UNIVERSIDAD CARLOS III

DATA WAREHOUSE: MARCO DE CALIDAD.

PROYECTO FIN DE CARRERA INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN

AUTOR: PABLO MARTÍN GUTIÉRREZ
TUTOR: ANTONIO GARCÍA CARMONA

TRIBUNAL

Presidenta: Pilar Arantzazu Herráez López.

Vocal: Ana Isabel González-Tablas Ferreres.

Secretario: Miguel Ángel Ramos González.

AGRADECIMIENTOS

Quiero agradecer a toda mi familia por haber creído en mí para la realización del proyecto final de carrera a pesar de las circunstancias con las que lo he tenido que hacerlo y que siempre estuvieran seguros de que iba a conseguirlo. Especialmente a mis padres (José Ángel y Delfi), a mis hermanos (David y Laura) y mi tía (Reyes).

Quiero agradecer especialmente a mi tutor Antonio García Carmona que confiara en mí para la realización del proyecto a pesar de que me encontrara en la distancia y activo laboralmente.

Por último quiero agradecer a mis amigos (que junto a mi familia) me han motivado una y otra vez para que no cejara en el empeño de la realización del proyecto.

Gracias a todos,

Pablo.

RESUMEN

"La memoria propone una serie de guías de actuación para abordar con garantías de éxito el desarrollo de sistemas que incluyan un almacén de datos central o Data Warehouse".

Con la memoria nos centramos en el desarrollo teórico del ámbito que rodea al Data Warehouse y sus relaciones con el resto de componentes que forman parte de un sistema de Business Intelligence.

Para poder conseguir las guías de actuación nos basamos en los estándares, normas de calidad o guías de buenas prácticas punteros de la actualidad.

El principal punto productivo de la memoria es que se propone un marco de calidad adaptado a las necesidades que un Data Warehouse necesita disponer para que pueda ser desarrollado con las mayores garantías de éxito posibles.

Se ha dividido la estructura del documento en **cuatro bloques**.

1. Con el **primer bloque** del texto hacemos un resumen del **conocimiento** que existe sobre soluciones de **Business Intelligence** centrándonos en primera instancia en qué son para llegar al detalle de su componente más crítico, el **Data Warehouse**.
   Se hace especial énfasis en conocer qué es un Data Warehouse, cómo se crea y mantiene y cómo puede explotarse para saber cómo adaptar los puntos fuertes de otros marcos de calidad para nuestros objetivos. Por ello, seguidamente pasamos a explicar el **contexto que rodea a los Data Warehouse** para saber la forma en la que se desarrollan y encajan sus componentes.
   Veremos un repaso histórico de cómo han ido evolucionando las soluciones que incorporan Data Warehouse para conocer a partir de ahí las metodologías más importantes que se han desarrollado.
   Las metodologías más importantes giran en torno a las filosofías de Kimball e Inmon cuyos enfoques son totalmente opuestos, ya que la de Kimball (o Bottom-Up) se centra en el detalle y construye la solución desde lo específico a lo genérico y la de Inmon (Top-Down) parte de construir la solución desde lo genérico para a partir de ahí propagar la solución detallada.

2. Dado que nos centramos en crear un marco de calidad, con el **segundo bloque** de la memoria hacemos un repaso de los **estándares, normas de calidad y guías de buenas prácticas** que podremos aplicar a través de un estudio sobre el estado del arte actual e identificando los puntos más adecuados para nuestras guidelines.
   No solo nos centraremos en la calidad de los datos sino que también haremos un estudio de la calidad de los procesos, desde la implementación del propio DW hasta su

explotación, gestión del proyecto de desarrollo etc. Por ello el segundo bloque tiene **dos divisiones** claras y diferenciadas, **la centrada en la gestión y los procesos y la centrada en el dato y sus características**.

3. Con el **tercer bloque** (y objetivo de la memoria) podemos ver el resultado depurado del resto de la memoria ya que **obtenemos las directrices** que nos sirven de punto de partida **para producir DW de calidad** que sean fiables para usar en soluciones de BI.

4. No se quería formar un documento puramente teórico, por lo que en la **cuarta parte** de la memoria nos centramos en el **uso de** dos **herramientas** de la familia Microsoft **que nos permiten acercar las guidelines al proceso** crucial de creación de un DW, el proceso de Extracción, Transformación y Carga de datos **(ETL)** de los sistemas origen en la base de datos destino que forma el propio DW.
   Las herramientas son Biztalk Server 2010 para solucionar problemas de interoperabilidad y asegurar que la tarea de Extracción se realice sin problemas y la segunda será SQL Server Integration Services, que propone muchas facilidades para completar el ETL con garantías.

ABSTRACT

"The document proposes an action guide for dealing with guarantees of success developing systems that include a Data Warehouse."

With memory we focus on the theoretical development of the area surrounding the Data Warehouse and its relations with the other components that are part of a business intelligence system.

For guidelines, we rely on quality standards and best practice guidelines pointers today.

The added value of the document as a framework adapted to the needs of a data warehouse needs to have in order to be developed with greater guarantees of success possible.

Structure has been divided into four blocks of the document.

1) The first block of text is a summary of existing knowledge on Business Intelligence solutions. We focus primarily on what are BI solutions to reach the detail of your most critical component, the Data Warehouse.
   We focus with a special interest in knowing what a Data Warehouse is, how it is created and maintained and how it can be exploited. Paragraph is important to know how to adapt the strengths of other quality frameworks for our purposes. Therefore, we will explain below the context around the data warehouse to know the manner in which they develop and fit components.
   We will see a historical review of how the solutions have evolved to incorporate Data Warehouse to know from there the most important methodologies that have been developed.
   The most important methods revolve around the Inmon and Kimball philosophies whose approaches are opposites, as Kimball (or Bottom-Up) focuses on the detail and build the solution from the specific to the generic and the Inmon (Top-Down) part of building the solution from the generic to propagate from there detailed solution.

2) As we focus on building a quality framework, with the second block of memory we review quality standards and best practice guidelines that can be applied through a study on the state of the art and identifying the most suitable to our guidelines.
   We will focus on both the quality of data and the quality of the process, since the implementation of the DW itself to exploitation, development project management etc. Thus the second block has two clear and distinct divisions, focused on managing the processes and the data-based and property.

3) The third block (and objective memory) shows the result of the rest of the memory and we get the guidelines that serve as a starting point for producing quality DW reliable for use in BI solutions.

4) We want to provide practical content to memory so in the fourth block, we focus on the use of two Microsoft tools that allow us to bring the guidelines to the process of creating a DW, the process of extraction, transformation and loading of data (ETL) from the source system to the target database that is itself DW. Tools are Biztalk Server 2010 to solve interoperability problems and ensure that the task of extraction goes smoothly and the second is SQL Server Integration Services, which offers many facilities to complete the ETL with guarantees.

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

INTRODUCCIÓN Y OBJETIVOS

1. INTRODUCCIÓN

En las últimas décadas se ha ido desarrollando una serie de técnicas que nos permiten extraer conocimiento sobre qué pasa en las empresas, porqué pasa y se realizan aproximaciones cada vez más complejas que permiten definir qué pasará en el negocio. Sobre éste principio se ha creado la llamada Inteligencia de Negocio o **Business Intelligence**.

Actualmente, desde el estallido de la crisis económica mundial, se intenta lidiar con la incertidumbre y los miedos de no poder controlar los procesos de nuestra organización. Es ahora cuando las grandes empresas ponen más empeño en seguir el camino del éxito, para evitar errar en momentos críticos dónde las decisiones estratégicas y tácticas de la organización pueden hacer equilibrar la balanza a favor de la empresa o enviar directamente la misma al abismo.

En la actualidad hay multitud de conceptos relacionados con el Business Intelligence y se han creado miles de proyectos que tratan de explotar los datos que manejan las empresas. Existen diferentes metodologías que nos indican cómo podemos llevar a cabo la construcción de sistemas de información o como deben de ser diseñados para explotar los datos de la empresa y generar información relevante.

Particularmente, he tenido la posibilidad de implementar o ver cómo se ha implementado alguna solución de Business Intelligence durante los últimos años sin que la tan mencionada crisis económica sobrevolara en las mentes de los analistas y diseñadores y he visto cómo se retoman proyectos abandonados o se aplican técnica de reingeniería sobre las soluciones realizadas para apurar hasta el mínimo detalle.

Por lo comentado anteriormente y porque la documentación publicada, sobretodo en castellano, no es tan densa como en otros campos relacionados con las TIC o simplemente no suele, por lo general, estudiarse tanto en detalle por las grandes empresas (incluidas las distintas organizaciones del sector público) ahora **nos encontramos en un momento de auge para las soluciones de business intelligence**.

Por otro lado, me he interesado en el componente que considero la piedra angular de todo proyecto de éxito de business intelligence, que es el Almacén de Datos o **data warehouse** destinado a ser explotado para aportar esa información que pueda desembocar en decisiones exitosas. No obstante, no hay que llevar al engaño y tenemos que saber que quién aporta la inteligencia al negocio es el analista de negocio, nunca el software o la plataforma, pero, ésta tiene un peso muy importante en hacer más productivo (e incluso posible) el trabajo de los analistas.

Tras leer las líneas actuales viene una pregunta a la cabeza muy clara, que nos dice: ¿Realmente, como te ayudan los sistemas de BI a tomar decisiones?

La solución es fácil e identifica dos vías:

A. Recopilando datos sobre los que trabajar.

B. Dotar a los analistas de negocio de herramientas para explotarlos de forma sencilla, útil y rápida.

1. OBJETIVOS

Con la memoria buscamos un objetivo principal sobre el que van a girar el resto de objetivos secundarios o indirectos, se pretende **establecer una guía de actuación basada en un marco de calidad que nos permita crear sistemas Data Warehouse con éxito**. No sólo queremos aislar el marco de calidad al propio DW sino que **se debe integrar a la perfección con el sistema completo de BI** que lo integre.

Con la guía **no nos queremos centrar únicamente en el proceso de desarrollo** como suele ocurrir en la mayoría de la documentación asociada a los DW, sino que vamos más allá a la hora de especificar directrices concretas sobre otras áreas de impacto. Con las áreas de impacto **nos referimos a establecer un marco de calidad completo que cubra todas las fases del ciclo de vida** en las que un proyecto de DW se integra en una solución de BI y por ello se han estudiado las normas y estándares de calidad y guías de buenas prácticas más destacadas de la actualidad para adaptarlas al proceso.

Se trata de tener claras una serie de premisas como son:

- ¿Quién tiene que tomar las principales decisiones sobre el DW?
- ¿Cómo gestionar el proyecto del DW?
- ¿Cómo saber si el DW es de calidad? Nos referimos en ésta caso a Calidad en el dato y saber cómo evaluarla.
- Saber cómo abarcar todos los procesos que forman parte del DW en base a unas pautas o directrices que nos garanticen la calidad de los procesos.
- Etc.

Con la guía tenemos el primer paso hacia saber cómo integrar un DW en una solución de BI y cómo hacerlo con las mayores posibilidades de éxito basándonos en un modelo de calidad

Dado el carácter teórico de la memoria se ha querido acercar a la práctica el proyecto en base a la adaptación de dos herramientas que se asocian con frecuencia a soluciones que incluyen DW pero que no se suele saber cómo encajar, se trata de Biztalk Server 2010 y SQL Server

Integration Services (SSIS). Con ellas especificamos como se pueden usar para el proceso más crítico en el desarrollo de un DW, la fase de consolidación de la información a través del flujo ETL. Por ello, uno de los objetivos secundarios, es aportar un acercamiento práctico de las directrices.

Usaremos Biztalk para solventar los problemas de interoperabilidad y facilitar que la extracción de datos de las fuentes sean propicias para que SSIS se pueda encargar de la transformación de datos para el sistema destino (Base de datos OLAP que forma el DW).

Como objetivo indirecto o académico se aporta el primer bloque de la memoria dónde se recoge un resumen bastante detallado de los DW y su contexto, formando una buena base para lograr desarrollar sistemas de BI que incluyan DW de calidad.

Además se hace un repaso a las normas y guías de buenas prácticas que hacen mayor hincapié en la calidad de los procesos y se depura un estudio sobre el estado del arte de la calidad del dato obteniendo las propiedades que los datos de nuestro DW deben asegurar.

2. FASES DE DESARROLLO

Para el desarrollo del proyecto se ha llevado a cabo una gran labor de ***investigación y documentación***. En un principio se ha estudiado el proceso de BI y cómo influye en el mismo el DW corporativo.

Para poder garantizar que el aseguramiento de la calidad que dota la guía se ha realizado un estudio en gran profundidad de la documentación actual en el ámbito de los DW, no sólo respecto a sus metodologías, sino con ejemplos prácticos dónde se puede ver los beneficios que aportan ciertas acciones que permiten obtener sistemas de calidad.

Además de ver las distintas métricas de calidad en el dato, se ha ideado la guía con el propósito de crear sistemas que sean eficaces en su cometido sin olvidarnos de que haciendo sistemas de calidad el rendimiento a largo plazo será mejor, más eficiente. Por ello se han seguido varios estudios dónde se realiza la construcción de distintos DW y puede verse una serie de ventajas e inconvenientes tras la toma de decisiones.

Dado que se quiere establecer un marco completo de calidad se han estudiado las normas y estándares de calidad de procesos, marcos de calidad y calidad del dato para adaptar en las *guidelines* los apartados más adecuados de ellas para cumplir nuestros objetivos.

Por último se intenta acercar el estudio teórico sobre el mundo del desarrollo de un DW a las personas que tienen que desarrollar uno desde cero, haciendo más fácil la creación de soluciones BI que incluyan DW usando las herramientas de las que este proyecto aporta información sobre cómo crear sistemas útiles sin perder de vista el objetivo de la calidad.

3. MEDIOS EMPLEADOS

La lista de medios empleados se puede ver desglosada en el presupuesto del proyecto, pero a continuación muestra un listado por categoría:

- Hardware:
  - Ordenador personal: HP ProBook 4540s
  - Memoria Flash USB 8GB Sony.
  - Disco duro externo 500GB Iomega.
- Software:
  - Microsoft Office 2007 Professional. Principalmente MS Word, MS Power Point y MS Excel.
  - Navegador Google Chrome.
  - Adobe Acrobat Reader.
- Laboratorio: Prácticas con un entorno de desarrollo bajo las siguientes características instalado en el ordenador personal con VMWARE.
  - Servidor con Windows Server 2008 R2.
  - SQL Server 2008.
  - MS Biztalk 2010.
  - SSIS para SQL Server 2008.
- Documentación:
  - Impresa: Puede consultarse en las referencias a la bibliografía. Textos base:
    - [12] W. H. Inmon: *"Building the Data Warehouse."*. (John Wiley & Sons Inc., 1992, 1ª Ed.).
    - [14] R. Kimball: *"The Data Warehouse Lifecycle Toolkit"*. (John Wiley & Sons Inc., 1998, 1ª Ed.).
    - [16] R. Kimball: *"The Data Warehouse ETL Toolkit (2nd Edition)"*. (John Wiley & Sons Inc., 2008, 2ª Ed.)

- Digital: Puede consultarse en las referencias a la la bibliografía. Textos base:
  - [21] ISO, I.O.o.S., ISO/IEC FDIS 25012 Software engineering - Software Product - Quality Requirements and Evaluation (SQuaRE) - Data quality model. 2008.
  - [80] ISO, I.O.o.S., ISO/IEC 90003:2004 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software, 2004.
  - [81] ISO, I.O.o.S., ISO/IEC 25000:2005 Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 2005.
  - [82] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition.

4. ESTRUCTURA DE LA MEMORIA

Bloque I: Data Warehouse, teoría y fundamentos – Es la teoría sobre el DW, sobre cómo se integra en una solución de BI y muestra el contexto en el que se encuentra explicando sus componentes más determinantes y el resto de componentes de BI que debemos conocer cuando hablamos de DW.

Bloque II: Calidad en los sistemas de BI – Se centra en el estudio de las normas, estándares calidad y guías de mejores prácticas sobre los procesos en los que participa el DW a lo largo de su ciclo de vida y aporta un estudio sobre las propiedades que debe cumplir un dato de calidad.

Bloque III: "*Guidelines* para el desarrollo de un Data Warehouse de calidad en un sistema BI. Calidad en el dato, calidad en el proceso" – Es la guía depurada de directrices que nos indica las bases para desarrollar DW de calidad en base a la información aprendida del desarrollo de los bloques I y II.

Bloque IV: Aplicación práctica: El Proceso ETL utilizando Microsoft Biztalk Server 2010 y Microsoft SQL SERVER 2008. Adaptación del modelo de calidad (En los datos y en el proceso) – Comenta como encajar las herramientas en el flujo EL y explica cómo usar los elementos que nos proporciona Biztalk para desarrollar nuestro cometido. Dado que Biztalk se entiendo como una herramienta de ayuda a la interoperabilidad orientada al administrador el nivel de detalle será menos específico que SSIS que está más orientado al desarrollador que debe conocer la estructura de la información que tiene que proporcionar para la base de datos destino que forma el DW.

BLOQUE I: DATA WAREHOUSE, TEORÍA Y FUNDAMENTOS.

1. CONCEPTOS PREVIOS.

A. DATOS, INFORMACIÓN Y CONOCIMIENTO.

En la actualidad se tiende a usar los términos de datos, información y conocimiento como sinónimos, y si bien en algunos ámbitos pueden llegar a serlo, con las tecnologías de la información se requiere de una serie de matices que eviten llevarnos al error. El utilizar indistintamente los términos puede llevar a una interpretación libre del concepto de conocimiento, que como veremos a continuación es mucho más que información quién a su vez aporta más que datos.

Si bien, los tres conceptos están plenamente relacionados y podríamos localizar a los datos como parte vital del mundo, localizar al conocimiento dentro de agentes (agentes de cualquier tipo, personas, empresas, universidades, etc.) y la información adoptaría un papel de nexo entre ambos, es decir, para que un dato pueda llegar a ser conocimiento debe convertirse en información.

Figura 1. Pirámide de datos, información y conocimiento. [1]

Un **dato** es la representación de hechos, conceptos e instrucciones, mostrada de modo formal, útil para comunicar, interpretar y procesar, por las personas, así como también por mecanismos automatizados. También se puede ver como un conjunto discreto de valores, que no dicen nada sobre el porqué de las cosas, y no están orientados para la acción.

La **información**, en el ámbito de los datos y de las empresas, se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quien debe tomar decisiones, al disminuir su incertidumbre.

Los datos se pueden transforman en información añadiéndoles valor [1]:

- **Contextualizando**: se sabe en qué contexto y para qué propósito se generaron.
- **Categorizando**: se conocen las unidades de medida que ayudan a interpretarlos.
- **Calculando**: los datos pueden haber sido procesados matemática o estadísticamente.
- **Corrigiendo**: se han eliminado errores e inconsistencias de los datos.
- **Condensando**: los datos se han podido resumir de forma más concisa (agregación).

Por tanto, la información es la comunicación de conocimientos o inteligencia, y es capaz de cambiar la forma en la que el receptor percibe algo, impactando sobre sus juicios de valor y sus comportamientos.

Información = Datos + Contexto (añadir valor) + Utilidad (disminuir la incertidumbre)

La información se convierte en **conocimiento** cuando se mezcla con experiencia y valores, y este último sirve como marca para la incorporación de nuevas experiencias e información, siendo útil para la acción.

En las organizaciones (empresas, universidades, etc.), con frecuencia, no sólo se encuentra dentro de documentos o almacenes de datos, sino que también está en rutinas organizativas, procesos, prácticas, y normas (arraigado a la organización).

Las organizaciones dependen de los datos y de la información, componentes que se deben saber extraer, analizar y explotar. Para ello se necesita el conocimiento para dirigir la organización, ya que si no se podría llegar a una situación en la que se avanza, se ejecutan procesos operacionales, se intentan alcanzar los objetivos marcados, pero si alguno de los componentes falla, los procesos se descontrolan, la coordinación empieza a desaparecer, y poco a poco, la organización fracasa o no consigue sus objetivos en tiempo, calidad o cantidad.

El **Business Intelligence**, como se verá en el siguiente apartado, se encarga entre otras cosas de extraer información de los datos provenientes de los sistemas operacionales del negocio (Business Operation), para convertirla en conocimiento para la empresa, y así dar soporte a la toma de decisiones. De alguna forma, facilita las pautas para dirigir los procesos, para que no se descontrolen, e intentar alcanzar los objetivos definidos.

2. BUSINESS INTELLIGENCE.

Una organización se debe entender como un sistema, que tiene unas entradas (materias primas, capital, trabajadores, etc.), un conjunto de procesos de transformación y unas salidas (como productos y servicios) [4].

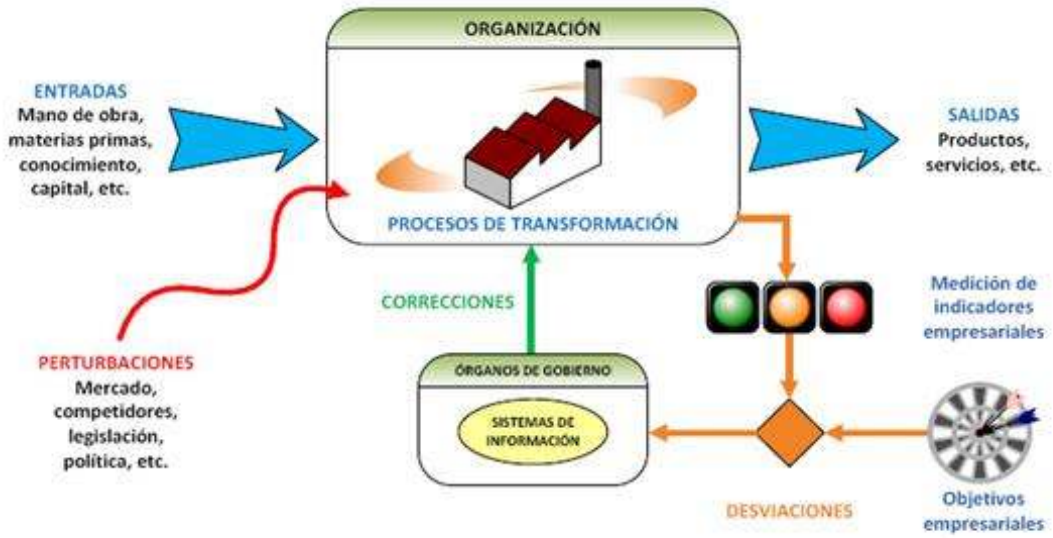

Figura 2. Una Organización desde el punto de vista de un sistema.

El conjunto de procesos de transformación es especialmente sensible a las perturbaciones del mercado, a los competidores, a la legislación vigente, entre otras. Es por eso, que se necesita lo que se denomina medición de indicadores empresariales, que tal y como su nombre indica, estos indicadores miden ciertos aspectos de los procesos, como por ejemplo cuántos productos se desarrollan en una hora. Además de tener mediciones, la empresa debe tener bien definidos unos objetivos empresariales, con los cuales se podrán comparar los indicadores, y así cuantificar las pertinentes desviaciones.

Por lo tanto, se puede decir que una organización/empresa es un sistema en el que se ejecutan procesos que se deben controlar y gestionar, teniendo en cuenta la información de que se dispone a partir de los datos extraídos de los procesos.

Con esta información, los órganos de gobierno de la compañía son capaces de tomar las decisiones necesarias para ejecutar las correcciones pertinentes sobre los procesos, a fin de reducir o erradicar las desviaciones.

Si la información es comprensible, y llega en el tiempo y formato adecuado, ésta permite a la empresa reducir su incertidumbre y tomar mejores decisiones con el objetivo de aportarle una ventaja competitiva.

La sociedad de la información se caracteriza por la utilización de esa información para generar conocimiento, con el fin de mejorar los procesos de cualquier empresa. La información es un bien cada vez menos restringido y más compartido, y la ventaja competitiva de las empresas radica en interpretarla y convertirla en un elemento diferencial, y en un activo productivo y rentable.

Es precisamente el Business Intelligence (la inteligencia de negocio), el que intenta proveer de información para el control y gestión de un proceso de negocio, independientemente de dónde se encuentre esta información almacenada, dando soporte a la toma de decisiones y proporcionado una capa semántica con la cual poder hablar en el lenguaje del negocio, es decir, que todo empleado de la empresa sepa de lo que se está hablando.

El término de **Business Intelligence** suele definirse como la transformación de los datos de la compañía en conocimiento para obtener una ventaja competitiva. Desde un punto de vista más pragmático, y asociándolo directamente a las tecnologías de la información, podemos definirlo como el conjunto de metodologías y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales y la información desestructurada (interna y externa a la compañía) en información estructurada, para su explotación directa o para su análisis y conversión en conocimiento, para finalmente dar soporte a la toma de decisiones sobre el negocio.

Esta definición pretende abarcar un conjunto muy amplio de tecnologías, acrónimos y disciplinas, pero las herramientas que son imprescindibles para el desarrollo del Business Intelligence en cualquier empresa, son los sistemas de almacenaje de datos, las bases de datos, los Data Warehouse (que serán el objeto de estudio en este proyecto) y los Data Marts.

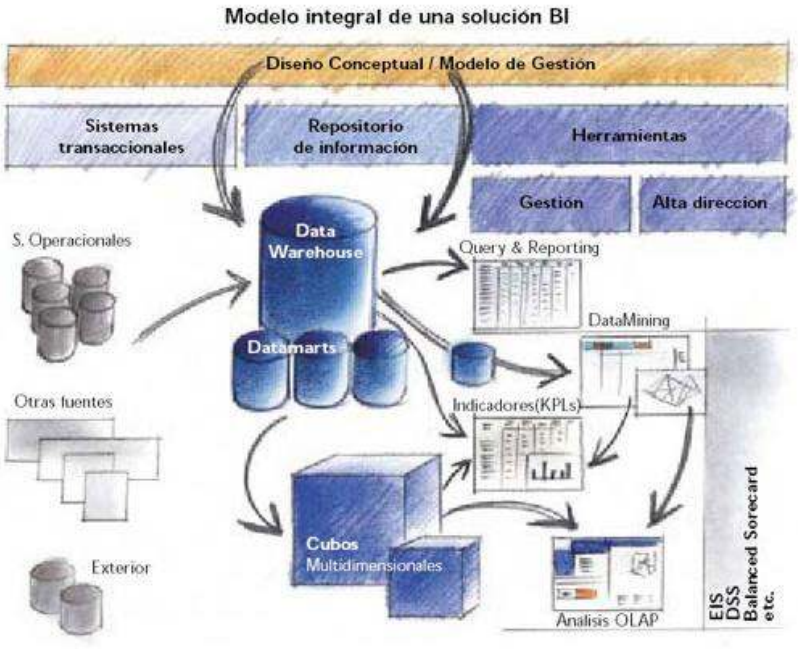

Figura 3. Modelo Integral de una solución de BI [2].

La figura 3 muestra cómo una solución integral de BI se compone de varios módulos que distinguen varios procesos con finalidades y ámbitos distintos.

Podemos ver que el diseño conceptual se divide en tres módulos, Sistemas Transaccionales-Repositorio de Información-Herramientas dónde existen una serie de componentes que forman cada módulo. Existen muchos diagramas que definen la arquitectura de un sistema BI, pero con la figura 3 sobretodo nos queda claro como es el flujo de información, que etapas se van cumpliendo (ETL sobre los sistemas transaccionales hacia el repositorio de información y uso de herramientas y técnicas avanzadas para ofrecer los entregables bien sea a la Alta Dirección o a la Dirección de Gestión de la compañía).

Cabe destacar que una organización puede utilizar por separado cada una de estas herramientas y personalizarlas a sus necesidades, o bien, implementar una solución completa estándar de Business Intelligence.

Este conjunto de herramientas y metodologías proporcionan a la empresa:

- Ahorro de costes.
- Acceso a la información.
- Apoyo en la toma de decisiones.
- Orientación al usuario final.
- Mayor rentabilidad.
- Mayor agilidad empresarial.
- Menores costes operativos.
- Mayor fidelidad de los clientes.
- Optimización de la adquisición de clientes.
- Interacción directa entre los usuarios.

A. BASES DE DATOS RELACIONALES.

El término de base de datos apareció por primera vez en 1963, en un simposio celebrado en California, en el cual se definió como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Las bases de datos son almacenes que nos permiten guardar datos relacionados entre sí y de forma organizada, que son recolectados para que posteriormente se puedan localizar, utilizar fácilmente, o sean explotados por un sistema de información de una empresa. [5].

En el ámbito de la informática, los sistemas de gestión de bases de datos (SGBDs) son programas desarrollados explícitamente para gestionar bases de datos, de forma que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada.

Aunque existen muchos tipos de bases de datos, en los dos siguientes apartados, se describe el más comúnmente utilizado: las bases de datos relacionales.

B. EL MODELO RELACIONAL

Un modelo de datos se podría definir como un modelo abstracto, que describe como se representan los datos y como se acceden a ellos. [6].

El tipo de base de datos más común, son las basadas en el modelo relacional (aunque existen otros modelos como el modelo jerarquizado o el modelo en red). El modelo relacional es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos, y tras su definición en 1970 por Edgar Frank Codd, en la actualidad es el modelo de bases de datos más utilizado por excelencia. [7].

Su idea fundamental es el concepto de relaciones. Una relación se define como un conjunto de *n-tuplas*. Una tupla es un conjunto no ordenado de valores de atributos, aunque estrictamente en matemáticas, una tupla tiene un orden y permite duplicados. Un atributo es una pareja ordenada, formada por un nombre de atributo y un tipo o dominio de valores, refiriéndose a todos los valores únicos que un elemento puede contener. Un valor de un atributo es un valor específico, válido según el tipo del atributo que lo define.

Una relación está compuesta por una cabecera y un cuerpo (también llamado extensión).La cabecera se compone de un conjunto de atributos no ordenados, y el cuerpo está formado por un conjunto no ordenado de n-tuplas. [8].

En este modelo, todos los datos se almacenan en relaciones, y como cada relación es un conjunto de datos, el lugar y el orden en el que estos se almacenen, no tiene mayor relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experimentado.

Para manipular y consultar los datos que se almacenan en diferentes relaciones, se utiliza lo que se denomina un lenguaje relacional, que ofrece una amplia flexibilidad y poder, para administrar la información. Actualmente se cuenta con dos lenguajes formales que son: el álgebra relacional y el cálculo relacional. El álgebra relacional permite describir la forma de realizar una consulta, en cambio, el cálculo relacional sólo indica lo que se desea devolver. [7].

▪ APLICACIÓN DEL MODELO RELACIONAL EN LAS BASES DE DATOS.

En el ámbito de las bases de datos relacionales, a los atributos se les denomina columnas o campos, a las tuplas se les denomina filas o registros, y al valor de un atributo, como la entrada específica formada por la intersección de una columna y una fila cualquiera. [8].

En la Figura se muestra la correspondencia de términos del modelo relacional, aplicados a las bases de datos relacionales.

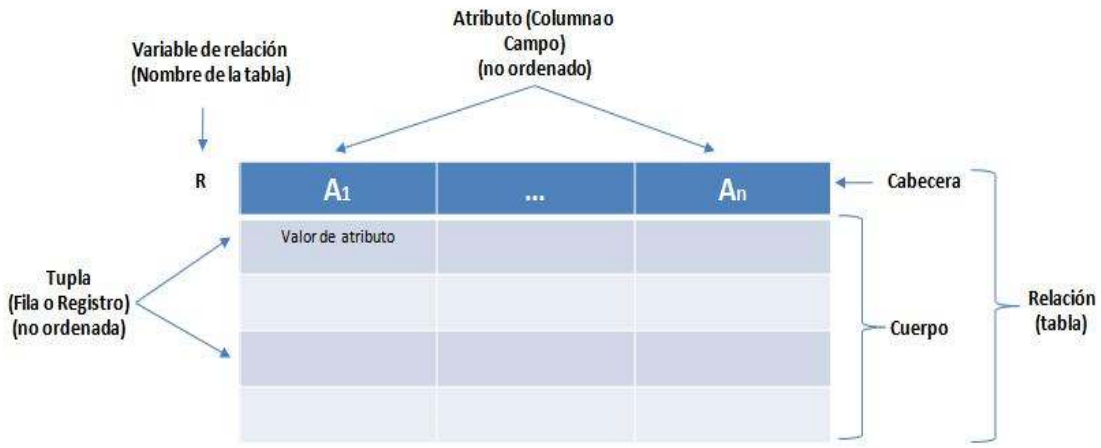

Figura 4. Correspondencia de términos entre el modelo relacional y bases de datos relacionales, adaptación [8].

Este modelo considera una base de datos relacional como una colección de relaciones. Cada relación, se define como una tabla formada por un conjunto de columnas y filas, donde se almacenan los distintos datos.

Las tablas se relacionan o vinculan entre sí por un campo en común, que normalmente se le denomina identificador o clave.

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización, el cual es entendido como el proceso necesario para que una base de datos se utilice de manera óptima y no exista redundancia de datos.
Las bases de datos relacionales se caracterizan por [7]:

- Garantizar que no existe la duplicidad de registros.
- Garantizar la integridad referencial, de forma que si se elimina una fila, se eliminan todas las filas dependientes.
- Favorecer la normalización, por ser más comprensible y aplicable.

El lenguaje relacional más común, para acceder y manipular los datos, de este tipo de bases de datos es el **SQL** (Structured Query Language), un estándar implementado por los sistemas de gestión de bases de datos relacionales, que permite entre otras operaciones la consulta, inserción, actualización y el borrado de datos.

En la Figura 5 se muestra un ejemplo sencillo de base de datos relacional, en el cual se muestra una tabla de Salidas y una tabla de Fechas de salida. Mediante una simple sentencia SQL, se vinculan las dos tablas a través de la clave *Código de salida*, se selecciona aquella

salida cuyo código es 24 y el resultado se ordena por fecha de salida. La sentencia SQL en concreto es la siguiente:

```
SELECT fs.Fecha, s.Salida, fs.[Guia acompañante]
      FROM Salidas s inner join [Fechas de Salida] fs
            on s.[Código de Salida] = fs.[Código de Salida]
      WHERE s.[Código de Salida] = 24
      ORDER BY fs.Fecha
```

Figura 5. Ejemplo de base de datos relacional adaptado de Wikipedia [8].

C. SISTEMAS DATA WAREHOUSE.

En las organizaciones, se generan grandes cantidades de datos constantemente, que son almacenados en algún sistema de información con múltiples fuentes de datos (internas o externas a la empresa). Esta multiplicidad de fuentes (documentos de texto, bases de datos, etc.) requiere que se puedan interconectar entre ellas mediante algún mecanismo único, ya que los datos existentes pueden estar almacenados en diferentes plataformas, formatos, lenguajes de acceso o consulta, distintos sistemas hardware y software base, etc.

Para una empresa, tener un simple almacén de datos no es suficiente para los procesos, ya que realmente lo que necesita es información para llegar al conocimiento con la finalidad de tomar decisiones de negocio, tal y como hemos visto en apartados anteriores.

Se llega a la conclusión de que las empresas necesitan información extraída a partir de los datos que generan sus procesos de negocio. Muchos de estos datos, pueden llegar a estar replicados en distintas bases de datos, y esto conlleva problemas de localización, comprobación y validación. Todos estos problemas hacen que los usuarios tengan dificultades para acceder a ellos, e incluso que existan diferentes visiones de los mismos. [9].

Para conseguir que los datos de negocio estén bien definidos y consolidados, que sean consistentes y se puedan acceder fácilmente, se estructuran y se almacenan en lo que se denomina un Data Warehouse, que son almacenes de datos corporativos que se cargan a través de un proceso de extracción, transformación y carga (ETL), para posteriormente analizarlos y obtener información relevante sobre los procesos mediante la realización de informes, estadísticas, minería de datos, entre otras técnicas. Son los sistemas que se nutren de todos los sistemas operacionales (bases de datos transaccionales) de la empresa, internos y externos a ella, y recogen, limpian, consolidan, unifican y dan formato a la información, para posteriormente analizarla desde infinidad de perspectivas y con gran velocidad de respuesta, para dar soporte a la toma de decisiones.

Bill Inmon, el padre de los Data Warehouse, definió este término como "la colección de datos, orientados a materias, integrados, cambiantes con el tiempo y no volátiles, para la ayuda al proceso de toma de decisiones de la dirección de una empresa". [9].

Otra definición podría ser: "sistema de información histórica e integrada proveniente de los distintos sistemas operacionales de la empresa, que refleja los indicadores clave, asociados a los procesos de negocio de la misma, y sirve de apoyo a la decisiones de gestión". [10]

La creación de un Data Warehouse representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de Business Intelligence.

El detonante del nacimiento de los sistemas Data Warehouse fueron, y siguen siendo, las killer queries o consultas asesinas hacía las bases de datos operacionales de las empresas. Una consulta asesina es aquella consulta a una base de datos que su resolución hace que el rendimiento de ésta empeore, hasta tal punto que puede llegar a dejarla inoperativa. No sólo por el hecho de empeorar rendimientos sino de la imposibilidad de disponer de cierta información en un instante, sin que tenga que ser calculada por procesos pesados y sin poder realizar comparaciones temporales, es decir, ver situaciones similares a los largo del tiempo para poder estudiar variaciones en los movimientos de la actividad de la empresa y anticipar acciones.

Para la toma de decisiones, los mandos estratégicos y los órganos de gobierno de las empresas necesitan información, y ésta se basa principalmente en los datos históricos de la operativa diaria de la empresa. Si la consulta de datos históricos es constante y excesiva, puede llegar a tumbar el sistema operacional, y con la consecuencia de parar la empresa o cómo se indica antes no poder ofrecer la información requerida en el momento deseado.

Es por eso que surge la necesidad de tener un entorno separado que ofrezca soporte a la decisión que no interfiera con la operativa del día a día. De esta forma nacieron los Data Warehouse, como herramienta principal de las soluciones Business Intelligence para

proporcionar información sobre los procesos, analizarla y posteriormente ayudar a la toma de decisiones, además de ser un elemento clave para la planificación y desarrollo de la empresa.

Mediante Data Warehouse, los usuarios pueden acceder a la información de negocio fácilmente, y de alguna forma, ser independientes en la generación de consultas e informes.

Además, pueden integrar los datos consolidados de diferentes aplicaciones operacionales, analizar datos históricos, flexibilizar el análisis de los cruces de información relevante en cada momento, y optimizar el rendimiento empleando un entorno diferente del operacional.

Los Data Warehouse son una herramienta que permite a las empresas que lo han implantado, tener una ventaja competitiva. No son un producto, y no hay que verlos sólo como una tecnología, sino como un enlace funcional de distintas herramientas (un sistema operativo, procesos ETL, un SGBD y herramientas de consulta).

El proceso de implantación de un Data Warehouse es largo y complejo, por eso normalmente se divide en un conjunto de fases iterativas, en que cada una de ellas tiene unos objetivos definidos y limitados.

Los resultados esperados no se ven ni a corto plazo, ni en la finalización de la primera fase.

Incluso se detectan insuficiencias, errores y nuevas necesidades, a medida que los usuarios finales empiezan a utilizar las primeras entregas del proyecto (como en la mayoría de procesos de desarrollo SW).

A medida que las distintas fases del proyecto de implantación se van completando, los cambios son menores, pero un Data Warehouse implantado con éxito no tiene fin, sino que está sometido a un continuo mantenimiento y revisión.

La implantación de un Data Warehouse estándar en una empresa no es una solución del todo adecuada, ya que presenta ciertas limitaciones frente a los desarrollos a medida, y perjudica uno de los objetivos principales de los Data Warehouse: la integración de todos los datos de una organización.

A modo de resumen, con un Data Warehouse los usuarios pueden acceder a los datos fácilmente, permitiendo agregar detalles de la operativa diaria de la empresa y tener una visión más amplia. Estos datos se convertirán en información mediante la realización de informes, estadísticas, etc. Remarcar el hecho de que un Data Warehouse no es una herramienta de reporting, y que mediante la implantación de éstos, se resuelven deficiencias que se pueden encontrar en los sistemas operacionales, que contestan a preguntas como “¿Durante qué meses hemos vendido la mayor cantidad del producto X?, ¿Qué incremento hemos tenido frente al año anterior?, ¿y para cada provincia?” y que sólo deben implantarse si la empresa ya ha consolidado sus sistemas de información para la operativa diaria (se necesita cierto nivel de madurez).

En la Figura 6 se muestra un diagrama con una arquitectura completa de los sistemas que se describirán en los siguientes apartados.

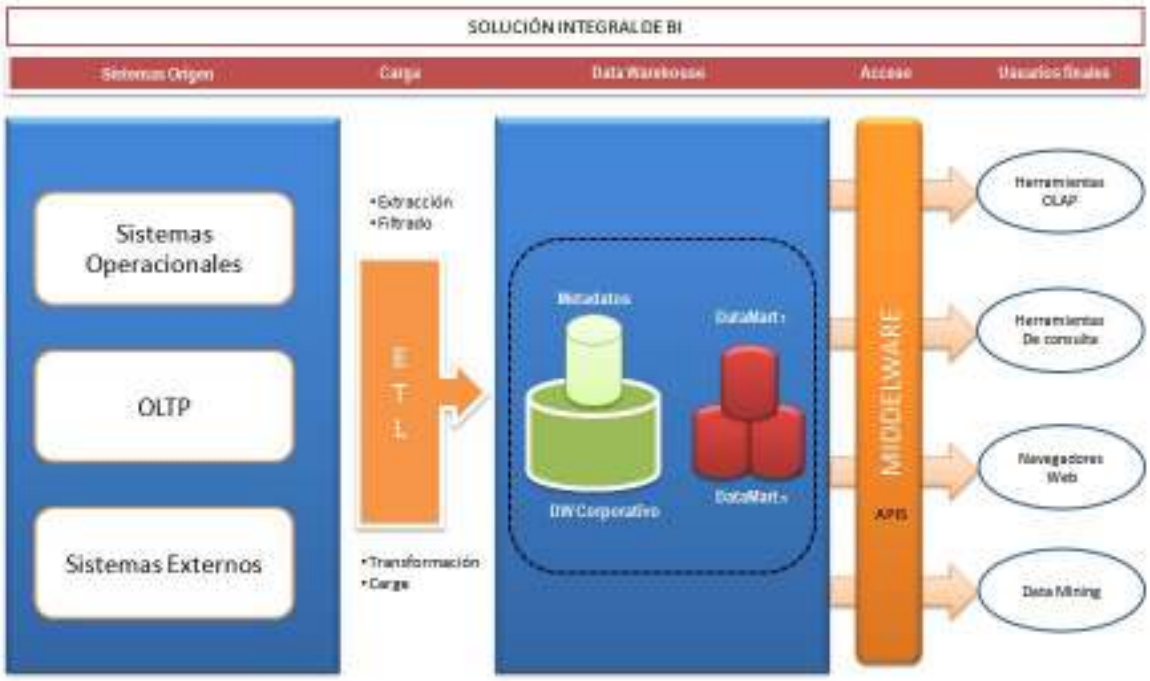

Figura 6. Solución integral de BI.

D. Bases de datos relacionales Vs. sistemas Data Warehouse.

Se puede caracterizar un Data Warehouse haciendo un contraste de cómo los datos operacionales almacenados en un sistema tradicional difieren de los datos almacenados en un Data Warehouse. La siguiente tabla muestra estas diferencias:

Base de Datos Operacional	Data Warehouse
Datos Operacionales	Datos del Negocio para Información
Orientado a la aplicación	Orientado al sujeto
Actual	Actual + Histórico
Detallada	Detallada + Resumida
Cambia continuamente	Estable

Figura 7. BBDD Operacional Vs Data Warehouse.[11].

Un Data Warehouse **no produce nuevos datos**, es decir, tan sólo sirve como repositorio para la información producida por otras aplicaciones.

Lo que se extrae de un Data Warehouse es información y conocimiento. Los servicios que ofrece el data Warehouse son usados por herramientas OLAP (On Line Analytical Processing) así como sistemas de ayuda a la toma de decisiones (DSS, Decision Support Systems), también llamadas aplicaciones de data mining. El data warehouse surge específicamente para dar soporte de almacenamiento a estos procesos.

En algunos casos el Data Warehouse es simplemente una base de datos junto a los procesos asociados para dar soporte a las aplicaciones de Data Mining. Pero en otros casos esto no es así. El Data Warehouse ha venido a poner solución al problema que surge como consecuencia de que muchas aplicaciones tengan diferentes versiones de la misma información distribuida en datos con diferentes formatos. El. Data Warehouse sirve como soporte de integración para esta problemática: Integrador de bases de datos.

