hacer clic en el paso de origen, mantener presionado el botón central del ratón y arrastrar el salto al paso de destino.
- Usar `<CTRL + clic izquierdo>` para seleccionar dos pasos, hacer clic con el botón derecho en el paso y elegir *Nuevo salto*.

Para dividir un salto, hay que insertar un nuevo paso en el salto entre dos pasos, arrastrando el paso sobre un salto. Hay que confirmar que se desea dividir el salto. Esta función solo es posible con pasos que aún no se han conectado a otro paso.

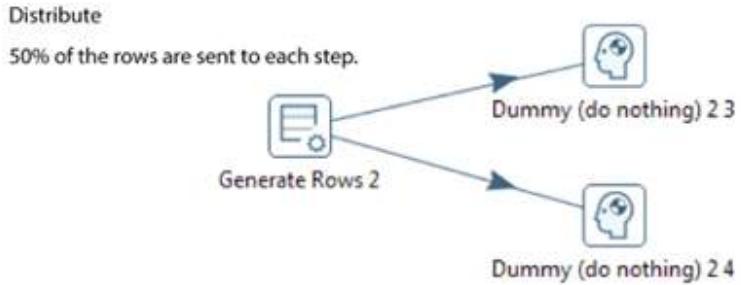
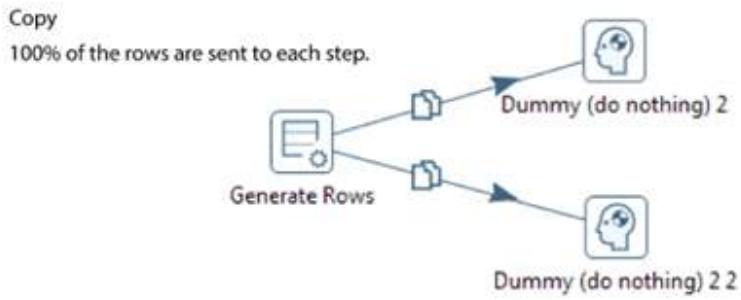
Los bucles no están permitidos en las transformaciones porque Spoon depende en gran medida de los pasos previos para determinar los valores de campo que transitan de un paso a otro. Permitir bucles en transformaciones puede dar como resultado bucles infinitos y otros problemas. Los bucles se permiten en trabajos porque Spoon ejecuta las entradas de trabajo secuencialmente, sin embargo, hay que asegurarse de no crear bucles infinitos.

Mezclar filas que tienen un diseño diferente no está permitido en una transformación, por ejemplo, si se tienen dos pasos de entrada de tabla que usan un número variable de campos. Mezclar diseños de filas hace que los pasos fallen porque los campos no se pueden encontrar donde se espera o el tipo de datos cambia inesperadamente. PDI muestra advertencias en el momento del diseño, si un paso recibe diseños mixtos.

Se puede especificar si los datos pueden ser copiados, distribuidos o *load balanced* entre saltos múltiples, dejando un paso. Para ello, hay que seleccionar el paso, hacer clic derecho y elegir la opción de tipo de movimiento de datos *Data Movement*.

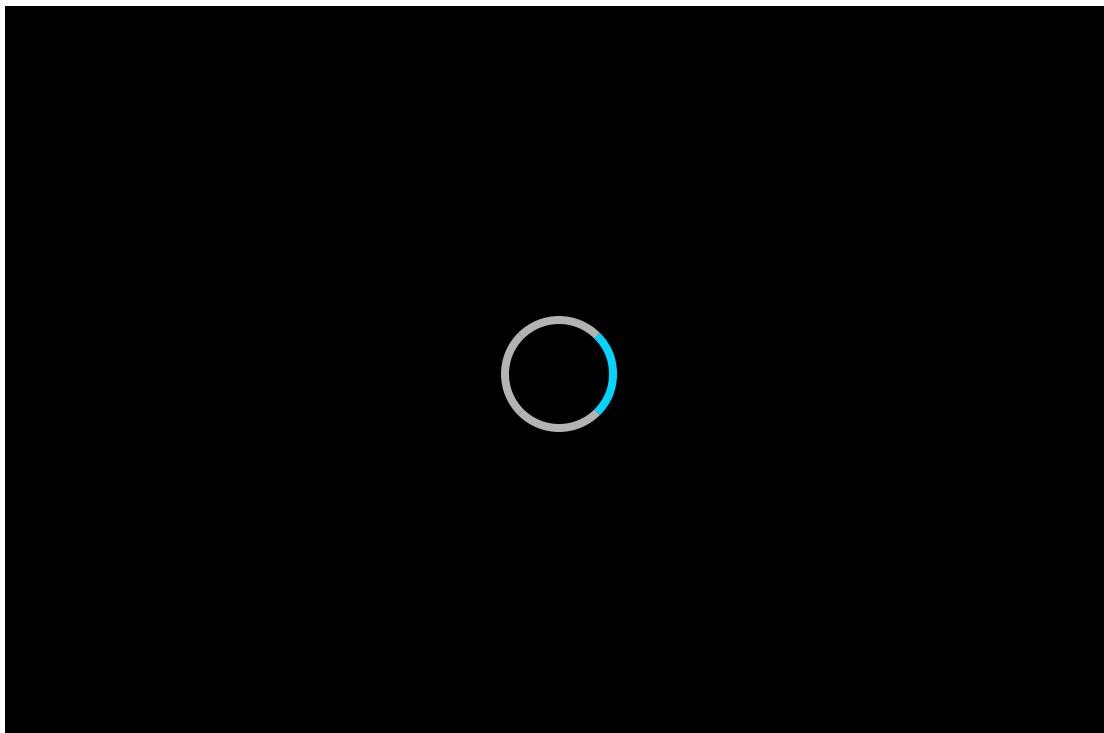
Figura 23. Tipos de movimiento de datos PDI.

Fuente: Official Pentaho Documentation.

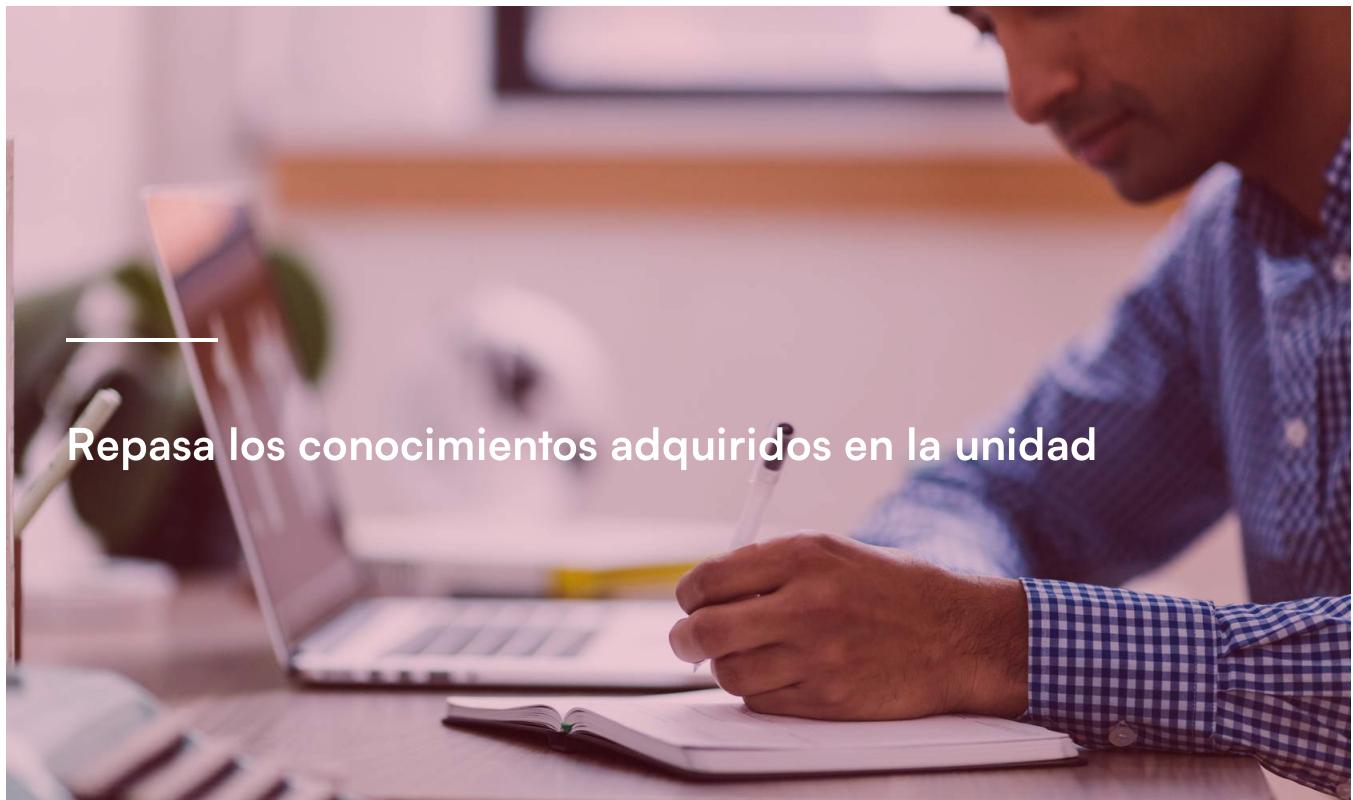


Un salto puede habilitarse o deshabilitarse —para fines de prueba, por ejemplo—. Para ello, hay que hacer clic derecho en el salto para mostrar el menú de opciones.

Se recomienda ver el *webinar* que ayudará a la conceptualización de la herramienta, y en el desarrollo del caso práctico. También se puede consultar el ejemplo definido en los anexos.



IX. Resumen



Repasa los conocimientos adquiridos en la unidad

En esta unidad se ha estudiado el proceso ETL y su importancia dentro de las soluciones de inteligencia de negocio.

Es evidente que una de las razones por las que el proceso ETL es uno de los más importantes, y clave en una solución de inteligencia de negocio, es que es el responsable directo del proceso de creación y mantenimiento del *data warehouse*, y el que se ocupa del procesamiento y mejora de la calidad del dato. Por ello cabe recalcar que en todos los proyectos de inteligencia de negocio del 70 al 80 % del tiempo se dedica a este tipo de tareas.

Gran parte de los beneficios de los sistemas de inteligencia de negocios se obtienen gracias a estos procesos ETL de integración de datos, que limpian, consolidan, enriquecen y validan la calidad de los datos que se llevan al *data warehouse* o al *data mart*.

Los procesos ETL primero extraen los datos a un repositorio común de datos, en formato original, aunque al estar en un repositorio común se facilita la siguiente tarea de transformación. En esta fase se hace un primer chequeo de la calidad de los datos.