Cabe destacar que un Data Warehouse no es un producto y no puede por tanto comprarse, debe ser construido paso a paso. Si bien, existen aplicaciones comerciales y productos que pueden personalizarse en cierta medida para un problema concreto, no hay un producto comercial actualmente, que se adapte a las necesidades de cualquier problema de esta índole (y tampoco tiene lógica que exista). El hecho de que haya que construirlo implica que requiera un período de tiempo, que variará en función de las necesidades de la organización, hasta que el sistema sea productivo.

Al igual que para la construcción de una casa son necesarios los planos de la misma, la construcción de un Data Warehouse (al igual que cualquier sistema de información) debe seguir al menos en cierta medida una serie de normas, que conforman una arquitectura.

En los sistemas de información actuales, una arquitectura es un valor añadido, al igual que lo son los planos para la construcción de una casa: mejora la comunicación y la planificación. Además aumentará la flexibilidad del sistema, mejorará la productividad y facilitará el aprendizaje del funcionamiento del sistema.

- **Mejora de la comunicación**: El esquema de la arquitectura es una herramienta excelente de comunicación a varios niveles. Ayuda a comunicar con el equipo de gestión encargado del proyecto, ayudando a entender la magnitud y complejidad del sistema. También ayuda a entender a cada equipo de trabajo en qué parte del sistema están trabajando.

- **Mejora de la planificación**: La arquitectura provee una tabla de verificación para la planificación del proyecto. Muchos detalles arquitectónicos terminan dispersándose y quedan enterrados en la planificación del proyecto. La arquitectura hace que estén todos visibles en un único lugar y muestra cómo encaja cada componente.

- **Aumento en la flexibilidad**: Al diseñar una arquitectura es necesario anticiparse a las posibles dificultades que puedan surgir, construyendo así un sistema que pueda manejar dichas dificultades de forma rutinaria antes de que puedan convertirse en problemas de difícil solución. La arquitectura se basa en modelos, herramientas y metadatos. Esto añade lo que se conoce como capa semántica al Data Warehouse. Esta capa describe los contenidos y los procesos y es usada por los procesos para crear, navegar y mantener el sistema. De esta manera, el Data Warehouse es más flexible y fácil de mantener.

- **Mejora de la productividad**: La arquitectura utiliza herramientas y metadatos como principal fuente de productividad y reutilización. La productividad se ve mejorada porque la arquitectura nos ayuda a elegir herramientas para automatizar parte de los procesos del Data Warehouse, en lugar de construir capas y capas de código personalizado. Al entender mejor los procesos y las bases de datos involucradas, se hace más fácil para un desarrollador reutilizar procesos existentes en lugar de crearlos de nuevo.

- **Facilidad para el aprendizaje del funcionamiento**: La arquitectura juega un rol importante como documentación del sistema. Ayuda a nuevos miembros del equipo a entender rápidamente los contenidos, los componentes y las conexiones entre ellos. Es importante que el diseño sea lo más preciso y riguroso posible para evitar malentendidos o interpretaciones poco precisas.

2. DATA WAREHOUSE.

1. DEFINICIÓN.

Según se ha anticipado en el apartado anterior con la introducción de los sistemas Data Warehouse queda definido el concepto de **Data Warehouse** como un almacén de datos que recoge toda la información que produce una empresa. Sobre la información que hay, de carácter histórico, se genera nueva información en forma de resúmenes, contabilizaciones e incluso se llega a un nivel superior de detalle para poder ofrecer información orientada a la toma de decisiones en cualquier momento, sin necesidad de esperar largos periodos en la realización de informes tediosos que tienen más probabilidad de error y acumulan un nivel muy superior de imprecisiones.

Es evidente el carácter integrador que ofrece un Data Warehouse, no sólo desde el punto de visto de recopilar información de todas las fuentes de datos que se use en la empresa (ya sean estas internas o externas) sino desde el punto de visto de la calidad de la información, cualidad principal sobre la que se centra éste estudio. Un Data Warehouse puede ser la mejor herramienta para conseguir información de calidad, dónde los datos que se muestren sean válidos, se encuentren correctamente formateados y tengan un propósito bien definido. Es decir, que se sepa para que se vaya a usar dicho dato, como usarlos y que el hecho de hacerlo no traiga errores de la base y en definitiva convertir los repositorios de la información en conocimiento útil a la empresa.

Conviene dejar claro que el Data Warehouse como su nombre indica es un almacén de datos, y que el resto de técnicas introducidas tanto para la construcción del mismo, como para su mantenimiento y explotación forman parte de los sistemas de Data Warehousing, el Business Intelligence y se pueden extender hasta dónde uno quiera.

Podríamos identificar el *Data Warehousing* como “una tecnología, cuyo propósito es reunir información de distintas fuentes y efectuar un proceso de implementación de un proyecto Data Warehouse”.

Una vez aclarado éste detalle conviene retomar las definiciones teóricas que se han ido estableciendo a lo largo de las últimas décadas de forma cronológica, ya que al ver los primeros pasos que se han ido dando en cuanto a los Data Warehouse, nos damos cuenta de que es un tema relativamente reciente pero que tiene fuerte impacto durante ligeros periodos y parece que se estanca, por lo que conviene siempre que se vaya a realizar cualquier tipo de actividad relacionada con un Data Warehouse informarse de las últimas tendencias y de cómo se han desarrollado a lo largo de los años los sistemas creados para prevenir error de diseño, no llevar a expectativas sobreestimadas ni desaprovechar toda la potencia de la que puede dotar el “tener” un Data Warehouse.

Así podemos ver cómo el Data Warehouse *pegó muy fuerte* durante los últimos años de la década de los 80 y primeros de los 90 cuándo en IBM se comenzó a desarrollar el concepto de *Business Data Warehouse*:

- **1988**: Barry Devlin y Paul Murphy publican el artículo "*Una arquitectura de sistemas de negocio y la información*" dónde se introduce el término "*Business Data Warehouse*".

- **1990**: Red Brick Systems lanza "*Red Brick Warehouse*" un sistema de gestión de base de datos específicamente para el almacenamiento de datos.

- **1991**: Soluciones Prisma introduce "*Prisma Warehouse Manager*" un Software para el desarrollo de un Data Warehouse.

- **1992**: ***Bill Inmon***, quién es conocido por muchos como el padre del Data Warehouse, publica el libro "***Building the Data Warehouse***". Marcando un hito en la materia. Inmon define un Data Warehouse como *"un conjunto de datos integrados, históricos, variantes en el tiempo y unidos alrededor de un tema específico, que es usado por la gerencia para la toma de decisiones*". [12]

  El propio Inmon introduce una metodología para la construcción de un Data Warehouse cuyo enfoque debe responder a todos los usuarios y no sólo a un determinado grupo, se debe afinar desde el primer momento los requisitos de negocio y de implementación del Data Warehouse para que haya el menor número de desviaciones posibles durante la construcción del sistema y su mantenimiento. Durante un lustro ésta metodología **Top-Down** fue estrella en la creación de los Data Warehouse.

- **1995**: Se crea la organización sin ánimo de lucro "*The Data Warehouse Institute"* que promueve el desarrollo de los Data Warehouse.

- **1996**: Ralph Kimball, quién es conocido junto con Bill Inmon como el otro gurú o padre de los Data Warehouse, publica el libro "*The Data Warehouse Toolkit*" que supone toda una revolución. Kimball aporta una definición más sencilla del Data Warehouse, lo define como "*una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis"* y propone una metodología **Bottom-Up** para su construcción. Para muchos extrapola el "*divide y vencerás"* al desarrollo del sistema para obtener en menor tiempo y a menos coste versiones del Data Warehouse que puedan ser de utilidad. [13]

En el apartado de metodologías para el diseño de un Data Warehouse se especifican tanto la propuesta por Inmon (Top-down) como la de Kimball (Bottom-up).

- **1996 +** : Definidas las técnicas propuestas por Inmon y Kimbal se decide utilizar uno u otro sistema de creación de Data Warehouse durante unos años y las publicaciones van encaminadas a las técnicas de BI que permiten sacar mayor productividad del Data Warehouse.

2. CARACTERÍSTICAS.

Si bien parece claro y definido el concepto de Data Warehouse, conviene agrupar una serie de características que siempre permitirán distinguir y ayudar a comprender qué es y cómo funciona un Data Warehouse. Así pues, a la pregunta de **¿Qué es un Data Warehouse?** Podríamos contestar lo siguiente:

- Es un ***depósito de datos***. Los datos son independientes de los sistemas operativos o de las aplicaciones existentes, simplemente satisfacen ciertos requerimientos.

- Es una forma de ***arquitectura de estructura de datos***. Permite atender consultas para la ***toma de decisiones*** ya que dota a los sistemas de explotación del DW de agregaciones y desagregación de datos de forma interactiva.

- Con el DW se realiza el análisis del problema en términos de ***dimensiones***. Por ejemplo, permite analizar los datos históricos a través del tiempo. El DW es orientado a sujetos.

- Incluye un proceso que ***integra datos*** provenientes de diversas fuentes, algunas internas y otras externas. Tiene la capacidad de integrar datos heterogéneos para conformar información homogénea y precisa, dónde el hecho de ***generar conocimiento*** sea más sencillo. No sólo usa datos heterogéneos en el origen en cuanto a tipo de tecnología o formato del dato, sino de ámbito, como pudieran ser bases de datos relacionales, documentales, geográficos, de archivos, etc.

- Los datos contenidos en un Data Warehouse constituyen la ***historia detallada*** de los negocios de la empresa y su relación con los clientes. Las empresas que sepan

aprovechar los recursos que ofrece un Data Warehouse estarán en mejor disposición para lograr ventajas competitivas.

- Un Data Warehouse es un sistema de aplicación empresarial que ***contiene su propia base de datos***. Es decir, se puede ver como un sistema aparte de los sistemas dónde la empresa mantiene su actividad empresarial primaria.

- La construcción y desarrollo de un Data Warehouse exitoso requiere la integración de varios componentes de tecnología y la habilidad para hacerlos funcionar todos juntos. Además ***debe tenerse muy claro el propósito*** por el que se creará el Data Warehouse y saber que requerimientos debe cubrir toma un papel crucial en el desarrollo del mismo.

- La finalidad de un Data Warehouse consiste en ayudar al usuario empresarial a conocer el pasado y poder planear el futuro***, ayuda a anticiparse***. Permite ***posicionar la empresa*** con respecto a los competidores.

Una frase recurrente en el mundo de los DW es el hecho de que "***Un Data Warehouse no se compra, se construye***", lo cual explica ese componente de individualidad que existe en cada compañía, el DW se debe adaptar y crear, si bien bajo ciertas metodologías y patrones, de forma individual para cada empresa.

Algunas de las **respuestas** que nos ofrecerá el DW son:

- ¿Cuál es el perfil de mis clientes?
- ¿Cómo se comportan mis clientes?
- ¿Cuál es la rentabilidad que dejan los clientes en mi empresa?
- ¿Cuál es el riesgo que existe con respecto a mis clientes?
- ¿Qué servicios y productos utilizan mis clientes? ¿Cómo puedo aumentarlos?
- Etc.

Algunas de las **aplicaciones** que ofrece un DW son las siguientes:

- Gestión de relaciones con clientes.
- Administración estratégica.
- Análisis de rendimiento con/de clientes.
- Administración de bienes corporativos.
- Etc.

Algunas de las **funciones** que puede tomar un DW son las siguientes:

- Planificación.
- Finanzas.
- Comercialización.
- Logística.
- Etc.

Dada la gran capacidad de conocimiento que puede producir un DW conviene siempre determinar en primera instancia el enfoque empresarial que se va a otorgar al DW, dado que la necesidad de determinar los requerimientos corporativos y traducirlos en consultas que puedan ser respondidas a través del DW será el aspecto de principal aplicación o uso que se dará al DW.

3. ARQUITECTURA DE UN DW.

Dada la aproximación realizada a las soluciones de BI como puede verse en la figura 3 (Modelo Integral de una solución de BI) o la figura 5 (Solución Integral de BI) el ámbito que rodea la creación, mantenimiento y explotación de un DW podría formar parte de la arquitectura de un Sistema de Data Warehousing, por ello vamos a estudiar más al detalle las arquitecturas expuestas en los conceptos previos.

Desde el ámbito teórico la arquitectura conllevaría los siguientes 5 componentes o fases:

- Fuentes de Datos.
- Consolidación.
- Almacenamiento.
- Acceso.
- Explotación.

El flujo de información seguiría el siguiente curso (De izquierda a derecha):

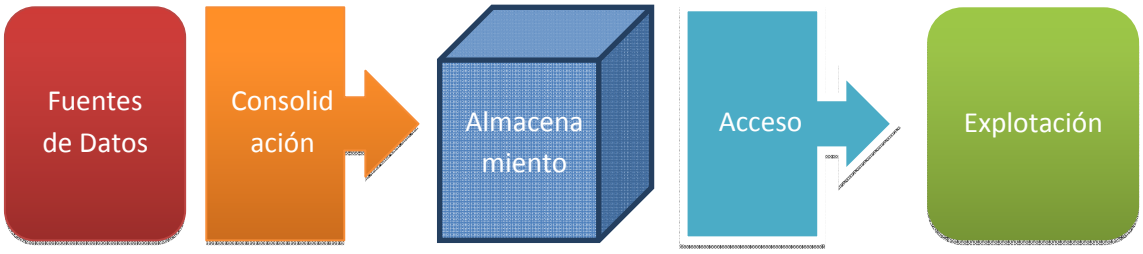

Figura 8. Arquitectura DW – teórica.

A continuación se hace una breve explicación sobre qué representa cada fase o componente que interviene en un sistema de Data Warehousing.

A. FUENTES DE DATOS

El primero de los 5 componentes o fases de la arquitectura de un DW son las ***fuentes de datos.*** Nunca existirá un DW sin Fuentes de datos, ya que la base de todo DW son los datos que trata para construir el sistema objetivo.

Las fuentes de datos pueden ser de varios tipos y se pueden hacer múltiples agrupaciones, la que se asemeja más según su proveniencia es la distinción entre fuentes internas y fuentes externas:

- **Fuentes Internas.**

  Se generan y mantienen en el ámbito de la empresa.

  - Contienen datos provenientes de aplicaciones transaccionales de la empresa (por tanto son bases de datos operacionales de la empresa).
  - Contienen datos heredados de aplicaciones en desuso de la empresa (son datos no necesarios para ejecutar procesos de operaciones actuales, pero que resultan importantes por su valor histórico. Se incluyen al DW incorporando su fecha de vigencia) o simplemente datos históricos que las bases de datos operacionales actualmente no necesitan para realizar su función.
  - Contienen datos que se usan en la empresa y pueden no ser generados por aplicaciones o no estar integrados directamente con el DW o las aplicaciones transaccionales (pueden ser **ficheros planos, XML, hojas de cálculo** –Excel - , cualquier tipo de documento no integrado con las aplicaciones transaccionales directamente, **Etc.**) y por tanto son susceptibles de incorporar niveles de calidad inferiores a los garantizados por las aplicaciones corporativas.

- **Fuentes Externas.**

  Se usan en el ámbito de la empresa pero no se crean ni mantienen dentro de la organización.

  - Contienen **datos maestros** de organizaciones gubernamentales (Tablas de códigos postales, referencias de productos de proveedor, Etc.).
  - Contienen **datos adquiridos** (pueden reflejar investigaciones de mercado, informes externos, auditorías, Etc.).

La arquitectura del componente de fuentes de datos se corresponde con el siguiente diagrama:

Figura 9. Arquitectura DW – Fuentes de Datos

Respecto a la calidad, conviene mencionar que la calidad de los datos que finalmente integran el DW depende directamente de la calidad encontrada en las fuentes de datos o el hecho de saber reconocer una carencia y cómo debe tratarse.

Por otro lado y hablando de aspectos más técnicos suele decirse que las fuentes de datos se forman por sistemas ***OLTP*** (On-Line Transactional Process) que orientan su trabajo a transacciones siendo los DW formados por sistemas OLAP (On-Line Analytical Process) que están orientados al análisis, por lo que su ámbito de gestión y operación es diferente, más adelante podemos ver más a fondo que significa cada técnica.

B. CONSOLIDACIÓN

La fase o componente de ***consolidación*** es la más importante para el propio DW porqué es el momento en el que realmente se crea y se actualiza su contenido.

En la consolidación se incluye el llamado proceso ***ETL***, (abreviatura de Extraction, Transformation and Load o Extracción, Transformación y carga en castellano) que se encarga de llevar el dato del sistema origen al DW en el formato deseado.

No sólo se trata de realizar los tres pasos de los que se componen las siglas ETL, es en éste momento cuándo tenemos la mejor oportunidad para, en primera instancia, ***generar un dato de calidad*** para el DW y poder conseguir que los procesos se hagan de forma eficaz y eficiente además de conseguir que realmente se almacene ***información en el DW*** y no sólo una copia formateada de los datos del origen.

Evidentemente el proceso de consolidación incorpora integración de datos, filtros, normalización y toda clase de técnicas de manipulación de datos. Será en el apartado ETL del capítulo*"3. Componentes, herramientas y conceptos"* dónde nos centremos en el detalle del proceso ETL y el resto de técnicas de limpieza de datos que lo conforman.

Después de la aproximación teórica mencionada puede verse un diagrama que representa el significado de la fase de consolidación:

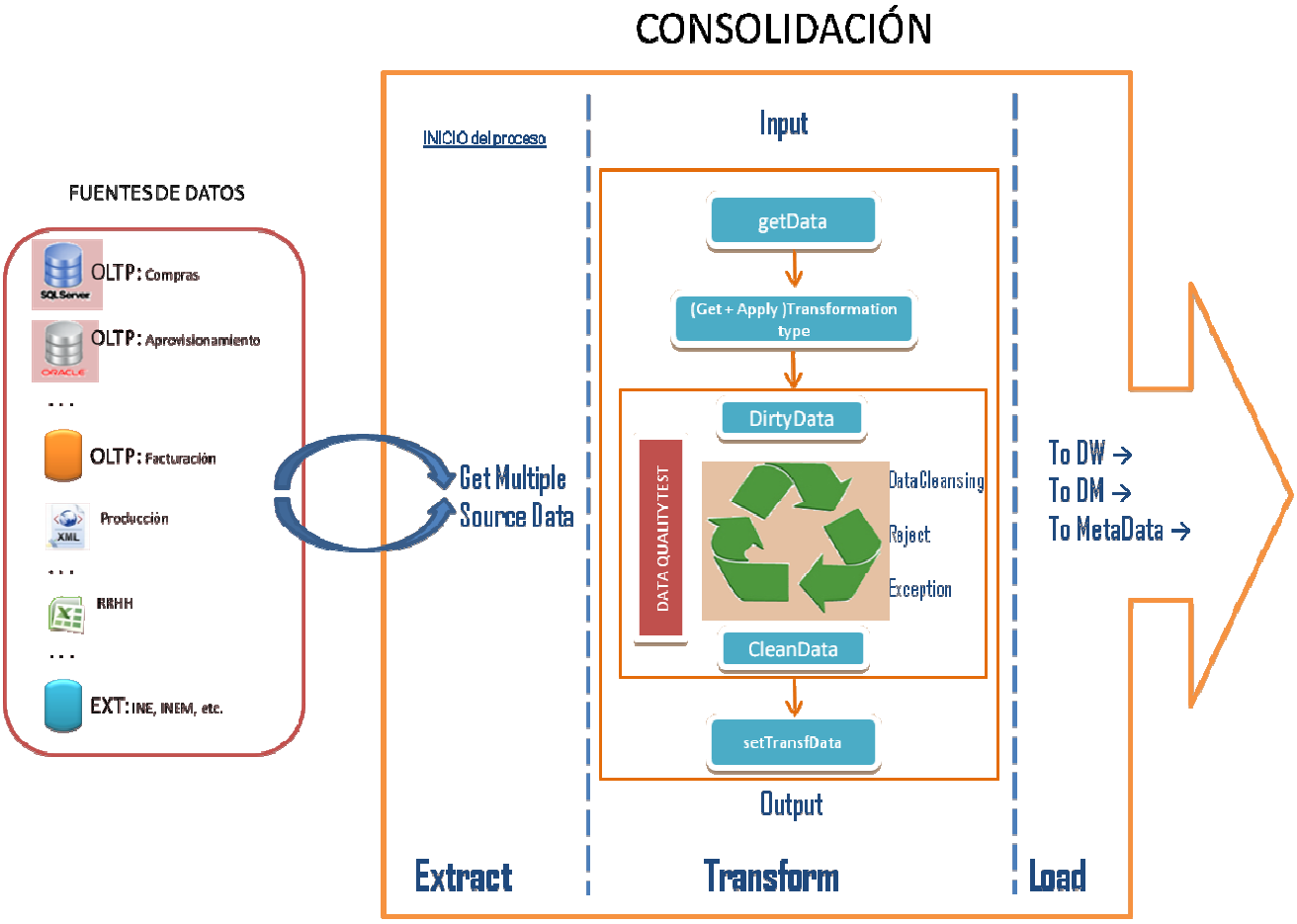

Figura 10. Consolidación

C. ALMACENAMIENTO

La fase o componente de ***almacenamiento*** puede verse como el propio DW, aunque no siempre estará únicamente formado por el DW en sí. Es decir, no será una única BBDD global.

El contenido de un DW se diferencia (estructura de datos aparte) en que es información dónde el dato llega al ***máximo nivel de detalle***, pudiendo contener ***agrupamientos*** y ***totalizaciones*** que respondan a las consultas más recurrentes para optimizar tiempos de respuesta y evitar realizar nuevos cálculos fuera del DW.

Como enuncia ***Inmon*** "***el DW constituye la única versión de la verdad de la empresa***" por el hecho de integrar todos los datos que aparecen en el ámbito de la organización y que no escapa detalle en su almacenamiento. [12]

Una parte importante en la fase de almacenamiento es el Diccionario de Datos, más conocido como ***Metadatos***, dónde se describen los datos almacenados con el objetivo de facilitar el acceso de los mismos a través de las herramientas de explotación del DW o bien como método de documentación. Estos metadatos establecen una correspondencia entre los datos almacenados y los conceptos a los que representan, de manera de facilitar la extracción por parte del usuario de negocios. Más adelante veremos cómo encajan los metadatos en todo el proceso del Data Warehousing.

En el almacenamiento se tiende a diferenciar ***dos estructuras*** de almacenamiento de datos dependiendo de la metodología que se haya seguido para la construcción del DW. Puede existir ***un DW y a partir de él varios DM*** o a la inversa se crearan ***varios DM que podrán formar el DW*** (o incluso obviarlo).

Cómo hemos visto anteriormente la metodología ***top-down*** que propone Inmon se centra en la creación de un DW corporativo, y a partir de él, nos centraríamos en la ***replicación y propagación*** de la información ***del DW para la creación de los DM*** que se necesiten, siendo cada uno orientado a su ámbito dónde tendremos un conjunto de información más reducido y adecuado para la explotación de cierto tipo de información.

Por tanto con la metodología top-Down tendríamos un DW corporativo y tantos DM departamentales o de ámbito como nos interesen, siendo nuestro DW corporativo la *única versión de la verdad de la empresa* y los DM replicaciones y propagaciones de información que dentro de su ámbito puede desarrollarse más a fondo u orientarse a ciertos tipos de consultas etc.

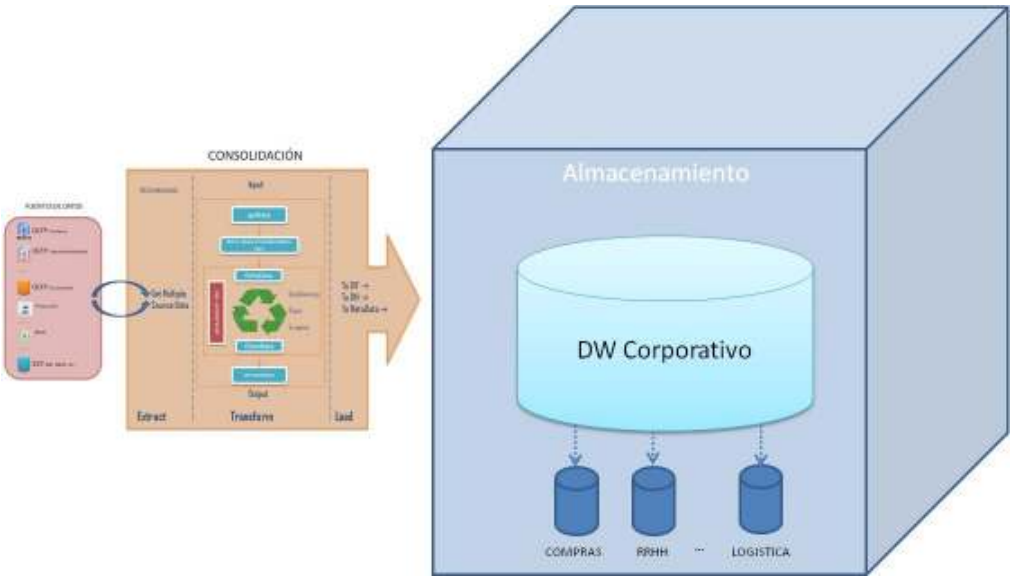

Figura 11. Almacenamiento – Modelo Inmon.

Si por el contrario nos centramos en la metodología ***bottom-up*** que propone Kimball el proceso de almacenamiento se centrará en la ***creación de DM departamentales*** y sería a partir de ellos dónde pudiera construirse el DW corporativo o no, ya sea por cuestiones de tiempo, económicas y porque no vayan a resultar de interés en la toma de decisiones.

Así, si hemos seguido la estructura bottom-up tendremos uno o varios DM departamentales y opcionalmente se habrá construido un DW a partir de la consolidación de los distintos DM en uno.

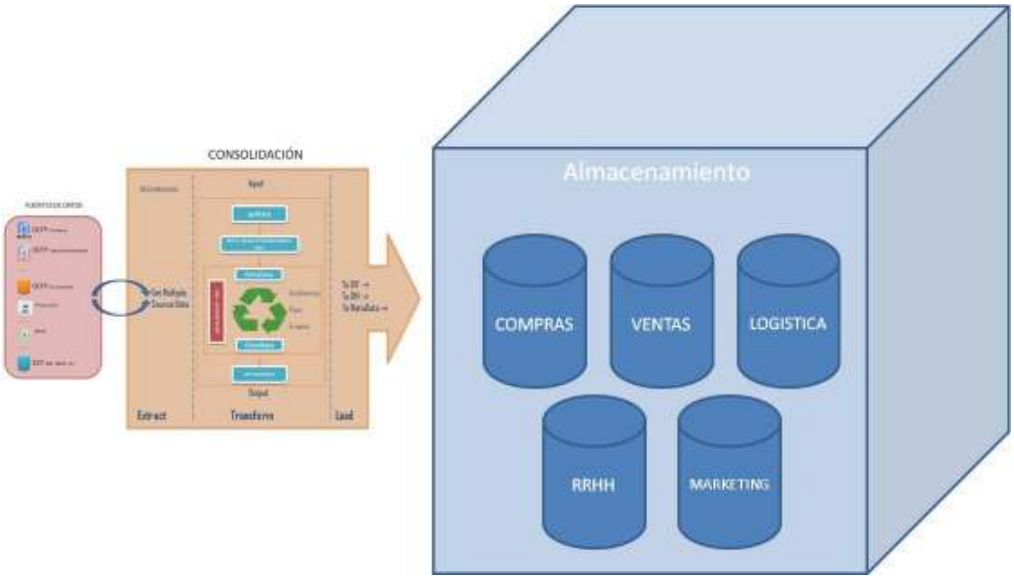

Figura 12. Almacenamiento - Modelo Kimbal.

En el apartado *4 Metodologías para el diseño de un DW* puede estudiarse más a fondo cómo serán ambas metodologías.

Algo que diferencia a los sistemas de Data Warehousing de los sistemas que usan base de datos operacionales es la disposición física y lógica de la información. Como se pudo ver en el apartado introductorio, los DW almacenan sus datos ***en estructuras multidimensionales*** en vez de en estructuras relacionales. Ésta técnica se gestiona a través de las aplicaciones llamadas ***OLAP*** (On-Line Analytical Process) que operarán las bases de datos multidimensionales como son los DW y los DM.

Dentro de un sistema OLAP conviene señalar el concepto de ***cubo OLAP*** (cubo multidimensional o hipercubo). Éste se compone de hechos numéricos llamados medidas (extraídos de las fuentes de datos, de los sistemas OLTP) que se clasifican por dimensiones. A partir de los cubos OLAP se podrán realizar las consultas de las que hablamos, pero veremos más en profundidad a los sistemas OLAP en su correspondiente apartado del capítulo 3 “Componentes, herramientas y conceptos”.

D. Acceso

La fase o componente de ***acceso*** es clave para que el componente de explotación pueda realizar su trabajo. Suele llamarse ***Middleware.*** En él nos encargamos de proveer una capa de acceso a los sistemas que se encarguen de generar ese conocimiento del que tanto hablamos de la información contenida en el componente de almacenamiento.

En Wikipedia podemos ver una definición totalmente acertada sobre la capa de acceso en referencia al Middleware: “*Middleware es un término genérico que se utiliza para referirse a todo tipo de* ***software de conectividad*** *que ofrece servicios u operaciones que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Estos servicios funcionan como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware puede verse como una capa API, que sirve como base a los programadores para que puedan desarrollar aplicaciones que trabajen en diferentes entornos sin preocuparse de los protocolos de red y comunicaciones en que se ejecutarán. De esta manera se ofrece una mejor relación costo/rendimiento que pasa por el desarrollo de aplicaciones más complejas, en menos tiempo. [19]*

La función del middleware en el contexto de los data warehouse es la de asegurar la conectividad entre todos los componentes de la arquitectura de un almacén de datos.” [18].

Para visualizar en mayor medida como la capa de acceso permite desacoplar el DW a una tecnología concreta o sistema de explotación del propio DW avanzamos hasta el estudio del 5º componente, el componente de explotación que usa la capa de acceso para llegar a los datos en las mejores condiciones independientemente del Sistema Operativo o del lenguaje de programación que use para la explotación de los datos.

E. EXPLOTACIÓN.

La fase o componente de ***explotación*** es el encargado de hacer tangible todo el proceso que conlleva un sistema de BI. Se encarga de generar conocimiento que pueda ser útil a la empresa a partir de la información que contiene el componente de almacenamiento. En ésta capa o componente se agrupan todos los interfaces de usuario.

Es decir, se encarga del manejo de las herramientas que obtienen resultados del DW y los DM. Serán herramientas de generación de informes ejecutivos (***EIS***, *Executive Information System*), soporte de toma de decisiones (***DSS***, *Decision Support Systems*), generación de modelos de predicción, análisis estadístico e incluso ***Data Mining***, Inteligencia Artificial, Redes Neuronales y herramientas ***OLAP.***

Como se ha explicado en la capa de acceso, éstas herramientas no tienen por qué utilizarse desde sistemas homogéneos o con la misma tecnología y serán independientes por tanto unas de otras.

Gracias a los Metadatos, se pueden recuperar los informes y consultas predefinidos de manera eficaz y cuyo rendimiento sea alto para que el sistema de explotación obtenga los datos requeridos en las mejores condiciones, tanto de tiempo cómo estructura.

Conviene destacar que la distinción entre los datos o información almacenados en el DW y su utilidad para generar conocimiento depende de los objetivos por los cuáles son concebidos, el sentido de cada dimensión de estudio dentro del DW y saber leer los significados que aporta su estudio, es decir, es en la capa de explotación dónde se debe tener una mayor visibilidad creativa para llegar a ciertas conclusiones no siendo trivial el hecho de explotación de los datos en todos los casos y habrá que usar diversas técnicas para la toma de decisiones, no es el DW el que te hace tomar una decisión sino es la persona que opera con los informes que se generar provenientes del DW.

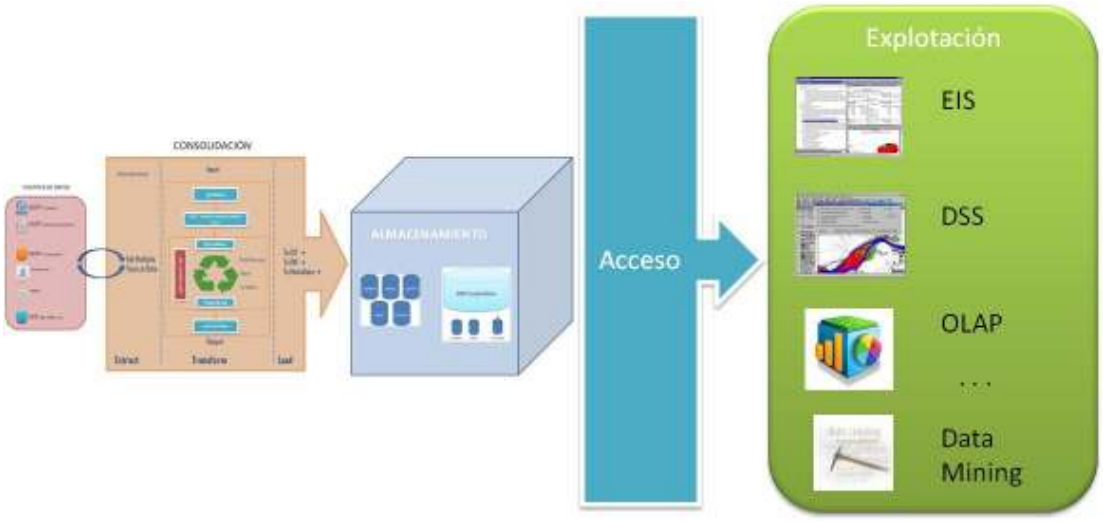

Figura 13. Arquitectura DW – Acceso y Explotación.

4. METODOLOGÍAS PARA EL DISEÑO DE UN DW.

A. INTRODUCCIÓN

Durante el proceso de diseño del DW se debe tener en cuenta todos los requisitos que establezcan los usuarios finales que explotarán la información del almacén para que la realización del mismo se amolde en el mayor grado posible a dichas necesidades (sea orientado a la creación de cierto tipo de informes), es decir, el enfoque sobre la arquitectura interna del propio DW debe hacer más fácil la presentación de informes y se debe adecuar a las tareas identificadas por los usuarios durante la fase de análisis.

Una vez detallados, comprendidos y aterrizados (o exportados) dichos requisitos al modelo físico del DW la explotación de la información podrá garantizar un rendimiento eficaz y más eficiente al ser definido con propósito y no cómo un DW con estructura estándar.

Para que nuestro DW pueda conseguir su objetivo, los procesos de negocio se seleccionan con el objetivo de modelarlos, estableciendo una granularidad para cada uno de ellos. Por este motivo es muy importante entender correctamente los datos de los diferentes sistemas dentro de la organización y las relaciones entre ellos. La gestión de estas relaciones durante la carga de almacenamiento de datos es esencial.

En la actualidad, existen dos metodologías tipo para el diseño de un DW. Éstas son la metodología ***Top-Down*** establecida por Inmon y la metodología ***Bottom-Up*** establecida por Kimball.

La experiencia en la materia ha dictaminado que debido al continuo cambio que se produce en el ámbito de los Sistemas de Información, la metodología de Ralph Kimball tiene mayor grado

de productividad y aceptación, por el hecho de que ofrece resultados más rápidos que la de Inmon. Suele haber un mayor porcentaje de Sistema de Business Intelligence creados con la metodología top-down y arquitectura de Data Marts.

Por el hecho anterior vamos a detallar **teóricamente la metodología Top-down** de Inmon para continuar a especificar **dos versiones de la metodología Bottom-Up** de Kimball, que son la ***"Rapid Warehousing Methodology"*** [17] propuesta por el instituto SAS (USA) y el estándar (de facto) que representa el ***"Ciclo de Vida"*** que propone Ralph Kimball en su libro *"The Data Warehouse Lifecycle Toolkit" [14].*

B. TOP-DOWN O DE INMON

El enfoque Top-Down se utiliza cuándo la tecnología y los problemas sobre el negocio son conocidos y se encuentran muy definidos en su estructura y posible gestión.

En el modelo Top-Down tendríamos un DW corporativo, y a partir de él, nos centraríamos en la ***replicación y propagación*** de la información ***del DW para la creación de los DM*** que se necesiten.

Cada Data Mart se encuentra orientado a su ámbito particular, dónde tendremos un conjunto de información más reducido y adecuado para la explotación de cierto tipo de datos. Por ello, estaría cada uno orientado (en cuanto a la existencia de agregaciones, recuentos, etc.) a la construcción de ciertos tipos de informes que no dependen del resto de áreas compuestas por el DW corporativo, sino que son muy específicas.

Por tanto, se trata de un método sistémico en el que se minimizan los problemas de integración, pero que dónde aumenta la complejidad de construcción debido a la gran cantidad de datos aportando poca flexibilidad. Por ello es menos frecuente la existencia de sistemas de BI creados con la metodología Top-Down en grandes empresas siendo más común en organizaciones de tamaño medio o cuyo proceso de negocio se encuentra muy definido y los procesos son bastante robustos y tienen a no cambiar y no fallar.

Con el método Top-Down de Inmon se formula un resumen del sistema, sin especificar detalles. Cada parte del sistema se refina diseñándola con mayor detalle. Después, cada parte nueva se redefine, cada vez con mayor detalle, hasta que la especificación completa sea lo suficientemente detallada como para validar el modelo.

Con el modelo de Inmones fácil utilizar en los diseños las "**cajas negras**" que se encargan de ciertas tareas permitiendo cumplir requerimientos, aunque no expliquen en detalle los componentes individuales.

Para **Bill Inmon**, el diseño de un DW comienza ya con la mera introducción de datos, ya que, la existencia de grandes cargas de datos obliga a definir ciertas políticas de gestión que serán usadas durante el traspaso de datos entre origen y el destino en favor de la eficiencia que pueda conseguirse en cuanto a tiempo y tamaño de los datos en el destino.

Por otro lado, se sustenta uno de los principio básicos y fundamentales del desarrollo de un DW, dónde **es imprescindible separar en varios entornos** alos orígenes de datos del sistema DW destino.

Será primordial que el DW resida en un sistema puramente independiente, con una base de datos independiente, alojado en un servidor independiente y dónde las características de seguridad, mantenimiento y operatividad se ajusten solamente al propósito del DW (y no dependan o interfieran en el trabajo cotidiano de ningún otro sistema).

Es necesario afirmar, que los sistemas actuales tienen gran cantidad de datos, lo que hace poco realista el intentar hacer cargas de datos en periodicidades muy bajas. Si el volumen de datos no está cuidadosamente gestionado y condensado, dicho volumen de datos impide que los objetivos del DW se alcancen. Por otro lado, hay que conocer cuándo esos datos serán definitivos y no sean propensos a contener errores.

Es una tarea muy importante el detallar cuidadosamente la periodicidad en las cargas de datos, los conjuntos de datos que se transfieren y el carácter que van a tener en el DW destino.

A Inmon se le asocia frecuentemente con los **DW a nivel empresarial**, que involucran desde un inicio todo el ámbito corporativo, sin centrarse en un incremento específico hasta después de haber terminado completamente el diseño del DW. En su filosofía, un DM es sólo una de las capas del DW y los DM son dependientes del depósito central de datos o DW Corporativo y por lo tanto se construyen después de él.

El enfoque de Inmon de desarrollar una estrategia de DW e identificar las áreas principales desde el inicio del proyecto es necesario para asegurar una solución integral. Inmon se apoya en que ésta medida puede ayudar a evitar la aparición de situaciones inesperadas que pongan en peligro el éxito del proyecto de creación del DW debido a que se conoce con antelación y bastante exactitud la estructura que presentarán los principales núcleos del desarrollo.

Inmon es defensor de utilizar el **modelo relacional para el ambiente en el que se implementará el DW Corporativo**, ya que como él mismo afirma, la creación de una base de datos relacional con una ligera normalización, son la base de los DM. O lo que es lo mismo, **a partir de los esquemas relacionales, a los que se les irá añadiendo complejidad, se obtendrán finalmente los DM**.

El desarrollo de la metodología propuesta por Inmon en [12] se aprecia en la siguiente figura:

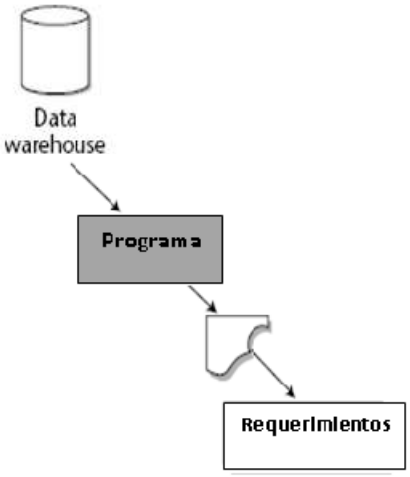

Figura 14. Desarrollo de un DW según Inmon.

La metodología de Inmon tiene un enfoque a modo de explosión en el sentido de que en cierto modo no viene acompañada del ciclo de vida normal de las aplicaciones, sino que los requisitos irán acompañando al proyecto según vaya comprobándose su necesidad. Esta visión de Inmon puede traer consigo mucho riesgo a la compañía, ya que invierte grandes esfuerzos en el desarrollo del DW y no es hasta la aparición de los DM cuando se empieza a explotar la inversión y obtener beneficios.

Esta estrategia se contempla en el marco de que es imposible conocer cuáles son las necesidades concretas de información de una empresa, el ambiente dinámico en que se mueve la organización, el cambio de estructura que conlleva el desarrollo de la nueva plataforma y los consiguientes cambios a los sistemas transaccionales que su introducción implica. Esto hace bastante probable que después de la gran inversión en tiempo y recursos en el desarrollo del DW, se haga patente la necesidad de cambios fundamentales que traen consigo altos desviaciones en el plan de desarrollo para la organización (desviaciones en cuanto a tiempo y coste), poniendo en evidente peligro el éxito de todo el proyecto en sí y que podían ser evitados tras una n una temprana puesta en explotación de un primer avance del DW.

Por ello, una metodología Top-Down tiene sentido cuándo los requisitos y problemas de negocio son muy conocidos y poco cambiantes.

Esta metodología para la construcción de un sistema de este tipo es frecuente a la hora de diseñar un sistema de información estándar (en referencia a cualquier aplicación de gestión o de operación, no de un DW), utilizando las herramientas habituales como el esquema Entidad/Relación.

Al tener un enfoque global, es más difícil de desarrollar un sistema DW que en un proyecto sencillo, pues estamos intentando abordar el **todo**, a partir del cual luego iremos al **detalle**. Esta es otra de las restricciones que trabajan en contra de la metodología de Inmon ya que

implica un consumo de tiempo mayor, teniendo como consecuencia que muchas empresas se inclinen por usar metodologías con las que obtengan resultados tangibles en un espacio menor de tiempo siempre minimizando el riesgo a favor de la productividad a medio plazo.

C. Bottom-Up o de Kimball

La metodología tipo restante, que propone el enfoque bottom-up es una metodología rápida que se basa en experimentos y prototipos. Es un método flexible que permite a la organización ir más lejos con menores costos. La idea es construir DM independientes para evaluar las ventajas del nuevo sistema a medida que avanzamos. En él, las partes individuales se diseñan con detalle y luego se enlazan para formar componentes más grandes, que a su vez se enlazan hasta que se forma el sistema completo. Las estrategias basadas en el flujo de información bottom-up se antojan potencialmente necesarias y suficientes porque se basan en el conocimiento de todas las variables que pueden afectar a los elementos del sistema.

a. Rapid Warehousing Methodology.

Entre las metodologías que se emplean actualmente está la técnica "*RapidWarehousing Methodology (RWM)*" propuesta por el Instituto SAS (EEUU) en el año 1998 [17]. Dicha metodología es iterativa y está basada en el desarrollo incremental del proyecto de DW dividido en cinco fases que se asemejan a las correspondientes en el ciclo de vida tradicional de desarrollo de un sistema informático, pero con pequeños matices y con carácter iterativo y de retroalimentación, son:

1. ***Definición de objetivos.***
2. ***Definición de los requerimientos de información.***
3. ***Diseño y modelización del sistema.***
4. ***Implementación.***
5. ***Revisión.***

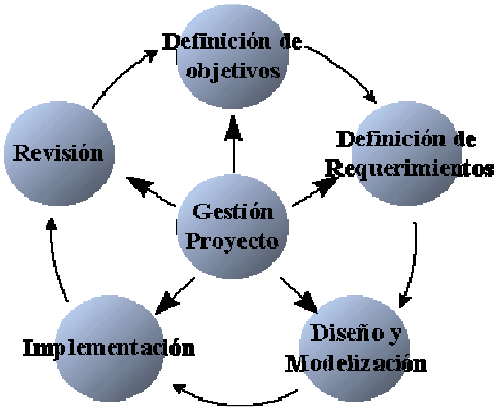

Figura 15. Rapid Warehouse Methodology

Todas irán acompañadas por una fase de gestión del proyecto que guiará los pasos a seguir entre revisiones.