La transformación de los datos es el proceso de ETL en que se aporta valor a los datos, enriqueciéndolos, consolidándolos en lo que se refiere a unidades de medida, mapeando maestros comunes de diferentes orígenes y añadiendo valor a los datos del data warehouse que se cargará en la última fase.

Cabe recalcar la importancia de establecer unas buenas prácticas antes de empezar a desarrollar o definir cualquier proceso de ETL. Estas prácticas acercarán al sistema a ser una necesidad fundamental para la empresa, generando procesos y sistemas escalables y fáciles de mantener.

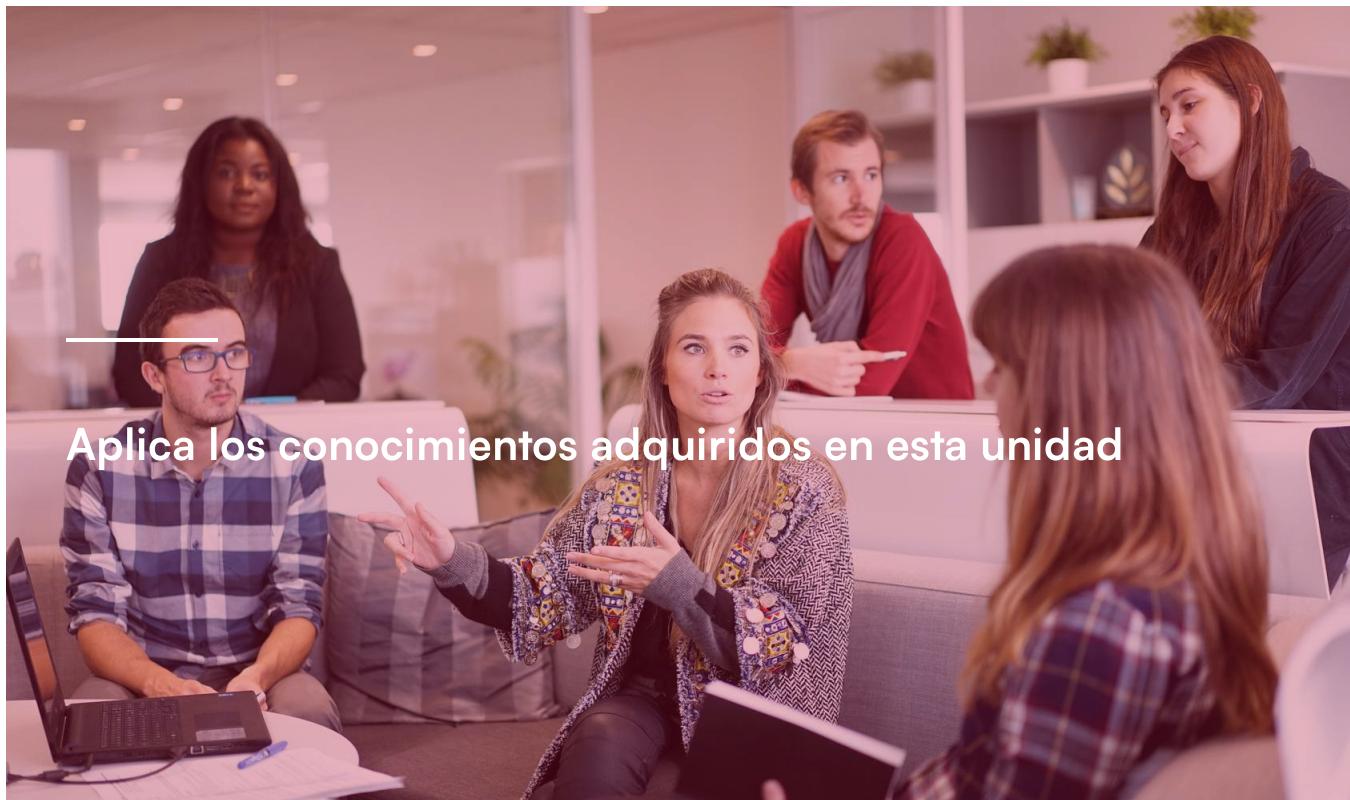
Sin embargo, también es importante identificar que las herramientas ETL no se tienen por qué usar solo en entornos de construcción y mantenimiento de un *data warehouse*, sino que son útiles para multitud de propósitos, como, por ejemplo, en los siguientes:

- Tareas de bases de datos: también se utilizan para consolidar, migrar y sincronizar bases de datos operativas.
- Migración de datos entre diferentes aplicaciones por cambios de versión o cambio de aplicativos.
- Sincronización entre diferentes sistemas operacionales (por ejemplo, el entorno ERP y la web de ventas).

- Consolidación de datos: sistemas con grandes volúmenes de datos que son consolidados en sistemas paralelos para mantener históricos o para procesos de borrado en los sistemas originales.
- Pasarela de datos con sistemas externos: envío de información a clientes, proveedores. Recepción, proceso e integración de la información recibida.
- Otros cometidos: actualización de usuarios a sistemas paralelos, preparación de procesos masivos (*mailings, newsletter*), etc.

Para entender el proceso de forma práctica, se ha introducido la herramienta Pentaho Data Integration. Se recomienda ver el *webinar* asociado a esta unidad y el ejemplo anexo antes de realizar el caso práctico.

X. Caso práctico con solución



Aplica los conocimientos adquiridos en esta unidad

ENUNCIADO

Una compañía del sector financiero realiza el *scoring* de numerosos clientes y entidades, a partir de los datos e información que recibe de numerosas entidades y fuentes de datos externas.