A continuación se describe con mayor detalle cada una de las fases que componen la metodología Rapid Warehousing Methodology.

Definición *de los objetivos*

Durante la primera fase se debe definir en primera instancia el alcance del proyecto y un responsable de proyecto. Una vez realizado este primer paso se debe organizar un equipo de trabajo tanto del lado de desarrollo del sistema como de los departamentos que vayan a usar los DM, serán los usuarios finales.

Conviene establecer cuáles son las funciones que cubrirá el DW para que sea el suministrador de información de negocio estratégica para la empresa. Se definirán así mismo, los parámetros que permitan evaluar el éxito del proyecto y métricas de control y gestión.

Definición *de los requerimientos de información*

El objetivo de ésta fase es definir todas las necesidades de negocio y funcionales que puedan requerir los usuarios finales del DW. Para ello se necesita convocar una serie de reuniones de tomas de requisitos que mantendrá el equipo de trabajo con los distintos representantes del departamento usuario final del sistema.

Se trata de dar sentido al sistema DW, porque los requerimientos identificados en ésta fase, serán los que trate de optimizar el diseño de DW. El resto de etapas que forman parte del proceso se nutren de la información definida en éste apartado, cuánto más claros sean los requerimientos menos dudas surgirán en el proceso de diseño e implementación a la hora de tomar decisiones sobre cómo dar solución a las necesidades requeridas por los usuarios.

Se realizará el estudio de los sistemas de información existentes, que ayudarán a comprender las carencias actuales en los sistemas encargados de generar los informes de negocio y explotación de los datos para que el sistema DW optimice su solución. Se conocerán cuáles son las consultas que más se realizan, la periodicidad con que se hacen, el perfil de los usuarios que la realiza etc.

Al finalizar esta fase se obtendrá el documento de definición de requerimientos en el que se reflejarán no solo las necesidades de información de los usuarios, sino cual será la estrategia y arquitectura de implantación del DWH.

Diseño *y modelización*

Tiene por objetico obtener el modelo lógico de datos del DW. Los requerimientos de información identificados durante la anterior fase proporcionarán las bases para realizar el diseño y la modelización del mismo.

Se identificarán las fuentes de datos (Internas y Externas), qué transformaciones son necesarias a llevar a cabo sobre los datos fuente para convertirse en información con la estructura del DW.

Se identificarán que agrupaciones son necesarias para dar solución a los requerimientos identificados en la fase anterior y se modelarán las dimensiones necesarias para que el sistema DW tenga una estructura ágil que permita obtener resultados en periodos óptimos de tiempo.

Este modelo estará formado por entidades y relaciones que permitirán resolverlas necesidades de negocio de la organización.

Implementación

La implantación de un DW lleva implícitos los siguientes pasos:

- Extracción de los datos de las distintas fuentes.
- Transformación de los datos origen en información contenida en las estructuras del DW.
- Carga de los datos validados en el DW. Deberá planificarse con una periodicidad que se adaptará a las necesidades durante las fases de toma de requerimiento y diseño del sistema.
- Explotación del DW.Dependiendo del tipo de aplicación que se quiera dar a la información se usarán unas técnicas u otras. Entre las más habituales podemos encontrar las siguientes:
    - Query & Reporting
    - On-line analytical processing (OLAP)
    - Executive Information System (EIS) o Información de gestión
    - Decission Support Systems (DSS)
    - Visualización de la información

Revisión

La construcción del DW no finaliza tras la implantación del mismo, sino que forma parte de un procesoiterativo dónde se quiere afinar cada vez más en las posibilidades que

ofrezca el sistema para generar conocimiento y a su vez realizar sus tareas de manera más eficiente posible.

Se debe realizar una serie de mejoras evolutivas que mantengan a la postre el DW como un sistema con utilidad. Debería realizarse una serie de revisiones del sistema planteando preguntas que permitan, después de los seis o nueve meses posteriores a su puesta en marcha, definir cuáles serían los aspectos a mejorar o potenciar en función de la utilización que se haga del nuevo sistema.

Gestión *del proyecto*

El eje principal que permite que durante el proceso de desarrollo, mantenimiento y evolución de un sistema DW todo salga bien, qué se cumpla con los objetivos planificados y con los tiempos y costes requeridos es la gestión del proyecto.

Ésta debe encargarse de la coordinación y ejecución de las distintas fases que conforman la construcción e implantación del DW, de las relaciones entre los departamentos del usuario final y el equipo de trabajo, etc.

Para la gestión del proyecto se usan metodologías específicas que permiten asegurar la calidad en el resultado final del proyecto y que el seguimiento del mismo siga dentro de los parámetros deseados.

Desde la gestión del proyecto se debe promover una fase de formación en la herramienta utilizadas o de incorporación de personal experto en la materia, para asegurar un máximo aprovechamiento del sistema. Seguir los pasos de la metodología y comenzar el DH por un área específica de la empresa permitirá obtener resultados tangibles en un corto espacio de tiempo.

b. Ciclo de Vida: Metodología de Ralph Kimball.

Ralph Kimball junto con Bill Inmon es el autor considerado como el "Gurú" del DW. Su metodología se ha convertido en el estándar de facto en el área de apoyo a las decisiones empresariales.

En el año 1998 dicha metodología se recoge como proceso a seguir en el desarrollo de un DW tras la publicación del libro: "*The Data Warehouse Lifecycle Toolkit*" [14].La siguiente figura muestra de forma esquemática las fases que componen la metodología propuesta por Kimball (y como se interrelacionan) y los siguientes apartados resumen el contenido de cada una de las fases.

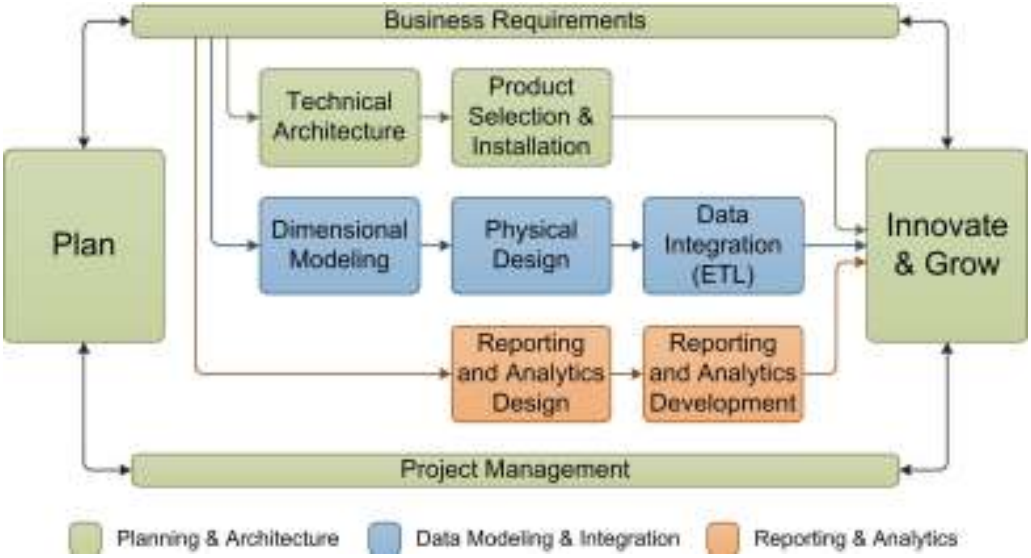

Figura 16. Lifecyle – Metodología Kimball

Como puede verse en el esquema se muestra el flujo general de implementación de un DW. Identifica una secuencia ordenada de tareas y actividad que deben producirse concurrentemente.

Para que la estructura de organización sea óptima, muchas necesidades deben ser adecuadas. No todos los detalles de las tareas del ciclo de vida deben ser ejecutadas en todos los proyectos, como siempre dependerá del alcance y planificación que se defina sobre el mismo, por ello la primera de las tareas sirve como base principal del desarrollo del sistema DW.

Planificación del Proyecto (Plan)

Con la planificación del proyecto se busca identificar los objetivos y el alcance del proyecto de desarrollo del sistema DW, incluyendo las justificaciones del negocio, las evaluaciones de factibilidad, de costes, etc.

Esta etapa se concentra sobre la **definición** del proyecto. Según sentencia Kimball: *"Antes de comenzar un proyecto de data warehouse o data mart, hay que estar seguro si existe la demanda y de dónde proviene. Si no se tiene un usuario sólido, posponga el proyecto".*

Como metodología, en esta etapa propone identificar el alcance preliminar basándose en los **requerimientos del negocio** y no en fechas límites, construyendo la justificación del proyecto en términos del negocio.

A nivel de planificación del proyecto se establece la identidad del mismo, el personal (los usuarios, gerentes del proyecto, equipo del proyecto), desarrollo del plan del proyecto, el seguimiento y la monitorización.

Definición de los requerimientos del negocio (Business Requirements)

Como puede verse en el diagrama, aparece ésta fase tras la planificación del proyecto y de él surgen tres flujos (tracks) concurrentes bien diferenciados que incluyen las definiciones sobre Tecnología, Datos y Aplicaciones de BI para concurrir en la tarea final de integración, más adelante estudiamos los flujos que surgen de los requisitos de negocio. Cabe destacar que se dibuja la fase como un rectángulo que horizontalmente abarca casi todo el diagrama ya que entre la fase de planificación y de revisión (innovación y crecimiento) siempre va a estar presente la definición o refinación de los requerimientos del negocio.

Un factor determinante en el éxito de un proceso de DW es la interpretación correcta de los diferentes niveles de requerimientos expresados por los distintos grupos de usuarios.

La técnica utilizada para revelar los requerimientos de los analistas del negocio difiere de los enfoques tradicionales guiados por los datos. Los diseñadores de los DW deben entender los factores claves que guían el negocio para determinar efectivamente los requerimientos y traducirlos en consideraciones de diseño apropiadas.

Los usuarios finales y sus **requerimientos** impactan **siempre** en la implementación de un DW. Según la perspectiva de Kimball, *"los requerimientos del negocio se posicionan en el **centro del universo del Data Warehouse**".* Como destaca siempre el autor, los **requerimientos del negocio deben determinar el alcance** del DW (qué datos debe contener, cómo deben estar organizados, cada cuánto tiempo deben actualizarse, quiénes y desde dónde accederán o manipularán el sistema, etc.).

Como puede apreciarse en el esquema, la dependencia entre tareas se indica por el alineamiento vertical y su secuencia interna por el alineamiento horizontal, dado que la definición de requerimiento está en el nivel superior hace que el resto de flujos o tareas dependan directamente de ella.

FLUJO TECNOLÓGICO:

1. Diseño de la arquitectura técnica. (Technical Architecture).

Los entornos de DW requieren la integración de numerosas tecnologías. Se debe tener en cuenta tres factores:

1. Los requerimientos del negocio.
2. Los entornos técnicos disponibles.
3. Las directrices técnicas y estratégicas planificadas por la compañía.

Para poder establecer el diseño de la arquitectura técnica que forme el entorno del sistema DW.

Algunos equipos de trabajo no entienden las ventajas de una arquitectura y tienen la sensación de que las tareas son demasiado opacas, por lo que entienden su diseño como una distracción y un obstáculo para el progreso del DW, así que optan por omitir el diseño de la arquitectura. Sin embargo, hay otros equipos de trabajo que dedican un tiempo demasiado grande para el diseño arquitectónico.

Kimball recomienda no irse a ninguno de los dos extremos, sino hacerlo de una forma intermedia. Para ello propone un proceso **de 8 pasos a seguir para asegurar un correcto diseño arquitectónico** sin extenderse demasiado en el tiempo.

Paso 1. Establecer un grupo de trabajo de arquitectura.

Es muy útil disponer de un pequeño grupo de trabajo de dos a tres personas que se centren en el diseño de la arquitectura. Por lo general, es el arquitecto técnico, trabajando con los datos de diseño, el que estará al frente de este grupo de trabajo. Este grupo necesita establecer la definición de sus tareas y orientar cómo será su la línea de prestaciones en el tiempo. También es necesario explicar al resto del equipo sobre la importancia que conlleva el diseño de la arquitectura y porqué se establece para que todo el grupo de trabajo vaya en la misma dirección.

Paso 2. Requisitos relacionados con la arquitectura

La arquitectura se crea para apoyar las necesidades del negocio, la intención no es comprar más productos sino crear un sistema que se adapte al cien por cien a los requerimientos del negocio identificados con anterioridad.

En consecuencia, el elemento fundamental para el proceso de diseño de la arquitectura proviene de los requerimientos de negocio obtenidos en esa fase de definición.

El enfoque principal es descubrir las implicaciones arquitectónicas asociadas a las necesidades críticas del negocio, por lo que además de aprovechar la definición de los requisitos del proceso de negocio, también se llevan a cabo reuniones adicionales dentro de la organización para comprender la normativa vigente dentro del marco tecnológico, instrucciones técnicas previstas y los límites no negociables. Cuánto antes se produzcan éstas reuniones menor desviación por culpa de incumplimiento o desconocimiento de detalles se producirá en el desarrollo del DW.

Paso 3. Documento de requisitos arquitectónicos.

Una vez definidos los requerimientos de negocio y llevado a cabo las entrevistas suplementarias es momento de documentar las conclusiones. La forma de hacerlo ha de ser sencilla pues el objetivo es tener una lista con cada requisito de negocio que tiene impacto en la arquitectura.

Como bien se comentaba antes del desglose de los 8 pasos a seguir para la correcta definición de la arquitectura técnica, hay que tratar de no extenderse demasiado en el detalle de los requerimientos hasta que el sistema quede en fases superiores ni obviar demasiado éste documento ya que sí conviene que todo requisito arquitectónico se documente, al menos para que no se pueda olvidar u obviar en fases futuras.

Paso 4. Desarrollo de un modelo arquitectónico de alto nivel.

Una vez que los requisitos de la arquitectura se han documentado es hora de empezar a formular modelos para apoyar las necesidades identificadas. Para ello se dividen los equipos de trabajo según los componentes principales, como el acceso a datos, metadatos y la infraestructura. A partir de aquí, los equipos definen y refinan el modelo arquitectónico de alto nivel.

Será necesario establecer reuniones de seguimiento e integración que ayuden a hacer ver a todos los integrantes del equipo de trabajo que se trabaja en una dirección y que persona que integre un grupo de trabajo sepa las decisiones que se van tomando en el resto de grupos y evitar situaciones inesperadas por falta de cohesión en el grupo con la definición de componentes que no se adapten o integren fácilmente los unos con los otros.

Paso 5. Diseño y especificaciones de los subsistemas.

Una vez llegados a este punto es momento de hacer un diseño detallado de los subsistemas. Para cada componente, el grupo de trabajo diseña una lista con las capacidades necesarias de dicho componente.

Por otro lado se tienen en cuenta las necesidades de seguridad, así como la infraestructura física y las necesidades de configuración. *En algunos casos, las opciones de infraestructura, tales como el hardware del servidor y el software de base de datos, están predeterminados por la propia empresa*.

El tamaño, escalabilidad, rendimiento y flexibilidad son factores clave a considerar al determinar el papel de los cubos OLAP en el conjunto de la arquitectura técnica.

Paso 6. Determinar las fases de aplicación de la arquitectura.

Es probable que no se puedan poner en práctica todos los aspectos de la arquitectura técnica a la vez. Algunos no son negociables, mientras que otros se pueden aplazar a una fecha posterior; éstos, son los requisitos de negocios para establecer las prioridades de la arquitectura.

Paso 7. Documento de la arquitectura técnica

Se debe de documentar la arquitectura técnica, incluyendo las fases de la implementación prevista. El documento de arquitectura incluirá información adecuada de manera que los profesionales cualificados puedan proceder con la construcción del sistema.

Paso 8. Revisar y finalizar la arquitectura técnica.

El plan de la arquitectura se debe comunicar con diferentes niveles de detalle: equipo de proyecto, sponsor y director del proyecto. Tras la revisión, la documentación debe ser actualizada y utilizada inmediatamente en el proceso de selección del producto.

2. Selección e instalación de SW. (Product selection & Installation).

Los entornos de DW requieren del uso de ciertos productos y herramientas SW adicionales y será ahora cuándo se seleccione cuáles, se instalen y forme al personal (en caso de que no conozca el SW) para su manipulación.

Éste apartado puede verse reducido por el hecho de que en muchas ocasiones las empresas que desean obtener un sistema DW ya tienen una serie de productos y herramientas estándar predefinidos para usar y será el equipo de trabajo el que se tiene que adaptar al mismo.

La selección estará basada en la arquitectura técnica diseñada. Para ello hay que ver que necesidades hemos definido para obtener una selección de productos y herramientas de la siguiente índole:

- Plataforma HW (en el caso de que no venga predefinido por la empresa requeridora del DW).
- SGBR: Sistema gestor de base de datos (en el caso de que no venga predefinido por la empresa requeridora del DW).
- Herramienta ETL.
- Herramienta de consultas (Query Tools).
- Herramienta de reporte.

A continuación, tras la selección de los productos y licencias convenientes, se debe proceder a la instalación de los mismos que más se adapte a las necesidades obtenidas en la definición de la arquitectura tecnológica documentando cómo se ha llevado a cabo y porqué se realizan los pasos que difieran de lo estándar.

Por último se debe verificar que la instalación es correcta y que no existen problemas en los productos y herramientas para asegurar la correcta integración *extremo a extremo* del sistema DW (Permisos, accesos, direcciones, etc.).

FLUJO DE DATOS:

1. Modelo dimensional (Dimensional Modeling).

La definición de los requerimientos del negocio determina los datos necesarios para cumplir los requerimientos analíticos de los usuarios. Diseñar los modelos de datos para soportar estos análisis requiere un enfoque diferente al usado en los sistemas operacionales.

Básicamente, se comienza con una matriz donde se determina la dimensionalidad de cada indicador y luego se especifican los diferentes grados de detalle dentro de cada concepto del negocio, así como la granularidad de cada indicador y las diferentes jerarquías que dan forma al modelo dimensional del negocio (MDN) o mapa dimensional.

El proceso de modelo dimensional se verá con detalle en el apartado *"3.2 OLAP: On-Line Analytical processing del siguiente apartado Componentes, herramientas y conceptos"*.

2. Diseño Físico (Physical Design).

El diseño físico de la base de datos se focaliza sobre la selección de las estructuras necesarias para soportar el diseño lógico. Un elemento principal de este proceso es la definición de estándares del entorno de la base de datos. La indexación y las estrategias de particionamiento se determinan también en esta etapa.

En la estrategia de particionamiento o agregación, el DW tiene, y debe tener, todo el detalle de información en su nivel atómico.

Así, por poner algún ejemplo, en los sectores de telecomunicaciones o banca es habitual encontrarse con DW con miles de millones de registros. Sin embargo, la mayoría de consultas no necesitan acceder a un nivel de detalle demasiado profundo.

Un jefe de producto puede estar interesado en los totales de venta de sus productos mes a mes, mientras que el jefe de área consulta habitualmente la evolución de ventas de sus zonas.

Incluso con el uso de índices, la compresión de las tablas, o con una inversión millonaria en hardware, estas consultas habituales deberían leer, agrupar y sumar decenas de millones de registros, lo que repercutiría directamente en el tiempo de respuesta y en el descontento de los usuarios.

Por tanto, muchas veces lo más complicado será realizar la correcta elección de las tablas agregadas necesarias. De nada sirve crear muchos agregados si estos no se utilizan, por lo que es necesario conocer las consultas habituales de los usuarios para hacer la selección de las tablas agregadas.

La solución ante estas situaciones pasa siempre por la preparación de tablas agregadas. Estas tablas deben ser versiones reducidas de las dimensiones asociadas con la granularidad de la tabla de hechos y añaden los indicadores de las tablas de detalle a un nivel superior.

Por ejemplo, las ventas podrían quedar pre-calculadas a nivel mensual, o por cliente, o por producto. De esta manera, las consultas típicas del jefe de producto o del jefe de área podrían ejecutarse en pocos segundos, sin necesidad de acceder a la tabla de ventas detalladas.

La existencia de estas tablas agregadas debe ser completamente transparente para el usuario de negocio. Es decir, tanto el jefe de área como el jefe de producto trabajarán con el indicador "Ventas", y la herramienta de BI hará el resto.

Por otro lado, en la estrategia de indexación los índices son estructuras opcionales optimizadas y orientadas a conjuntos de operaciones.

Según Kimball, "*las tablas de dimensión deben tener* ***un único índice sobre las claves primarias****y sería recomendable que el índice estuviera compuesto de un único atributo"*. Además recomienda el uso ***de índices de tipo árbol-B en atributos de alta cardinalidad***y aplicar ***los índices de mapas de bits en atributos de cardinalidad media o baja***.

La clave principal de la tabla de hechos es casi siempre un subconjunto de las claves externas, de manera que se elegirá un índice concatenado de las principales dimensiones de la tabla de hechos y *dado que muchas consultas tienen relación con la dimensión fecha, ésta debería liderar el índice definido*.

Además, el atributo fecha en la primera posición permitirá aumentar la velocidad de los procesos de carga de datos que se agrupan por fecha y, dado que la mayoría de los optimizadores de consulta de los sistemas de gestión de bases de datos permiten que se utilice más de un índice a la hora de resolver una consulta, es posible construir diferentes índices en las demás claves ajenas de la tabla de hechos.

3. Integración de los de datos: ETL. (Data Integration: ETL).

Esta etapa es típicamente la más subestimada de las tareas en un proyecto de DW.

Las principales actividades de esta fase del ciclo de vida (ETL process) son:

- La extracción de datos.
- La transformación de los datos origen para que se convierta en la información estructurada que se ha definido en el modelo dimensional.
- La carga de datos.

Se definen como ***procesos de extracción*** aquellos requeridos para obtener los datos que permitirán efectuar la carga del Modelo Físico diseñado.

Se definen como ***procesos de transformación*** los procesos para convertir o recodificar los datos fuente a fin de poder efectuar la carga efectiva del Modelo Físico.

Por otra parte, los ***procesos de carga*** de datos son los procesos requeridos para poblar el DW.

Todas estas tareas son altamente críticas pues tienen que ver con la materia prima del DW: los datos y la información. La desconfianza y pérdida de credibilidad del DW provocará efectos inmediatos e inevitables si el usuario se encuentra con información inconsistente. Es por ello que la calidad de los datos es un factor determinante en el éxito de un proyecto de DW. ***Es en esta etapa donde deben sanearse todos los inconvenientes relacionados con la calidad de los datos fuente***. Para cumplir con estas premisas es necesario tener en cuenta ciertos parámetros a la hora de desarrollar las tablas de dimensión y la tabla de hechos.

FLUJO DE INTEGRACIÓN DE APLICACIONES DE BI

1. Diseño del análisis y reporte de la información

Se refiere al conjunto de aplicaciones que consultan, analizan y presentan la información que reside en el modelo dimensional.

Desde un punto de vista funcional, las aplicaciones de BI serían las que aportarían valor al negocio del DW ya que la meta es dotar de nuevas capacidades o posibilidades al negocio para soportar y mejorar la toma de decisiones.

Nunca debe subestimarse ésta fase ya que será la que de forma visible y tangible ofrezca al usuario final la potencia de la que se dota al sistema DW.

Durante el diseño se identifican las aplicaciones de BI candidatas a consumir el DW y los interfaces de navegación apropiados para ello.

Siempre será orientado a las necesidades de los usuarios finales.

El diseño produce la especificación de las aplicaciones de BI.

2. Desarrollo del análisis y reporte de información

Se trata de la configuración de los metadatos del negocio y la infraestructura de las herramientas de BI.

Conlleva la construcción y validación de aplicaciones BI analíticas y operacionales junto con un portal de navegación para su uso.

Despliegue, innovación y crecimiento (Innovate & grow).

Por último tenemos la fase de ***despliegue***. Si la planificación se ha ejecutado correctamente y se ha asociado a las pautas indicadas se puede asegurar:

- Los resultados de los flujos de tecnología, datos y aplicaciones de BI
- La disponibilidad de la infraestructura de capacitación y apoyo.

El despliegue debe quedar bien sincronizado. Deberá ser aplazado si todas las piezas, tales como, formación, documentación o validación de datos no están listos para la liberación a producción del sistema al cien por cien.

Tras el despliegue se inicia una fase de ***mantenimiento*** del sistema cuándo se encuentra en producción. Incluye tareas técnicas operacionales que son necesarias para mantener en estado óptimo el flujo de trabajo habitual del sistema. Se deberá realizar tareas cómo:

- Monitorización del uso del sistema.
- Tunning del desempeño.
- Mantenimiento de la tabla de índices.
- Procedimiento de BackUps del sistema.
- Resolución de posibles incidencias.

En ésta fase siempre quedará patente un sentimiento de apoyo permanente, capacitación y comunicación con los usuarios finales.

Finalmente llegamos a la zona de **innovación y crecimiento**, que inevitablemente se tiene que producir de continuo en un sistema de DW.

Si el DW es exitoso y productivo para la empresa siempre tiende a extenderse, una extensión se considera como síntoma de éxito (siempre y cuando existan requerimientos de negocio que permitan extender el sistema).

Se deberá priorizar los nuevos requerimientos para iniciar un nuevo ciclo como el visto anteriormente pero partiendo de una solución robusta y puesta en producción.

3. COMPONENTES, HERRAMIENTASY CONCEPTOS

1. OLTP: ON-LINE TRANSACTIONAL PROCESSING.

El procesamiento de transacciones en línea u **OLTP** (**On-Line Transactional Processing**) es un tipo de sistemas que facilitan y administran aplicaciones transaccionales destinadas usualmente para entrada y recuperación de datos o el procesamiento de transacciones. [51]

Se apoyan en **gestores transaccionales**, que son componentes que procesan información descomponiéndola de forma unitaria en operaciones indivisibles llamadas transacciones (serían los INSERT, UPDATE, DELETE, etc.). Dónde cada transacción finalizará de forma correcta o incorrecta por unidad completa, no hay estados intermedios. Se suelen confirmar las transacciones mediante la aceptación de ellas (COMMIT) o devolviendo la mase de datos al estado anterior (ROLLBACK). [52]

Gracias a éste diseño los sistemas OLTP infieren un tiempo de procesamiento muy rápido, manteniendo la integridad de los datos en entornos multi-acceso y propiciando una tasa de eficacia muy elevada en la gestión de transacciones por segundo.

Sin embargo en las bases de datos OLTP no suele haber datos detallados o resumidos, siendo el esquema utilizado para almacenar bases de datos transaccionales el modelo de entidad, por lo general 3FN (3ª Forma Normal).

El modelo OLTP cada vez necesita más recursos para las transacciones que se propagan por una red y con frecuencia integrar datos de varias empresas simultáneamente. Por ésta razón el software actual (Sistemas Gestores de Bases de Datos) utilizan procesamiento cliente-servidor y capas middleware para que las diferentes plataformas empresariales puedan ejecutar sus transacciones.

En grandes aplicaciones, la **eficiencia** del OLTP puede depender de lo sofisticado que sea el software de gestión de transacciones o de las tácticas de optimización que se utilicen para optimizar la concurrencia de transacciones contra la base de datos. Gracias a la definición que se extiende del modelo OLTP se puede crear entornos descentralizados para las bases de datos más exigentes dónde residen programas de intermediación OLTP que distribuyen el procesamiento de transacciones entre carios ordenadores en una red. Por lo tanto, se suele usar el modelo OLTP para integrarse en una arquitectura orientada a servicios SOA.

Por lo comentado hasta ahora se identifican dos claros **beneficios** del procesamiento OLTP, la simplicidad y la eficiencia. [51]

- Proporciona una base concreta para la estabilidad de una organización gracias a las actualizaciones oportunas y fiables →**Previene anomalías de actualización.**
- Amplia de forma sencilla el número de consumidores sobre una base de datos por lo que extiende con facilidad el uso de la misma →**Alta escalabilidad.**

- Los procesos individuales se ejecutan mucho más rápido →**Asegura la consistencia de los datos a través de las transacciones.**
- Permite generar en tiempos muy bajos grandes estructuras de datos para su persistencia y manipulación con tiempos de respuesta altos. Además reduce el esfuerzo en cuánto a la modificación de aplicaciones. →**Optimiza la eficiencia en los procesos de la aplicación.**
- La documentación es sencilla gracias a que las estructuras de informaciones no son complejas y las posibles operaciones que se puedan obtener sobre ellas son atómicas.→**Facilidad de aprendizaje.**

Algunos **inconvenientes** están destinados al reporte y análisis de información sobre la explotación de éstos sistemas.

Si bien son sistemas que facilitan la rápida producción de estructuras para su uso, esto produce que el análisis de información a través de toda la estructura de datos sea complejo.

Además las consultas analíticas que resumen grandes volúmenes de datos afectan negativamente la capacidad del sistema para responder a las transacciones en línea, limitando su mayor beneficio.

Existe por otro lado un arma de doble filo con respecto a la facilidad de adaptación del modelo OLTP en la integración de arquitecturas SOA lo que aumenta la complejidad en cuando a la definición de directrices de seguridad sobre la base de datos.

En definitiva, se suele usar sistemas OLTP para el procesamiento diario del flujo de información empresarial y se combina, con sistemas OLAP para el análisis de los datos generados. Los sistemas OLTP suelen ser fuentes de datos para los sistemas DW.

2. OLAP: ON-LINE ANALYTICAL PROCESSING.

EL procesamiento analítico en línea u OLAP (**On-Line Analytical Processing**) ha sido creado como solución para agilizar el análisis de datos en grandes estructuras de información. El modelo OLAP va ligado al Business Intelligence o Inteligencia de Negocio y suele ser referenciado como "Cubos OLAP". [54]

El ámbito empresarial de un sistema OLAP suele ir ligado a la creación de **informes** en las áreas de ventas, marketing, minería de datos o resúmenes de dirección.

La razón de usar sistemas OLAP reside en la rapidez de respuesta que se obtiene sobre datos resumidos, agrupados y detallados en contraposición con los sistemas OLTP. Los sistemas OLAP priman las sentencias SELECT mientras que los sistemas OLTP priman las sentencias INSERT, UPDATE o DELETE.

Los **Cubos OLAP** (o hipercubos) son estructuras multidimensionales que contienen datos resumidos de grandes bases y fuentes de datos (Sistemas OLTP, ficheros de datos, etc.). Estos cubos se componen de hechos numéricos llamados **medidas** que se clasifican por dimensiones. A los cubos se asocian **metadatos** que indican cómo obtener los datos y se pueden definir siguientes diferentes esquemas de organización de la información (esquema en **estrella** o esquema en **copo de nieve**).

El **esquema en estrella** es un modelo de datos que tiene una tabla de hechos que contiene los datos para el análisis rodeada de las tablas de dimensiones. La tabla de hechos al situarse en el centro, y de un tamaño superior al resto, asemeja el dibujo que provoca el esquema a una estrella.

El **esquema en copo de nieve** es una estructura algo más compleja que el esquema en estrella. Se forma cuándo alguna de las dimensiones se implementa para más de una tabla de datos. La finalidad es normalizar las tablas y así reducir el espacio de almacenamiento al eliminar posibles redundancias de datos. Al tener que crear más tablas de dimensiones y más relaciones entre tablas provoca un peor rendimiento.

Tradicionalmente los sistemas OLAP se clasifican en las siguientes categorías:

- **ROLAP: OLAP Relacional.**

La implementación OLAP almacena los datos en un sistema relacional apoyado de un motor OLAP. Típicamente los datos son detallados evitando las agregaciones y las tablas se encuentran sin normalizar. Aunque suelen usarse con esquemas en estrella o en copo de nieve, es posible trabajar sobre esquemas relacionales.

La arquitectura de un sistema ROLAP está compuesta por un servidor de base de datos relacional y el motor OLAP se encuentra aislado en un servidor dedicado.

La principal ventaja de esta arquitectura es que permite el análisis de una enorme cantidad de datos.

- **MOLAP: OLAP Multidimensional**.

La implementación OLAP almacena los datos en un sistema multidimensional. Para optimizar tiempos de respuesta, el resumen es usualmente calculado por adelantado.

Los valores pre-calculados o agregaciones son la base de las ganancias de desempeño de este sistema. Algunos sistemas utilizan técnicas de comprensión de datos para disminuir el espacio de almacenamiento en disco debido a los valores pre-calculados.

- **HOLAP: OLAP Híbrido**.

La implementación OLAP se divide para algunos datos en un sistema relacional y para otros en un sistema multidimensional. Algunos de los motivos por los que se usa el sistema es para primar las ventajas de un sistema ROLAP o MOLAP por cada tipo de dato o la facilidad que se tenga de extender el sistema OLAP hacia una arquitectura u otra.

Cada sistema OLAP tiene distintos beneficios y no siempre son los mismos dependiendo del proveedor que se use.

Algunas implementaciones **MOLAP** son propensas a la "explosión" de la base de datos; este fenómeno provoca la necesidad de grandes cantidades de espacio de almacenamiento para el uso de una base de datos **MOLAP** cuando se dan ciertas condiciones: elevado número de dimensiones, resultados pre-calculados y escasos datos multidimensionales. Las técnicas habituales de atenuación de la explosión de la base de datos no son todo lo eficientes que sería deseable.

Por lo general **MOLAPofrece mejor rendimiento** debido a la especializada indexación y a las optimizaciones de almacenamiento. **MOLAP** también necesita menos espacio de almacenamiento en comparación con los especializados **ROLAP** porque su almacenamiento especializado normalmente incluye técnicas de compresión.

ROLAP es generalmente **más escalable**. Sin embargo, el gran volumen de pre-procesamiento es difícil de implementar eficientemente por lo que con frecuencia se omite; por tanto, el rendimiento de una consulta **ROLAP** puede verse afectado.

Desde la aparición de **ROLAP** van apareciendo nuevas versiones de bases de datos preparadas para realizar cálculos, las funciones especializadas que se pueden utilizar tienen más limitaciones.

HOLAP (OLAP Híbrido) engloba un conjunto de técnicas que tratan de combinar MOLAP y ROLAP de la mejor forma posible. Generalmente puede pre-procesar rápidamente, escala bien, y proporciona una buena función de apoyo.

Por otro lado, existen algunas catalogaciones optimizadas según su ámbito basadas en las anteriores como son **WOLAP** (Web OLAP, orientado para la Web), **DOLAP** (Desktop OLAP), **RTOLAP** (Real Time OLAP, orientado al tiempo real) y **SOLAP** (Spatial OLAP).

3. COMPARATIVA ENTRE SISTEMAS OLTP Y SISTEMAS OLAP.

Comúnmente podemos dividir los sistema IT en transaccionales OLTP y analíticos OLAP sin miedo a equivocarnos indicando que los sistemas OLTP producen los datos de la operativa diaria y sirven de fuente de datos para los sistema DW mientras que los sistemas OLAP ayudan a analizar los datos.

El sistema OLTP va ligado a la operación de datos y el sistema OLAP va ligado con la información extraída del dato (orientado a sistemas Data Warehouse).

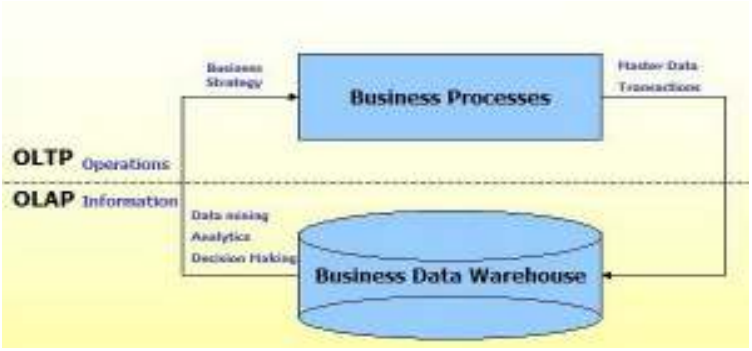

Figura 17. Comparativa entre OLAP y OLTP [54]

Con la siguiente tabla se identifican las diferencias más comunes entre sistemas diseñados con OLTP o con OLAP.

Diferencias entre Sistemas OLAP y OLTP		
Concepto	**OLTP**	**OLAP**
Fuente de datos	Los datos son operacionales; OLTPs son la fuente original de los datos.	Consolidación de los datos, OLAP datos normalmente proceden de las distintas bases de datos OLTP.
Propósito de los datos	Para controlar y ejecutar las tareas fundamentales del negocio	Para ayudar con la planificación, resolución de problemas y apoyo a las decisiones
Sobre el dato	Revela una instantánea de los procesos de negocio en curso	Son datos multidimensionales, diversos puntos de vista de tipos de actividades comerciales
Actualizaciones e inserciones	Inserciones cortas y rápidas y actualizaciones iniciadas por los usuarios finales	Trabajos que se ejecutan en batch actualizan por lotes los datos.
Consultas	Las consultas quedan relativamente estandarizadas y sencillas. Retornan relativamente pocos registros	Suelen ser consultas complejas que implican agregaciones
Rapidez de procesamiento	Suelen ser muy rápidas	Depende de la cantidad de datos involucrados, la actualización de datos por lotes puede tomar mucho tiempo.
Necesidades de espacio	Puede ser relativamente pequeño, si los datos históricos se archivan	Más grande debido a la existencia de estructuras de agregación de datos y la historia, requiere más que los índices de OLTP
Diseño de la BBDD	Suele ser mucho con bastantes tablas normalizadas.	Normalmente es menor, aunque mucho más complejo, con tablas sin normalizar y uso de esquemas en estrella o copo de nieve
BackUp y restauración	El backUp se fácil y seguro. La restauración de los datos conlleva pérdidas de información.	En lugar de copias de seguridad periódicas, algunos entornos pueden considerar simplemente volver a cargar los datos OLTP como un método de

Figura 18. Comparativa entre OLAP y OLTP (2)

4. ETL: EXTRACT, TRANSFORM AND LOAD.

El proceso ***ETL***, (abreviatura de Extraction, Transformation and Load o Extracción, Transformación y carga en castellano) que se encarga de llevar el dato del sistema origen al DW en el formato deseado.

No sólo se trata de realizar los tres pasos de los que se componen las siglas ETL, es en éste momento cuándo tenemos la mejor oportunidad para, en primera instancia, ***generar un dato de calidad*** para el DW y poder conseguir que los procesos se hagan de forma eficaz y eficiente además de conseguir que realmente se almacene ***información en el DW*** y no sólo una copia formateada de los datos del origen.

Evidentemente el proceso de consolidación incorpora integración de datos, filtros, normalización y toda clase de técnicas de manipulación de datos. En el siguiente diagrama podemos ver cuál sería el flujo natural de recorrido de un Dato desde la fuente hasta el destino. Explicaremos cada paso del proceso ETL para identificar las acciones más comunes a realizar:

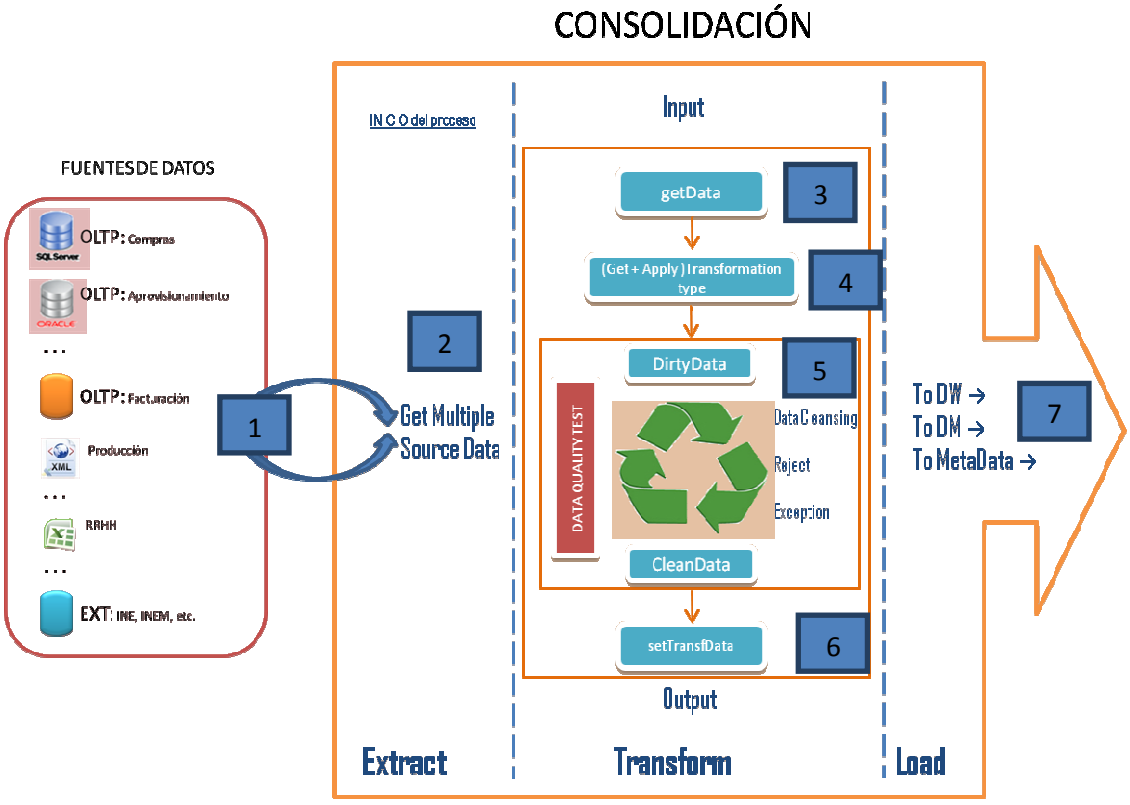

Figura 19. El proceso ETL detallado.

A continuación explicamos los puntos identificados sobre el diagrama de flujo de proceso ETL genérico dividiendo las mismas en 4 grupos diferenciados. Por otro lado, hay que tener muy claro que grupo de actividades sigue a la precedente dado que en un flujo de éste tipo cada tarea facilita datos a la siguiente para que se vaya ejecutando el proceso completo, es decir,

sin realizar las tareas previas no se puede avanzar en el flujo, siempre vamos a seguir una estructura secuencial.

A. COMUNICACIÓN CON LAS FUENTES DE DATOS: INTEROPERABILIDAD

Incluye el punto 1.

Se ha querido diferenciar el proceso de comunicación con las fuentes de datos por motivos prácticos, es decir, en la teoría no se suelen contemplar técnicas o definir reglas de comunicación con las fuentes de datos porque directamente se suele ir al detalle de la tarea principal que queremos acometer, que es el proceso ETL. Pero se ha demostrado que para que el proceso ETL pueda comenzar sin inconvenientes y no conlleve retrasos innecesarios o sin contabilizar se debe tener en cuenta una serie de cuestiones como son:

- ¿Dónde se encuentran las fuentes de datos?
- ¿Accedemos a copias de las fuentes de datos o a las fuentes de datos de producción?
- ¿Cuándo podemos acceder a las fuentes de datos? Y ¿Durante cuánto tiempo?
- ¿Qué restricciones de seguridad residen en las fuentes de datos para poder comenzar a extraer datos de ellas?
- ¿Qué tipo de tecnología forma la fuente de datos? En base a ello hay que determinar cómo conectar el proceso de extracción con la fuente de datos.

Una vez habiendo tenido especificadas las preguntas anteriores y subsanadas las cuestiones que pudieran surgir, sobretodo en el ámbito de seguridad e interoperabilidad con las fuentes de datos podremos dar comienzo a la fase teórica inicial del flujo ETL, la extracción.

Es muy importante prevenir todos los problemas de **interoperabilidad** y definirlos desde un inicio ya que pueden ser problemas muy comunes que no se definen en las planificaciones de los proyectos que lleven a que la resolución del proyecto global no termine en éxito.

En cuánto a los posibles requisitos de **seguridad** suelen ser necesarias la aplicación de distintas configuraciones indicadas por la política de seguridad de las organizaciones.

Se recomienda en la medida de lo posible delegar el proceso de gestión de la configuración de la seguridad e interoperabilidad con fuentes de datos a un módulo ESB, o Enterprise Service Bus o Servicio de Bus Empresarial que permita la conexión de todos los componentes de la empresa a través de una plataforma genérica dónde se puede aplicar políticas de seguridad a todos los componentes de una forma escalable y común y habilitando todos los componentes necesarios para evitar problemas de interoperabilidad. En el último bloque de la memoria se indica cómo como habilitar un ESB a través de la creación de aplicaciones con MS BizTalk Server.

B. EXTRACT (EXTRACCIÓN)

Incluye el punto 2.

El proceso de extracción debe hacerse con un alto grado de estudio, ya que es la primera piedra en el camino que debemos franquear para poder obtener la información deseada.

Es importante conocer todas las estructuras de información que contienen las fuentes de datos para definir cómo recuperar la información necesaria. Por tanto la definición de la fase de extracción no basta simplemente con lanzar consultas a base de datos externas para recuperar datos, sino que debe de hacerse para realizar el proceso de extracción de acuerdo a varias de las cuestiones definidas anteriormente como son el ver si se accede a las fuentes de datos de producción o copias para evitar bajar el rendimiento de las fuentes de datos origen o ver durante cuánto tiempo se tiene abierta una ventana de consulta a las mismas.

Deberán de ejecutarse procesos de recuperación de información rápidos y que no tengan que repetirse por posibles fallos en la recuperación de la información, es decir, una vez que se va a la fuente del origen, debemos traernos todos los datos necesarios para a partir de aquí operar por cuenta propia.

Comúnmente se suele recuperar la información necesaria (incluso algunos datos extra) y cargar esa información en el denominado **Staging Area** o área de pruebas que será desde dónde comience la siguiente fase del proceso ETL. Por tanto, el proceso ETL no recupera los datos y directamente los formatea, sino que se trata de recuperar toda la información necesaria en el menor tiempo posible y almacenarla en el Staging Area.

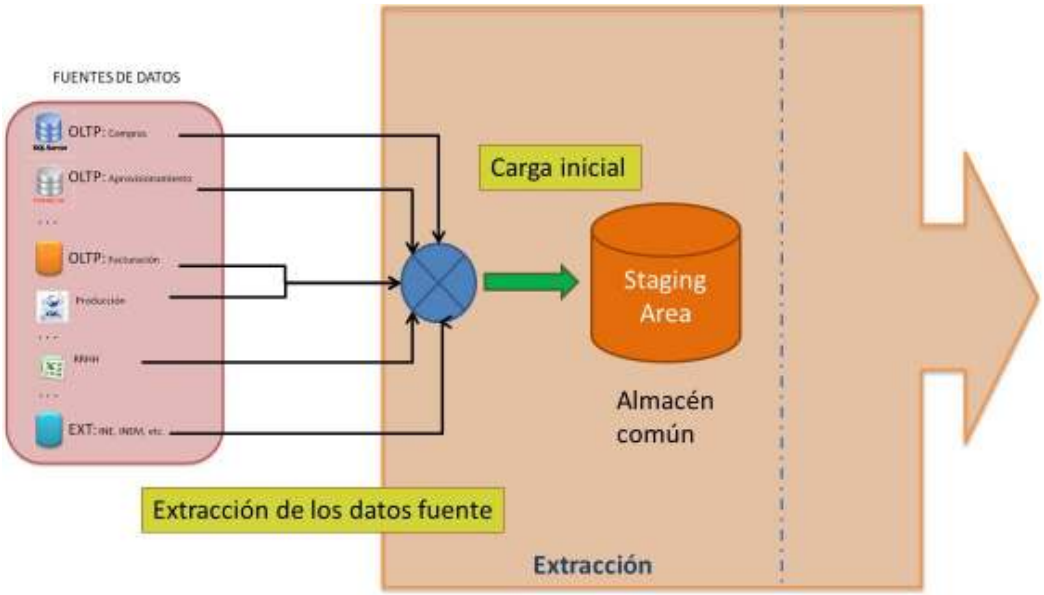

Figura 20. ETL Detallado. Extracción.

C. TRANSFORM (TRANSFORMACIÓN)

Incluye los puntos 3, 4, 5 y 6.

El proceso de transformación puede verse como una caja negra que recibe entradas de datos (inputs) y genera nuevos datos (outputs) que serán filtrados, transformados, formateados y estarán acorde con lo que se espera de ellos en el proceso de BI.

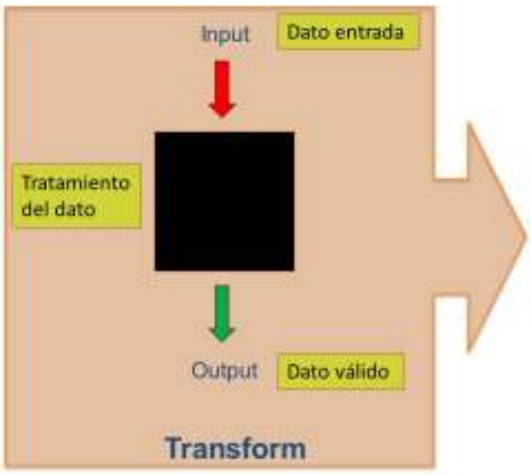

Figura 21. La caja negra la formarían a grandes rasgos las siguientes tareas:

- **GetData.**

Figura 22. ETL. Transformación, tarea Get Data.

Se trata de obtener los datos que van a formar parte de la transformación.

Generalmente los datos vienen del staging área.

Deberá estipularse si la obtención de los datos se puede realizar en lote o bien se hace de forma unitaria.

Sin ésta tarea no puede realizarse el resto de la transformación.

- **Tipo de transformación.**

En base al dato final que queramos obtener y el dato de entrada que tengamos deberemos consultar los metadatos (o tabla de metadatos) para que nos indiquen que tipo de transformación tenemos que realizar.

El tipo de transformación serán, entre otras muchas, de los siguientes tipos:

Dividir un dato para producir datos finales nuevos.

Concatenación de varios datos para conseguir un campo final.

Obtener nuevos valores calculados: p.ej. TotalVenta = cantidad * precio, etc.

Conversión entre tipos de dato: conversión de una cadena a un entero, etc.

Formateo de datos: p.ej. añadiendo/quitando prefijos/sufijos.

Recuperar el id de una tabla en base a una descripción: p.ej. Madrid = 28)

etc.

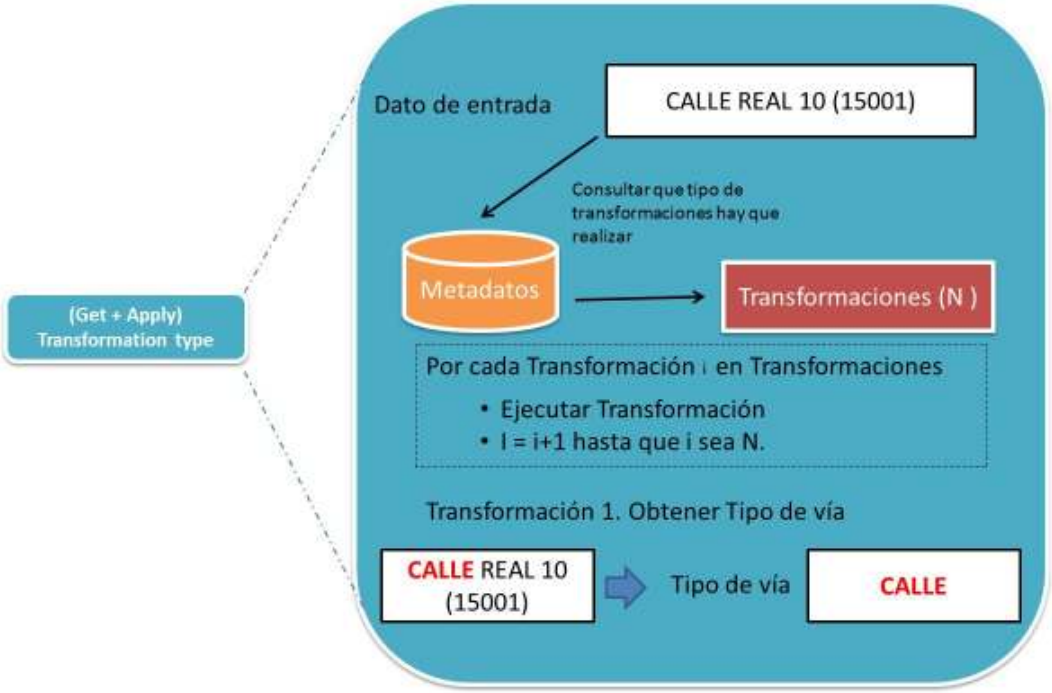

Figura 23. ETL. Transformación, ejemplo de tipo de transformación.

Una vez obtenido que realizar con el dato, aplicamos la modificación y procedemos a pasar el filtro de calidad sobre él.

Test de calidad de datos.

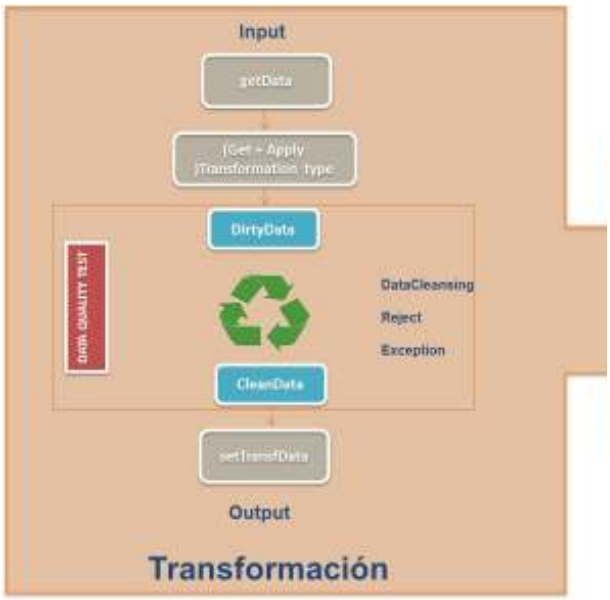

Figura 24. ETL, Transformación. Test de calidad de los datos.

Durante ésta tarea tenemos la opción de aplicar diferentes filtros de calidad a los datos.

Se consulta a la tabla de metadatos para conocer que filtros corresponden al juego de datos actual y en base a ellos se irán aplicando uno tras otro.

Uno de los filtros a realizar suele ser el llamado **DataCleanSing**

(que tiene diferentes definiciones) pero que suele ser como veremos a continuación.

Se suelen encontrar datos que debemos rechazar porque sean duplicados o no estén completos.

Puede haber datos a los que aplicar excepciones de formato y migrarlos con valores por defecto.

DataCleanSing

La limpieza de datos (Data Cleansing) debe considerarse como una tarea más compleja y con un alcance mayor que la de actualizar registros con información libre de errores.

Para realizar un proceso de limpieza exhaustivo, se requerirá una descomposición, análisis y posterior montaje del conjunto de datos que tratemos. Por ello se entiende el proceso como una sola tarea, completa y no cómo acciones individuales y puntuales pero que generalmente deben, al menos, tener las siguientes **tres fases**:

1. Detección y definición de la tipología de errores.
2. Búsqueda e identificación de los casos de error.
3. Corrección de errores detectados.

Cada una de las fases constituye un problema complejo en sí mismo, aunque sin un alto grado de desarrollo de las dos primeras fases, la tercera carece de sentido práctico, ya que si no somos capaces de encontrar los errores no podremos solucionarlos.

La mayoría de soluciones de data cleansing se centran exclusivamente en el análisis de la integridad de los datos para detectar errores. Esta tipología de análisis (parece más enfocada a bases de datos relacionales) es la operativa más sencilla en una tarea de limpieza de datos. Es importante saber que actualmente nuestra fuente de datos, staging área, será una base de

datos relacional y debemos realizar éste tipo de limpieza sobre ella para no llevar errores a la bases de datos OLAP.

Para un conjunto de datos (base de datos), el análisis de integridad incluirá algunas opciones más como son el análisis de integridad referencial, referencias, relación entre entidades, integridad por columna, etc. dónde se podría obtener con consultas SQL directamente sobre el conjunto de datos analizado.

La función de análisis de integridad de datos permite destapar un gran número de errores, so bien no es capaz de identificar errores más complejos.

Errores que involucran relaciones entre uno o varios campos, son, a menudo, más complicados de encontrar. Esta tipología de errores en datos requiere un análisis más profundo basado en métodos más complejos.

Digamos que un gran porcentaje de los datos (**99.5%**) se comportan de forma similar, por lo tanto el resto (**0.5%**) podrían ser candidatos a ser erróneos. Estos datos se nombran como ***Outsiders***.

El proceso para llegar a este conjunto de datos se compone de dos partes:

1. Identificación de tendencias de normalidad de los datos.
2. Identificación de tendencias de los datos outsiders o variaciones extrañas en sus valores.

En el mundo real, para llegar a determinar una tendencia de normalidad de los datos, rara vez basta con un único modelo de distribución. Este proceso suele basarse en varios métodos diferentes:

- **Modelo Estadístico:** Identifica los valores erróneos en base a medias, desviaciones estándar, rangos, etc. (basado en el teorema de Chebyshev).
- **Modelo de Clustering:** Son modelos **de Minería de Datos** (Datamining) que permiten agrupar conjuntos de datos con patrones comunes, determinados también por el propio algoritmo. Veremos más adelante información complementaria de minería de datos.
- **Modelo Basado en Patrones:** Búsqueda de valores que no conforman un patrón específico, bien manual o bien obtenido como combinación de técnicas matemáticas (particionado, clasificación y clustering). El patrón se define como el grupo de registros que cumplirán el mismo comportamiento según un % de confianza definido por el usuario.
- **Modelo de Reglas de Asociación:** Son reglas de asociación con altos intervalos de confianza que definen diferentes tipos de patrones. Como en el modelo anterior, los registros que no sigan estos patrones serán considerados **outliners**. Este modelo se recomienda cuando se trata con datos de diferentes tipos. Habitualmente se utilizan reglas de asociación ya definidas, como modelos estándar (basado en el modelo de

patrones) pero podría extenderse a otros tipos de asociación como las correlaciones estadísticas y otros.

Dentro de cada herramienta a utilizar o software base, se puede usar el data cleansing (suele ser un componente del mismo) para estandarizar información en base a patrones conocidos y definidos en el SW y tenemos la posibilidad de crear otros modelos diseñados por nosotros.

- **Marcar la transformación como finalizada**.

Una vez realizadas las tareas anteriores se deberá marcar los datos como limpios y formateados para su carga o como no válido para reportar de ello al usuario. Es importante conocer en qué puntos se pueden marcas los datos como corruptos para poder detectar errores en la fuente o simplemente para ver que se ha realizado de forma correcta el proceso de testado de los datos.

Por otro lado, hay que estudiar una política de excepciones en los datos que pudieran ser necesarios de migrar aunque el proceso de filtrado haya devuelto el dato como no válido, pero que su carga con datos aproximados o por defecto pueda generar datos que no hagan el juego resultando uno con una tasa de calidad menor.

Durante la fase de marcado se generará (opcional y recomendablemente) un informe sobre cómo se forma la carga de datos que procederá a incorporarse en la fase siguiente en la base de datos OLAP.

D. LOAD (CARGA)

La fase de carga es el momento en el cuál los datos de la fase anterior (transformación) son cargados en el sistema destino. Dependiendo de los requerimientos de la organización, este proceso puede abarcar una amplia variedad de acciones diferentes. [56]

En algunas bases de datos se sobrescribe la información antigua con los datos nuevos. Normalmente, un DW mantiene un histórico de los registros de manera que se pueda hacer una auditoría de los mismos y disponer de un rastro de toda la historia de un valor a lo largo del tiempo.

Existen dos formas básicas de desarrollar el proceso de carga:

1. **Acumulación simple:** La acumulación simple es la más sencilla y común, y consiste en realizar un resumen de todas las transacciones comprendidas en el período de tiempo seleccionado y transportar el resultado como una única transacción hacia el data warehouse, almacenando un valor calculado que consistirá típicamente en un sumatorio o un promedio de la magnitud considerada.

2. **Rolling:** El proceso de rolling por su parte, se aplica en los casos en que se opta por mantener varios niveles de granularidad. Para ello se almacena información resumida a distintos niveles, correspondientes a distintas agrupaciones de la unidad de tiempo o diferentes niveles jerárquicos en alguna o varias de las dimensiones de la magnitud almacenada (p.ej. totales diarios, totales semanales, totales mensuales, etc.)

La fase de carga interactúa directamente con la base de datos OLAP. Al realizar esta operación se aplicarán todas las restricciones y disparadores que se hayan definido en ésta durante la fase de transformación (valores únicos, integridad referencial, campos obligatorios, rangos de valores).

Estas restricciones y triggers (disparadores) contribuyen a que se garantice la calidad de los datos en el proceso ETL siempre que se haya tomado la importancia necesaria en la tarea de test de la calidad de los datos de la fase de transformación.

E. PROCESAMIENTO PARALELO

Actualmente el SW ETL se ha concentrado en el desarrollo de procesamiento en paralelo. Esto ha permitido desarrollar una serie de métodos para mejorar el rendimiento general de los procesos ETL cuando se trata de grandes volúmenes de datos. Existen principalmente 3 tipos de paralelismos que se pueden implementar en procesos ETL. [56]

1. **De datos:** Consiste en dividir un único archivo secuencial en pequeños archivos de datos para proporcionar acceso rápido.
2. **De segmentación (pipeline):** Permitir el funcionamiento simultáneo de varios componentes en el mismo flujo de datos. Un ejemplo de ello sería buscar un valor en el registro número 1 a la vez que se suman campos en el registro número 2.
3. **De componente:** Consiste en el funcionamiento simultáneo de múltiples procesos en diferentes flujos de datos, pertenecientes todos ellos a un único flujo de trabajo. Esto es posible cuando existen porciones dentro de un flujo de trabajo que son totalmente independientes entre ellas a nivel de flujo de datos.

Esos tres tipos de paralelismos no son excluyentes, sino que pueden ser combinados para realizar una misma operación ETL.

Una dificultad adicional es asegurar que los datos que se cargan sean relativamente consistentes. Las múltiples bases de datos de origen tienen diferentes dichos de actualización. En un sistema de ETL será necesario que se puedan detener ciertos datos hasta que todas las fuentes estén sincronizadas. Del mismo modo, cuando un almacén de datos tiene que ser actualizado con los contenidos en un sistema de origen, es necesario establecer puntos de sincronización y de actualización.

Es muy común trabajar sobre copias de las bases de datos para facilitar tareas de sincronización y no interferir en los niveles de rendimiento de las bases de datos de producción.

F. RIESGOS

Los procesos ETL pueden ser muy complejos. Un sistema ETL mal diseñado puede provocar importantes problemas operativos. [56]

En un sistema operacional el rango de valores de los datos o la calidad de éstos pueden no coincidir con las expectativas de los diseñadores a la hora de especificarse las reglas de validación o transformación. Es recomendable realizar un examen completo de la validez de los datos (**Data profiling**) del sistema de origen durante el análisis para identificar las condiciones necesarias para que los datos puedan ser tratados adecuadamente por las reglas de transformación especificadas. Esto conducirá a una modificación de las reglas de validación implementadas en el proceso ETL.

Normalmente los DW son alimentados de manera asíncrona desde distintas fuentes, que sirven a propósitos muy diferentes. El proceso ETL es clave para lograr que los datos extraídos asíncronamente de orígenes heterogéneos se integren finalmente en un entorno homogéneo.

La escalabilidad de un sistema de ETL durante su vida útil tiene que ser establecida durante el análisis. Esto incluye la comprensión de los volúmenes de datos que tendrán que ser procesados según los acuerdos de nivel de servicio - ANS (SLA: Service level agreement).

El tiempo disponible para realizar la extracción de los sistemas de origen podría cambiar, lo que implicaría que la misma cantidad de datos tendría que ser procesada en menos tiempo.

Algunos sistemas ETL son escalados para procesar varios terabytes de datos para actualizar un data warehouse que puede contener decenas de terabytes de datos. El aumento de los volúmenes de datos que pueden requerir estos sistemas pueden hacer que los lotes que se procesaban a diario pasen a procesarse en micro-lotes (varios al día) o incluso a la integración con colas de mensajes o a la captura de datos modificados (CDC: change data capture) en tiempo real para una transformación y actualización continua.

5. METADATOS.

Los metadatos o el repositorio o base de datos de los metadatos es uno de los componentes más importantes de un DW. Su importancia radica en el hecho de que todo el conocimiento sobre la creación de un DW es almacenado en los metadatos.

Aunque el rol que se imprime a los metadatos en algunos proyectos de DW no le hace ser un componente crítico, es obvio que al ser el componente sobre el que gira el proceso ETL para poder generar el DW se tiene que dedicar mucho tiempo y esfuerzo a construir un repositorio de metadatos bien diseñado y alineado con los propósitos de nuestro proyecto. [57]

En general, los metadatos son definidos como información sobre los datos, es decir, serán información sobre la estructura, contenido e interdependencias de los componentes del DW.

En un DW, los metadatos describen entre otros la siguiente información:

- Tipos de datos en el DW.
- Definición física y lógica de los datos.
- Consultas predefinidas.
- Reportes predefinidos.
- Reglas de validación y orientadas al tema.
- Definiciones de fuentes de datos.
- Rutinas de transformación.
- Secuencias de procesamiento.
- Información del usuario.
- Etc.

Los metadatos se refieren a todo elemento o tarea que define un objeto del DW, es decir, puede referirse desde a tablas, columnas, temas como a filtros o validaciones.

Los metadatos guían los procesos de ETL y sirven para que las herramientas de consulta y los generadores de informes o herramientas de reporting hagan su trabajo. En otras palabras, las operaciones de mantenimiento para poder aplicar ciertos cambios en el flujo del ETL o sobre las consultas y reporting, por lo general, se realizan sobre el repositorio de metadatos.

Clasificación de los Metadatos.

Por lo general, se suelen crear dos tipos de metadatos:

1. **Metadatos técnicos.**

Los desarrolladores y administradores de un DW se interesan principalmente en los metadatos a un nivel de implementación técnica. Por tanto, los desarrolladores de SW usan los metadatos técnicos para conocer las definiciones físicas y lógicas de los datos para poder diseñas e implementar aplicaciones, mientras que los administradores accederán a los metadatos

técnicos para ejecutar las tareas de administración como gestión de objetos y usuarios del DW, afinamiento de la base de datos etc.

2. **Metadatos semánticos (orientados al tema).**

Los usuarios finales, por ejemplo los analistas o gerentes, que no tienen por qué estar familiarizados con los formatos de descripción del DW como los archivos SQL de la base de datos, están interesados en saber y entender la semántica orientada al tema y por lo tanto necesitan representaciones semánticamente ricas sobre la estructura y los contenidos a explotar del DW.

- **Gestión de los Metadatos.**

A menudo, un repositorio de metadatos es usado para almacenar y gestionar todos los metadatos asociados a un DW. El repositorio permite compartir los metadatos entre las diversas herramientas y procesos utilizados para diseñar, establecer, usar, operar y administrar un DW.

El beneficio de gestionar los metadatos técnicos de un DW es similar al beneficio que se obtiene de gestionar los metadatos en un ambiente de procesamiento de transacciones OLTP.

Los datos técnicos integrados y consistentes crean un ambiente de desarrollo más eficiente para el staff técnico responsable de construir y mantener los sistemas de procesamiento de decisiones.

Un beneficio adicional en el ambiente DW es la habilidad de rastrear como cambian los metadatos a lo largo del tiempo.

Por su parte, los beneficios obtenidos gracias a la gestión de los metadatos semánticos son exhaustivos de un ambiente de procesamiento de decisiones y son clave para explotar el valor de un DW una vez que ha sido puesto en operación.

- **Arquitectura de los Metadatos.**

En todo proyecto de BI que incluya la creación de un DM o una estructura DW, el proceso de diseño de los metadatos debería de incluirse alineado con la estructura del mismo. El hecho de desarrollar una arquitectura de metadatos al inicio del proyecto ayuda a tener una visión de futuro, guiando al equipo de trabajo a través de las diferentes fases de desarrollo del DW.

Los usuarios de un DW deberían contar no sólo con metadatos que sean precisos, sino que sean contextuales, ya que de otra manera se podría obtener información engañosa y ambigua que puede conducir a decisiones equivocadas.

La arquitectura de los metadatos en un proyecto de BI debería ser un punto obligatorio y bien planificado que se incluya en el proceso de diseño de la arquitectura del DW.

Nuestro repositorio de Metadatos puede estar centralizado o distribuido dependiendo de las necesidades y requerimientos de la organización. Se puede identificar al repositorio como un grupo de base de datos compuestos por objetos y datos relacionales.

El gestionador de Metadatos es el componente esencial de la arquitectura de metadatos e idealmente consistirá en los siguientes componentes.

- **Captura de Metadatos:** Captura inicial de los metadatos desde diversas fuentes.
- **Sincronizador de Metadatos:** Procesos para mantener los metadatos actualizados.
- **Motor de búsqueda de Metadatos:** Sería el componente Front-End para los usuarios que requieren consultar y acceder a los metadatos
- **Gestionador de resultados de los Metadatos:** Procesaría los resultados de la búsqueda de metadatos y permitiría al usuario hacer una selección apropiada.
- **Notificador de Metadatos:** Notificaría a los suscriptores sobre cualquier cambio en el contenido de los metadatos dependiendo del perfil del usuario.
- **Disparador de consultas de Metadatos:** Dispararía una herramienta de consulta apropiada para obtener datos desde un DW o cualquier otra fuente basada en la selección hecha por el usuario en el gestionador de resultados de metadatos.

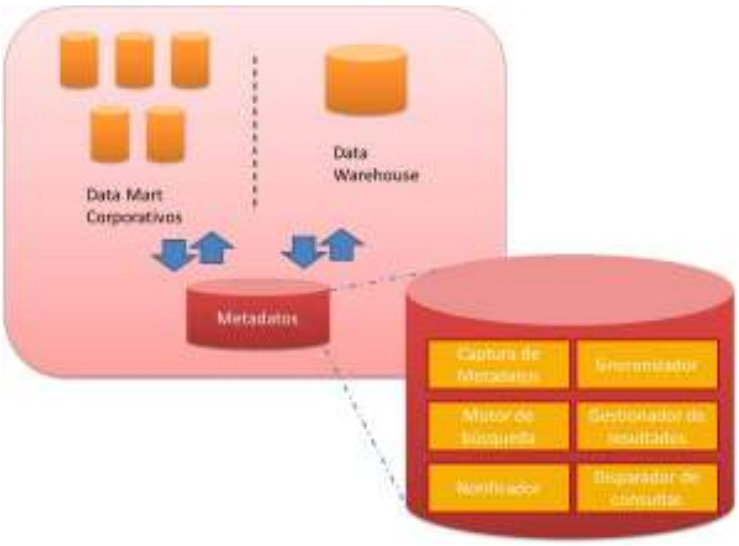

Figura 25. Metadatos.

Actualmente se suelen almacenar los Metadatos en una base de datos relacional con un diseño orientado a objetos. La base de datos se usa como una caja negra que sirve para obtener información y no como una extensión de las aplicaciones comerciales.

- **Conclusiones.**

Los metadatos son un componente fundamental de un DW, ya que estos se encargan de guiar los procesos de ETL.

La arquitectura de los metadatos debería ser una parte integral y bien planificada de toda la arquitectura de un DW en su conjunto, ya que de ello dependerá la buena gestión de los metadatos.

Una representación explícita de los metadatos da soporte a los usuarios orientados al tema en las tareas de navegación, consultas ad-hoc y datamining.

Se entiende que en el futuro, los metadatos pueden adquirir mucha importancia dada la unión complementaria que puede asociarse a los DW con tecnología Web. Se podría proponer una serie de browsers de metadatos como punto de acceso de la información orientada al tema. Por tanto los metadatos serían algo crítico en un sistema de BI con orientación Web.

6. DATA MINING.

Para hablar de Datamining se suelen hacer introducción sobre el KDD o proceso de extracción del conocimiento, que se refiere al proceso no-trivial de descubrir conocimiento e información potencialmente útil dentro de los datos contenidos en algún repositorio de información.

Sobre el KDD se analizan 5 etapas que serían:

1. Selección de datos.
2. Preprocesamiento.
3. Transformación.
4. Datamining.
5. Interpretación y evaluación.

Dado que estamos hablando sobre el mundo BI y concretamente sobre el desarrollo de DW, fácilmente vemos que el proceso o la etapa del Datamining encajarían en nuestro modelo en la etapa de Explotación, justo después de definir el almacenamiento y proveer una capa middleware durante la etapa de acceso para llegar a los datos.

El **Data Mining o Minería de Datos**, es **la extracción de información oculta y predecible** de grandes bases de datos (en nuestro caso los DW o DM).

Es una poderosa tecnología con gran potencial para ayudar a las compañías a concentrarse en la información más importante de su DW. Las herramientas de Data Mining predicen futuras tendencias y comportamientos, permitiendo en los negocios tomar decisiones proactivas y conducidas por un conocimiento acabado de la información.

Los análisis prospectivos automatizados ofrecidos por un producto así van más allá de los eventos pasados provistos por herramientas retrospectivas típicas de sistemas de soporte de decisión.

Las herramientas de Data Mining pueden responder a preguntas de negocios que tradicionalmente consumen demasiado tiempo para poder ser resueltas y a los cuales los usuarios de esta información casi no están dispuestos a aceptar. Estas herramientas **exploran las bases de datos en busca de patrones** ocultos, encontrando información predecible que un experto no puede llegar a encontrar porque se encuentra fuera de sus expectativas.

Los **objetivos** del Data Mining son:

- **Predicción:** p.ej. que comparan los clientes bajo determinados descuentos.
- **Identificación:** p.ej. secuencias
- **Clasificación:** p.ej. clientes que buscan descuentos, fieles, ocasionales, etc.
- **Optimización:** ante recursos limitados de tiempo, espacio, presupuesto, etc.

Las técnicas de Data Mining pueden ser implementadas rápidamente en plataformas de software base. Se pueden ser integrar con nuevos productos y sistemas ya que los datos se recuperan on-line.

Una vez que las herramientas de Data Mining han sido implementadas en modelo cliente-servidor de alta rendimiento o de procesamiento paralelo, se pueden analizar grandes bases de datos para conseguir respuesta a preguntas tales como:

"¿Cuáles clientes tienen más probabilidad de responder al próximo mail promocional, y por qué? y presentar los resultados en formas de tablas, con gráficos, reportes, texto, HTML, etc.

El nombre de Data Mining deriva de las similitudes entre buscar información valiosa de negocios en grandes bases de datos y minar una montaña para encontrar metales valiosos. Ambos procesos requieren examinar una inmensa cantidad de material, o investigar inteligentemente hasta encontrar exactamente donde residen los valores.

Dadas bases de datos de suficiente tamaño y calidad como los DW, la tecnología de Data Mining puede generar nuevas oportunidades de negocios al proveer las siguientes **capacidades**:

- **Predicción automatizada de tendencias y comportamientos.**

El Data Mining automatiza el proceso de encontrar información predecible en grandes bases de datos. Preguntas que tradicionalmente requerían un intenso análisis manual, ahora pueden ser contestadas directa y rápidamente desde los datos.

Un típico ejemplo de problema predecible es el marketing apuntado a objetivos (targeted marketing). Data Mining usa datos en mail promocionales anteriores para identificar posibles objetivos para maximizar los resultados de la inversión en futuros mail.

Otros problemas predecibles incluyen pronósticos de problemas financieros futuros y otras formas de incumplimiento, e identificar segmentos de población que probablemente respondan similarmente a eventos dados.

- **Descubrimiento automatizado de modelos previamente desconocidos.**

Las herramientas de Data Mining barren las bases de datos e identifican modelos en un sólo paso. Otros problemas de descubrimiento de modelos incluyen detectar transacciones fraudulentas de tarjetas de créditos e identificar datos anormales que pueden representar errores de tipado en la carga de datos.

Las técnicas de Data Mining pueden producir los **beneficios de automatización** en las plataformas de hardware y software existentes y pueden ser implementadas en sistemas nuevos a medida que las plataformas existentes se actualicen y nuevos productos sean desarrollados.

Cuando las herramientas de Data Mining son implementadas en sistemas de procesamiento paralelo de alta rendimiento, pueden analizar grandes bases de datos en minutos.

Procesamiento más rápido significa que los usuarios pueden automáticamente experimentar con más modelos para entender datos complejos.

Alta velocidad hace que sea práctico para los usuarios analizar inmensas cantidades de datos.

Por tanto, **Grandes bases de datos, a su vez, producen mejores predicciones**.

Las bases de datos pueden ser grandes tanto en profundidad como en ancho:

- **Más columnas.** Los analistas muchas veces deben limitar el número de variables a examinar cuando realizan análisis manuales debido a limitaciones de tiempo. Sin embargo, variables que son descartadas porque parecen sin importancia pueden proveer información acerca de modelos desconocidos. Un Data Mining de alto rendimiento permite a los usuarios explorar toda la base de datos, sin preseleccionar un subconjunto de variables.

- **Más filas.** Muestras mayores producen menos errores de estimación y desvíos, y permite a los usuarios hacer inferencias acerca de pequeños pero importantes segmentos de población.

Las **técnicas** más comúnmente usadas en **Data Mining** son:

- **Redes neuronales artificiales**: Modelos predeciblesno lineales que aprenden a través del entrenamiento y semejan la estructura de una red neuronal biológica.

- **Arboles de decisión:** Estructuras de forma de árbol que representan conjuntos de decisiones. Estas decisiones generan reglas para la clasificación de un conjunto de datos. Métodos específicos de árboles de decisión incluyen Arboles de Clasificación y Regresión (CART: Classification And Regression Tree) y Detección de Interacción Automática de Chi Cuadrado (CHAI: Chi Square Automatic Interaction Detection)

- **Algoritmos genéticos:** técnicas de optimización que usan procesos tales como combinaciones genéticas, mutaciones y selección natural en un diseño basado en los conceptos de evolución.

- **Método del vecino más cercano:** una técnica que clasifica cada registro en un conjunto de datos basado en una combinación de las clases del/de los k registro (s) más similar/es a él en un conjunto de datos históricos (donde k ? 1). Algunas veces se llama la técnica del vecino k-más cercano.

- **Regla de inducción:** la extracción de reglas *if-then* de datos basados en significado estadístico.

Muchas de estas tecnologías han estado en uso por más de una década en herramientas de análisis especializadas que trabajan con volúmenes de datos relativamente pequeños. Estas capacidades están ahora se integran directamente con herramientas OLAP y de DW.

7. DM: DATA MART.

Una **Data Mart** es un almacén de datos limitado a un área concreta de la organización. Hay diversas definiciones que identifican al Data Warehouse como un almacén centralizado que alimenta una serie de data marts.

El enfoque de un data mart sería el de complementar los requerimientos específicos de un determinado grupo de usuario en términos de análisis, contenido, presentación y facilidad de uso.

Los usuarios de un data mart pueden tener datos que se presentan en términos que le son familiares ya que irán orientados a su área de negocio en un grado más específico que el que pudiera encontrarse directamente en el DW.

Los **data marts se pueden generar obteniendo datos de un DW corporativo o pueden ser creados independientemente** de fuentes de datos externas directamente con un proceso similar al de la creación de un DW.

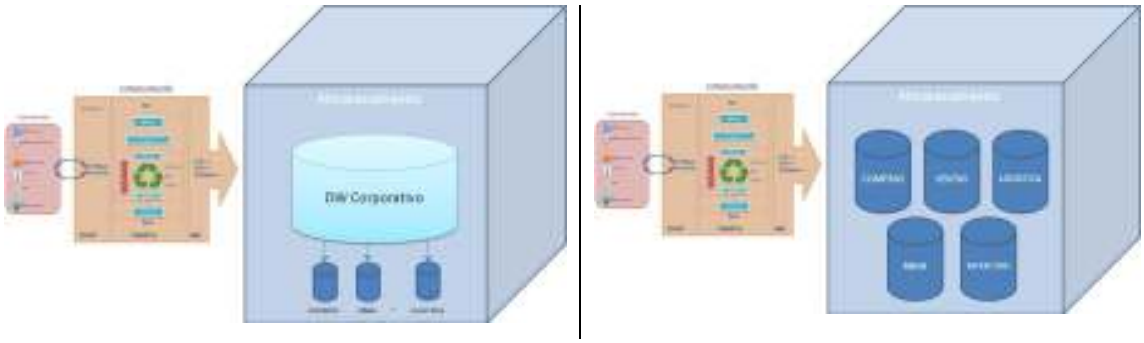

Figura 26. Tipos de Data Mart.

Los **data marts independientes** no son lo más adecuado, no son recomendables, ya que se originan islas de información, siendo esto precisamente lo que los almacenes de datos intentan evitar.

Existen también otro tipo de **data mart** que podemos llamar **personales,** que son subconjuntos de datos extraídos de data marts departamentales o de unidades de negocio específicas e incluso DW que responden a los requerimientos de un único usuario o grupo reducido de ellos.

Los usuarios individuales se suelen suscribir a centros de distribución que periódicamente actualizan sus data marts a medida que también se actualiza el almacén de datos.

Puede ser complejo justificar la creación de un data mart viendo la potencia global que produce un DW y entender que se puede afinar un poco más el proceso de explotación del DW para que no fuera necesario el DM, pero vamos a explicar una serie de **razones** por las cuáles puede ser muy conveniente construir un DM.

1. Fácil acceso a los datos que se necesitan frecuentemente.
2. Se crea una vista colectiva para el grupo de usuarios al que va dirigido.
3. Mejora el tiempo de respuesta del usuario final.
4. Partiendo de un DW es fácil de crear o extender.
5. Creado independientemente su costo siempre será inferior al de un DW completo.
6. Los usuarios potenciales se identifican mejor que para un DW completo y por tanto la complejidad del negocio estará más acotada.

Por otro lado conviene prevenir de malos entendidos a la hora de comprender el concepto de data mart. Al hablar de data marts, es inevitable la comparación con los DW y al final es fácil que se acabe diciendo que los DM son DW en pequeño, y aunque en cierto modo puede ser así, la idea representa errores sobre su implementación y funcionamiento que vamos a enumerar.

1. Un DM no siempre es más simple de implementar que un DW ya que el proceso es muy similar y deben proporcionar las mismas funcionalidades. La ventaja se obtiene cuándo el DW ya existe de antemano.

2. Los DM no son pequeños conjuntos de datos que propician una menor necesidad de recursos ya que por lo general los sistemas encargados de explotar un DM o un DW suelen consumir los mismos recursos.

3. Las consultas no son más rápidas porque el volumen de datos es menor. Qué el volumen de datos sea menor se debe a que no se almacenan todos los datos de la empresa, pero sí se tienen todos los datos de un determinado sector de la empresa, por lo que una consulta sobre dicho sector tarda lo mismo si se hace sobre un DM que sobre un DW.

4. El proceso de actualización de un DM no añade tiempos innecesarios. Actualizar el DM desde el DW cuesta menos que actualizar el DW desde sus fuentes primarias (ya que

los formatos de datos son o suelen ser idénticos), dónde es necesario realizar una serie de pasos previos a la carga en el DW (ETL, etc.).

8. DSS: DECISION SUPPORT SYSTEM.

Un DSS, es un sistema informático que utiliza información y modelos matemáticos para ayudar a los usuarios TIC de las organizaciones a tomar decisiones empresariales adecuadas según las condiciones del mercado y la situación interna de la compañía.

El término **DSS** es el acrónimo de "**Decision Support System**" en inglés**Sistema de Soporte a Decisiones** es castellano, es decir, se refiere a los sistemas para el apoyo a la toma de decisiones. Se trata de un término que se popularizó a mediados de los 90 pero que sin embargo ha caído en desuso con la misma facilidad con la que se popularizó.

Como la propia palabra indica, un DSS es un sistema, es decir, no es una herramienta, ni un software, ni un concepto ni una metodología. Es un sistema y hay que entenderlo como tal. Un **sistema** es un conjunto de componentes relacionados entre sí que contribuyen a un determinado objetivo. Un sistema se caracteriza habitualmente a través de sus entradas y salidas, y su resultado se ve afectado por las condiciones externas al sistema, y por parámetros internos del mismo.

Pues bien, exactamente esta definición es aplicable para referirse al término DSS. Un DSS es un sistema informático que utiliza información y modelos matemáticos para ayudar a los trabajadores de la información a tomar decisiones empresariales adecuadas según las condiciones del mercado y la situación interna de la compañía.

Como sabemos, en el día a día de las empresas se toman continuamente decisiones de muy distinto tipo, y de manera muy diferente. Existen decisiones que por su naturaleza se toman de manera racional y absolutamente informada, y otras que se toman de manera menos sistemática, casi por instinto o corazonadas.

Como veremos a continuación los DSS no están ni mucho menos acotadas a una solución normalizada y mi elección dependería del tipo de decisiones que queramos obtener, el SW de mercado y el conocimiento de que dispongan los usuarios que van a tomar las decisiones.

Los actuales DSS utilizan metodologías OLAP, y ofrecen un **soporte pasivo** a la toma de decisiones. Es decir, los sistemas DSS actuales ayudan a la toma de decisiones proporcionando información confiable y actualizada, pero raramente aportan valor añadido a la información y decisión resultante.

Un **soporte activo** a la toma de decisiones requiere modelos matemáticos y estadísticos avanzados que descubran patrones ocultos en la información (para diseñar mejores campañas de marketing, para optimizar una cadena de suministro, para orientar mejor los productos a mercados específicos, etc.), y todo eso se está haciendo todavía muy poco, y de manera poco estructurada.

Podemos clasificar los **tipos de decisiones empresariales según el alcance** que puede tomar una decisión. Merece la pena señalar que esta clasificación no dice nada sobre la importancia de las decisiones ya que todas ellas son importantes y necesarias.

Una mala decisión operativa puede costar millones (del mismo que una buena decisión puede suponer suculentos beneficios)aunquetambién existen decisiones "estratégicas" que resultan ser irrelevantes desde el punto de vista económico.

La clasificación puede realizarse de la siguiente forma:

- **Decisiones estratégicas.**

Son aquellas que afectan a toda la empresa (o a una buena parte de la misma) durante un largo periodo de tiempo. Influyen, por lo tanto, en los objetivos generales de la empresa y en su modelo de negocio.

Estas decisiones son tomadas por los máximos responsables de las compañías (CEO, presidentes, directores generales, comités de dirección, etc.).

- **Decisiones tácticas.**

Afectan únicamente a parte de la empresa, o a parte de sus procesos, y generalmente se toman desde un solo departamento (o de unos pocos). Tienen un impacto relevante a medio plazo (1 o 2 años, como máximo), y son tomadas por cargos intermedios (jefes de departamento, gerentes, etc.)

- **Decisiones operativas.**

Afectan a actividades específicas, con un alcance muy claro, y su efecto es inmediato o muy limitado en el tiempo. Estas decisiones son responsabilidad de los niveles bajos de la jerarquía empresarial (jefes de equipo, encargados de área, dependientes, etc.)

Por otro lado, se suelen clasificar las **decisiones empresariales en lo referente a su naturaleza**, dónde podemos distinguir los siguientes tres tipos:

- **Decisiones estructuradas.**

  En este caso, las variables que afectan a la decisión son perfectamente conocidas, y en muchos casos el proceso de decisión puede representarse mediante un diagrama de flujo, e implementarse mediante un algoritmo. En casos extremos, ni siquiera es necesaria la intervención humana, aunque no es lo habitual.

- **Decisiones desestructuradas.**

  Son aquellas decisiones en la que no es posible diseñar un "flujo de decisión" en detalle, no es evidente que modelo se debe aplicar, ni cómo se debe diseñar el proceso o al menos con qué criterios decidir. Suelen ser decisiones que se toman

ante eventos inesperados o que ocurren muy esporádicamente. En estos casos, evidentemente, la intervención humana es insustituible.

- **Decisiones semi-estructuradas.**

Es el caso intermedio. En cierto sentido, serían todas o casi todas las decisiones, ya que se encuentran en algún punto intermedio entre los dos extremos descritos anteriormente. En este caso, algunos pasos del proceso de decisión están claros y pueden definirse razonablemente, aunque existen otros aspectos inciertos que es necesario valorar.

Combinando estas tres clasificaciones, nos aparecen 9 clases de decisiones. En función de lo ambicioso de nuestro DSS, deberíamos tratar de cubrir el máximo número de estos casos, y cómo veremos a continuación todavía nos quedan muchas lagunas en él proceso.