A diario, esta compañía recibe miles de ficheros que son depositados por fuentes externas en repositorios para ser procesados a continuación y cargarlos en el *data warehouse* corporativo de la

compañía.

Sin embargo, la compañía pretende diseñar un proceso de *data quality* que permita descartar de forma automática los ficheros fuentes que tengan algún error y cargar únicamente la información que sea correcta.

SE PIDE

Realizar el diseño e implementación de una solución sencilla de proceso de *data quality* que permita verificar, en un proceso previo a la carga en el DW, que la información que se carga cumple unas reglas de formato y negocio definidas por la compañía. En concreto, se solicita:

- Proceso automatizado de *data quality*.
- Lectura periódica de fichero en formato Excel, .XLS.
- Notificación de error de lectura de fichero o no existencia de fichero en repositorio de destino.
- Aplicación de cuatro reglas de negocio definidas por la compañía.
- Cargar únicamente la información válida en el DW corporativo.
- Enviar notificación a la compañía origen del fichero, para reenvío de información errónea.

Para ello, se facilita una muestra ejemplo de entrada de datos, "Fichero_Ejemplo_Entrada.xlsx", que se puede descargar en [este siguiente enlace](#).

Así como la definición de las reglas que deben cumplirse:

- Regla 1:
 - Que CL_RAMO_MOD contenga alguno de estos valores: "14A", "14D", "14E", "14H", "14L", "14M", "14P", "14R", "14T", "14U", "14Z".

- Regla 2:
 - Que FECHA_EFECTO sea mayor que FECHA_CALCULO.
- Regla 3:
 - En los casos en los que el campo AGE_AT_ENTRY < 14 -> hay que realizar un recálculo del campo AGE_AT_ENTRY == ROUND((FECHA_EFECTO - FECHA_NACIMIENTO) / 365.25)
- Regla 4:
 - El campo POLIZA debe tener una longitud igual a 9.

ID_SEC	CL_RAMO_MOD	FECHA_EFECTO	FECHA_CALCULO	AGE_AT_ENTRY	FECHA_NACIMIENTO	POLIZA
1	14A	17/05/2016	15/05/2016	12	17/05/2010 xxxxxxxxxx	
2	14D	10/05/2016	12/05/2016	15	10/05/2010 xxxxxxxxxx	
3	14E	08/05/2016	01/05/2016	12	08/05/2010 xxxxxxxxxx	
4	14F	01/05/2016	11/05/2016	10	01/05/2010 xxxxxxxxxx	
5	14H	17/04/2016	17/05/2016	14	17/04/2010 xxxxxxxxxx	
6	14L	23/03/2016	03/03/2016	7	23/03/2010 xxxxxxxxxx	
7	14M	15/02/2016	10/02/2016	8	15/02/2010 xxxxxxxxxx	
8	14N	25/02/2016	25/02/2015	11	25/02/2010 xxxxxxxxxx	
9	14P	11/02/2016	11/03/2016	30	11/02/2010 xxxxxxxxxx	
10	14Q	01/02/2016	01/01/2016	11	01/02/2010 xxxxxxxxxx	
11	14U	17/05/2015	07/05/2015	13	17/05/2010 xxxxxxxxxx	
12	14Y	17/01/2016	27/01/2016	10	17/01/2010 xxxxxxxxxx	



Fichero_Ejemplo_Entrada.xlsx

17.9 KB



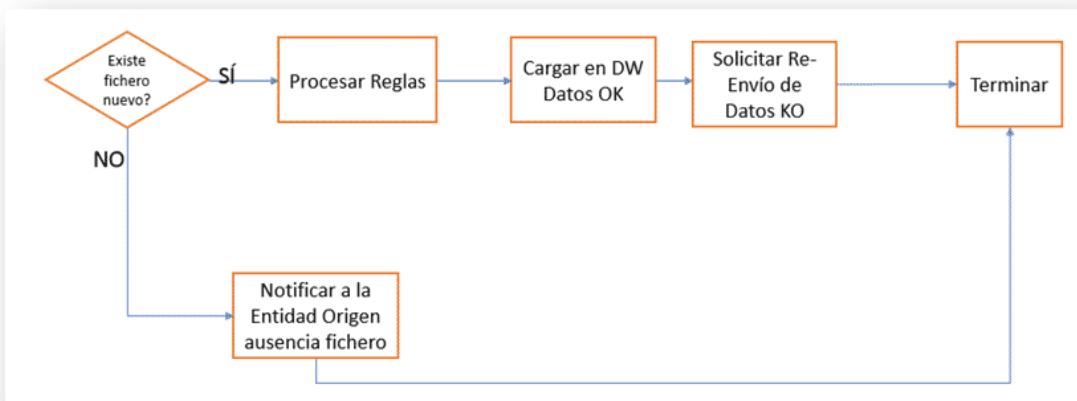
VER SOLUCIÓN

SOLUCIÓN

Este caso se va a afrontar con la herramienta de procesamiento y ETL de Pentaho, Pentaho Data Integration.