Si por el contrario nos movemos en el entorno SW según la entrada de los programas que se pueden manejar y los modelos de decisión que pueden soportar podemos obtener las siguientes categorías.

- **DSS dirigidos por la comunicación.**

Disponen de soporte para varias personas en una misma tarea compartida. Ej.: chats y mensajería instantánea, sistemas de colaboración, etc.

Podemos encontrar el SW ***"Facilitador"*** con licencia Open-Source en http://facilitator.sourceforge.nete incluso "***MS Office Groove"*** con licencia comercial e integrada con la suite de Office.

- **DSS dirigidos por datos.**

Destacan el acceso y la manipulación de series temporales de datos internos de la organización, son utilizados para consultar una base de datos o data Warehouse variable el tiempo.

P. ej.: sistemas de información geográfica, el cuál puede ser usado para representar datos geográficamente dependientes usando mapas.

Podemos encontrar el SW ***"MicroStrategy"*** con licencia comercial, que está basado en el BI e incluye un buen componente DSS de datos. E incluso "***Sybase"*** o "***SAP"*** pueden entenderse como dirigidos por datos, aunque sobretodo SAP puede ir más dirigido por modelos.

- **DSS dirigidos por Documentos.**

Gestionan, recuperan y manipulan la información no estructurada en variedad de formatos electrónicos y están dirigidos a un gran número de usuarios.

El propósito de estos DSS es buscar en páginas webs y encontrar documentos.

P. ej.: Distintos software de análisis de textos o Wikis (sitio Web cuyas páginas Web pueden ser editadas por múltiples voluntarios a través de la Web).

- **DSS dirigidos por el Conocimiento.**

Proporcionan experiencia acumulada en forma de hechos, normas, procedimientos, o en estructuras similares especializados para la resolución de problemas. Son utilizados esencialmente para proveer consejos a las gerencias sobre la selección de productos o servicios.

P. ej.: procesar enormes volúmenes de datos, identificar patrones ocultos y presentar recomendaciones basadas en esos patrones.

- **DSS dirigidos por Modelos.**

Centrado en el acceso y manipulación de un modelo estadístico, financiero, de optimización o simulación.

P. ej.: predecir los cambios en procesos de negocios, utilizando datos del pasado para responder a preguntas complejas del estilo (what if) “que-pasa-si” a los usuarios en tomas de decisiones.

Podemos encontrar el SW ***“SAP”*** con licencia comercial o ***“EGADSS”*** con licencia Open-Source en http://dicodess.sourceforge.net

Figura 27. Pantallas DSS.

9. EIS: EXECUTIVE INFORMATION SYSTEM.

Un **EIS** (SIE) acrónimo de **Executive Information System** en inglés o Sistema de Información para Ejecutivos o **Sistema de Información Ejecutiva** en castellano, es una herramienta software, basada en un DSS, que provee a los gerentes de un acceso sencillo a información interna y externa de su compañía, y que es relevante para sus factores clave de éxito.

La finalidad principal es que el ejecutivo tenga a su disposición un panorama completo del estado de los indicadores de negocio que le afectan al instante, manteniendo también la posibilidad de analizar con detalle aquellos que no estén cumpliendo con las expectativas establecidas, para determinar el plan de acción más adecuado.

De forma más práctica, se puede definir un EIS como una aplicación con soporte de querying y reporting sobre las diferentes áreas de negocio, de forma consolidada, para facilitar la monitorización de la organización o área específico.

El EIS se caracteriza por ofrecer al ejecutivo un acceso rápido y efectivo a la al DW o DM, utilizando interfaces gráficas visuales e intuitivas, con un interfaz gráfico que permite que los usuarios no sean técnicos.

Suele incluir alertas e informes basados en excepción, así como históricos y análisis de tendencias. También es frecuente que permita el envío por diferentes canales (correo, sistemas de ficheros, etc.) de los informes más relevantes.

A través de esta solución se puede contar con un resumen del comportamiento de una organización o área específica, y poder compararla a través del tiempo. Es posible, además, ajustar la visión de la información a la teoría de Balanced Scorecard o **Cuadro de Mando Integral** o bien a cualquier modelo estratégico de indicadores que maneje la compañía.

Existen muchas soluciones de SW que incluyen componentes EIS de productividad contrastada como ***son "Microsoft Business Intelligence"***, ***"Oracle Business Intelligence Suite"*** o ***"i68 Business Intelligence"*** con licencia comercial o ***"Pentaho"*** con licencia Open-Source.

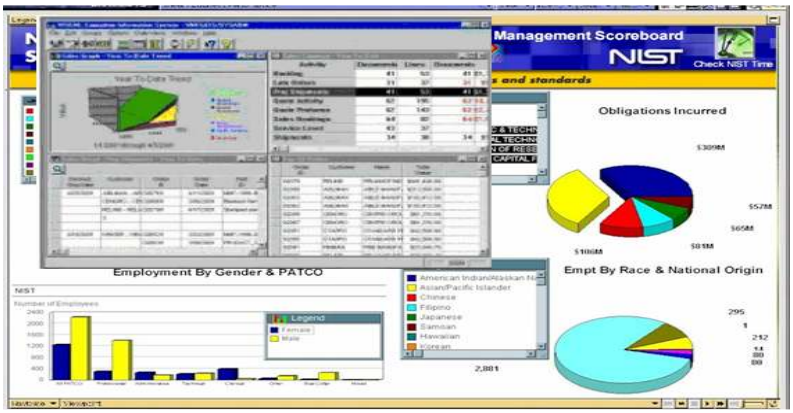

Figura 28. Pantallas EIS

10. CMI: CUADRO DE MANDO INTEGRAL.

El **Cuadro de Mando Integral (CMI),** también conocido como ***Balanced Scorecard (BSC) o dashboard***, es una herramienta de control empresarial que permite establecer y monitorizar los objetivos de una empresa y de sus diferentes áreas o unidades.

También se puede considerar como una aplicación que ayuda a una compañía a expresar los objetivos e iniciativas necesarias para cumplir con su estrategia, mostrando de forma continuada cuándo la empresa y los empleados alcanzan los resultados definidos en su plan estratégico.

El Cuadro de Mando Integral se diferencia de otras herramientas de Business Intelligence, como los Sistemas de Soporte a la Decisión (DSS) o los Sistemas de Información Ejecutiva (EIS), en que está más **orientado al seguimiento de indicadores** que al análisis minucioso de información.

Por otro lado, es muy común que un CMI sea **controlado por la dirección general** de una compañía, frente a otras herramientas de Business Intelligence más enfocadas a la dirección departamental. El CMI requiere, por tanto, que los directivos analicen el mercado y la estrategia para construir un modelo de negocio que refleje las interrelaciones entre los diferentes componentes de la empresa (plan estratégico).

Una vez que lo han construido, los responsables de la organización utilizan este modelo como mapa para seleccionar los indicadores del CMI.

Tipos de Cuadro de Mando

- **CMO: Cuadro de Mando Operativo**

Es una herramienta de control enfocada al seguimiento de variables operativas, es decir, variables pertenecientes a áreas o departamentos específicos de la empresa. La periodicidad de los CMO puede ser diaria, semanal o mensual, y está centrada en indicadores que generalmente representan procesos, por lo que su implantación y puesta en marcha es más sencilla y rápida. Un CMO debería estar siempre ligado a un DSS (Sistema de Soporte a Decisiones) para indagar en profundidad sobre los datos.

- **CMI: Cuadro de Mando Integral.**

Representa la ejecución de la estrategia de una compañía desde el punto de vista de la Dirección General.

Existen diferentes tipos de cuadros de mando integral, si bien los más utilizados son los que se basan en la metodología de Kaplan y Norton. Las principales características de esta

metodología son que utilizan tanto indicadores financieros como no financieros, y que los objetivos estratégicos se organizan en **cuatro áreas** o perspectivas:

1. **Financiera**.

   Incorpora la visión de los accionistas y mide la creación de valor de la empresa. Responde a la pregunta: ¿Qué indicadores tienen que ir bien para que los esfuerzos de la empresa realmente se transformen en valor? Esta perspectiva valora uno de los objetivos más relevantes de organizaciones con ánimo de lucro, que es, precisamente, crear valor para la sociedad.

2. **Cliente.**

   Refleja el posicionamiento de la empresa en el mercado o, más concretamente, en los segmentos de mercado donde quiere competir. Por ejemplo, si una empresa sigue una estrategia de costes es muy posible que la clave de su éxito dependa de una cuota de mercado alta y unos precios más bajos que la competencia. Dos indicadores que reflejan este posicionamiento son la cuota de mercado y un índice que compare los precios de la empresa con los de la competencia.

3. **Interna.**

   Recoge indicadores de procesos internos que son críticos para el posicionamiento en el mercado y para llevar la estrategia a buen puerto. En el caso de la empresa que compite en coste, posiblemente los indicadores de productividad, calidad e innovación de procesos sean importantes. El éxito en estas dimensiones no sólo afecta a la perspectiva interna, sino también a la financiera, por el impacto que tienen sobre las rúbricas de gasto.

4. **Aprendizaje/crecimiento**.

   Para cualquier estrategia, los recursos materiales y las personas son la clave del éxito. Pero sin un modelo de negocio apropiado, muchas veces es difícil apreciar la importancia de invertir, y en épocas de crisis lo primero que se recorta es precisamente la fuente primaria de creación de valor: se recortan inversiones en la mejora y el desarrollo de los recursos.

A continuación mostramos unos ejemplos de CMI, en el primero vemos un esquema de como segmentar nuestros indicadores dentro de cada área.

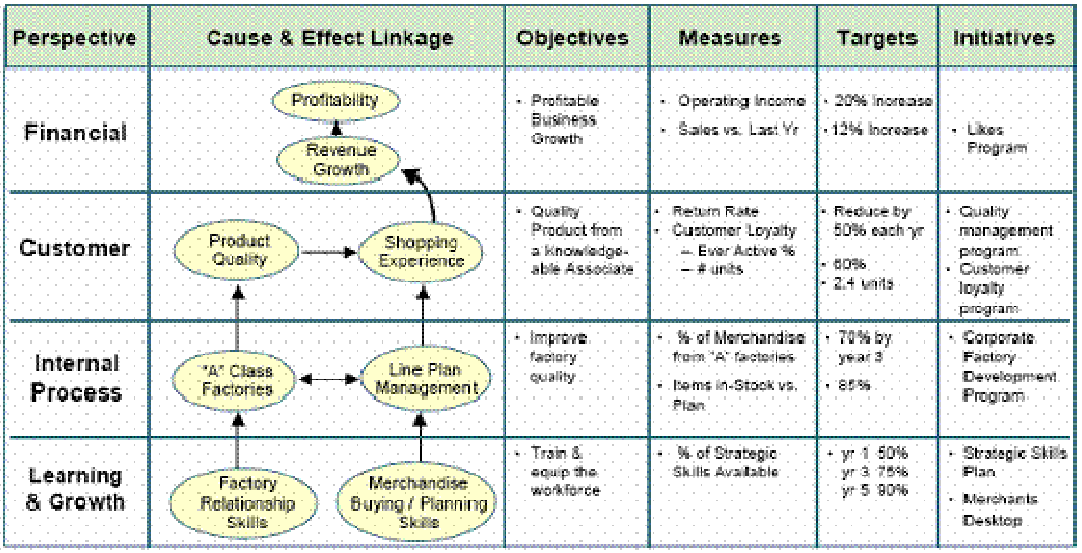

Figura 29. Ejemplos CM (1).

En el siguiente ejemplo podemos ver la definición de una sería de indicadores para el área de Aprendizaje y crecimiento.

PERSPECTIVA DE APRENDIZAJE Y CRECIMIENTO				
Factores-clave	**Indicadores**	**Alarmas**		
		R	**A**	**V**
Formación y capacitación	• Empleados formados / Total de empleados • Promociones / Puestos de trabajo (7)	< 0.4 < 0.3	0.4 a 0.5 0.3 a 0.4	> 0.5 > 0.4
Incentivos	• Premios • Salario mínimo / Salario máximo	A definir < 0.3	A definir 0.3	A definir > 0.3
Motivación	• Sugerencias por empleado • Índice de motivación	A definir < 0.4	A definir 0.4	A definir > 0.4
Productividad	•1-Δ de costes salariales / Δ de ventas netas	< 0.3	0.3 a 0.4	> 0.3
Tecnología	• Inversión en Software (8) • Investigadores / Empleados • Inversión 1+D / Gastos totales	< 80.000€ < 0.1 < 0.25	80.000€ 0.1 0.25	> 80.000€ > 0.1 > 0.25
Nuevos productos	• Productos nuevos / Productos totales • % de ventas nuevos productos por u. de tiempo	< 0.08 A definir	0.08 A definir	> 0.08 A definir

Figura 30. Ejemplos CMI (2).

Es muy común realizar diagramas de balance con los CMI para facilitar la comprensión de los informes.

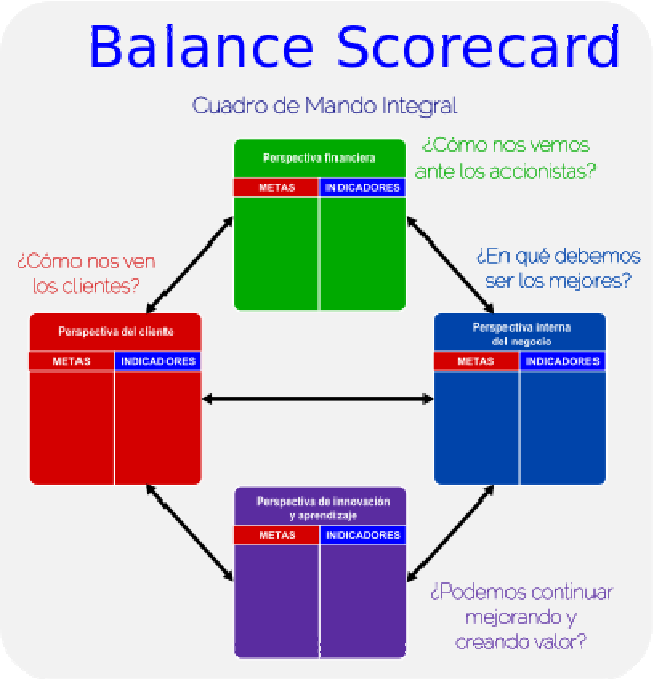

Figura 31. Ejemplos CMI (3).

Pese a que estas cuatro son las perspectivas más genéricas, no son obligatorias. Por ejemplo, una empresa de fabricación de ropa deportiva tiene, además de la perspectiva de clientes, una perspectiva de consumidores. Para esta empresa son tan importantes sus distribuidores como sus clientes finales.

Una vez que se tienen claros los objetivos de cada perspectiva, es necesario **definir los indicadores** que se utilizan para realizar su seguimiento. Para ello, debemos tener en cuenta varios criterios: el primero es que el número de indicadores no supere los siete por perspectiva, y si son menos, mejor. La razón es que demasiados indicadores difuminan el mensaje que comunica el CMI y, como resultado, los esfuerzos se dispersan intentando perseguir demasiados objetivos al mismo tiempo. Puede ser recomendable durante el diseño empezar con una lista más extensa de indicadores. Pero es necesario un proceso de síntesis para disponer de toda la fuerza de esta herramienta.

No obstante, la aportación que ha convertido al CMI en una de las herramientas más significativas de los últimos años es que se cimenta en un modelo de negocio. El éxito de su implantación radica en que el equipo de dirección se involucre y dedique tiempo al desarrollo de su propio modelo de negocio.

Beneficios de la implantación de un Cuadro de Mando Integral

- La fuerza de explicitar un modelo de negocio y traducirlo en indicadores facilita el consenso en toda la empresa, no sólo de la dirección, sino también de cómo alcanzarlo.

- Clarifica cómo las acciones del día a día afectan no sólo al corto plazo, sino también al largo plazo.
- Una vez el CMI está en marcha, se puede utilizar para comunicar los planes de la empresa, aunar los esfuerzos en una sola dirección y evitar la dispersión. En este caso, el CMI actúa como un sistema de control por excepción.
- Permita detectar de forma automática desviaciones en el plan estratégico u operativo, e incluso indagar en los datos operativos de la compañía hasta descubrir la causa original que dio lugar a esas desviaciones.

Riesgos de la implantación de un Cuadro de Mando Integral

- Un modelo poco elaborado y sin la colaboración de la dirección no tiene suficientes fundamentes como para triunfar con certeza y el esfuerzo será en vano.
- Si los indicadores no se escogen con cuidado, el CMI pierde una buena parte de sus virtudes, porque no comunica el mensaje que se quiere transmitir.
- Cuando la estrategia de la empresa está todavía en evolución, es contraproducente que el CMI se utilice como un sistema de control clásico y por excepción, en lugar de usarlo como una herramienta de aprendizaje.
- Existe el riesgo de que lo mejor sea enemigo de lo bueno, de que el CMI sea perfecto, pero desfasado e inútil.

11. DW Vs Vistas.

Antes de profundizar en el proceso de BI y los DW se puede venir a la mente la creación de vistas para obtener datos tras consultas más rápidas que crear un nuevo modelo de BBDD. Pero además de todas las funcionalidades que provee un DW que no puede producir una vista conviene señalar unas diferencias básicas que sirven para entender que el uso de uno y otro tipo de almacenamiento no sirven para el mismo uso.

Un DW es un almacenamiento permanente mientras que las Vistas se construyen sólo cuándo son necesarias.

Los DW son multidimensionales pero las Vistas suelen ser relacionales, por lo que no encaja totalmente con el modelo de explotación OLAP.

Los DW son indexados para optimizar su rendimiento mientras que las Vistas son indexadas dependiendo de la BD subyacente, no son un elemento por si solas.

Los DW dan unas funcionalidades específicas y con gran nivel de detalle, Las vistas no pueden hacerlo ya que dependen al 100% de la base de datos (no existe un proceso de ETL).

Los DW poseen grandes cantidades de datos integrados y temporales mientras que para las Vistas son extractos de la BD.

12. FACTORES DE ÉXITO EN EL PROCESO DE DESARROLLO DE UN DW.

Una gran cantidad de proyectos de BI aparecen constantemente, pero la experiencia global en los últimos años no es tan buena. Por lo general, parece que algo no va bien en la ejecución de proyectos de BI, un estudio realizado por la Universidad de Monash [31] concluye que el 85 % de los proyectos de BI han fracasado en la consecución de los objetivos. Así pues, ¿qué determina el éxito o fracaso de esos proyectos? A continuación se realiza un análisis de los factores críticos de éxito de estos proyectos para dar un principio de solución a la pregunta realizada con anterioridad.

Se introducen una serie de factores de éxitos agrupados por categorías [32] según el aspecto del proyecto de BI que contemplan. Por otro lado re realiza una revisión bibliográfica sobre el estado del arte de las publicaciones que hacen referencia a aquellos aspectos que los autores identifican cómo influyentes directamente para el éxito o fracaso final de un proyecto de BI.

Estos factores irán encaminados hacía el hecho de elegir cierta metodología y forma de trabajo para crear el sistema de BI, dónde se trata de que se cumplan las siguientes características finales [39]:

1. La metodología ha de estar orientada al cambio y no a la consecución de un producto final.
2. La gestión del proyecto se debe realizar de forma global y transversal a toda la empresa.
3. Se debe de poder manejar múltiples subproyectos a la vez y en paralelo.
4. Todos los procesos (tareas) de la empresa deben de tenerse en cuenta, sean críticos o no.
5. La metodología y forma de trabajo debe de ajustarse en la gestión de los cambios críticos del workflow empresarial.
6. La solución BI debe estar orientada a las personas y relaciones entre ellas.
7. Se debe de alinear metodología, solución BI y recursos con las necesidades de la empresa.

A. FACTORES CRÍTICOS DE ÉXITO

A. Relativos a las Herramientas de BI.

En ocasiones se culpa a las herramientas de BI de problemas con respecto a los que se espera de ellas, para verificar que las herramientas de BI han estado fallando B. Azvine et al. [33] nos propones los siguientes tres puntos principales:

1. En la visualización de la información de lo que ha pasado.
2. En ayudar a comprender la razón o las causas de lo que ha pasado.
3. En la predicción de lo que sucederá.

Si realmente depende de la validez de la herramienta de BI, entonces, no es necesario por la metodología a utilizar en el desarrollo de un DW, está visión podría ser demasiado alarmista.

B. *Relativos a la Organización.*

Tanto la cultura organizacional como el nivel de madurez de la empresa son factores básicos y críticos por tanto para el éxito del desarrollo de una solución de BI. La gestión (incluso la creación) de una cultura y un nivel de madurez en una empresa requieren mucho tiempo, K. R. Quinn [34] ofrece cinco reglas para el éxito y otras cinco causas del fracaso de los proyectos de BI relacionados con la organización.

Figura 32. Factores críticos de éxito Relativos a la Organización en el Desarrollo de una Solución de BI

C. *Relativos a la Gestión del Conocimiento.*

Hay que asumir que los propios departamentos de sistemas de información tienen limitaciones y deficiencias en la gestión del conocimiento. J. Becker et al. [35] enumeran esas principales deficiencias.

- Existe gran **dificultad** para definir estructuras adecuadas para la **localización de información**. Entorno al 30% de los documentos con **información** relevante se almacena y manipula en **ordenadores personales o portátiles aislados** por lo que el acceso a dicha información está limitado y es accesible solamente por un conjunto

limitado de usuarios dentro de la organización. Además no se facilitan mecanismos para compartir esa información o acceso con el resto del personal interesado.

- Existe gran **dificultad** por el personal respecto al **acceso al conocimiento** para ejercer sus funciones satisfactoriamente. Esta adquisición del conocimiento **requiere tiempo y dinero** de la empresa repercutido por iniciativas formativas, pero estos recursos son de difícil reutilización. Si además, sumamos una **alta rotación de empleados**, las pérdidas en éste ámbito pueden ser considerables.

D. *Relativos a Aspectos Intangibles.*

Como en toda solución TIC, un proyecto de BI debe tener presente más si cabe, el componente intangible del proyecto por las lecturas de la información que pueden obtenerse a posteriori. A. Counihan et al. [36] ya identificaban la dificultad de evaluar los aspectos intangibles de los SI estratégicos. Por otra parte, M. Gibson et al. [37] identificaban **seis** primeros **criterios para evaluar los factores críticos intangibles** de un proyecto de BI, que son:

1. Determinar la criticidad de las cuestiones intangibles.
2. Separar los requisitos de los usuarios, de los aspectos intangibles propios del proyecto.
3. Convencer de la importancia de los intangibles a los gerentes y directivos de la empresa.
4. Clasificar adecuadamente los intangibles para hacer más fácil su evaluación.
5. Gestionar el proyecto orientado a resultados rápidos (quick win).
6. Medir el nivel de cumplimiento de los intangibles.

E. *Relativos al Personal y al Liderazgo.*

Sin lugar a dudas, para la conducción de cualquier proyecto en todo ámbito es muy importante el perfil, experiencia y formación de las personas involucradas en el proyecto, dadas las características del BI el liderazgo es una función especialmente crítica para el éxito en la implantación. En torno a ésta idea A. Faulkner y A. MacGillivray [38] identifican **doce aspectos** (casi requisitos**) que debe satisfacer el líder de todo proyecto de BI** para afrontar con las mayores garantías de éxito un proyecto de BI, estos son:

1. Debe tener la capacidad de saber reflexionar antes que actuar sobre los valores de la organización.

2. Focalizar los objetivos del proyecto en las necesidades más urgentes de la organización, a fin de que los resultados sean visibles lo más pronto posible.
3. En cierto sentido, el líder deber ser considerado como un antropólogo amateur, identificando las necesidades del negocio y quienes las están gestionando, para proveerles de herramientas fáciles de usar según sus habilidades, necesidades y preferencias personales.
4. Debe planificar para el éxito, comprendiendo como la organización evalúa el éxito para poder dirigirse hacia él.
5. Debe cuestionar todas las decisiones para conocer el porqué de la elección de un camino en detrimento del resto de opciones.
6. Debe ser capaz de conseguir que el proyecto sea una iniciativa de cambio e innovación para toda la organización, desde el primero hasta el último miembro que forma parte de ella.
7. El diálogo debe ser un arman a favor que debe usarse sin descanso.
8. El líder debe ser capaz de integrar e involucrar a los ejecutivos y directores intermedios como colíderes (patrocinadores) del proyecto, para que lo sientan como suyo.
9. Debe ser proactivo anticipando la resistencia del cambio y convertirse en el defensor de la causa de BI.
10. Como todo líder capaz, deber saber aprender del resto, recuperar lo mejor de cada uno para saber elegir la mejor vía, ya sea propuesta por él o por otro miembro del equipo.
11. Evaluar el coste y el riesgo de la alternativa de no usar herramientas de BI para poder usarlo como argumento y dar mayor sentido al proyecto.
12. Tener una mente abierta y una visión global de la evolución que puede tener el BI dentro de la organización.

Por otro lado, T. Chenoweth et al. [40] afirman que la interacción entre la tecnología y el contexto social de las empresas claramente determina el éxito o el fracaso de un banco de datos común a la organización. Y al mismo tiempo, reconocen que esta misma interacción puede determinar la extensión y evolución de un sistema de BI. En este sentido proponen **siete cuestiones clave**, como una lista ordenada de preguntas **básicas a considerar en todo proyecto de BI**. Son las siguientes:

1. ¿El proyecto tiene el apoyo de la dirección?
2. ¿Los futuros usuarios apoyan el proyecto?
3. ¿Los usuarios pueden acceder a un amplio abanico de datos con el sistema?
4. ¿Los usuarios necesitan herramientas concretas?
5. ¿Los usuarios entienden la relación existente entre la información que les proporcionará el sistema de BI y los procesos de negocio que llevan a cabo?

6. ¿Los usuarios perciben al departamento de SI como soporte y ayuda en la realización de sus proyectos de negocio?
7. ¿Existen usuarios estrella o con dominio sobre el resto?

B. PUBLICACIONES.

Con ésta aproximación a las publicaciones, realmente no identificamos un apartado diferente al de factores críticos de éxito, la diferencia estriba en que los autores no agrupan por temas específicos dichos factores sino que se trata de recomendaciones generales, son aspectos más genéricos pero igualmente relevantes y críticos.

A. *M.D. Solomon.*

El autor en [41] nos propone **una guía de aspectos a considerar** cuando se lleva a cabo la implementación de una solución de BI o se crea un DW. Es la que sigue:

- El usuario tiene que participar en la definición del nivel del servicio y los requisitos de información.
- Identificar los sistemas de provisión de datos.
- Definir el plan de calidad.
- Elegir un modelo de diseño adecuado.
- Escoger la herramienta ETL a utilizar.
- Realizar cargas de datos incrementales preferentemente.
- Escoger cuidadosamente la plataforma de desarrollo de BI y el SGBDR adecuado.
- Realizar procesos de conciliación de datos.
- Revisar y modificar periódicamente la planificación.
- Proveer soporte al usuario.

B. *L.T. Moss.*

El autor en [39] nos resume los **diez errores más frecuentes** en la gestión de proyectos de BI y creación de DW. Son los siguientes:

1. No usar ninguna metodología.
2. No disponer del equipo adecuado,
3. Los usuarios y los responsables de las decisiones importantes no participan en el proyecto.
4. Descomposición del proyecto en etapas inadecuadas.
5. No hay planificación del proyecto.
6. No hay plan de calidad.
7. Las pruebas de calidad no están finalizadas o son incompletas.
8. No se prevé el volumen correcto de datos a monitorizar o depurar.

9. Ignorar metadatos y semántica de datos.
10. Depender en exceso (alto acoplamiento) de la herramienta de gestión del proyecto.

C. *D. Briggs et al.*

En las obras [42] y [43] se proponen una serie de **factores críticos de éxito** para sistemas decisionales, que son:

- Necesidad de patrocinio del proyecto.
- Gestión de las expectativas del usuario.
- Uso de prototipos.
- Búsqueda del resultado rápido (quick win).
- Escoger un parámetro de la organización medible.
- Modelización y diseño del data warehouse con precisión.
- Selección del caso de negocio adecuado.
- Alineación con la estrategia organizativa.
- Selección cuidadosa de las herramientas.
- Usuario finales involucradas activamente.
- Realización de una gestión del cambio organizativa.
- Centrarse en la gestión de los datos.
- Nivel de escalabilidad y flexibilidad del proyecto y la solución.
- Transmitir el conocimiento en proyectos subcontratados.
- Uso de estándares.
- Aprovechamiento de la experiencia de los miembros del equipo.
- Soporte al usuario final.

D. *B.H. Wixom y H.J. Watson.*

En las obras [44] y [45] identifican los siguientes **factores** como los **más relevantes** para el éxito de un proyecto de BI:

- Apoyo a la gestión de la organización.
- Existencia de un líder del proyecto.
- Uso adecuado de los recursos.
- Participación del usuario final.
- Equipo con competencias adecuadas.
- Disposición de fuentes de datos adecuadas para su explotación.
- Considerar la información y su análisis como parte de la cultura de la organización.
- Alineamiento con la estrategia de la organización.
- Gestión y control eficaz del BI.

E. *D. Sammon y P.Finnegan.*

A través de la publicación [46] los autores proponen los **diez mandamientos del proyecto BI**:

1. Iniciativa ligada a las necesidades del negocio.
2. Existencia del patrocinio de la dirección.
3. Gestión de las expectativas del usuario.
4. Proyecto transversal a la organización.
5. Control de calidad.
6. Flexibilidad del modelo de datos.
7. Gestión orientada a los datos.
8. Proceso automático de extracción de datos.
9. Conocimiento.
10. Experiencia.

F. *R. Weir et al.*

Define a través de la publicación [47] las **mejores prácticas** en el proceso de desarrollo de un proyecto de BI:

- Realización de cambios incrementales.
- Construcción de un sistema adaptable.
- Gestionar las expectativas del usuario.
- Equipo mixto entre técnicos y usuarios finales.
- Contacto directo con la organización y el negocio.
- No perseguir la perfección.

G. *R.S. Abdullaev y I.S. Ko*

A través de la publicación [48] analizan las lecciones aprendidas de diversas **experiencias** en la construcción de BI:

- La centralización de datos en un DW y su agregación en varios DM permiten un acceso rápido y de confianza a la información solicitada.
- La definición de listados estandarizados para todos los usuarios favorece el intercambio de información entre departamentos de una manera más clara y consistente.
- Algunos modelos de informes predefinidos se han de implementar con el fin de proporcionar a los decisores la funcionalidad para añadir o eliminar elementos necesarios y crear informes específicos.

- Es necesario un equipo responsable de alinear las especificaciones de informes estándar con las necesidades locales y que facilite la ejecución del proyecto de BI.
- Debe existir un fuerte compromiso de la dirección para resolver cualquier conflicto durante el desarrollo del proyecto.
- La integración de técnicas "Six Sigma" en la infraestructura TI de la Organización contribuye a la construcción de un sistema de BI robusto.
- La infraestructura de TI ha de centrarse en una sola plataforma proporcionada por proveedores conocidos.

Además a través de la publicación [49] identifican los **aspectos** considerados como especialmente **críticos** durante el desarrollo de proyectos BI:

- Deben prevalecer los requisitos del mercado y del cliente a los requisitos internos.
- Se debe tener personal responsable de cada departamento dedicados al proyecto.
- El equipo debe ser formado por personal competente en el proyecto.
- Adopción de una metodología de desarrollo de proyectos BI.
- Realizar y seguir una planificación del proyecto.
- Estandarización de los datos.
- Control de calidad de los datos.
- Existencia de los datos.
- Usar sólo las herramientas necesarias.

H. *W. Yeoh et al.*

Con la publicación [50] se hace una **recopilación de los factores críticos de éxito** para proyectos de BI que enuncia las siguientes afirmaciones que deben producirse:

- Soporte al proyecto por parte de la alta dirección.
- Disponer de los recursos adecuados.
- Apoyo comprometido por parte de la organización.
- Participación formal del usuario a lo largo de todo el proyecto.
- Soporte, formación y entrenamiento.
- Caso de negocio establecido y consensuado.
- Visión estratégica de BI integrada con las iniciativas de la compañía.
- Ámbito del proyecto claramente definido.
- Adopción de un enfoque de resultados incrementales.
- Proyecto enfocado a resultados rápidos (quick win).
- Equipo poseedor de la perfecta combinación de capacidades.
- Participación de consultoría externa en las fases iniciales del proyecto.
- Experiencia en el dominio del negocio.
- Equipo multifuncional.

- Sistemas proveedores de datos estables.
- Entorno técnico estratégico, escalable y extensible.
- Prototipo usado como prueba de concepto.
- Disponer de fuentes de datos de calidad.
- Métricas y clasificaciones comunes establecidas por la organización.
- Modelo de datos escalable,
- Gobierno de los datos por la organización.

C. CLASIFICACIÓN, CRÍTICA Y VALORACIÓN

La revisión de la bibliografía realizada para identificar factores críticos de éxito, mejores prácticos, errores comunes y recomendaciones de cómo realizar un proyecto de BI proponen una serie de directrices que se deben de acoplar en la gestión del proyecto y metodología utilizadas para el desarrollo más adecuado de un proyecto de BI.

Se han analizado y clasificado agrupando los factores críticos de éxito, así, una vez realizada la agrupación se establecen los factores críticos de éxito en distintos grupos de importancia contabilizando el número de proposiciones sobre cada factor que aparecen por los distintos autores. Además identificamos que siempre debe considerarse un plan de calidad, por lo que dicho factor no lo agruparemos en ninguno de los grupos indicados.

Los grupos son:

- **Factores Primarios**

  Serán los aspectos o factores que aparecen propuestos por más de cinco autores. Serán básicos y se tendrán que atender obligatoriamente.

- **Factores Secundarios**

  Serán los aspectos o factores que aparecen propuestos por entre dos y cinco autores. Formaran parte de la mayoría de proyectos de BI, sobretodo cuándo el tamaño del proyecto no sea reducido.

- **Factores de autor.**

  Serán los aspectos o factores que aparecen propuestos por un único autor. Serán recomendaciones a tener en cuenta, pero en la mayoría de las ocasiones no se puede exigir su obligatoriedad.

A continuación se muestra un cuadro resumen indicando que factores pueden encontrarse en las distintas publicaciones para los factores que forman parte de los grupos de Factores Primarios y Factores Secundarios:

A. Factores Primarios

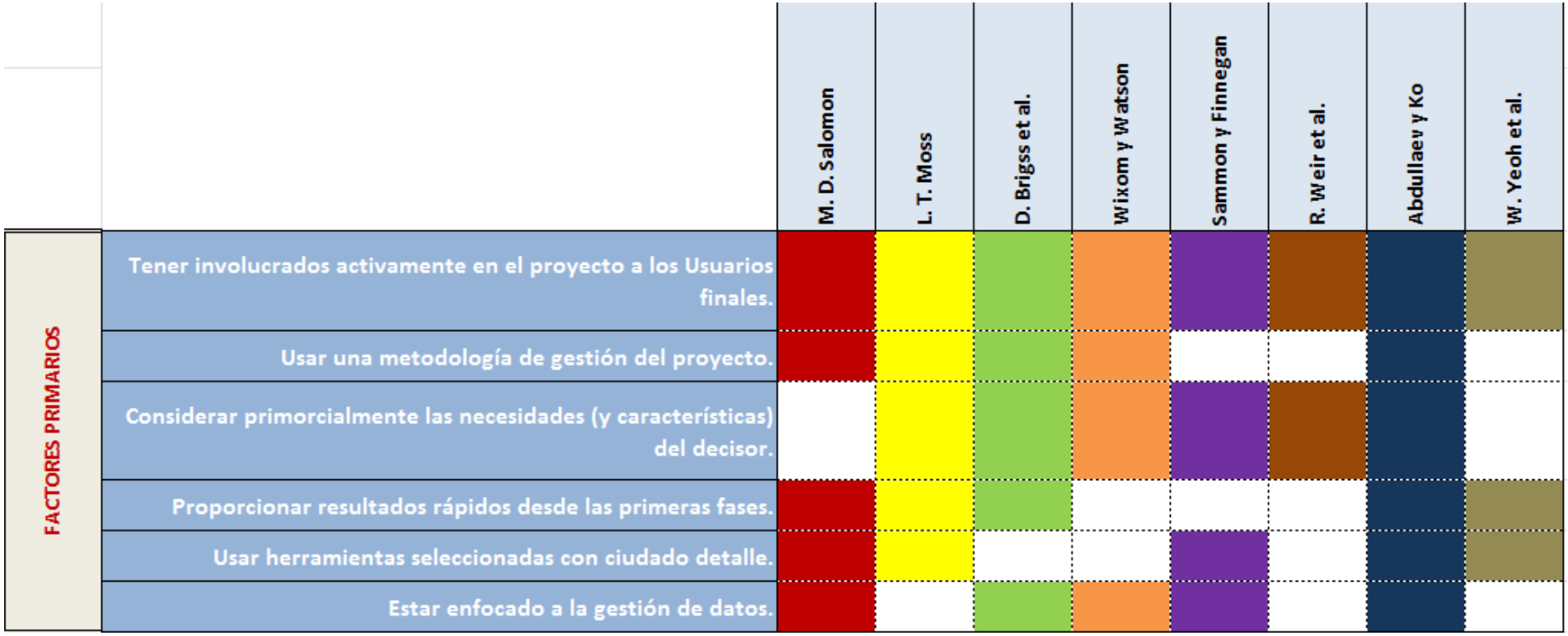

Figura 33. Comparativa de los FCE en un Proyecto de BI (Primarios)

B. **Factores Secundarios**

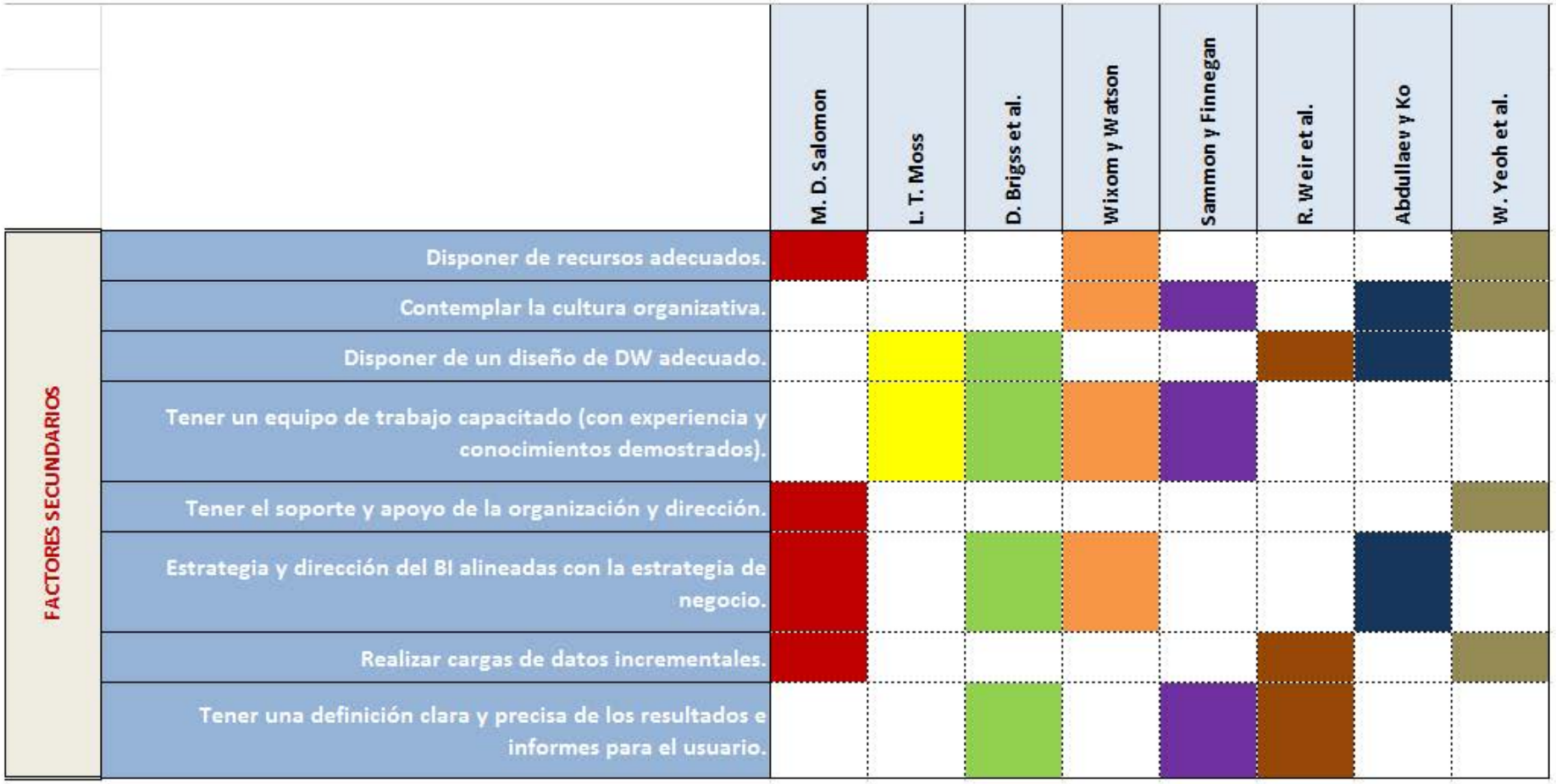

Figura 34. Comparativa de los FCE en un Proyecto de BI (Secundarios)

D. CONCLUSIONES

Con el surgimiento del BI en los últimos años, todas las organizaciones de cierto nivel de estructura hacen esfuerzos para crear o mejorar sus procesos de decisión y sistemas decisionales. Aparecen muchos proyectos de BI pero la experiencia no es buena, ya que en torno al 85% de ellos no alcanzan los objetivos iniciales y no porque sean demasiado ambiciosos.

Una gran diversidad y heterogeneidad de enfoques metodológicos para la gestión de proyectos de BI muestra el estado de la novedad y la inexperiencia que todavía existe en este ámbito. Elegir una metodología de BI (entendiendo también como metodología las directrices principales y características de gestión del proyecto) no es tarea fácil ya que en un proyecto de BI se suponen cientos de tareas a realizar y por el momento queda muy lejana una metodología que de soporte a todas las cuestiones independientemente del tipo de proyecto de BI con el que tratemos.

La principal recomendación es extraer una serie de factores claves que deben contener/ evitar los proyectos de BI para afrontar con las máximas garantías de éxito el desarrollo de un proyecto de BI.

Por otro lado, la principal conclusión sobre el apartado de factores de éxito es que **el usuario es el centro de todo** y su implicación en el proyecto de BI va a determinar claramente el éxito o fracaso del mismo. El usuario tiene que saber que quiere, para poder hacerlo.

BLOQUE II: CALIDAD EN LOS SISTEMAS DE BI.

1. INTRODUCCIÓN

La calidad en los SI se puede ver desde varias perspectivas, como son el uso de metodologías, aumentando las posibilidades de realizar nuestro cometido con eficacia y una mayor eficiencia en el proceso de desarrollo de un SI; proporcionandouna perspectivareferente, aumentando la precisión final del producto que puede referirse a que el contenido del SI sea de calidad en cuanto al dato manejado y por consiguiente la información que se extrae de ella.

En los sistemas data Warehouse se puede ir un paso más allá, cuando hablamos de la calidad en término del conocimiento, entendiendo un DW como el desarrollo de un SI orientado a generar conocimiento. Ahora no nos vale simplemente realizar los sistemas con calidad de procesos o generando un resultado de calidad sino que debemos poder servir de herramienta para la generación de conocimiento, serán pieza fundamental el dato, la información que seamos capaz de extraer ycómocombinar el juego de procesos y fases de implementación que se incluyen durante el ciclo de vida de un DW ya que sus características como vimos al comienzo del texto hacen que mínimos desvíos en el diseño o la implementación lleven al fracaso del sistema, no sólo en tiempo de desarrollo o medidas de almacenamiento sino a las capacidades de las que se presupone al mismo con lo que las decisiones tomadas en base a él serán erróneas.

Por todo lo comentado en los párrafos anteriores se hace indispensable tener en cuenta varias normas de calidad que se deben aplicar en cualquier sistema DW. Cada una reflejará unos aspectos a tener en cuenta con respecto a las demás en favor de su propósito. Nuestro cometido es repasar las normas de calidad existentes y abstraer la información contenida en ellas para integrar sus indicaciones en un SI de DW para el desarrollo de BI.

Dependiendo del ámbito de calidad en el que nos movamos podemos consultar diferentes normas de calidad, en cuanto a los **procesos** (ISO 12207, ISO 15504 o CMMi), en cuanto al **producto** (ISO 9126, ISO 14598 o ISO 25000), en cuanto a la **Gestión y servicios** (ISO 9001 e ISO 90003, ISO 20000 e ITIL, COBIT y PMBOK), en cuanto a calidad en las variables del **ciclo de vida** (ISO 27001, PMBOK), incluso metodologías **ágiles** como SCRUM o sobre los planes de **aseguramiento/Medición/Evaluación de la calidad** del SW (IEEE 730, ISO 2502n, ISO 2504n) o calidad, pero vamos a centrarnos en hacer un breve repaso sobre las que para el propósito del desarrollo del DW corporativo de soluciones de BI vemos más significativas a la hora de la calidad en el proceso de desarrollo y de la calidad del dato.

Por tanto, no debemos olvidarnos que nos encontramos dentro de un proceso de desarrollo de software que difiere de lo estándar, dónde cómo se ha visto en los apartados anteriores requiere de un punto de vista diferente a los procesos de desarrollo de software convencional.

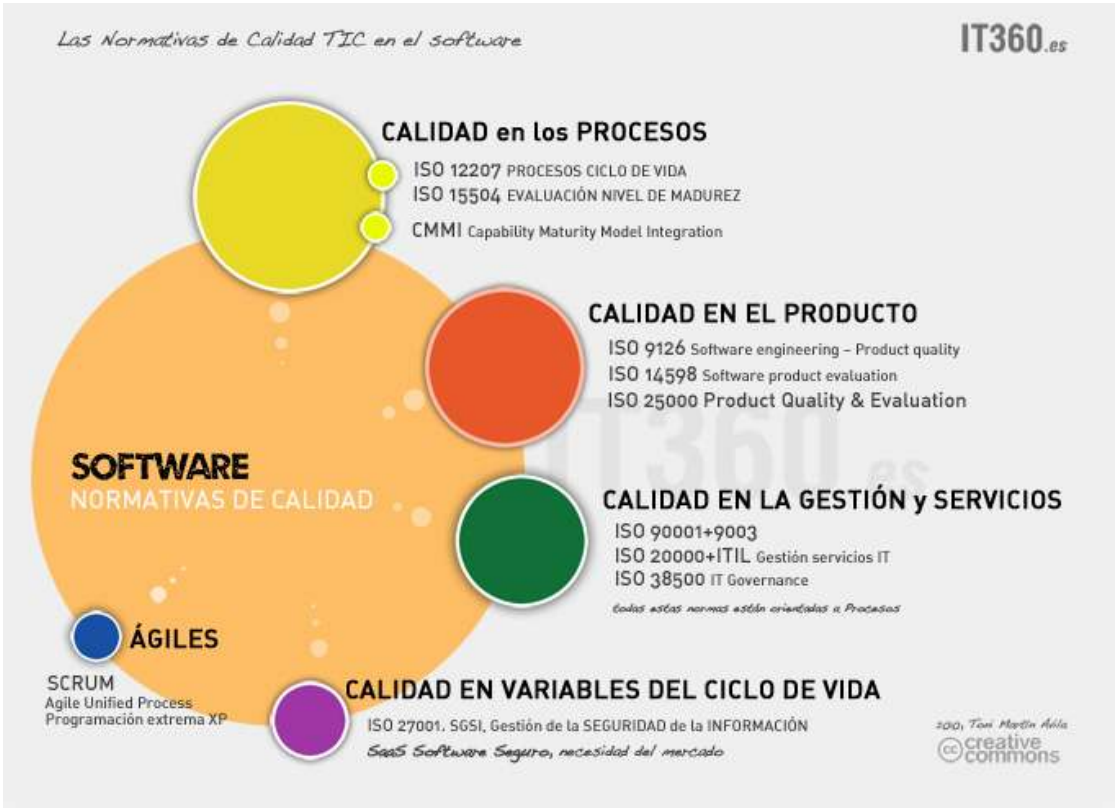

Figura 35. Normas de calidad en el SW [67]

El objetivo de la gestión de la calidad en un desarrollo de software para la construcción de un sistema DW debe comprender aquellos factores críticos que se enumeran en los modelos de calidad conocidos tanto para la calidad del dato como para el desarrollo, gestión y mantenimiento del propio DW durante todo su ciclo de vida.

Para ello mostraremos unrepaso sobre la bibliografía existente en la actualidad sobre métricas de calidad en sistemas DW, sobre calidad en el dato y sacaremos una serie de conclusiones sobre las que a partir de los estudios reconocidos seamos capaces de crear una serie de directrices dónde siempre que se pueda adaptar dichas directrices se pueda asegurar que un DW resultante será de una calidad alta, será diseñado de forma eficiente y si se emplean las técnicas de extracción oportunas siempre será eficaz en la consecución de los objetivos primordiales de un DW.

El apartado se centrará en repasar las normas y estándares de calidad que puedan tener mayor impacto en cuanto a la calidad del proceso y a continuación nos centraremos en la calidad del dato. En base a los análisis y descripciones mostradas a continuación se genera el Bloque III: “Guidelines para el desarrollo de un Data Warehouse de calidad en un sistema BI. Calidad en el dato, calidad en el proceso”.

2. CALIDAD EN EL PROCESO. NORMAS Y ESTÁNDARES DE CALIDAD.

1. ISO 9001.

A. *La Norma*

La ISO 9001 (Quality management systems – Requirements o Sistemas de gestión de la calidad. Requisitos) es una norma internacional que se aplica a los sistemas de gestión de calidad (SGC) y que se centra en todos los elementos de administración de calidad con los que una empresa debe contar para tener un sistema efectivo que le permita administrar y mejorar la calidad de sus productos o servicios.

- Sección 1 – ***Ámbito***

  Guías y descripciones generales, no se enuncia ningún requisito.

- Sección 2 - ***Referencias normativas***

  Guías y descripciones generales, no se enuncia ningún requisito.

- Sección 3 - ***Términos y Definiciones***

  Guías y descripciones generales, no se enuncia ningún requisito.

- Sección 4 - ***Requisitos del Sistema de Gestión***.

  Contiene los requisitos generales y los requisitos para gestionar la documentación.

- Sección 5 - ***Responsabilidades de la Dirección***

  Contiene los requisitos que debe cumplir la dirección de la organización, tales como definir la política, asegurar que las responsabilidades y autoridades están definidas, aprobar objetivos, el compromiso de la dirección con la calidad, etc.

- Sección 6 - ***Gestión de Recursos***

  La Norma distingue 3 tipos de recursos sobre los cuales se debe actuar: RRHH, infraestructura, y ambiente de trabajo. Aquí se contienen los requisitos exigidos en su gestión.

- Sección 7 - ***Realización del Producto***

  Aquí están contenidos los requisitos puramente productivos, desde la atención al cliente, hasta la entrega del producto o el servicio.

- Sección 8 ***- Medición, Análisis y Mejora***

Aquí se sitúan los requisitos para los procesos que recopilan información, la analizan, y que actúan en consecuencia. El objetivo es mejorar continuamente la capacidad de la organización para suministrar productos que cumplan los requisitos. El objetivo declarado en la Norma, es que la organización busque sin descanso la satisfacción del cliente a través del cumplimiento de los requisitos.

En cuanto al propósito de éste documento no nos centraremos demasiado en la norma ISO 9001 ya que su ámbito es más extenso que el cometido del proyecto, no obstante sería recomendable que se repasaran las secciones 4, 5 y 6 del mismo en la planificación del desarrollo, implementación y operación de los sistemas de BI por la ligadura que tendrán tanto los responsables de la dirección de definir los sistemas de BI como de la gestión correcta de recursos para un proyecto de éstas características.

Para extender la norma, se publica ISO/IEC 900003 que proporciona una guía para que las organizaciones puedan aplicar de forma correcta la norma ISO 9001 en cuanto a la adquisición, suministro, desarrollo, operación y mantenimiento de software y servicios de soporte. Nos centraremos algo más en ella dado que se adapta más a las necesidades que se detectaran en las primeras fases de decisiones sobre la planificación de un sistema BI a la hora de estudiar posibles adquisiciones de productos ya que para el propio desarrollo de SW existen normas mejor preparadas para nuestro propósito.

B. *ISO 90003*

La norma ISO/IEC 90003 (Software engineering - Guidelines for the application of ISO 9001:2000 to computer software ó Ingeniería del software - Directrices para la aplicación de la Norma ISO 9001:2000 al software de ordenador) también conocida como ISO 9000-3 fue preparada por el comité técnico conjunto ISO/ IEC. La primera edición de ISO/IEC 90003 canceló y sustituyó a la norma ISO/ IEC 9000-3: 1997, que se ha actualizado en conformidad con ISO 9001:2000.

Mientras que el objetivo de la norma ISO 9001 es la de establecer un sistema de calidad, la ISO 90003:2004 provee las especificaciones de cómo aplicar la ISO 9001 a los procesos de software, entre ellos los procesos de adquisición, provisión, desarrollo, operación y mantenimiento y servicios de ayuda relacionados. Además, partiendo del hecho de que es muy poco probable obtener buenos resultados en un proyecto de desarrollo de software en organizaciones de tamaño mediano, algunos temas como la Administración de la Configuración o Planeación de Proyectos que no están presentes en las normas de la familia ISO 9000, son incluidos en la ISO 90003. Esta característica implica que para ciertos productos y/o servicios,

la especificación de requerimientos contenida en las normas de ISO 9001 no sea la suficiente para asegurar la calidad, justificando la necesidad de otras normas o guías más específicas como complemento.

Esta norma identifica los puntos que se recomienda considerar y es independiente de la tecnología, modelos de ciclo de vida, procesos de desarrollo, secuencia de actividades y estructura utilizados por una organización.

Por lo tanto, puede ser combinado con otros enfoques más específicos, como por ejemplo el modelo Espiral de Boehm [64].

Este estándar no agrega o cambia los requerimientos de la ISO 9001:2000. Por lo tanto ISO 90003:2004 es una guía (y por tanto no es certificable).

La aplicación de ISO 90003:2004 es apropiada para un software que cumple las siguientes condiciones:

1. Forma parte de un contrato comercial con otra organización.
2. Es un producto disponible para un sector del mercado.
3. Es usado para soportar/apoyar los procesos de una organización.
4. Se encuentra embebido en un producto hardware.
5. Está relacionado con servicios de software.

Unas de las razones para implantar la norma son las siguientes:

- Introducirse en el mercado europeo.
- Cubrir las expectativas de los clientes.
- Obtener beneficios de calidad.
- Como estrategia de mercado.
- Reducir costes de producción.

Además se entiende que una organización que tiene implantada la norma consigue los siguientes beneficios:

- Mejor documentación de los sistemas.
- Cambio cultural positivo.
- Incremento en la eficiencia y productividad.
- Mayor percepción de calidad.
- Se amplía la satisfacción del cliente.
- Se reducen las auditorías de calidad.
- Agiliza el tiempo de desarrollo de un sistema.

La Estructura de la Norma ISO 90003:2004 destacar las siguientes clausulas:

- **Administración de la Responsabilidad**

  La dirección de la empresa debe definir y documentar su política y sus objetivos con respecto a la calidad.

  Las responsabilidades, autoridades y relaciones entre todo personal, cuyo trabajo afecte la calidad del producto, deben ser definidas.

- **Sistemas de Calidad**

  La empresa debe establecer y mantener un sistema de calidad documentado que debe incluir:

  - Directrices para la preparación de procedimientos del sistema de calidad.
  - Instrucciones para la aplicación efectiva de los procedimientos.

- **Auditorías internas de calidad**

  La empresa debe llevar un sistema de auditorías internas de calidad y sus resultados deben ser documentados.

- **Revisión de Contratos**

  La empresa debe establecer y mantener procedimientos para la revisión de los contratos y para la coordinación de estas actividades.

  De esta forma asegurar que la empresa tiene la capacidad de cumplir con todos los requerimientos contractuales.

- **Control de Documentos y Datos**

  La empresa debe establecer y mantener procedimientos para controlar todos los documentos y datos.

  Todo documento debe estar disponible, y los documentos obsoletos deben ser eliminados.

- **Planificación del diseño y el desarrollo**

  Esta cláusula exige la definición de una metodología.

- **Inspección y pruebas**

  La empresa debe asegurar que los productos no se utilicen o procesen hasta que sean inspeccionados y verificados.

  Se deben establecer y mantener registros que contengan el criterio de aceptación del producto.

- **Control, Calibración y Mantenimiento de los equipos de inspección, medición y pruebas utilizados por la empresa.**
- **Control del producto no conforme**

  Los productos que no cumplan los requerimientos, especificados no deben ser instalados, usados o puestos en producción.

- **Acciones correctivas y preventivas**
- **Control de los registros de calidad**
- **Capacitación**

  La empresa debe establecer y mantener procedimientos para identificar las necesidades de capacitación y proveer entrenamiento a todo el personal, que además debe ser calificado con base a su educación, entrenamiento oexperiencia.

- **Estadísticas**

  La empresa debe mantener un registro de estadísticas adecuadas para verificar el estado del proceso.

En resumen, esta norma supone sin lugar a dudas un cambio cualitativo para que losambientes tecnológicos más especializados avancen, especialmente los enfocados al diseño y desarrollo de software.

2. ISO/ IEC 9126

A. *La Norma*

ISO 9126 (Software engineering — Product quality ó Ingeniería de software - Calidad del producto) es un estándar internacional para la evaluación del Software. Está supervisado por el proyecto SQuaRE, ISO 25000:2005, el cual sigue los mismos conceptos y reemplaza al indicado, haremos mención del mismo debido al gran impacto que provocó durante unos años y debido a que aún hay organizaciones que no han dado el paso de implantar la gama de normas ISO 250xx.

El estándar está dividido en cuatro partes las cuales dirigen, respectivamente, lo siguiente:

1. Modelo de calidad.
2. Métricas externas.
3. Métricas internas.
4. Calidad en las métricas de uso.

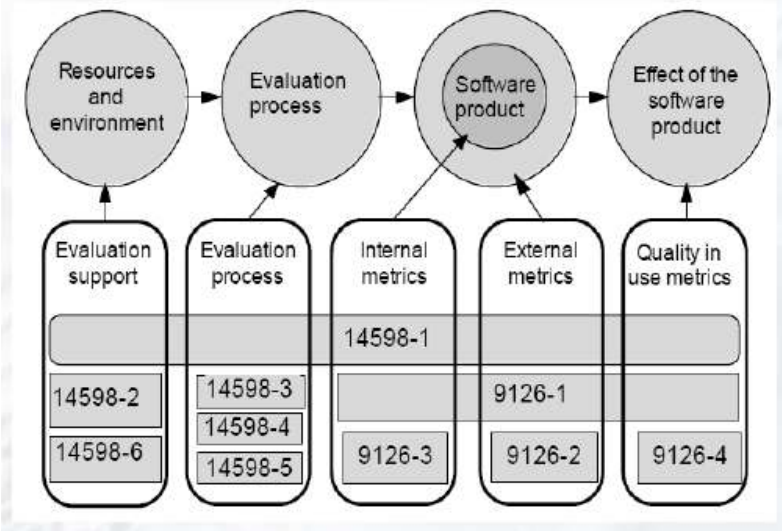

Figura 36. Arquitectura de las series 9126

El modelo de calidad establecido en la primera parte del estándar, ISO 9126-1 ha sido desarrollado como un intento de identificar los atributos clave de calidad para el software. Sólo la primera parte es un estándar aprobado y publicado, siendo el resto de partes de la norma informes que se encuentran en la llamada Technical Report (TR).

Modelo de calidad (ISO 9126-1):

Se identifican 6 ***atributos clave de calidad***:

Calidad del software (interna y externa)					
Funcionalidad	***Fiabilidad***	***Usabilidad***	***Eficiencia***	***Mantenibilidad***	***Portabilidad***
Adecuación	*Madurez*	*Fácil comprensión*	*Comportamiento frente al tiempo*	*Facilidad de análisis*	*Adaptabilidad*
Exactitud	*Tolerancia a fallos*	*Fácil aprendizaje*	*Uso de recursos*	*Capacidad para cambios*	*Facilidad de instalación*
Interoperatividad	*Capacidad de recuperación*	*Operatividad*	*Adherencia a normas*	*Estabilidad*	*Coexistencia*
Seguridad	*Adherencia a normas*	*Software atractivo*		*Facilidad para pruebas*	*Facilidad de reemplazo*
Adherencia a normas		*Adherencia a normas*		*Adherencia a normas*	*Adherencia a normas*

Figura 37. Atributos de calidad SW ISO 9126-1.

1. Funcionalidad:

El grado en que el software satisface las necesidades indicadas por los siguientes sub-atributos:

- **Idoneidad**
- **Corrección**
- **Interoperabilidad**
- **Conformidad**
- **Seguridad**

2. Fiabilidad:

Cantidad de tiempo que el software está disponible para su uso. Está referido por los siguientes sub-atributos:

- **Madurez**
- **Tolerancia a fallos**
- **Facilidad de recuperación**

3. Usabilidad:

Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes sub-atributos:

- **Facilidad de comprensión**
- **Facilidad de aprendizaje**
- **Operatividad**

4. Eficiencia:

Grado en que el software hace óptimo el uso de los recursos del sistema. Está indicado por los siguientes sub-atributos:

- **Tiempo de uso**
- **Recursos utilizados**

5. Mantenibilidad:

Facilidad con que una modificación puede ser realizada. Está indicada por los siguientes sub-atributos:

- **Facilidad de análisis**
- **Facilidad de cambio**
- **Estabilidad**
- **Facilidad de prueba**

6. Portabilidad:

La facilidad con que el software puede ser llevado de un entorno a otro. Está referido por los siguientes sub-atributos:

- **Facilidad de instalación**
- **Facilidad de ajuste**
- **Facilidad de adaptación al cambio**

El atributo ***Conformidad*** no está listado arriba ya que se aplica a todas las características. Ejemplos son conformidad a la legislación referente a usabilidad y fiabilidad.

Un atributo es una entidad la cual puede ser verificada o medida en el producto software. Los atributos no están definidos en el estándar, ya que varían entre diferentes productos software.

Un producto software está definido en un sentido amplio como: los ejecutables, código fuente, descripciones de arquitectura, etc. Como resultado, la noción de usuario se amplía tanto a operadores como a programadores, los cuales son usuarios de componentes como son bibliotecas software.

El estándar provee un entorno para que las organizaciones definan un modelo de calidad para el producto software. Sin embargo esto lleva a cada organización la tarea de especificar precisamente su propio modelo. Esto podría ser hecho, por ejemplo, especificando los objetivos para las métricas de calidad las cuales evalúan el grado de presencia de los atributos de calidad.

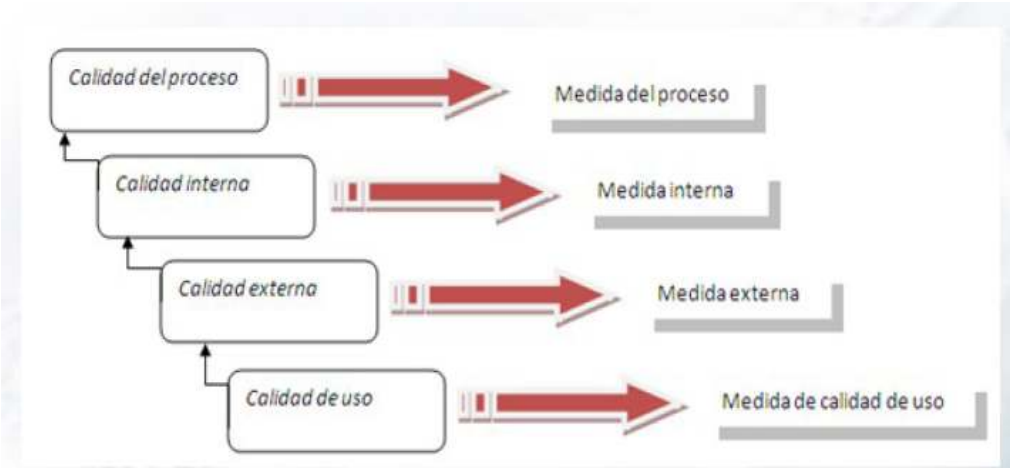

Figura 38. Calidad en el Ciclo de Vida ISO 9126

Métricas internas son aquellas que no dependen de la ejecución del software (medidas estáticas).

Métricas externas son aquellas aplicables al software en ejecución.

La calidad en las métricas de uso están sólo disponibles cuando el producto final es usado en condiciones reales.

Idealmente, la calidad interna determina la calidad externa y esta a su vez la calidad en el uso.

Este estándar proviene desde el modelo establecido en 1977 por McCall y sus colegas, los cuales propusieron un modelo para especificar la calidad del software. El modelo de calidad McCall está organizado sobre tres tipos de Características de Calidad:

1. **Factores**(especificar)

   Ellos describen la visión externa del software, como es visto por los usuarios.

2. **Criterios** *(construir)*

   Ellos describen la visión interna del software, como es visto por el desarrollador.

3. **Métricas** *(controlar)*

   Ellas son definidas y usadas para proveer una escala y método para la medida.

ISO 9126 distingue entre fallos y no conformidad, siendo un ***fallo*** el no cumplimiento de los requisitos previos, mientras que la ***no conformidad*** afecta a los requisitos especificados. Una distinción similar es hecha entre la validación y la verificación.

B. *Utilidad*

Este estándar está pensado para los desarrolladores, adquirentes, personal que asegure la calidad y evaluadores independientes, responsables de especificar y evaluar la calidad del producto software.

Por tanto, puede servir para validar la completitud de una definición de requisitos, identificar requisitos de calidad de software, objetivos de diseño y prueba, criterios de aseguramiento de la calidad, etc.

La calidad de cualquier proceso del ciclo de vida del software (estándar ISO 12207) influye en la calidad del producto software que, a su vez, contribuye a mejorar la calidad en el uso del producto.

La calidad del software puede evaluarse midiendo los atributos internos (medidas estáticas o productos intermedios) o atributos externos (comportamiento del código cuando se ejecuta).

La "**calidad de uso**" es definida como "la capacidad del software que posibilita la obtención de objetivos específicos con efectividad, productividad, satisfacción y seguridad".

Figura 39. Calidad de uso 9126-1.

3. ISO / IEC 15504 (SPICE)

A. *La Norma*

El ISO/IEC 15504, también conocido como Software Process Improvement Capability Determination, abreviado **SPICE**, en español, «**Determinación de la Capacidad de Mejora del Proceso de Software**» es un modelo para la mejora y evaluación de los procesos de desarrollo y mantenimiento de sistemas de información y productos de software. [65]

Las siglas SPICE significan: Software Process Improvement and Capability Determination, es decir, Determinación de la capacidad y mejora de los procesos de software.

El proyecto SPICE tenía tres objetivos principales [65]:

1. Desarrollar un borrador de trabajo para un estándar de evaluación de procesos de software.
2. Llevar a cabo los ensayos de la industria de la norma emergente.
3. Promover la transferencia de tecnología de la evaluación de procesos de software a la industria del software a nivel mundial.

Las empresas de desarrollo y mantenimiento software que implanten la norma se las supone las siguientes ventajas:

1. Pueden contar con una norma ISO, internacional y abierta.
2. En España, la norma cuenta con el respaldo del Ministerio de Industria de España ya que existen ayudas para la certificación de las PYMES y de AENOR.

3. Integración más fácil con otras normas ISO del sector TIC, como son: ISO 27000 de seguridad, ISO 20000 de servicios de IT e ISO 9000.
4. Evalúa por niveles de madurez, la evaluación más extendida entre los modelos de mejora.
5. Normalmente, tiene un menor coste de certificación que otros modelos similares (Ver Informe de INTECO).
6. Existen certificaciones de prestigio, como por ejemplo la otorgada por AENOR.

En la actualidad queda reemplazado por la familia de normas ISO 250xx, pero dada la importancia que ha tenido el estándar y que muchas empresas se han esforzado para seguir sus indicaciones vemos relevante el mencionar que puntos fuertes tiene.

La norma ISO 15504 permite realizar evaluaciones usando niveles de madurez, la evaluación más extendida en la actualidad [68].

Tiene una arquitectura basada en **dos dimensiones**: de proceso y de capacidad de proceso (Madurez). Define que todo modelo de evaluación de procesos debe definir.

1. **La dimensión de procesos.**

   El modelo de procesos de referencia (dimensión de las abscisas).

2. **La dimensión de la capacidad o Madurez**

   Niveles de Madurez y atributos de los procesos.

Los **niveles de madurez** son conjuntos predefinidos de procesos que ayudan a una organización a mejorar en el desarrollo software evolucionando por los distintos niveles.

En esta norma, se han establecido **6 niveles** que indican la madurez de la organización. Como se observa en la siguiente figura, el nivel inferior (nivel 0) se corresponde con una organización inmadura, los siguientes niveles van haciendo crecer a la organización en su madurez, hasta el máximo nivel, el nivel 5.

Figura 40. Niveles de Madurez de la Norma ISO 15504.

La consecución de los niveles de madurez es de forma escalonada, esto significa que para alcanzar un determinado nivel de madurez deben haberse alcanzado también los niveles inferiores.

Cada nivel de madurez estará formado por un conjunto de procesos, estos procesos se definen en los esquemas de certificación.

Características de la Norma [68]:

- Establece un marco y los requisitos para cualquier proceso de evaluación de procesos y proporciona requisitos para los modelos de evaluación de los procesos.
- Proporciona también requisitos para cualquier modelo de evaluación de organizaciones.
- Proporciona guías para la definición de las competencias de un evaluador de procesos.
- Actualmente tiene 10 partes: de la 1 a la 7 completas y de la 8 a la 10 en fase de desarrollo.
- Comprende: evaluación de procesos, mejora de procesos, determinación de capacidad.
- Proporciona, en su parte 5, un Modelo de evaluación de procesos para los procesos de ciclo de vida del software definidos en el estándar ISO/IEC 12207 que define los

procesos del ciclo de vida del desarrollo, mantenimiento y operación de los sistemas de software.

- Proporciona, en su parte 6, un Modelo de evaluación de procesos para los procesos de ciclo de vida del sistema definidos en el estándar ISO/IEC 15288 que define los procesos del ciclo de vida del desarrollo, mantenimiento y operación de sistemas.
- Proporcionará, en su parte 8, un Modelo de evaluación de procesos para los procesos de servicios TIC que serán definidos en el estándar ISO/IEC 20000-4 que definirá los procesos contenidos en la norma ISO/IEC 20000-1.
- Equivalencia y compatibilidad con CMMI. ISO forma parte del panel elaborador del modelo CMMI y SEI y viceversa, y se mantiene la compatibilidad y equivalencia de ésta última con 15504. Sin embargo CMMI-DEV aún no es un modelo conforme con esta norma (según lo requiere la norma ISO 15504 para todo modelo de evaluación de procesos).

4. ISO/IEC 250xx(SQUARE)

A. *Introducción*

El objetivo general de la creación del estándar ISO/IEC 25000 ***SQuaRE*** (Software Product Quality Requeriments and Evaluation) es organizar, enriquecer y unificar las series que cubren dos procesos principales:

1. Especificación de requerimientos de calidad del software.
2. Evaluación de la calidad del software, soportada por el proceso de medición de calidad del software.

Las características de calidad y sus mediciones asociadas pueden ser útiles no solamente para evaluar el producto software sino también para definir losrequerimientos de calidad.

La serie ISO/IEC 25000:2005 reemplaza a dos estándares relacionados: ISO/IEC 9126 (Software Product Quality) e ISO/IEC 14598 (Software Product Evaluation). [23]

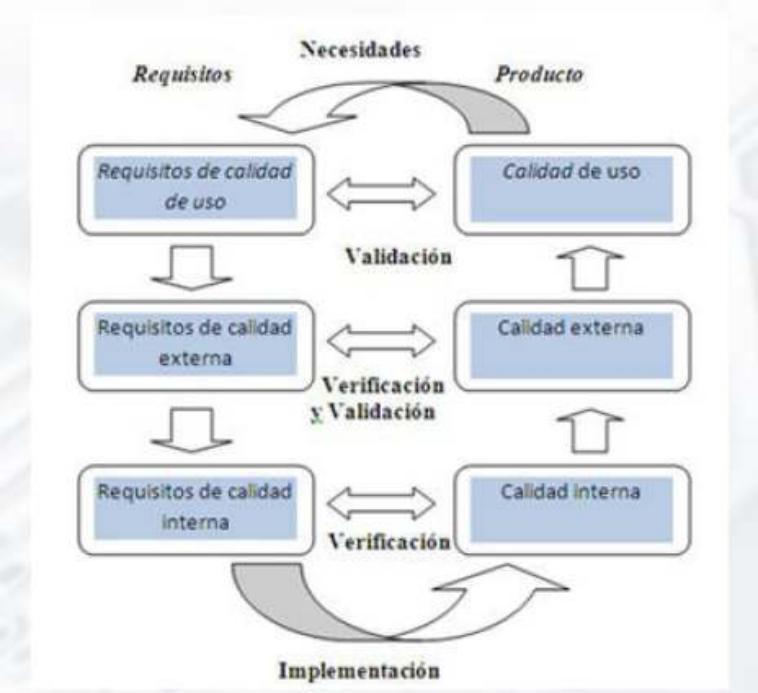

Figura 41. Modelo del Ciclo de Vida - Calidad del producto SW

Divisiones

- ***ISO/IEC 2500n***. División de **gestión de calidad**. Los estándares que forman esta división definen todos los modelos comunes, términos y referencias a los que se alude en las demás divisiones de SQuaRE.

- ***ISO/IEC 2501n.*** División del **modelo de calidad**. El estándar que conforma esta división presenta un modelo de calidad detallado, incluyendo características para la calidad interna, externa y en uso.

  Estudiaremos la norma ***ISO 25012*** en el apartado de calidad del dato.

- ***ISO/IEC 2502n.*** División de **mediciones de calidad**. Los estándares pertenecientes a esta división incluyen un modelo de referencia de calidad del producto software, definiciones matemáticas de las métricas de calidad y una guía práctica para su aplicación. Presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.

- ***ISO/IEC 2503n.*** División de **requisitos de calidad**. Los estándares que forman parte de esta división ayudan a especificar los requisitos de calidad. Estos requisitos pueden ser usados en el proceso de especificación de requisitos de calidad para un producto software que va a ser desarrollado o como entrada para un proceso de evaluación. El

proceso de definición de requisitos se guía por el establecido en la norma ISO/IEC 15288 (ISO, 2003).

- ***ISO/IEC 2504n.*** División de **evaluación de la calidad**. Estos estándares proporcionan requisitos, recomendaciones y guías para la evaluación de un producto software, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores.

- ***ISO/IEC 25050–25099.*** Estándares de extensión SQuaRE. Incluyen **requisitos para la calidad de productos de software** "Off-The-Self" y para **el formato común de la industria** (CIF) para informes de usabilidad.

Se han reservado los valores desde ISO/IEC 25050 hasta ISO/IEC 25099 para extensiones y "Technical Reports".

El contenido sigue lasiguiente estructura:

- Términos y definiciones
- Modelos de referencia
- Guía general
- Guías por división
- Estándares internacionales para especificación de requerimientos, planificación y gestión, medición y evaluación de la calidad del producto.

B. *Estado actual*

Para hacernos una idea de todo lo que abarca el proyecto SQuaRE se muestra el estado actual del mismo, dónde nos vamos a centrar en la norma ISO/IEC 25012:2008 Data Quality Model ya que es la más importante para nuestros objetivos.

ISO/IEC	Nombre	Stage	Status
25000:2005	Guide to SQuaRE	60.60	Published
25001:2007	Planning and management	60.60	Published
CD 25010	Quality model and guide	30.60	Under Development
25012:2008	Data Quality model	60.60	Published
25020:2007	Measurement reference model and guide	60.60	Published
TR 25021:2007	Quality Measure elements	60.60	Published
25030:2007	Quality Requirements	60.60	Published
FDIS 25045	Evaluation module for recoverability	50.20	Under Development
25051:2006	Requirements for quality of Commercial Off-The-Shelf (COTS) software product and instructions for testing	60.60	Published
DTR 25060	Common Industry Format (CIF) for Usability -- General Framework for Usability-related Information	40.20	Under Development
25062:2006	Common Industry Format (CIF) for usability test reports	60.60	Published

Figura 42. Estado actual ISO 25000 – SQUARE

Establece criterios para la especificación de requisitos de calidad de productos software, sus métricas y su evaluación. El siguiente esquema ofrece una visión de la gama de atributos y características de calidad que recoge el modelo.

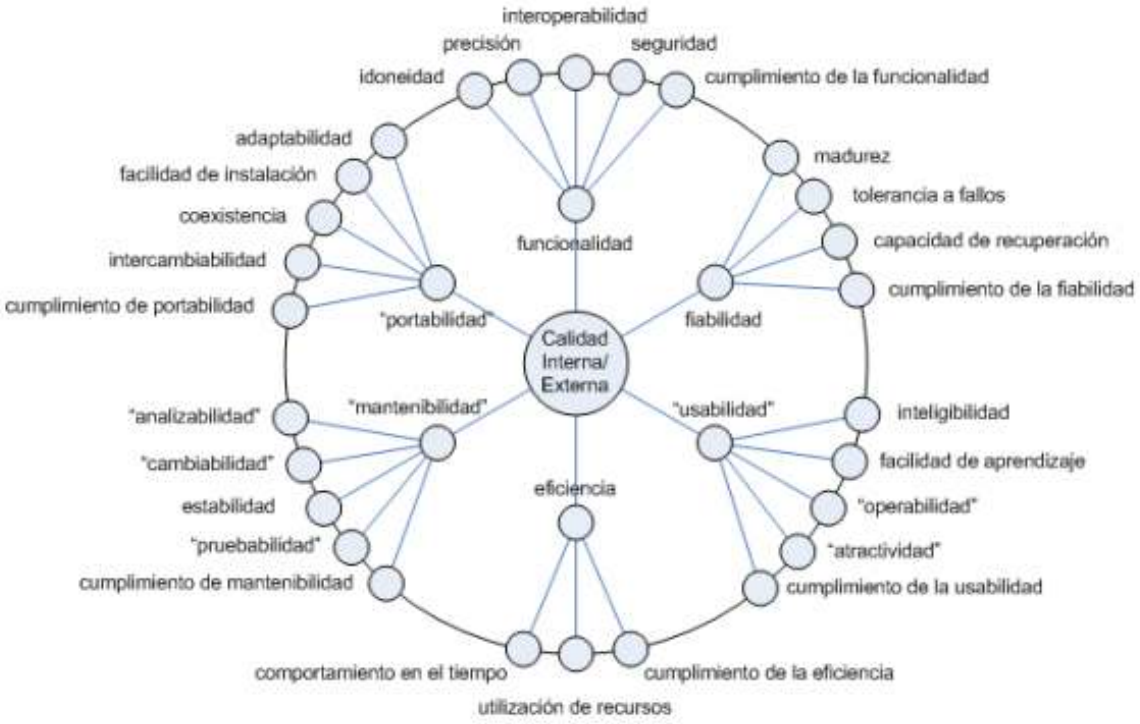

Figura 43. Atributos de calidad ISO 250xx

E. ISO 250xx Vs ISO 9126.

A. *Comparativa*

En el siguiente apartado estableceremos una serie de comparaciones entre la norma predecesora ISO 9126 centrándonos en su primer tomo y ISO250xx o ISO/IEC 25000.

- ISO250xx es una revisión de la ISO/IEC 9126:2001 y conserva las mismas características de calidad del SW centrándose en el producto.

- ISO 9126 pertenece a la primera generación de estándares de calidad de un producto SW mientras que ISO250xx pertenece a la segunda.

- La herencia del modelo de calidad de ISO 9126 en ISO250xx propone basar la calidad del producto SW en tres fases principales del ciclo de vida del producto SW (Producto bajo desarrollo, producto en operación y producto en uso).

- **ISO 9126 (en conjunto con la norma ISO 14598) tiene una serie de problemas que comentamos a continuación**:

  Problemas causados por cambios en el entorno y avances en las tecnologías de la información. Necesidades de una nueva arquitectura única de guías (difícil recordar todos los números de las normas).

  El estándar de requisitos de calidad no es propuesto como una parte de las series pero si como un estándar independiente esto provocará confusión a los usuarios.

- **Las diferencias principales entre ISO250xx y sus predecesores (ISO 9126 y ISO/IEC 14598) son los siguientes**:

  - Introducción a un nuevo modelo de referencia general.
  - Introducción de guías dedicadas y detalladas para cada división.
  - Introducción de elementos de medida de calidad dentro de la división de medida de calidad.
  - Introducción de la división de requisitos de calidad.
  - Incorporación y revisión de los procesos de evaluación.
  - Introducción de guías para uso práctico en forma de ejemplos.
  - Coordinación y armonización del contenido con la ISO/IEC 15939.

- **Las diferencias entre las características y subcaracterísticas del modelo de calidad interno y externo de la ISO 9126-1 y el modelo de calidad del producto software de ISO 250xx**:

Diferencias entre características y Subcaracterísticas		
SQuaRE	ISO/IEC 9126-1	Características
Adecuación funcional	Funcionalidad	El nuevo nombre es más preciso y no provoca confusiones con otros significados de funcionalidad
	Interoperabilidad	Movido a Compatibilidad
	Seguridad	Característica propia de ISO 9126-1
Disponibilidad	Madurez	Disponibilidad es mucho más importante que madurez
Robustez		Subcaracterística de SQuaRE.
Eficiencia de rendimiento	Eficiencia	Renombrado para no provocar conflictos con otras definiciones
Operabilidad	Usabilidad	Renombrado para no provocar conflictos con otras definiciones
Reconocimiento de adecuación	Comprensibilidad	El nuevo nombre de SQuaRE es mucho más preciso.
Facilidad de uso	Operabilidad	Simplemente se ha renombrado.
Util		Nueva subcategoría de SQuaRE
Accesibilidad técnica		Nueva subcategoría de SQuaRE
Seguridad	Seguridad	En SQuaRE es una característica, en la ISO 9126-1 es una subcategoría.
Compatibilidad		No estaba suficientemente declarado en las subcategorías de Portabilidad en la ISO 9126-1
Interoperabilidad		En la ISO 9126-1 es una subcaracterística de Funcionalidad.

Figura 44. Diferencias entre características y Subcaracterísticas

- **Para el modelo de calidad de uso, existen las siguientes diferencias entre las normas 9126-1 y ISO250xx:**

Diferencias para el modelo de calidad de uso		
SQuaRE	ISO/IEC 9126-1	Características
Usabilidad en uso		Nueva subcaracterística de SQuaRE
Flexibilidad en uso		Nueva característica de SQuaRE
Conformidad del contexto de uso		Nueva subcaracterística de SQuaRE
Extensión del contexto de uso		Nueva subcaracterística de SQuaRE
Salud y seguridad de operador		Nueva subcaracterística de SQuaRE
Salud y seguridad de público		Nueva subcaracterística de SQuaRE
Daño del entorno de uso		Nueva subcaracterística de SQuaRE
Daños comerciales de uso		Nueva subcaracterística de SQuaRE

Figura 45. Diferencias para el modelo de calidad de uso

B. *Conclusiones:*

ISO 9126 está dividida en tres partes:

9126-1 que contiene un modelo de calidad interna y externa, y un modelo de calidad de uso, 9126-1 que está formado por conjunto de métricas externas y finalmente 9126-3 que está formado por un conjunto de métricas internas.

ISO250xx (SQuaRE) es una revisión de 9126-1 y tiene las mismas características de calidad del software. Se centra en el producto.

Tiene tres revisiones, y cada una de ellas tiene un modelo de calidad del producto software con distintas características y subcaracterísticas.

Desde la primera revisión de Mayo de 2007 a la última revisión de Julio de 2008 ha habido una remodelación de las características y subcaracterísticas siendo la última versión la que tiene más precisión en dicho aspecto.

C. *Tendencia futura.*

ISO250xx es la siguiente generación a ISO 9126. Se seguirá utilizando la ISO 9126 hasta que pueda ser reemplazado totalmente por la familia ISO250xx. Actualmente se está trabajando en nuevas revisiones de la familia ISO250xx y es un estándar más extendido que ISO 9126.

F. ISO/IEC 25012

A. *Introducción*

Con la Norma ISO/IEC 25012:2008 "**Data Quality Model**" o Modelo de Calidad de Datos, se define la calidad de datos cómo el grado en que las características de los datos guardan las condiciones y sugiere las necesidades cuando es usado bajo condiciones específicas.

Las características de calidad de datos son las categorías de atributos de calidad de datos que llevan a la calidad de datos y que el modelo de calidad de datos es el conjunto de características de calidad que proveen un marco de trabajo con requerimientos específicos de calidad de datos y su evaluación.

La calidad de datos es un concepto multidimensional, por lo cual, para medirla es necesario descomponerla en características observables llamadas dimensiones de calidad de datos, en base a las cuales sea posible definirla, identificarla y medirla. El modelo propuesto por el estándar ISO/IEC 25012 [21] categoriza los atributos de calidad de datos en 15 características o dimensiones considerados desde dos puntos de vista:

- **Inherente al sistema**.

La calidad de datos inherente se refiere al grado en el cual las características de calidad del dato tienen el potencial intrínseco para satisfacer las necesidades implicadas cuando el dato es usado bajo condiciones específicas.

- **Dependiente del sistema**.

La calidad de datos dependiente del sistema se refiere al grado en el cual la calidad del dato es enriquecida y preservada dentro de un sistema de cómputo cuando el dato es usado bajo condiciones específicas.

B. *Dimensiones de calidad de datos*

A continuación se describen cada una de las dimensiones de calidad de datos:

Dimensiones de calidad de datos ISO/IEC 25012	
Inherentes	
Dimensión	**Descripción**
Exactitud (Accuracy)	El grado en el cual el dato tiene atributos que correctamente representan el valor correcto del atributo intencionado de un concepto o evento en un contexto específico de empleo.
Completitud (Completeness)	El grado al cual el dato del sujeto asociado con una entidad tiene valores para todos los atributos esperados e instancias de entidad relacionadas en un contexto específico de uso.
Consistencia (Consistency)	El grado en el cual el dato tiene los atributos que son libres de contradicción y son coherente con otros datos en un contexto específico de uso.
Credibilidad (Credibility)	El grado en el cual el dato tiene atributos que son considerados como verdaderos y creíbles por usuarios en un contexto específico de uso.
Actualidad (Currentness)	El grado en el cual el dato tiene los atributos que son del período correcto en un contexto específico de uso.
Inherentes y dependientes del sistema	
Accesibilidad (Accessibility)	El grado en el cual el dato puede ser accedido en un contexto específico de uso, en particular por la gente que necesita el soporte de tecnología o una configuración especial debido a alguna inhabilidad (incapacidad).

Conformidad (Compliance)	El grado en el cual el dato tiene atributos que se adhieren a normas, convenciones o regulaciones vigentes y reglas similares relacionadas con la calidad de datos en un contexto específico de uso.
Confidencialidad (Confidentiality)	El grado en el cual el dato tiene los atributos que aseguran que éste es sólo accesible e interpretable por usuarios autorizados en un contexto específico de uso.
Eficiencia (Efficiency)	El grado en el cual el dato tiene los atributos que pueden ser procesados y proporciona los niveles esperados de funcionamiento (desempeño) usando las cantidades y los tipos de recursos apropiados en un contexto específico de uso.
Precisión (Precision)	El grado en el cual el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso.
Trazabilidad (Traceability)	El grado en el cual el dato tiene atributos que proporcionan un rastro de auditoría de acceso a los datos y de cualquier cambio hecho a los datos en un contexto específico de uso.
Entendibilidad (Understandability)	El grado en el cual el dato tiene atributos que le permiten ser leído e interpretado por usuarios, y es expresado en lenguajes apropiados, símbolos y unidades en un contexto específico de uso.
Dependientes del sistema	
Disponibilidad (Availability)	El grado en el cual el dato tiene atributos que le permiten ser recuperados por usuarios autorizados y/o aplicaciones en un contexto específico de uso.
Portabilidad (Portability)	El grado en el cual el dato tiene los atributos que le permiten ser instalado, substituido o movido de un sistema a otro conservando la calidad existente en un contexto específico de uso.
Recuperabilidad (Recoverability)	El grado en el cual el dato tiene atributos que le permiten mantener y conservar un nivel especificado de operaciones y calidad, aún en caso de falla, en un contexto específico de uso.