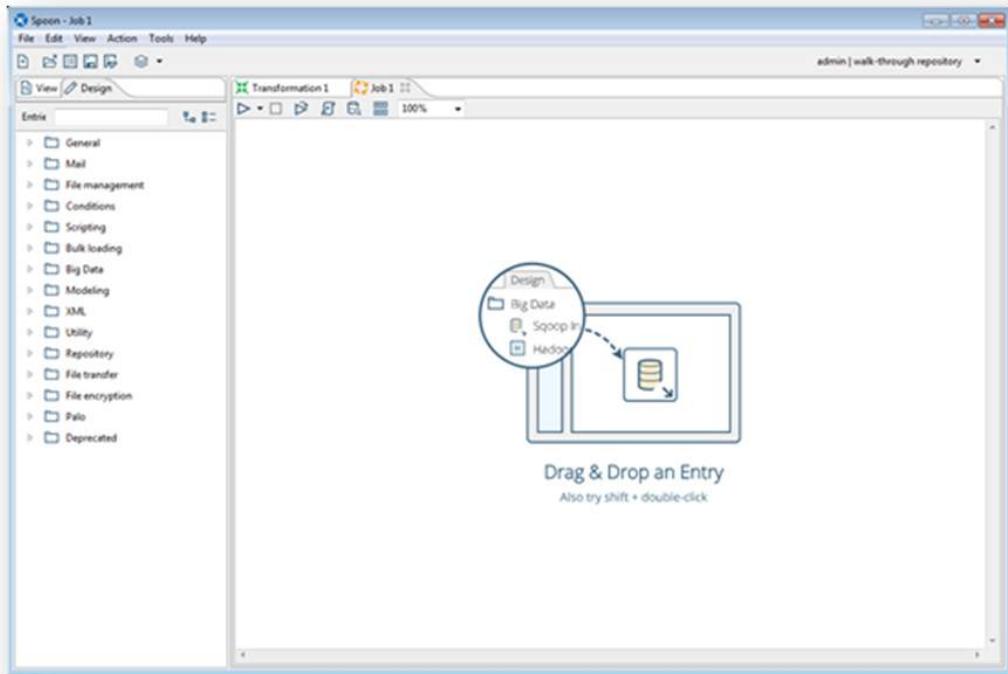
Y, para ello, primero se va a realizar un diseño *top-down* de la posible solución e implementación, teniendo en cuenta que es una solución recomendada, orientada a que el alumno entienda y asimile algunos conceptos básicos.

Se comienza con el diseño a nivel conceptual, mediante diagramas de flujo, de cómo se van a procesar los datos y qué acciones hay que incluir en el proceso de información:

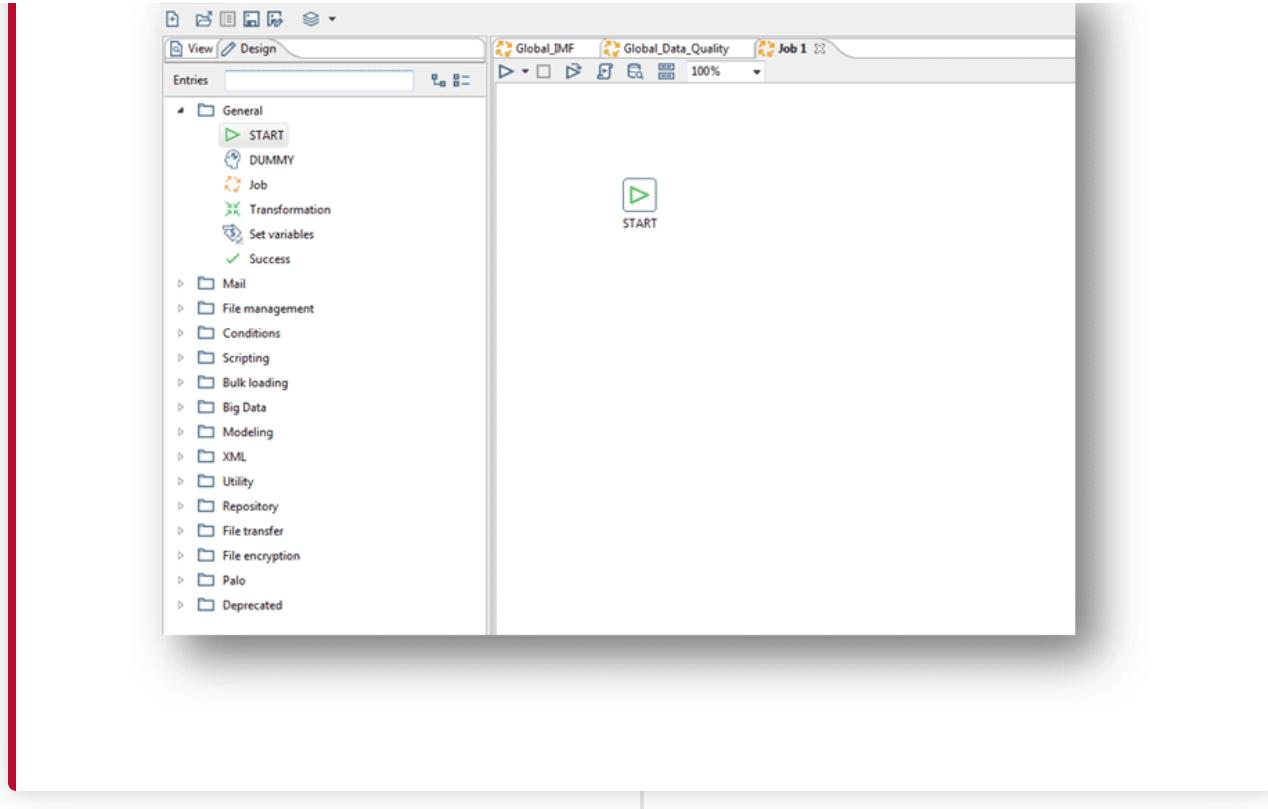


Para implementar el flujo anterior, se crea un nuevo “trabajo” en PDI:

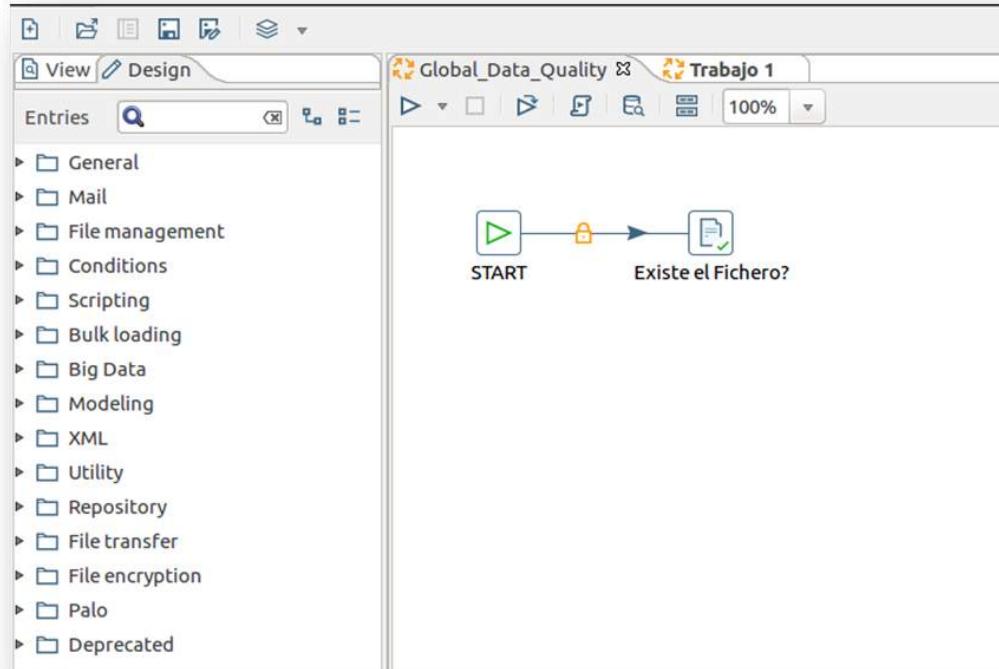
1. Ir a Archivo > Nuevo > Trabajo.



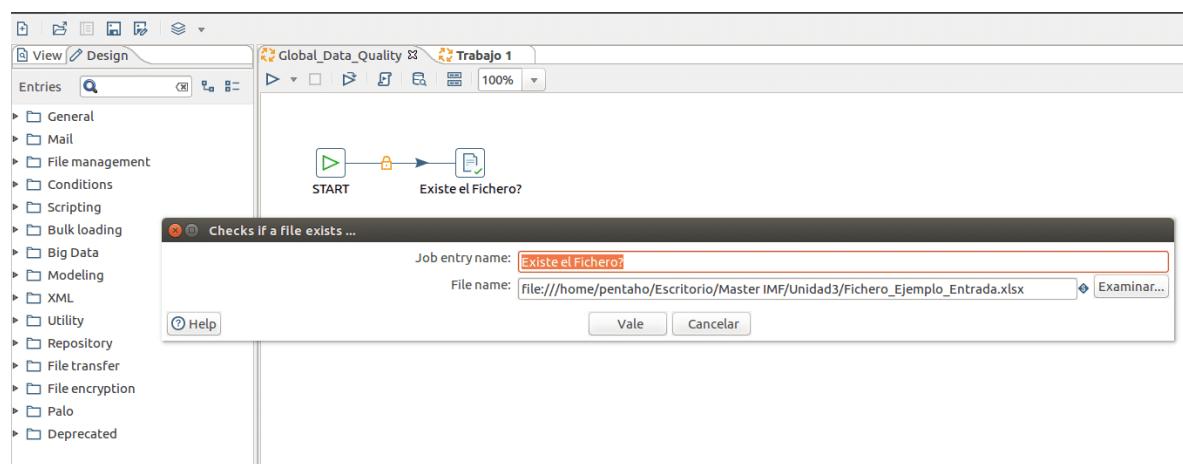
2. Expandir la carpeta "General" y arrastrar una entrada de trabajo de inicio al espacio de trabajo gráfico.
La entrada del trabajo de inicio define dónde comenzará la ejecución.
3. Seleccionar y arrastrar una entrada de inicio o "Start", que definirá el inicio del trabajo.



4. Se incluye una entrada para comprobar si el fichero existe en una ubicación específica. Esta entrada falla si no encuentra el nombre exacto del fichero.