Figura 46. Dimensiones de calidad de datos ISO IEC 2012

Dado que en SO 25012 se denominaran características de calidad muchas publicaciones posteriores usan él mismo término, aunque varios autores las denominan como dimensiones, atributos, factores.

G. ISO/IEC12207

A. *La Norma*

ISO/IEC 12207 Information Technology / Software Life Cycle Processes, es el estándar para los **procesos de ciclo de vida del software** [70].

Dado que ya hemos visto durante la memoria dos metodologías que incluyen como realizar un proyecto de BI que incluye almacenes DW no vamos a profundizar mucho en la misma, pero si la vamos a conocer para comprender mejor el porqué de algunas de las medidas que incluyen dichas metodologías.

El modelo ISO 12207:2008 establece un conjunto de buenas prácticas para guiar a las organizaciones en la mejora de sus procesos de desarrollo y mantenimiento software.

La estructura del estándar ha sido concebida de manera que pueda ser adaptada a las necesidades de cualquiera que lo use. Para conseguirlo, el estándar se basa en dos principios fundamentales:

1. **Modularidad.**

Se pretende conseguir procesos con un mínimo acoplamiento y una máxima cohesión

2. **Responsabilidad.**

Se busca establecer un responsable para cada proceso, facilitando la aplicación del estándar en proyectos en los que pueden existir distintas personas u organizaciones involucradas, no importando el uso que se le dé a este.

En concreto, define 43 procesos que pueden aplicarse durante la **adquisición** de unproducto o servicio software y durante **el suministro, desarrollo, operación, mantenimiento y evolución de productos software.**

Figuran en el siguiente esquema representativo, para hacernos una idea del alcance de la norma:

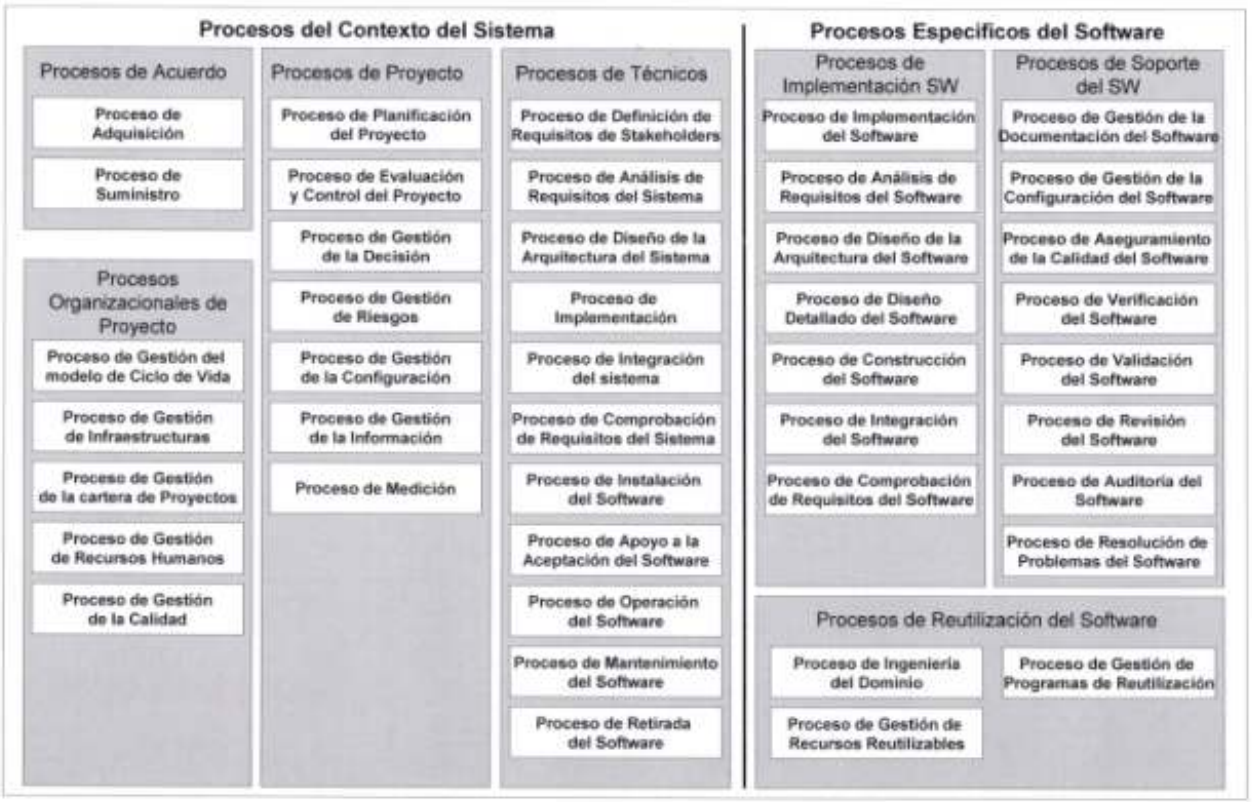

Figura 47. ProcesosISO12207: 2008

Como puede verse la norma es muy extensa y certificar a las compañías en todos los procesos es una tarea casi imposible, por ello en AENOR se ha realizado un esquema de certificación que veremos a continuación:

B. *Esquema de certificación de AENOR*

Para ayudar a las organizaciones que se encuentren en un proyecto de mejora de procesos a definir e implantar los procesos siguiendo las buenas prácticas y recomendaciones especificadas en el modelo ISO 12207:2008, en el esquema de certificación de AENOR se ha desarrollado una "Guía básica para la implantación de procesos ISO 12207:2008".

En este contexto, actualmente la guía contiene los procesos y sus resultados del proceso (RP) requeridos para implantar los niveles 1 y 2 de madurez.

El contenido de esta guía se corresponde con una traducción realizada por Kybele Consulting del modelo de procesos de referencia para la industria del software ISO 12207:2008, ya que en la actualidad no se dispone de una traducción oficial.

En este sentido, los procesos que se encuentran en el alcance son los siguientes:

- Proceso de Suministro (SUM)
- Proceso de Definición de los Requisitos de Usuario (RQU)
- Proceso de Análisis de los Requisitos del Sistema (RQSIS)
- Proceso de Gestión del Modelo del Ciclo de Vida (MCV)
- Proceso de Planificación del Proyecto (PP)
- Proceso de Evaluación y Control del Proyecto (ECP)
- Proceso de Gestión de la Configuración del Software (GCS)
- Proceso de Gestión de la Configuración (GC)
- Proceso de Medición (MED)
- Proceso de Aseguramiento de la Calidad Software (ACS)

De los cuales destacamos que pueden ser importantes de conocer para proyectos de BI, con respecto a otras normas y publicaciones los de Definición de los Requisitos de Usuario (RQU) por la importancia que toma en la construcción del DW, el de Planificación del Proyecto (PP) para no olvidarnos que un proyecto de BI sigue siendo un proyecto de desarrollo de SW y el proceso de Medición (MED).

H. IEEE 730

A. *La Norma.*

El estándar **IEEE 730 "Software Quality Assurance Plans"** (SQAP), es una norma que permite estandarizar, proporcionando los requisitos mínimos aceptables, la preparación y contenido de los **Planes de Aseguramiento de la Calidad del Software**.

El estándar es muy corto, de unas 10 páginas en las que define una lista de 16 secciones para el documento, las cuales consideran las distintas actividades que se deben llevar a cabo con el fin de definir y documentar el plan de aseguramiento de la calidad del producto software.

También plantea la posibilidad de agregar secciones adicionales al plan de calidad, según se requiera.

Cada sección viene definida con una serie de directrices a llevar a cabo para la creación de un plan de aseguramiento de la calidad conforme a la norma.

Las 16 secciones son las siguientes:

1. Purpose – Propósito.
2. Reference documents – Documentos de referencia.
3. Management – Administración.
4. Documentation – Documentación.
5. Standards, practices, conventions, and metrics - Estándares, prácticas, convenciones y métricas.
6. Software reviews – Revisiones de SW
7. Test - Planes de pruebas
8. Problem reporting and corrective action - Informes de problemas y corrección de errors.
9. Tools, techniques, and methodologies – Herramientas, técnicas y metodologías.
10. Media control – Control de medios.
11. Supplier control - Control de proveedores
12. Records collection, maintenance, and retention – Recolección de registros, mantenimiento y retenciones.
13. Training - Formación.
14. Risk management – Gestión de riesgos
15. Glossary
16. SQAP change procedure and history – SQAP histórico y cambios de procedimiento.

Entre esas 16 secciones destacamos las siguientes que pueden ser interesantes para nuestros objetivos:

- **Propósito** (Sección 1).

Se deja muy claro que el propósito de la norma es establecer los requisitos mínimos uniformes aceptables para la reparación de los planes que garanticen de calidad de software especificando que su propósito no es sustituir, revisar o modificar las normas existentes dirigidas a industrias o aplicaciones específicas.

Por tanto, al ser nuestro objetivo una aplicación muy específica, sólo identificaremos los puntos más importantes de la misma que pueden interesarnos.

Un plan de pruebas deberá comprender las secciones que se indican a partir de ésta para asegurar la calidad del producto SW. Nosotros destacamos lo información que aporta sobre las que recogemos ya que las definiciones son a muy bajo nivel.

- **Documentación** (Sección 4).

Se especifica que se llevan a cabo dos funciones:

1. Identificar la documentación que rige el desarrollo, verificación, validación, uso y mantenimiento del software.
2. Lista los documentos que deben ser auditados o revisados para verificar la adecuación a la norma.

- **Estándares, prácticas, convenciones y métricas** (Sección 5).

Identifica las normas, los métodos, convenciones y técnicas estadísticas que se deben utilizar, los requisitos de calidad y las métricas que se deben aplicar.

Establece que podemos apoyarnos en otras normar como IEEE1219 y IEEE1228 para las medidas del producto y el proceso.

I. CMMI

A. *Introducción*

Capability Maturity Model Integration o CMMI es un modelo para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas de software.

Tiene por objetivo obtener mejores productos software.

Recoge las mejores prácticas identificadas agrupadas en tres áreas que son cubiertas por los siguientes modelos o constelaciones de CMMI:

- Desarrollo o CMMI-DEV.

Recoge las buenas prácticas para la mejora y evaluación de procesos para el desarrollo, mantenimiento y operación de sistemas software.

- Adquisición o CMMI-ACQ.

Recoge las buenas prácticas para la adquisición de productos y servicios.

- Servicios o CMMI-SVC.

Recoge las buenas prácticas para organizaciones proveedoras de servicios.

Los procesos se agrupan en cinco niveles que proporcionan el camino para mejorar la visibilidad y el control de las actividades.

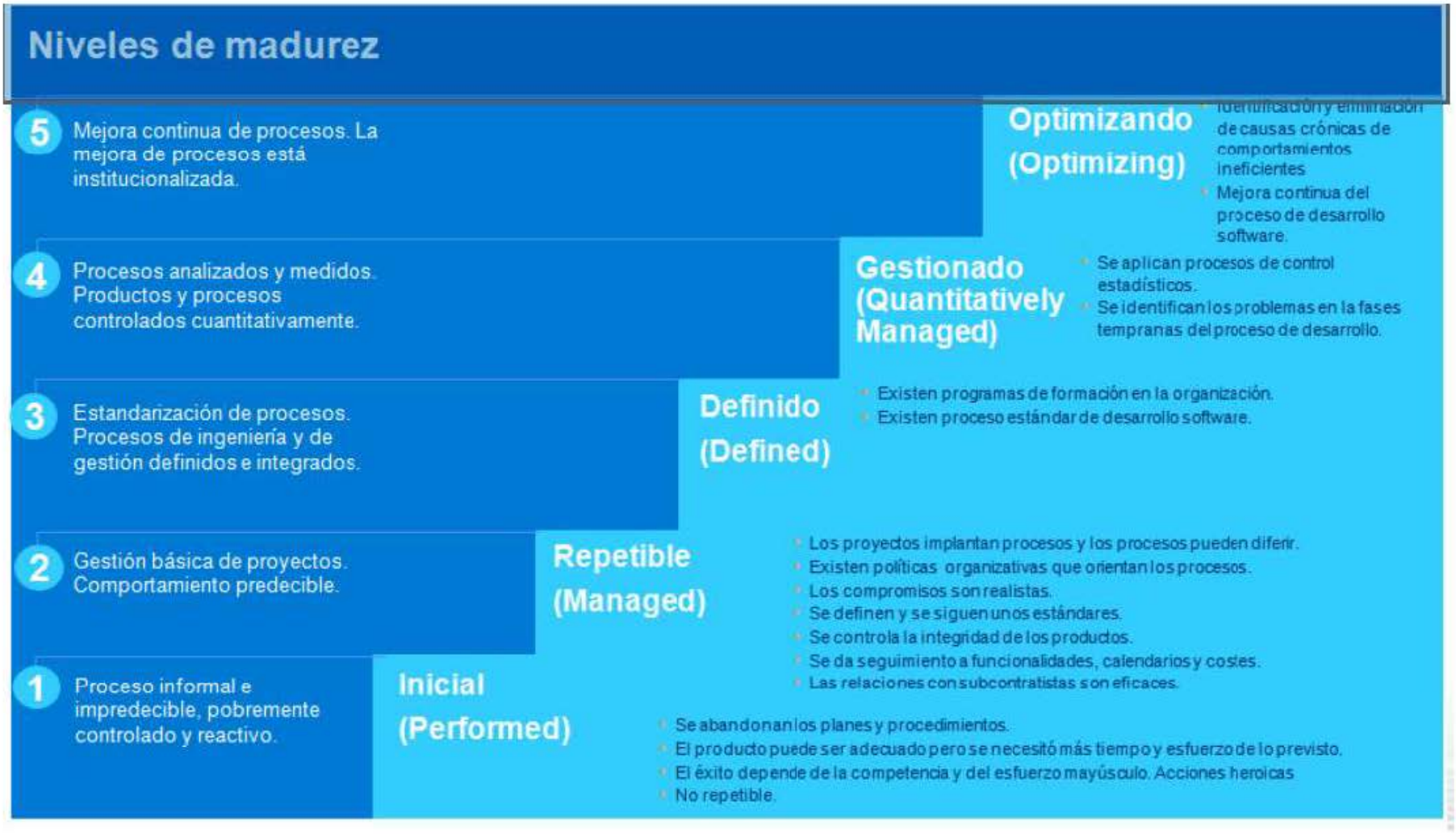

Figura 48. CMMI – Niveles de madurez.

Para nuestros objetivos nos interesa especialmente CMMI-DEV y en el caso de optar para contratar soluciones del mercado que refuercen la base de nuestro sistema de BI podemos seguir CMMI –ACQ. Por tanto mostramos un resumen del mapeo sobre cada nivel de madurez y el enfoque de estas dos constelaciones o modelos.

B. *CMMI-ACQ*

Tiene por objetivo proporcionar una guía para la aplicación de buenas prácticas en el proceso de adquisición de productos y servicios que satisfagan las necesidades del cliente.

NIVEL		ENFOQUE	ÁREAS DE PROCESO
5	En optimización	Mejora continua del proceso	Innovación y despliegue organizativo Análisis causal
4	Gestionado cuantitativamente	Gestión cuantitativa	Performance de procesos organizativos Gestión cuantitativa de proyectos
3	Definido	Estandarización del proceso	Desarrollo de requisitos Gestión técnica de la adquisición Integración de producto Verificación de la adquisición Validación de la adquisición Enfoque al proceso organizativo Definición del proceso organizativo Formación organizativa Gestión integrada del proyecto Gestión de riesgos Análisis y toma de decisiones
2	Gestionado	Gestión de proyectos básica	Gestión de requisitos Planificación del proyecto Seguimiento y control del proyecto Licitaciones y acuerdos con proveedores Desarrollo de requisitos para adquisición Gestión de contratos Medición y análisis Aseguramiento de la calidad Gestión de la configuración
1	Inicial	Sin áreas de proceso – ¡el trabajo se realiza de alguna manera!	

Figura 49. Cuadro resumen CMMI-ACQ.

C. *CMMI-DEV*

Tiene por objetivo proporcionar una guía para la aplicación de buenas prácticas en el proceso de desarrollo de productos y servicios software que satisfagan las necesidades del cliente.

NIVEL		ENFOQUE	ÁREAS DE PROCESO
5	En optimización	Mejora continua del proceso	Innovación y despliegue organizativo Análisis causal
4	Gestionado cuantitativamente	Gestión cuantitativa	Performance de procesos organizativos Gestión cuantitativa de proyectos
3	Definido	Estandarización del proceso	Desarrollo de requisitos Solución técnica Integración de producto Verificación Validación Enfoque al proceso organizativo Definición del proceso organizativo Formación organizativa Gestión integrada del proyecto Gestión de riesgos Análisis y toma de decisiones
2	Gestionado	Gestión de proyectos básica	Gestión de requisitos Planificación del proyecto Seguimiento y control del proyecto Gestión de acuerdos con proveedores Medición y análisis Aseguramiento de la calidad Gestión de la configuración
1	Inicial	Sin áreas de proceso – ¡el trabajo se realiza de alguna manera!	

Figura 50. Cuadro resumen CMMI-DEV.

J. PMBOK

La Guía PMBOK® o Project Management Body of Knowledge es un estándar en la

Administraicón de proyectos desarrollado por el Project Management Institute (PMI) que comprende dos grandes secciones, la primera sobre los procesos y contextos de un proyecto y la segunda sobre las áreas de conocimiento específico para la gestión de un proyecto. [73]

Para el objetivo de nuestro proyecto nos interesaremos más en el segundo área, para la gestión de proyectos.

La guía PMBOK estables un flujo sobre el que basar nuestra gestión de proyectos que contempla las siguientes fases o etapas y que puede verse la interrelación entre ellas en la siguiente figura.La fases son una variación del ciclo de Deming para la mejora continua.

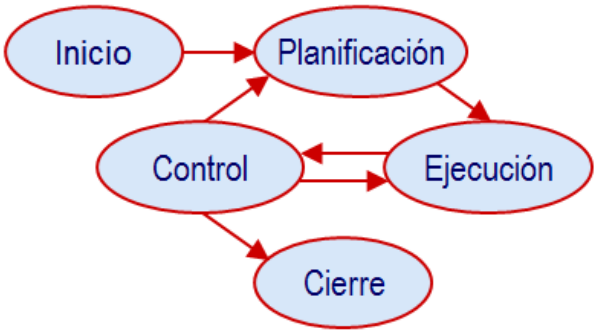

Figura 51. Flujo de gestión según PMBOK

La guía nos propone una serie de documentación en forma de documentos, entregables y formularios que se deberán ir cubriendo a lo largo del ciclo de vida del proyecto.

1. **Fase de iniciación**

Define y autoriza el proyecto o una fase del mismo. Está formado por dos procesos.

La documentación a completar es:

- Ficha preliminar
- Documento de Plan de Viabilidad (opcional)
- Documento de Ficha inicial

2. **Fase de planificación**

Define, refina los objetivos y planifica el curso de acción requerido para lograr los objetivos y el alcance pretendido del proyecto. Está formado por veinte procesos.

La documentación a completar es:

- Documento de Plan de proyecto

3. **Fase de Ejecución**

En la fase de ejecución se especifica un **modelo de ciclo de vida**. Está basado en el modelo secuencial, pero se da especial importancia a la etapa de especificación del software, y sobre todo el análisis de requisitos. En concreto, las subfases que permiten la especificación de la solución, individualmente pueden ser evolutivos, pero como etapa global debe ser secuencial, por ejemplo, no se debe empezar el análisis funcional sin obtener la aprobación de los requisitos, y no debe empezar el diseño técnico sin la aprobación del análisis funcional. Sin embargo, dentro de la subfase de requisitos, se podrán hacer tantas iteraciones con el usuario como sean necesarias. Lo que se pretende con esta técnica es mitigar el riesgo de una mala especificación del software y como consecuencia una mala construcción y evitar en la medida de lo posible, continuos retrabajos. Está formada por 8 procesos.

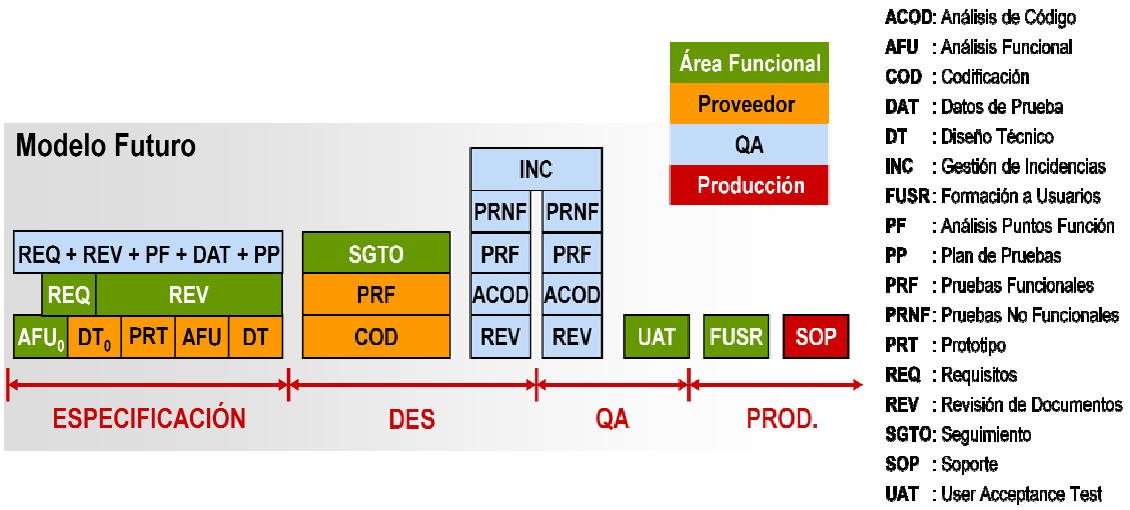

Figura 52. Ciclo de Vida PMBOK

Se definen cuatro grandes actividades dentro del modelo de ciclo de vida.

1. **Especificación.**

Responsable del análisis y diseño de la aplicación.

2. **Desarrollo.**

Responsable fundamentalmente de la construcción de los requisitos definidos en la fase de especificación.

3. **Calidad.**

Responsable de todas las tareas relativas al aseguramiento y control de la calidad.

4. **Producción.**

Es la etapa de implantación en los entornos de producción.

Los documentos de la fase de ejecución serán los siguientes:

- Documento de Análisis Funcional y Plan de Pruebas Global (opcional)
- Documento de Arquitectura Técnica (opcional)
- Documento de Análisis Funcional y Plan de Pruebas
- Documento de Arquitectura Técnica
- Documento de Diseño Técnico
- Documento de Instalación
- Documento de Plan de Pruebas de Carga y Estrés
- Documento de Administración de la aplicación
- Documento de Manual de Usuario

5. Fase de Seguimiento y Control

Mide, supervisa y regula el progreso y desempeño del proyecto, para identificar áreas en las que el plan requiera cambios. Está formado por diez procesos.

La documentación a completar es:

- Documento de Seguimiento y Control

6. Fase de Cierre

Formaliza la aceptación del producto, servicio o resultado, y termina ordenadamente el proyecto o una fase del mismo. Está formado por dos procesos.

La documentación a completar es:

- Documento de Cierre de Proyecto

K. COBIT

COBIT son las siglas de **"Control Objetives for Information and related Technology"** en inglés. Es una guía de buenas prácticas para el buen Gobierno Corporativo de las TIC.

Dado el alto grado de ligamiento que existe entre los proyectos de BI y la dirección de las empresas se cree conveniente al menos hacer referencia al marco para tener en cuenta que nos propone.

La misión de COBIT es "investigar, desarrollar, publicar y promocionar un conjunto de objetivos de control generalmente aceptados para las tecnologías de la información que sean autorizados (dados por alguien con autoridad), actualizados, e internacionales para el uso del día a día de los gestores de negocios (también directivos) y auditores". [76]

Gestores, auditores, y usuarios se benefician del desarrollo de COBIT porque les ayuda a entender sus Sistemas de Información (o tecnologías de la información) y decidir el nivel de seguridad y control que es necesario para proteger los activos de sus compañías mediante el desarrollo de un modelo de administración de las tecnologías de la información. [76]

Para nuestro propósito se puede utilizar el marco metodológico de buenas prácticas que aporta COBIT, para el establecimiento de indicadores, basado en un encadenamiento en cascada de objetivos e indicadores.

El marco enumera una serie de tareas que se pueden acometer agrupadas en diferentes procesos que indican cómoactuar en respuesta a las necesidades que se identifiquen.

Los grandes procesos a los que se hace referencia agrupando tareas son los siguientes:

- Información
- Planificar y evaluar.
- Adquirir e implantar.
- Entregar y soportar.
- Monitorizar y evaluar.
- Recursos de TI.

Es siguiente esquema muestra las relaciones existentes entre los procesos indicados y las tareas asociadas a algunos de ellos.

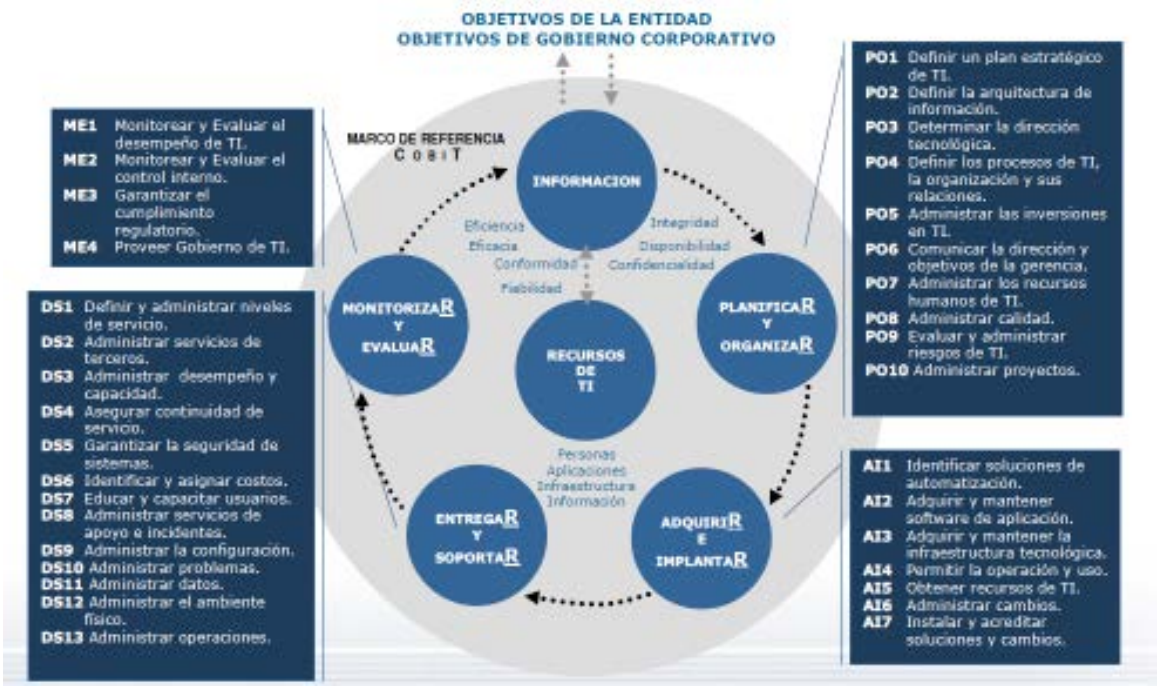

Figura 53. Marco de buenas prácticas de COBIT

El modelo COBIT ayudará en la identificación y puesta en marcha de una serie de objetivos de control y controles que tengan por meta la implantación de políticas, procedimientos, prácticas y estructuras organizativas diseñadas para garantizar, de forma razonable los siguientes hechos:

- Que se alcancen los objetivos del negocio.
- Que se prevengan o se detecten y corrijan los eventos no deseados.

Asimismo, el marco de referencia COBIT ayudará:

- Haciendo posible mapear las metas de TI a las metas del negocio y viceversa
- Aportando una mejor alineación (sincronización), basada en un enfoque del negocio
- Proporcionando una visión, comprensible para el resto de Sub. y Gerencias, de lo que es TI
- Identificando con claridad la propiedad y las responsabilidades, sobre la base de su orientación a procesos
- Aportando la aceptabilidad general por parte de terceras partes y entidades reguladoras
- Favoreciendo un entendimiento compartido entre todos los involucrados, basado en un lenguaje común
- Y cumpliendo con los requisitos de otros marcos de control interno de nivel superior, como COSO, pero para el entorno de TI.

L. SELECCIÓN DE DIRECTRICES

Para la depuración de las *guidelines* se han estudiado las Normas, Estándares y Guías de buenas prácticas identificadas en la memoria y se ha elegido algunas de ellas en detrimento del resto, vamos a identificar con que apartados de qué documentos nos quedamos para nuestra selección y vamos a identificar el porqué de la elección.

Figura 54. Marco de calidad

Se ha tenido desde un primer momento, que la necesidad de establecer un marco general dónde sustentar la base de nuestro SI en cuanto a calidad era algo necesario sin tener en cuenta especificaciones que lo pudieran limitar, por ello nos basamos en **la Norma ISO 9001 y concretamente en ISO 90003 para el establecimiento del marco general**.

La norma ISO 9001 es muy conocida y se reconoce en gran medida a las empresas que se certifican en ella o adoptan su metodología, por ello creemos que para establecer el marco general nos debemos basar en la norma ISO 90003 que nos indica las especificaciones de cómo aplicar la norma ISO 9001 a los procesos de Software.

El hecho de basarnos en la norma para las guidelines reside en el aspecto que aporta para que la organización adopte un sentimiento de cultura de empresa en base a la implicación de todos los miembros de la misma, por ello nos quedaremos con su sección cuarta **"Responsabilidad de la dirección"** ya que expresa que "La dirección de la empresa debe definir y documentar su política y sus objetivos con respecto a la calidad. Las responsabilidades, autoridades y

relaciones entre todo personal, cuyo trabajo afecte la calidad del producto, deben ser definidas."

Además especifica una serie de directrices adecuadas para la **gestión de recursos** a través del apartado de revisión de contratos dónde indica que "La empresa debe establecer y mantener procedimientos para la revisión de los contratos y para la coordinación de estas actividades. De esta forma asegurar que la empresa tiene la capacidad de cumplir con todos los requerimientos contractuales.", hecho significativo en proyectos de BI dónde la volatilidad de la plantilla puede resultar muy dañina para la consecución de los objetivos.

Una vez establecido el marco general, queremos ir poco a poco aumentando el grado de detalle, por ello el siguiente paso será identificar el sistema. Con la palabra identificar queremos decir conocer el alcance que tendrá el mismo y los **requerimientos** que lo formarán. Sin duda la mejor opción, debido su alto grado de aceptación y adaptabilidad, es la familia de **normas ISO 25000**.

Concretamente nos basaremos en la división **ISO 2503n** de **requisitos de calidad**. Con ella se usan las indicaciones aportadas en el proceso de especificación de requisitos de calidad para un producto SW que va a ser desarrollado o cómo entrada para un proceso de evaluación, característica que tendremos en cuenta para el aseguramiento de la calidad. Nos basaremos en la división **ISO 2504n** para la **evaluación de la calidad** que nos proporciona requisitos, recomendaciones y guías para la evaluación de un producto SQ, tanto si la llevan a cabo evaluadores, como clientes o desarrolladores. **ISO 2502n** para la **medición de la calidad** ya que presenta aplicaciones de métricas para la calidad de software interna, externa y en uso.

Para la evaluación de la calidad necesitaremos de más parámetros para poder establecer indicadores y medir. Por ello en el siguiente apartado nos vamos a centrar en la **calidad del dato**, conociendo las propiedades que creemos más importantes para que un dato sea de calidad dónde si aseguramos la calidad de la fuente los procesos dependen únicamente de ellos y no arrastran deficiencias en el resto del proceso.

Un aspecto del que se ha hablado mucho en la memoria pero en el que no nos hemos centrado directamente son los **riesgos y la viabilidad de los sistemas de BI**, por ello nos basaremos en COBIT para la gestión de los riesgos. Creemos que es mejor basarnos en una guía del nivel de gobierno de TI por la repercusión económica y estructural que conlleva este tipo de proyecto que en una norma más centrado en un solo proyecto, por ello nos basamos en COBIT.

A partir de aquí surgen dos incógnitas bien diferenciadas que será la implementación del producto y la posterior gestión del mismo.

Para la **metodología de desarrollo** del sistema nos basaremos en el ciclo de vida de Kimball por su alto grado de repercusión sobre los DW haciendo mención a una posible ventana de

adaptación de la filosofía de Inmon para aquellas entidades que se encuentren con un modelo de procesos de gran madurez y poco cambiante.

Para la **gestión del proyecto** usaremos la guía PMBOK, que aporta una gran variedad de buenas prácticas que adoptar en todos los sectores a los que puede afectar el proyecto de desarrollo. Dadas las características de un desarrollo de BI no podemos olvidarnos de establecer un **proceso de mejora continua** dónde usaremos la constelación CMMI-DEV para saber el nivel de madurez de los procesos y guiarnos a través de sus resultados para optimizar las decisiones a tomar en base a la mejora en busca de objetivos reales.

3. Calidad de los datos: Estado del Arte y Normas de referencia

1. Introducción

Debido al carácter de importancia que se imprime sobre las decisiones tomadas en base a los DW se han publicado varias investigaciones acerca de características de calidad de datos y sobre los factores críticos a tener en cuenta a la hora de desarrollar un sistema de BI.

En la mayoría de ellas se toman como referencia las normas o estándares antes mencionados y aportan la definición de ciertas características en la calidad de los datos basadas en estudios, aproximaciones y encuestas o se basan en la experiencia de proyectos finalizados para aportar factores críticos de éxito.

A continuación vamos a mostrar un resumen de la publicaciones que tienen más seguidores o han aportado un mayor peso específico en la calidad de los datos para los DW.

En primer lugar nos centraremos en las características de calidad de los datos dónde y las vamos a comparar con la norma ISO/IEC 25012 para sacar una serie de conclusiones dónde se define con mayor homogeneidad las características de calidad de los datos.

A continuación haremos un repaso sobre las publicaciones que hacen hincapié sobre los factores de éxito para agruparlos de forma genérica y establecer un modelo basado en la experiencia.

2. Características del dato de calidad.

A. Publicaciones.

A continuación vamos a enunciar una serie de publicaciones agrupadas en cinco apartados sobre características en la calidad de los datos que se corresponden con el **DWQ Project**, las publicaciones de **Wang y Strong** sobre el marco conceptual de calidad del dato, las publicaciones del grupo de **Leo L. Pipino** sobre las evaluaciones de calidad del dato, las publicaciones de **Rudra y Yeo** sobre las cuestiones claves para el logro de la calidad en los datos junto a la consistencia en los DW, además veremos una aproximación del marco de Wang y Strong que realiza Leithesier R aplicado al entorno de la salud.

Estas publicaciones tienen fuerte dedicación a las características de los datos para que sean de calidad, por lo que vamos a compararlas con la norma ISO/IEC 25012 que define una serie de características agrupadas para intentar definir cuál serían un dato de calidad.

Se podría concluir con que un DW cuyos datos siguieran las características de calidad que se indican en el siguiente apartado, tienen bastantes más probabilidades de ser de calidad que sistemas que no lo tengan en cuenta.

A. *DWQ Project.*

El proyecto de Calidad de almacenes de Datos (***DWQ Project***) [24] realizó varias publicaciones dónde la más destacada ***es "Architecture And Quality In Data Warehouses: An Extended Repository Approach"*** [25] dónde se definen varias características de calidad de datos.

Los **objetivos** del proyecto son:

- Enriquecer semánticamente las meta bases de datos con modelos formales de calidad de información.
- Enriquecer semánticamente los modelos de las fuentes de datos.
- Enriquecer semánticamente los modelos de esquemas de almacenes de datos.
- Ofrecer ventajas de calidad en la implementación de estos almacenes

Las **características** que define son:

- **Accesibilidad (Accesibility):** está relacionada con la posibilidad de acceder a los datos por medio de consultas.

- **Seguridad (Security):**describe las políticas de autorización y privilegios que cada usuario tiene para consultar los datos.

- **Disponibilidad del sistema (System Availability):** describe el porcentaje de tiempo que la fuente o el sistema de almacén de datos está disponible.

- **Disponibilidad transaccional (Transactional Availability)**: describe el porcentaje de tiempo que la información en los almacenes de datos o fuentes está disponible debido a la ausencia de procesos de actualización de datos, los cuales bloquean la escritura en estos.

- **Utilidad (Usefulness):**describe la característica temporal (oportunidad) de los datos así como la receptividad del sistema.

- **Actualidad (Currency):**describe cuando la información fue ingresada en la fuente o/y en el almacén de datos.

- **Volatilidad (Volatility):**describe el período de tiempo en que la información es validad a en el mundo real.

- **Interpretabilidad del modelo (Interpretatibility of model)**: describe el grado para el que el almacén de datos es eficientemente modelado para ser repositorio de información. Es decir, lo más fácil en que se pueden realizar las consultas. (Esta interpretabilidad no se orienta tanto a los datos como al esquema).

- **Completitud (Completeness):** describe el porcentaje en el que la información ingresada a las fuentes y/o almacenes representa al mundo real. Por ejemplo, la completitud podría ser el índice de extensión el cual una cadena describe una dirección y que actualmente encaja en el tamaño del atributo que representa la dirección.

- **Credibilidad (Credibility):** describe la credibilidad de la fuente que proporciona la información. ***En este caso se refiere explícitamente a la fuente por lo cual se podría denominar reputación.***

- **Exactitud (Accuracy):**describe la precisión y/o exactitud de los datos de entrada después de haber realizado el proceso de ingreso en las fuentes.

- **Consistencia (Consistency):**describe la coherencia lógica de la información.

- **Interpretabilidad de los datos (Data interpretability):**concierne con la descripción de los datos (es decir, disponer de datos de los sistemas heredados y datos externos, tablas de descripción de las bases de datos relacionales, llaves primarias y foráneas, alias, predefinidas, dominios, explicación de códigos de valores, etc.).

B. *Wang y strong*

Una obra importante para el estudio de las características de calidad de datos es el marco conceptual de calidad de datos de Wang y Strong [26]. En el marco se analizan varios atributos de calidad de datos desde la perspectiva de las personas que utilizan los datos. Wang y Strong identificaron un conjunto completo de características de calidad de datos y se añadieron varias características sobre los mismo como son credibilidad, valor añadido, interpretabilidad, accesibilidad entre otras. Estas características fueron agrupadas en cuatro amplias categorías que son intrínsecas, contextuales, representacionales y accesibilidad dando como resultado su conocido marco conceptual de datos que puede verse a continuación:

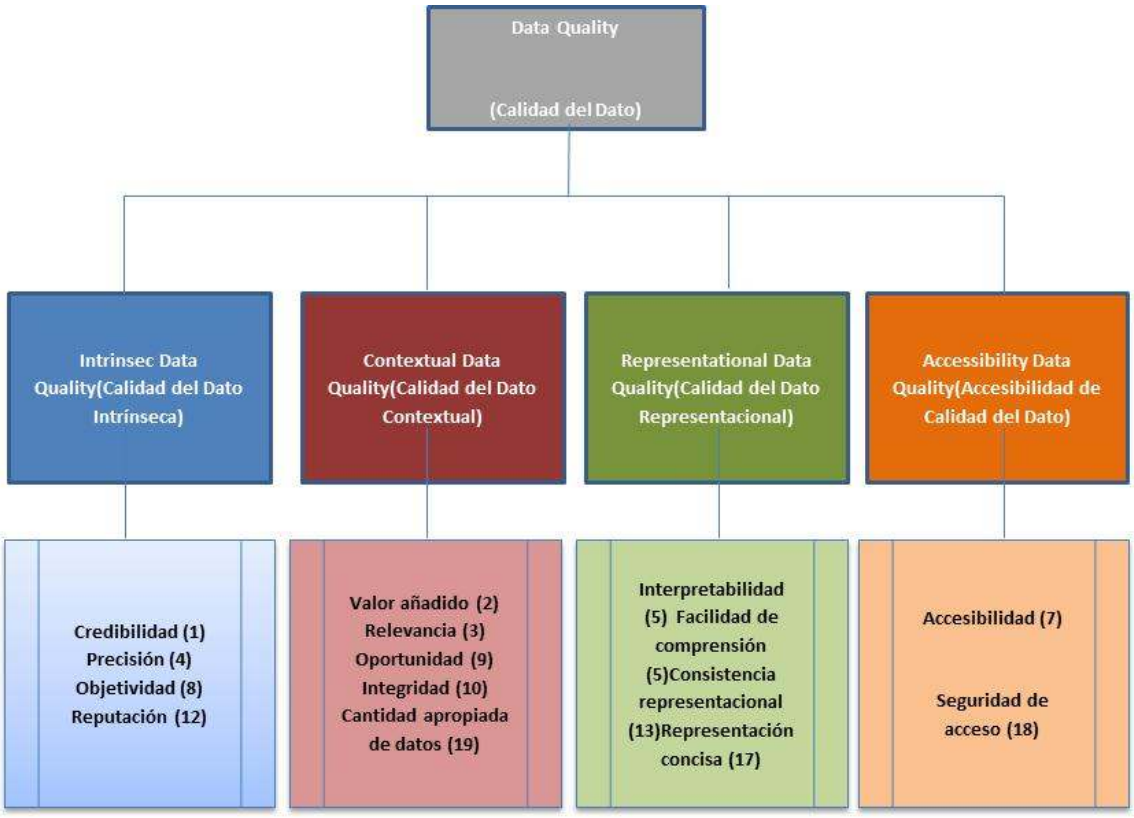

Figura 55. Marco de calidad de Wang y Strong

Las definiciones sobre las cuatro categorías y sus características son las siguientes:

- **Calidad de datos intrínseca:** Es la que denota que los datos tienen calidad por ellos mismo, es decir, calidad inherente a ellos por su razón de existencia.

    - **Credibilidad (Believability):**El hecho de que los datos sean aceptados por considerarse como verdaderos, reales y creíbles.
    - **Exactitud o precisión (Accuracy):** El hecho de que los datos sean correctos, fiables y certificados cómo libres de errores.
    - **Objetividad (Objectivity)**:el hecho de que los datos no tienen sesgos (sin prejuicios) o imparcialidades.
    - **Reputación (Reputation):**El hecho de que los datos son verdaderos o considerados altamente creíbles en términos de las fuentes o contenidos de origen.

- **Calidad de datos contextual:**Serían los requerimientos destacados de la calidad de datos que deben ser considerados dentro del contexto de la tarea actual, es decir, los

datos deben ser relevantes, oportunos, completos y apropiados en términos de cantidad así como de valor añadido.

- **Valor añadido (Value-added):**El hecho de que los datos son beneficiados y proporcionan ventajas en su propio uso.
- **Relevancia (Relevancy):**El hecho de que los datos son aplicables y útiles para la tarea actual.
- **Oportunidad (Timeliness):**El hecho de que la vigencia de los datos sea apropiada para la tarea actual.
- **Completitud (Completeness):**el hecho de que los datos son suficientemente amplios, profundos y están en el ámbito para la tarea actual.
- **Cantidad apropiada de datos (Appropiate amount of data):**El hecho de que la calidad y el volumen habilitado para los datos es el apropiado.

- **Calidad de datos representacional**: Hace énfasis en la importancia del rol en el sistema, el sistema debe presentar los datos de una manera que sean interpretables, fácil de entender, que los datos se representen de forma concisa y consistente.

  - **Interpretabilidad (Interpretability):**El hecho de que los datos son expandibles, adaptables pueden añadirse con facilidad a otras necesidades.
  - **Fácil entendimiento (Ease to understanding):**el hecho de que los datos sean limpios, sin ambigüedad y que realmente proporcionen fácil comprensión.
  - **Consistencia representacional (Representational cosistency):**El hecho de que los datos son siempre presentados en el mismo formato y son compatibles con los datos previos,
  - **Representación concisa (Concise representation):**El hecho de que los datos son representados compactamente sin ser imprecisa, es decir, su presentación debe ser breve pero completa.

- **Accesibilidad**: Hace hincapié en el hecho de que el sistema debe ser muy accesible pero muy seguro dónde se pueda gestionar cada nivel de acceso con seguridad.

  - **Accesibilidad (Accesibility):**El hecho de que los datos están disponibles o se pueden recuperar fácil y rápidamente.
  - **Acceso seguro (Access security):**El hecho de que el acceso a los datos pueda ser restringido y además mantenga la seguridad.