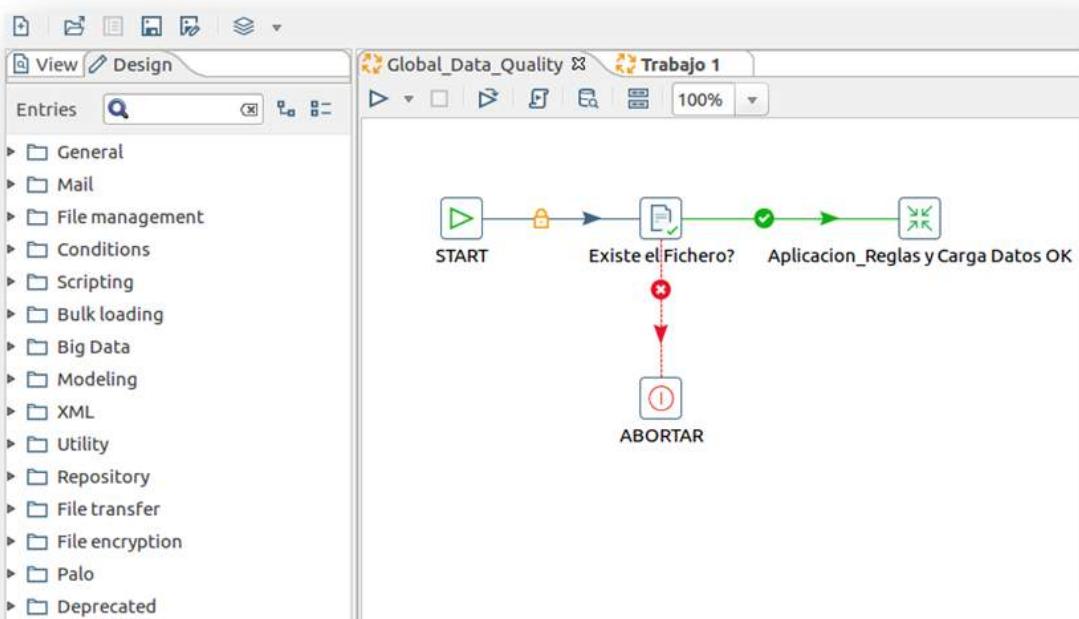


5. Se configura la entrada.



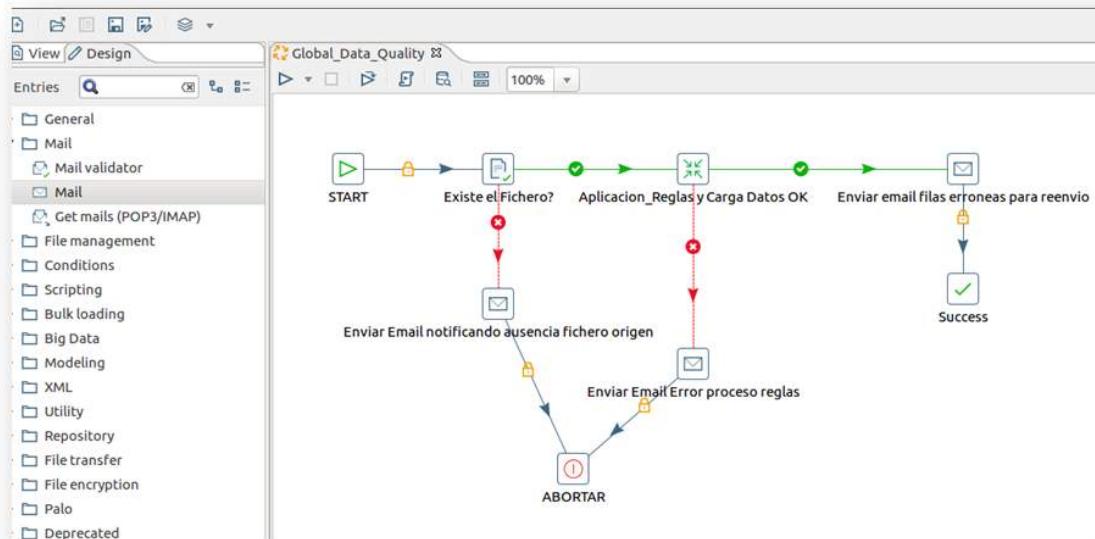
6. Se incluye una entrada a una transformación para el caso de que sí exista el fichero de datos de entrada. Esta entrada de transformación se diseñará más adelante.

Del mismo modo, se incluye una llamada a *Abort* para abortar el proceso, en caso de que el fichero de entrada no esté disponible.



7. El siguiente paso consiste en incluir las notificaciones que necesite el proceso. A continuación, se identifican tres:

1. Enviar un correo electrónico de notificación a la entidad que envía el fichero, cuando este no se encuentra en la carpeta prevista.
2. Enviar un correo electrónico a soporte, cuando el proceso de aplicación de las reglas no sea correcto o falle por cualquier motivo.
3. Tras el procesado de las reglas, filas de datos correctas son almacenadas en el DW, pero las que son erróneas se envían por correo electrónico a las entidades origen.



En [este enlace](#), se puede descargar el archivo de solución “**Global_Data_Quality.kjb**”.

Pero ¿cuál es el diseño e implementación de la transformación que aplica las reglas? A continuación, se muestra una posible solución e implementación del proceso.

Para realizar el diseño, hay que tener claro cuál es el objetivo que se debe cumplir y plantear una estrategia de diseño. En este caso, se parte de un fichero de entrada, tal y como se muestra a continuación:

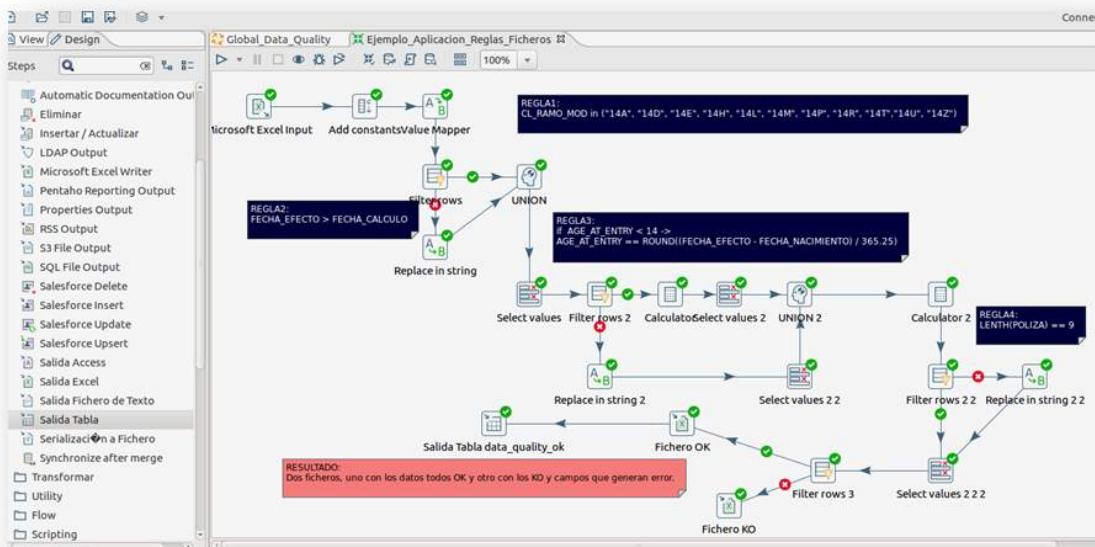
ID_SEC	CL_RAMO_MOD	FECHA_EFECTO	FECHA_CALCULO	AGE_AT_ENTRY	FECHA_NACIMIENTO	POLIZA
1 14A		17/05/2016	15/05/2016	12	17/05/2010 xxxxxxxxx	
2 14D		10/05/2016	12/05/2016	15	10/05/2010 xxxxxxxxx	
3 14E		08/05/2016	01/05/2016	12	08/05/2010 xxxxxxxxx	
4 14F		01/05/2016	11/05/2016	10	01/05/2010 xxxxxxxxx	
5 14H		17/04/2016	17/05/2016	14	17/04/2010 xxxxxxxxx	
6 14L		23/03/2016	03/03/2016	7	23/03/2010 xxxxxxxxx	
7 14M		15/02/2016	10/02/2016	8	15/02/2010 xxxxxxxxx	
8 14N		25/02/2016	25/02/2015	11	25/02/2010 xxxxxxxxx	
9 14P		11/02/2016	11/03/2016	30	11/02/2010 xxxxxxxxx	
10 14Q		01/02/2016	01/01/2016	11	01/02/2010 xxxxxxxxx	
11 14U		17/05/2015	07/05/2015	13	17/05/2010 xxxxxxxxx	
12 14Y		17/01/2016	27/01/2016	10	17/01/2010 xxxxxxxxx	

En donde las filas marcadas en rojo no cumplen con alguna de las reglas que hay que aplicar.

La transformación tendrá, pues, como objetivo verificar fila a fila si cumple o no con cada una de las reglas definidas. El resultado final de esta transformación debe ser diferente para cada caso de fila. Es decir, para las filas que cumplen con todas las reglas, la salida debe ser un fichero igual al de entrada, y se debe incluir esta información en una tabla para su procesamiento en el DW.

Las filas que incumplan una o más reglas deben ser separadas y agrupadas para crear un fichero de salida que contenga la información de entrada, así como la o las reglas que ha incumplido, con el propósito de poder reenviarlo al origen para su corrección.

A continuación, se muestra una posible implementación:



La implementación está disponible en la [máquina virtual del módulo](#).

Como se puede apreciar, se han ido aplicando una a una cada una de las reglas, con una implementación

muy sencilla de pasos, y partiendo de la premisa inicial de incluir varios *switches* de control que indican que las reglas se cumplen, con el fin de ir comprobándolo una a una y cambiar solo aquellos casos que incumplen la regla.

Fichero de solución de transformación: "Ejemplo_Aplicacion_Reglas_Ficheros.ktr".



Global_Data_Quality.kjb

17.5 KB



Ejemplo_Aplicacion_Reglas_Ficheros.ktr

63.5 KB



XI. Glosario



El glosario contiene términos destacados para la comprensión de la unidad

ETL

Extracción, transformación y carga. Proceso responsable de la extracción de datos, su limpieza, conformación y localización en el almacén de datos.

Pasos

Son los componentes básicos de una transformación; por ejemplo, una entrada de archivo de texto o una salida de tabla. Cada paso de una transformación está diseñado para realizar una tarea específica, como leer datos de un archivo plano, filtrar filas e iniciar sesión en una base de datos.

PDI

Pentaho Data Integration. Set de herramientas que permite diseñar ETL, mediante transformaciones y trabajos, que pueden ser ejecutados por las herramientas Spoon, Pan y Kitchen.

Saltos

Vías de datos que conectan los pasos entre sí y permiten que los metadatos del esquema transiten de un paso a otro. Los saltos determinan el flujo de datos a través de los pasos, no necesariamente por la secuencia en la que se ejecutan.

Spoon

Interfaz gráfica para diseño de trasformaciones y trabajos ETL.

Trabajos

Son modelos similares al flujo de trabajo para coordinar los recursos, la ejecución y las dependencias de las actividades de ETL.

Transformación

Es una red de tareas lógicas llamadas pasos. Las transformaciones son esencialmente flujos de datos. La transformación es, en esencia, un gráfico dirigido de un conjunto lógico de configuraciones de transformación de datos. Los nombres de archivo de transformación tienen una extensión .KTR.

XII. Anexos

Anexo I. Acceso a Pentaho Data Integration

Descarga el documento a continuación:



Anexo_I.pdf

301.1 KB



Anexo II. Ejemplos

Descarga el documento a continuación:



Anexo_II.zip

2.7 MB