C. *Leo L. Pipino, et. al*

Otra publicación dónde se hace especial hincapié en las características generales de calidad de los datos es “Data Quality Assessment” [27], dónde se define a la ***calidad*** cómo un concepto multidimensional que depende fundamentalmente de los principios necesarios que definen en cada caso, puede depender del propósito del sistema o de la empresa.

La subjetividad de la evaluación de la calidad de los datos refleja la necesidad y experiencia de los stakeholders (ya sean productores, administradores o consumidores de datos). En la publicación se identifican las siguientes características:

- **Accesibilidad (Accessibility):** El grado en que los datos están disponibles, o son fácil y rápido recuperables.

- **Cantidad apropiada de datos (Appropriate Amount of Data):** El grado que el volumen de datos es apropiado para la tarea a mano.

- **Credibilidad (Believability):** El grado en que los datos se consideran como verdaderos o creíbles.

- **Completitud (Completeness):** El grado en que los datos no son faltantes y son suficiente amplios y profundos para la tarea a mano.

- **Representación Concisa (Concise Representation):** El grado en que los datos son representados concisamente.

- **Representación Consistente (Consistent Representation):** El grado en que el dato es representado en el mismo formato.

- **Facilidad de manipulación (Ease of Manipulation):** El grado en que el dato es fácil de manipular y añadir a tareas diferentes.

- **Libre de error (Free-of-error):** El grado en que el dato es correcto y fiable.

- **Interpretabilidad (Interpretability):** El grado en que el datosestá en un lenguaje apropiado, símbolos, y unidades, y una clara definición.

- **Objetividad (Objectivity):** El grado en que el dato es objetivo, sin perjuicios e imparcial.

- **Relevancia (Relevancy):** El grado que el dato se aplica y es útil a la tarea a mano.

- **Reputación (Reputation):** El grado en que el dato es muy bien considerado en términos del contenido de la fuente.

- **Seguridad (Security):** El grado en que el acceso al dato es apropiadamente restringido para mantener su seguridad.

- **Oportunidad (Timeliness):** El grado en que el dato está lo suficientemente al día para la tarea a mano.

- **Entendibilidad (Understandability):** El grado en que el dato es fácil de comprender.

- **Valor añadido (Value-added):** El grado en que el dato es beneficioso y proporciona ventajas para su uso.

D. *Rudra y Yeo*

Amit Rudra y Emilie Yeo ha publicado varios artículos de sobre el mundo de la calidad con respecto a los DW, podemos destacar dos el artículo *"Key Issues in Achieving Data Quality and Consistency in Data Warehousing among Large Organizations in Australia"*, [28] y *"Issues in user perceptions of data quality and satisfaction in using a data warehouse-an Australian experience",* [29]. Brevemente los siguientes artículos expresaban lo siguiente:

- ***"Key Issues in Achieving Data Quality and Consistency in Data Warehousing among Large Organizaciones in Australia",*** [28] tiene por objetivo definir los temas claves que logran la calidad de datos en los ambientes de almacenes de datos, relacionando la consistencia con la calidad de datos.

Indica que la calidad de datos se refiere a las siguientes características sobre el dato, ya que ha de ser:

- **Relevante (relevant).**
- **Preciso (precise).**
- **Útil (useful).**
- **Entendible (understandable).**
- **Oportuno (timely data).**
- **Exacto (accuracy).**
- **Debe proporcioar salidas oportunas (output timeliness).**
- **Fiabilidad (reliability).**
- **Completitud (completeness).**
- **Relevancia (relevance).**
- **Precisión (precision).**

Define a su vez el problema de la ***inconsistencia de los datos***, como la ocurrencia de estos cuando existen diferentes versiones en la misma base de datos, esto puede ser causado por varias estados de actualización o cuando se cambia algo que ha sido tratado en un archivo. La inconsistencia de los datos almacenados son unos de los más comunes de los errores de las fuentes en los sistemas de información. Se puede llegar a obtener la consistencia de los datos controlando o eliminando la redundancia, conjuntamente con la buena administración que puede llegar a promover un alto nivel de integridad de los datos.

"Issues in user perceptions of *data quality and satisfaction in using a data warehouse- an Australian experience",* [29] el objetivo fue identificar la percepción de la calidad de datos entre los usuarios de los DW, y las tasas de satisfacción de los usuarios de los DW.

En esta publicación definen nuevamente las mismas características de calidad de datos que en la publicación pasada.

También comentan el problema de la inconsistencia de datos tal como lo hicieron en el anterior, pero además comentan de un mini estudio en que realizaron donde por medio de un cuestionario que enviaron a 16 personas vía correo, de los cuales solo 10 contestaron y así lograron identificar varios factores de la percepción de la calidad de los datos a su vez indican la media y desviación estándar, así:

Preguntas	Media	Desviación Estándar
Usted piensa que las salidas son de utilidad	4.86	1.46
¿La información es clara?	4.33	1.22
¿La información contiene lo que usted necesita?	3.86	1.68
¿Usted obtiene la información que necesita en tiempo?	3.83	2.14
¿El sistema le proporciona información al día?	3.67	2.35
¿El sistema es exacto?	3.63	1.6
¿El sistema le proporciona la información precisa que usted necesita?	3.57	1.72
¿El sistema le proporciona información suficiente?	3.43	1.4
¿Está usted satisfecho con la exactitud del sistema?	3.38	1.69

Figura 56. Encuesta de Rudra y Yeo [29]

Las preguntas iban orientadas hacia las características definidas en la primera publicación.

E. *Leithesier R.*

La publicación de la que vamos a hablar a continuación no pretende definir un modelo de calidad como tal, sino que utiliza las características del marco de Wang y Strong para adaptarlo de forma genérica a los DW del ámbito de la medicina. Por tanto podemos ver como los diferentes modelos de calidad pueden tener distintas aproximaciones o extensiones que serán adaptados en cada caso. Robert L. Leitheiser en su **publicación *"Data quality in health care data warehouseenvironments"*** [30] definió el terminó de calidad como apto para el uso, lo cual sugiere que el concepto de calidad de datos es relativo.

Además indica que la calidad de datos es considerada para cada uno de los usuarios, puede tratarse como una necesidad que está direccionada a las necesidades de cada uno de estos. Su aproximación de calidad al ámbito de la medicina del Marco de Wang y Strong usa las siguientes características:

Intrínsecas (Intrinsic)	Exactitud (Accuracy)
	Objetividad (Obectivity)
	Credibilidad (Believability)
	Reputación (Reputation)
Accesibilidad (Accessibility)	Acceso (Access)
	Seguridad (Security)
Contextual (Contextual)	Relevancia (Relevancy)
	Valor añadido (Value-added)
	Oportunidad (Timeliness)
	Completitud (Completeness)
	Cantidad de datos (Amount of data)
Representacional (Representational)	Interpretabilidad (Interpretability)
	Fácil entendimiento (Ease of understanding)
	Representación concisa (Concise Representation)
	Representación consistente (Consistent Representation)

Figura 57. Aproximación al marco de Wang y Strong de Leithesier [30]

F. *Ley Orgánica de Protección de datos de Carácter Personal 15/1999.*

La Ley Orgánica 15/1999 contiene entre sus principios generales, el principio de calidad de los datos, que, ligado al principio de proporcionalidad de los datos, exige que los mismos sean adecuados a la finalidad que motiva su recogida. [63]

La recogida y tratamiento de datos de carácter personal debe efectuarse desde su subordinación a los principios de calidad de los datos y de proporcionalidad que establece la Ley.

No puede obviarse que estamos tratando de un auténtico derecho fundamental, cuyo contenido el Tribunal Constitucional ha terminado de perfilar en la Sentencia 292/2000, de 30 de noviembre, denominándolo derecho de autodeterminación informativa o de libre disponibilidad de los datos de carácter personal. Así, en dicha sentencia se indica que este derecho fundamental 'persigue garantizar a esa persona el poder de control sobre sus datos personales, sobre su uso y destino, con el propósito de impedir su tráfico ilícito y lesivo para la dignidad y derecho del afectado', estableciendo, en cuanto a su ámbito, que 'el objeto de protección del derecho fundamental a la protección de datos no se reduce sólo a los datos íntimos de la persona, sino a cualquier tipo de dato personal, sea o no íntimo, cuyo conocimiento o empleo por tercero pueda afectar a sus derechos sean o no fundamentales, porque su objeto no es sólo la intimidad individual, que para ello está la protección que el artículo 18. 1 CE otorga, sino los datos de carácter personal'.

Aun concretando más el contenido del derecho, se establece que el poder de disposición y control sobre los datos personales que tal derecho implica 'se concretan jurídicamente en la facultad de consentir la recogida, la obtención y el acceso a los datos personales, su posterior almacenamiento y tratamiento, así como su uso o usos posibles, por un tercero, sea el Estado o un particular. Y ese derecho a consentir el conocimiento y el tratamiento, informático o no, de los datos personales, requiere como complementos indispensables, por un lado, la facultad de saber en todo momento quién dispone de esos datos personales y a qué uso los está sometiendo, y, por otro lado, el poder oponerse a esa posesión y usos'.

Artículo 4. Calidad de los datos.

- Los datos de carácter personal sólo se podrán recoger para su tratamiento, así como someterlos a dicho tratamiento, cuando sean adecuados, pertinentes y no excesivos en relación con el ámbito y las finalidades determinadas, explícitas y legítimas para las que se hayan obtenido.
- Los datos de carácter personal objeto de tratamiento no podrán usarse para finalidades incompatibles con aquellas para las que los datos hubieran sido recogidos. No se considerará incompatible el tratamiento posterior de éstos con fines históricos, estadísticos o científicos.
- Los datos de carácter personal serán exactos y puestos al día de forma que respondan con veracidad a la situación actual del afectado.
- Si los datos de carácter personal registrados resultaran ser inexactos, en todo o en parte, o incompletos, serán cancelados y sustituidos de oficio por los correspondientes datos rectificados o completados, sin perjuicio de las facultades que a los afectados reconoce el artículo 16.

- Los datos de carácter personal serán cancelados cuando hayan dejado de ser necesarios o pertinentes para la finalidad para la cual hubieran sido recabados o registrados No serán conservados en forma que permita la identificación del interesado durante un período superior al necesario para los fines en base a los cuales hubieran sido recabados o registrados. Reglamentariamente se determinará el procedimiento por el que, por excepción, atendidos los valores históricos, estadísticos o científicos de acuerdo con la legislación específica, se decida el mantenimiento íntegro de determinados datos.
- Los datos de carácter personal serán almacenados de forma que permitan el ejercicio del derecho de acceso, salvo que sean legalmente cancelados.
- Se prohíbe la recogida de datos por medios fraudulentos, desleales o ilícitos.'

Constituye infracción de carácter grave, de acuerdo con lo dispuesto en el artículo 44.3.f):

"Mantener datos de carácter personal inexactos o no efectuar las rectificaciones o cancelaciones de los mismos que legalmente procedan cuando resulten afectados los derechos de las personas que la presente Ley ampara".

Doctrina Judicial

Sentencia de la Sala de lo Contencioso Administrativo de la Audiencia Nacional, de 6 de julio de 2001.

"La conducta de la entidad recurrente recabando o intentando recabar unos datos de carácter personal (en concreto los relativos a la cuenta bancario o VISA) para su tratamiento automatizado que resultaban completamente innecesarios e inadecuados en relación con el ámbito y finalidades legitimas para las que se hayan obtenido, debe ser constitutiva de infracción (....) Y aunque los datos no llegaron a ser incorporados a los ficheros (...), ello no implica que falte el necesario tratamiento automatizado de los mismos para que se produzca el tipo sancionador."

Sentencia de la Sección Novena de la Sala de lo Contencioso Administrativo del Tribunal Superior de Justicia de Madrid, **de 5 de noviembre de 1998**.

"Sin embargo, aunque (...) no hubo (...) una intención de dañar ni enriquecimiento injusto (...) los hechos han tenido una doble perturbación para la perjudicada: (...) imputarle una deuda inexistente (.....) lo más grave fue su inclusión en un Registro Informático de Morosos y además sin conocimiento de la perjudicada (....) y de esa inclusión indebida en el Registro de Morosos no eran responsables los que llevan el Registro sino los que suministraron el dato".

Sentencia de la Sección Octava de la Sala de lo Contencioso Administrativo del Tribunal Superior de Justicia de Madrid, **de 18 de octubre de 2000**.

"(....) para incluir en un fichero de solvencia patrimonial el dato relativo a una deuda, ésta, además de cierta, vencida y exigible, ha de haber resultado efectivamente impagada (....) debiendo además, el acreedor (como requisito previo a la inclusión del dato en un fichero de estas características) proceder en la forma más arriba descrita y cuya finalidad no es otra que garantizar la exactitud de los datos que se pretender incluir".

Sentencia de la Sección Octava de la Sala de lo Contencioso Administrativo del Tribunal Superior de Justicia de Madrid, **de 26 de mayo de 1999**.

"Si los datos registrados se han obtenido de una fuente accesible al público (como en el caso de autos) el medio más efectivo para mantener actualizados aquellos será notificando al

afectado la existencia del dato a fin de que éste (si el dato obtenido de esa fuente de acceso público es incorrecto o la situación ha variado) pueda instar las rectificaciones pertinentes en el momento en que el dato registrado no responda a la realidad y si el afectado declina realizar las oportunas rectificaciones, entendemos, su inactividad exculpará al titular del fichero de toda responsabilidad en orden a la actualización de los datos, en la medida que esa actualización no pueda obtenerse de la misma forma en la que se obtuvo el dato".

Sentencia de la Sala de lo Contencioso Administrativo de la Audiencia Nacional, **de 19 de enero de 2001**.

"(...) incluido el dato erróneo, la infracción se produce hasta que el mismo haya sido erradicado del fichero".

Sentencia de la Sección Octava de la Sala de lo Contencioso Administrativo del Tribunal Superior de Justicia de Madrid, **de 9 de febrero de 2000**.

"(...) el dato registrado (...) es transcripción del contenido en dos edictos publicados en el BOCAM, por lo que ignoramos si son o no exactos dichos datos y, en todo caso, la inexactitud del mismo nunca sería imputable a la actora. Si (..) para los datos obtenidos de fuente accesibles al público la LORTAD no exige la notificación del registro al afectado, difícilmente puede saber el titular del fichero si el dato obtenido de una fuente accesible al público es o no correcto y, además, en el caso de autos, dado que en el edicto no consta otro datos que el nombre y apellidos de los demandados, nunca hubiera sido posible efectuar tal notificación, ni averiguar la exactitud del dato publicado, ni de lo actuado puede afirmarse que dicho dato se refiera siquiera al denunciante, por lo que en la medida que no conste al titular del fichero la inexactitud del dato registrado, inexactitud que, reiteramos, no consta, no existe para éste la obligación legal de cancelar el dato".

Sentencia de la Sala de lo Contencioso Administrativo de la Audiencia Nacional, **de 9 de marzo de 2001**.

"Uno de los principios que inspira la legislación sobre tratamiento automatizado de datos de carácter personal es el de calidad de datos. Este principio implica, entre otras cosas, que los datos sean necesarios y pertinentes para la finalidad para la cal hubieran sido recabados o registrados (art. 4.5 de la LO 5/1992) y que sean exactos y completos art. 4.4 de la LO 5/1992. Por lo tanto, si los datos han dejado de ser necesarios para los fines para los cuales fueron recabados o registrados o resultan inexactos, se debe proceder (...) a su cancelación, sin necesidad de solicitud del afectado. Y así se infiere del propio tenor literal de los artículos 4.4 y 4.5 de la LO 5/1992, que utiliza la expresión imperativa 'serán cancelados' y sin condicionarla a la existencia de una previa solicitud del afectado. En suma, la norma establece la obligación del responsable del fichero de proceder de oficio y con la debida diligencia a cancelar los datos inexactos o que han dejado de ser necesarios para la finalidad del fichero y sin necesidad de solicitud previa del afectado".

B. CLASIFICACIÓN, CRÍTICA Y VALORACIÓN

Se ha realizado un cuadro resumen dónde se indica que características de calidad del dato aparecen en las publicaciones referentes del estado del arte.

Dado que la dimensión legal no aparecía en ninguno de los estudios o normas vistos no aparece en el cuadro comparativo, pero si se enumera en el resumen depurado de características de la calidad del dato.

Cuadro comparativo

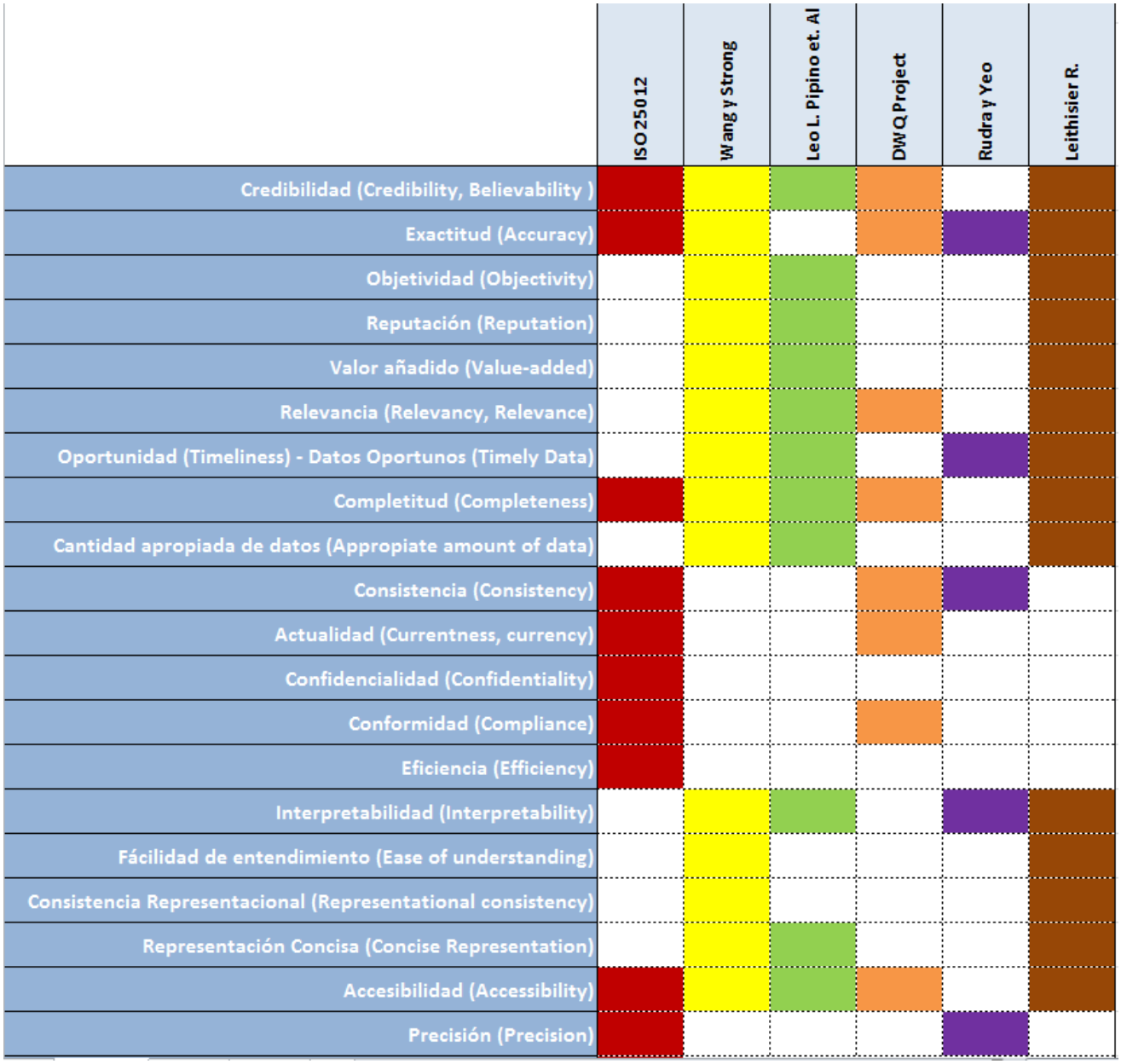

Figura 58. Características de calidad del dato (I)

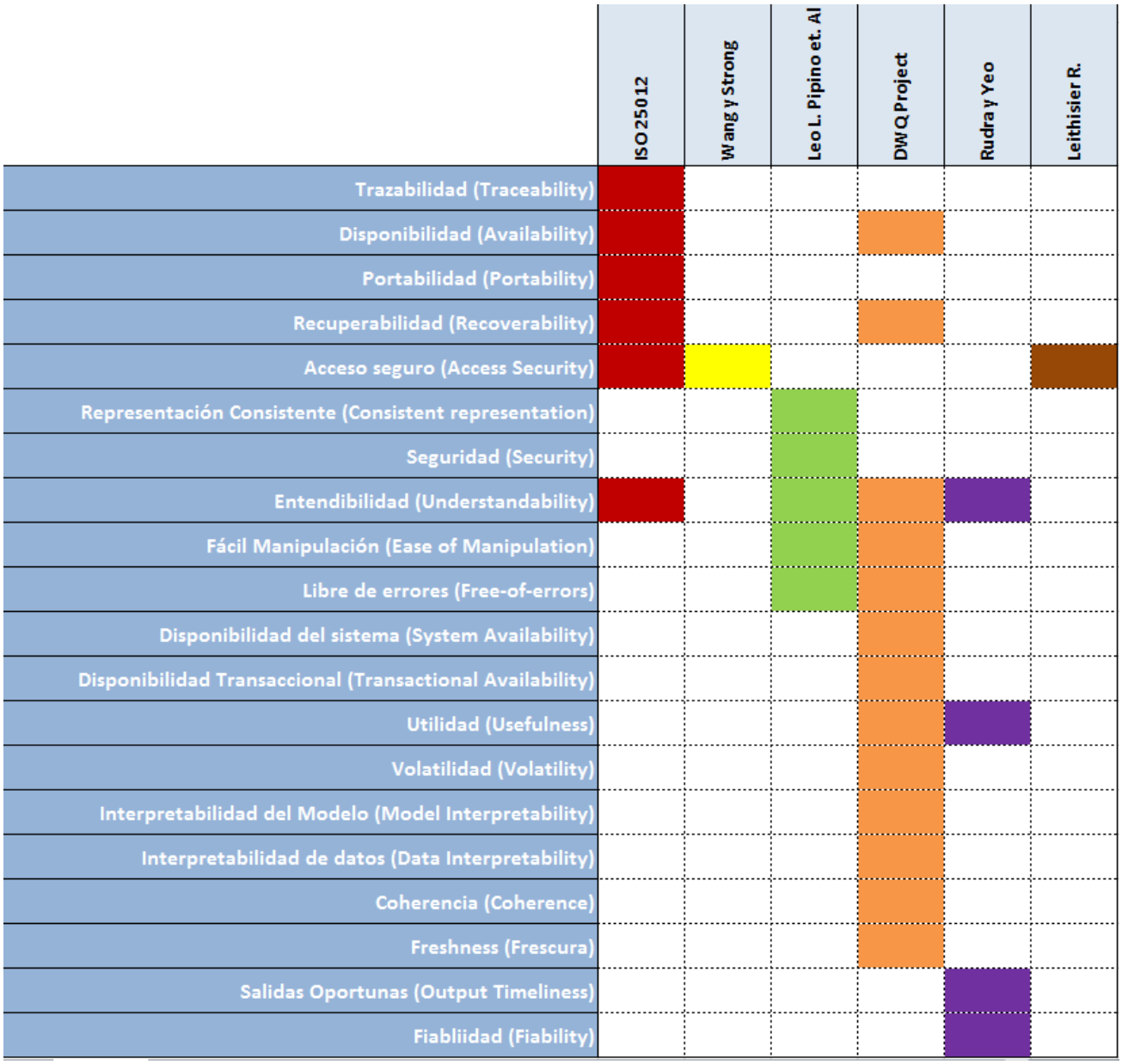

Figura 59. Características de calidad del dato (II)

Ahora analizamos las distintas definiciones de los conceptos asociados a características de calidad de datos que mencionan los diversos autores en sus publicaciones, con el objetivo de identificar conceptos similares y recuperar la definición más adecuada para redefinir un concepto, reconstruir una definición en torno al resto y que será utilizado como base para las directrices de calidad de datos en el desarrollo de un DW.

c. CARACTERÍSTICAS DEPURADAS.

A. *Credibilidad*

Para de la definición del concepto de credibilidad (credibility, believabilitiy), se analizó la definicióndel concepto de credibilidad (credibility) de la ISO 25012 con el de credibilidad (credibility) del modelo de Wang y Strong, el de credibilidad (believability) de Pipino et. al. En cuyo caso todos se orientaron a considerar verdaderos y creíbles los datos; sin embargo el concepto que expuso el personal del proyecto DWQ credibilidad (credibility)se orienta a la credibilidad de la fuente el que se analizará en la característica de reputación, en este caso se utilizará la definición de la ISO 25012 por ser un estándar.

Credibilidad (Credibility): Es el grado en el que el dato tiene atributos que son considerados como verdaderos y creíbles por usuarios en un contexto específico de uso.

B. *Exactitud*

Para la definición del concepto de exactitud (accuracy) se analizó la definición el concepto de exactitud (accuracy) de la ISO 25012 con el de exactitud (accuracy) de Wang yStrong, el de libre de errores (free-of-error) de Pipino et. al., exactitud (accuracy) del personal del proyecto DWQ, en este caso se utilizará la definición que da la ISO 25012 por ser un estándar.

Exactitud (Accuracy): El grado en el que el dato tiene atributos que representan correctamente el valor verdadero del atributo instanciado en un concepto o evento en un contexto específico de uso.

C. *Objetividad*

Para de la definición del concepto de objetividad (objectivity) se analizó la definición del concepto de objetividad (objectivity) de Wang y Strong con el concepto de objetividad (objectivity) de Pipino et. al., siendo ambos similares, para este caso se utilizará el concepto de Wang y Strong por la forma como fue validado el mismo.

Objetividad (Objectivity): El hecho que los datos no tiene sesgos (sin prejuicios) e imparcialidades.

D. *Reputación*

Para de la definición del concepto de reputación (reputation), se analizaron las definiciones de reputación (reputation) de Wang y Strong con el de reputación (reputation) de Pipino et. al., que son similares al de credibilidad (credibility) del personal del proyectoDWQ, sin embargo Rudra y Yeo mencionan el término de fiabilidad (reliability) no existiendo definición por lo cual se toma

como la fiabilidad de la fuente; por lo anterior para este concepto utilizaremos la definición de Wang y Strong por la forma como fue validada.

Reputación (Reputation): El hecho que los datos son verdaderos o considerados altamente creíbles en términos de las fuentes o contenidos de origen.

E. *Valor Añadido*

En el caso de la definición del concepto de valor añadido (value-added), se analizaron las definiciones de valor añadido (value-added) de Wang y Strong con el de valor añadido (value-added) de Pipino et. al., siendo la misma razón por la cual se utilizará la definición de Wang y Strong por la forma como fue validada.

Valor añadido (Value-added): El hecho que los datos son beneficiados y proporcionan ventajas en su propio uso.

F. *Relevancia*

Para la definición del concepto de relevancia (relevancy, relevance), se analizó la definición del concepto relevancia (relevancy) de Wang y Strong con el de relevancia (relevance) de Pipino et. al., razón por la cual se tomará el de Wang y Strong por la forma como fue validada.

Relevancia (Relevancy): El hecho que los datos son aplicable y útiles para la tarea actual.

G. *Oportunidad, actualidad y volatilidad*

Para la definición del concepto de oportunidad (timeliness) se analizó con las definiciones de actualidad (currentness) de la ISO 25012 con el de oportunidad (timeliness) de Wang y Strong, el de oportunidad (timeliness) de Pipino et. al., las de frescura (freshness) y utilidad (usefulness) del personal del proyecto de DWQ identificando que son las mismas definiciones aunque el nombre de algunos conceptos es distinto. A su vez Rudra y Yeo mencionan únicamente las características de datos oportunos (timely data) y salidas oportunas (output timeliness) no dan definición por lo que se consideran bajo este concepto. Sin embargo los conceptos de actualidad (currency) y volatilidad (volatility) que definen el personal del proyecto DWQ seven como subcaracterísticas de esta característica. En resumen, se utilizará la definición de Wang y Strong para el concepto oportunidad por su forma de validación ya que abarca el concepto de todos los demás autores, adicionalmente se utilizaran las definiciones delas subcaracterísticas como las propone el personal del proyecto DWQ, así:

Oportunidad (Timeliness): El hecho que la edad de los datos es apropiada para la tarea actual.

Actualidad (Currency): describe cuando la información fue ingresada en la fuente o/y en el almacén de datos.

Volatilidad (Volatility): describe el período de tiempo en que la información es validad en el mundo real.

H. *Completitud*

Para la definición del concepto de completitud (completeness) se analizó la definición de completitud (completeness) de la ISO 25012, con la de completitud (completeness) de Wang y Strong, la de completitud (completeness) de Pipino et. al., también la de completitud (completeness) del personal del proyecto de DWQ, y el significado es el mismo, razón por la cual se tomará la definición de Wang y Strong para el concepto de completitud por su forma de validación y a su vez abarca el concepto de todos los demás autores.

Completitud (Completeness): El hecho que los datos son suficientemente amplios, profundos y están en el ámbito para la tarea actual.

I. *Cantidad apropiada de datos*

En el caso de la definición del concepto de cantidad apropiada de datos (appropriateamount of data), se analizó la definición de cantidad apropiada de datos (appropriateamount of data) de Wang y Strong y la definición de cantidad apropiada de datos (appropriate amount of data) de Pipino et. al., en este caso las definiciones son similares, razón por la cual se utilizará la definición que da Wang y Strong por la forma como fue validada.

Cantidad apropiada de datos (Appropriate amount of data): El hecho en que la calidad y el volumen habilitado para los datos es apropiado para la tarea actual.

J. *Consistencia*

Para la definición del concepto de consistencia (consistency), se analizó las definiciones de los conceptos de consistencia (consistency) de las ISO 25012, la de consistencia(consistency) del personal del proyecto DWQ como está la expuso en diferente publicación a la de consistencia (consistency), se verificó que la coherencia se refería a la integridad de los datos. Para este concepto utilizaremos la definición de la ISO 25012 por ser un estándar.

Consistencia (Consistency): El grado en el que el dato tiene atributos que son libres de contradicción y son coherente con otros datos en un contexto específico de uso.

K. *Accesibilidad*

En el caso de accesibilidad (accessibility) se analizó la definición de accesibilidad(accessibility) de la ISO 25012, con el de accesibilidad (accessibility) del personal del proyecto de DWQ, estas son muy similares, sin embargo, la definición de accesibilidad(accessibility) de Wang y Strong y la de accesibilidad (accessibility) de Pipino et. al., se orientan hacia la definición del concepto de disponibilidad (availability) de la ISO 25012.Por lo anterior se toma la definición de la ISO 25012 por ser un estándar.

Accesibilidad (Accessibility): El grado en el que el dato puede ser accedidos en un contexto específico de uso, particularmente por la gente que necesita el soporte de tecnología o una configuración especial porque tiene alguna indisponibilidad.

L. *Confidencialidad*

En el caso de la definición del concepto de confidencialidad (confidentiality) solo se encontró la que define la ISO 25012.

Confidencialidad (Confidentiality): El grado en el que el dato tiene atributos que aseguran que éste es sólo accesible e interpretable por usuarios autorizados en un contexto específico de uso.

M. *Disponibilidad.*

Para la definición del concepto de disponibilidad (availability) se tiene únicamente la definición de la ISO 25012, sin embargo, como se mencionó en la característica de calidad de datos de accesibilidad, esta definición es similar a la de accesibilidad(accessibility) de Wang y Strong y la de accesibilidad (accessibility) de Pipino et. al., razón por la cual se tomará para este concepto la definición de la ISO 25012. Además a este concepto se ve influenciado por dos subcaracterísticas que fueron definidas por el personal del proyecto DWQ que son disponibilidad del sistema (system availability) y disponibilidad transaccional (transactional availability).

Disponibilidad (Availability): El grado en el que el dato tiene atributos que le permiten ser recuperados por usuarios autorizados y/o aplicaciones en un contexto específico de uso.

Disponibilidad del sistema (System Availability): describe el porcentaje de tiempo que la fuente o el sistema de almacén de datos está disponible.

Disponibilidad transaccional (Transactional Availability): describe el porcentaje de tiempo que la información en los almacenes de datos o fuentes está disponible debido a la ausencia de procesos de actualización de datos, los cuales bloquean la escritura en estos.

N. *Conformidad*

Para la definición de conformidad (compliace) se analizó la definición de conformidad

(compliace) de las ISO 25012 y la de conformidad (compliace) del personal del proyectoDWQ, siendo similares, razón por la cual se utilizará la de la ISO 25012 por ser un estándar.

Conformidad (Compliace): El grado en el que el dato tiene atributos que se adhieren a las normas, convenciones o regulaciones vigentes y reglas similares relacionadas con localidad de datos en un contexto específico de uso.

O. *Eficiencia*

Para la definición de eficiencia (efficiency) utilizaremos la de las ISO 25012 por ser la única que definió el concepto, así:

Eficiencia (Efficiency): El grado en el que el dato tiene atributos que pueden ser procesados y proporciona los niveles esperados de desempeño al utilizar las cantidadesy los tipos de recursos apropiados en un contexto específico de uso.

P. *Interpretabilidad*

En el caso de la definición de Interpretabilidad (interpretability) se analizó la definición deinterpretabilidad (interpretability) de Wang y Strong, con la definición de robustez, pero con la de interpretabilidad (interpretability) de Pipino et. al., las definiciones eran distintas, sin embargo la definición de Pipino et. al., concuerda con la definición de entendibilidad (understandability) de la ISO 25012. Además el personal del proyecto DWQ definió dossubcaracterísticas para este concepto la interpretabilidad de los datos (datainterpretability) y la interpretabilidad del modelo (interpretability of model). Para este caso se utilizará la definición de interpretabilidad de Wang y Strong por la forma como fuevalidada y para las subcaracterísticas la definición del personal del proyecto DWQ, así:

Interpretabilidad (Interpretability): El hecho que los datos son expandibles, adaptables y fácil adición para otras necesidades.

Interpretabilidad de los datos (Data interpretability): concierne con la descripción delos datos (es decir, disponer de datos de los sistemas heredados y datos externos, tablas de descripción de las bases de datos relacionales, llaves primarias y foráneas, alias, predefinidas, dominios, explicación de códigos de valores, etc.).

Interpretabilidad del modelo (Interpretatibility of model): describe el grado para el que el almacén de datos es eficientemente modelado para ser repositorio de información. Es decir, lo más fácil en que se pueden realizar las consultas. (Esta interpretabilidad no seorienta tanto a los datos como al esquema).

Q. *Entendibilidad*

Para el caso de la definición de entendibilidad (understandability) se analizó la definición de entendibilidad (understandability) ISO 25012, la de fácil entendimiento (ease ofunderstanding) de Wang y Strong, a su vez las definiciones de entendibilidad (understandability) e interpretabilidad (interpretability) de Pipino et. al., (ya que la unión de ambas se ve cubierta por las definiciones de entendibilidad de los otros autores), asimismo la de entendibilidad (understandability) del personal del proyecto de DWQ, y todas estas son similares, razón por la cual se utilizará la definición de la ISO 25012 por ser un estándar.

Entendibilidad (Understandability): El grado en el que el dato tiene atributos que le permiten ser leído e interpretado por usuarios, y es expresado en lenguajes apropiados, símbolos y unidades en un contexto específico de uso.

R. *Representación consistente*

Para la definición de consistencia representación (representational concistency) se analizó la definición de consistencia representación (representational concistency) de Wang y Strong y la de coherencia (coherence) del personal del proyecto DWQ y la de representación consistente (consistent representation) de Pipino et. al. Siendo todas las definiciones similares y que se ampliamente cubiertas por la de Wang y Strong que por la forma como fue validada es la que se utilizará para este concepto.

Consistencia representacional (Representational consistency): El hecho que los datos son siempre presentados en el mismo formato y son compatibles con los datos previos.

S. *Representación Concisa*

En el caso de la definición representación concisa (concise representation) se analizó la definición de representación concisa (concise representation) de Wang y Strong con la de representación concisa (concise representation) de Pipino et. Al. Por lo cual se utilizará la definición de Wang y Strong por la forma como fue validada.

Representación concisa (Concise representation): El hecho que los datos son representados compactamente sin ser imprecisos (es decir, breve en la representación, completa y al punto).

T. *Precision*

En el caso de la definición del concepto de precisión (precision) solo se encontró la que define la ISO 25012.

Precisión (Precision): El grado en el que el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso.

U. *Trazabilidad*

En el caso de la definición del concepto de trazabilidad (traceability) solo se encontró la que define la ISO 25012.

Trazabilidad (Traceability): El grado en el que el dato tiene atributos que proporcionan un rastro de auditoría al acceso a los datos y de cualquier cambio realizado a los datos en un contexto específico de uso.

V. *Facilidad de manipulación*

En el caso de la definición del concepto de facilidad manipulación (ease of manipulation) solo se encontró la que define Pipino et. al.

Facilidad de manipulación (Ease of Manipulation): El grado en que el dato es fácil de manipular y añadir a tareas diferentes.

W. *Acceso seguro*

En el caso de la definición del concepto de acceso seguro (access security) se analizó con las definiciones de acceso seguro (access security) de Wang y Strong, la de seguridad (security) de Pipino et. al., y la de seguridad (security) del personal del proyecto de DWQ, siendo similares, por lo cual se toma la de Wang y Strong por la forma como fue validada.

Acceso seguro (Access security): El hecho que el acceso a los datos pueda ser restringido y además mantenga la seguridad.

X. *Recuperabilidad*

Respecto a la definición del concepto de recuperabilidad (recoverability) se analizó la definición de recuperabilidad (recoverability) la ISO 25012 y la de recuperabilidad (recoverability) del personal del proyecto DWQ, siendo similares, por lo cual se utiliza lade la norma ISO 25012 por ser un estándar.

Recuperabilidad (Recoverability): El grado en el que el dato tiene atributos que le permiten mantener y conservar un nivel especificado de operaciones y calidad, aún en caso de falla, en un contexto específico de uso.

Y. *Portabilidad*

En el caso de la definición del concepto de portabilidad (portability) solo se encontró la que define la ISO 25012.

Portabilidad (Portability): El grado en el que el dato tiene los atributos que le permiten ser instalado, substituido o movido de un sistema a otro conservando la calidad existente en un contexto específico de uso.

Z. *Legalidad*

Dependiendo de las leyes en las que se mueva nuestra organización deberemos atender a los requisitos establecidos por las leyes vigentes, así, para el caso de España se debe atender a la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal.

d. CONCLUSIONES

La calidad de datos ha sido definida por varios autores, y la mayoría de estos concuerda en la definición que proporciona el estándar ISO 25012, como el grado en que las características de los datos guardan las condiciones y sugiere las necesidades cuando es usado bajo condiciones específicas, y que las características de calidad de datos son las categorías de los atributos de calidad de datos que llevan a la calidad de datos.

Puede verse que existen áreas específicas como la medicina dónde también se hace uso especial de los DW, por lo que la calidad de los datos en este ámbito es muy importante, ya que las decisiones tomadas con datos erróneos pueden llevar a desenlaces fatales.

En este trabajo se analizaron las diferentes publicaciones encontradas que tenían relación con características de calidad de datos, se identificaron 111 características, a las cuales se les realizó un proceso de análisis para identificar definiciones similares; luego de realizar este proceso dio como resultado 25 características.

Las características de calidad de datos para DW es un tema poco estudiado, por lo cual este análisis debe ser complementado con otras características de calidad que se orienten a bases de datos o sistemas de información, para robustecer el modelo conceptual de calidad obtenido.

Deben definirse medidas para cada una de las características de calidad de datos en el entorno de los almacenes de datos.

Con las características encontradas identificar las relaciones y la implicación de una característica con otra, para definir una métrica que, de cómo resultados la calidad de datos que tienen un almacén de datos.

Hay que validar trabajar sobre varias de estas dimensiones hasta hacer que nuestro objetivo no sea tan solo proveer información de calidad, sino el de apoyar a la organización a hacer buen uso de la información para apoyar y mejorar el uso de los recursos y las operaciones del negocio.

BLOQUE III: "*GUIDELINES* PARA EL DESARROLLO DE UN DATA WAREHOUSE DE CALIDAD EN UN SISTEMA BI. CALIDAD EN EL DATO, CALIDAD EN EL PROCESO".

1. ESTABLECIMIENTO DEL MARCO GENERAL DE UN SI DE CALIDAD

1. INTRODUCCIÓN.

El desarrollo de sistemas BI de calidad, a menudo se ve postrado a un segundo plano por la falta de guías de mejores prácticas que agrupen todos los procesos que forman parte del proceso de ciclo de vida del proyecto y se toman referencias de normas y estándares de calidad únicamente ligados a algunos de sus apartados.

Con el siguiente texto se agrupa todo el proceso por el que fluye una solución de BI desde la fase de desarrollo hasta incluso la gestión del proyecto.

Se ha estructurado un marco de calidad cimentado en cuatro grandes bloques, que son la identificación del SI, la metodología de desarrollo del SI, la metodología de gestión del SI y el aseguramiento de la calidad del SI.

Figura 60. Marco de calidad de una solución de BI

Pero no debemos olvidar que por las características que se confiere a una solución de BI y el impacto que puede tomar la explotación del Data Warehouse corporativo en busca de

decisiones estratégicas y tácticas se debe establecer un marco general de calidad desde la responsabilidad de la dirección hacia el resto de miembros de la corporación.

2. RESPONSABILIDAD DE LA DIRECCIÓN.

La Dirección General deberá suministrar pruebas para apoyar y desarrollar la implementación del Sistemas de Gestión de calidad y su mejora continua a través de una serie de acciones como:

- Comunicando a la empresa la importancia de satisfacer los requisitos predefinidos tanto por el cliente (incluida la propia organización) como de Ley vigente.
- Establecer la política de calidad en base a la implantación del plan de calidad.
- Asegurando que los objetivos de calidad son conocidos y se cumplen.
- Conduciendo las revisiones de la Dirección y llevando el timón de las decisiones que se toman para reconducir desviaciones, ya sean de costes, temporales o de calidad.

Se debe detallar con claridad desde la dirección quién llevará el liderazgo del proyecto y como guiar el mismo para la consecución de los objetivos, que deberán ser claros y concisos desde el primer momento y conocidos por todos los miembros que formen parte del equipo de trabajo.

La dirección general debe involucrarse durante todo el proyecto y conocer el estado del mismo en cada fase, se encargará de que la planificación del proyecto se cumpla y que todos los recursos estén disponibles responsabilizándose de lo contrario.

Nos basaremos en las indicaciones de la norma ISO 9000-3 para saber cómo responsabilizar a la dirección en el proyecto.

2. IDENTIFICACIÓN DEL SI

1. REQUERIMIENTOS.

La calidad del producto resultante no se va a centrar en la satisfacción del cliente, sino que nos basaremos en un modelo de evaluación de la calidad. En base a las propiedades especificadas en el apartado “5.1 Calidad del dato”, se debe especificar cuáles de las características de la calidad del dato serán de mayor impacto para que cada requisito identificado se considere como superado.

Así una vez establecidos los requisitos que debe proporcionar nuestro sistema de BI, se aplicará que características deberá contener con mayor índice el propio requisito y se contralará su estadística a través de métricas.

Se debe definir que métricas se aplicarán para establecer si se cumplen los requisitos establecidos en base a la ponderación individual de cada característica y ver si la ponderación total del grupo correspondiente a cada requisito supera los niveles mínimos establecidos, por lo

tanto se debe definir un sistema de niveles que nos permita saber a través de los indicadores que se creen si se cumplen los objetivos o debemos reconducir el modelo usado.

Debe figurar un histórico de la medida en la que se cumplen los requisitos y la evolución del sistema, además de verificar el cumplimiento de los niveles establecidos, estos niveles deberán revisarse cada cierto tiempo.

La mejor forma de completar el trabajo, en primera instancia, es a través de la creación de un cuadro de mando integral destinado a mostrar la información que necesitamos. Aunque podremos usar herramientas de automatización de pruebas, dado el carácter particular de un sistema BI como el indicado, en principio bastaría con adecuar nuestro CMI.

Deberemos llevar a cabo las siguientes tres tareas para poder conseguir el análisis de cumplimiento de los requerimientos:

1. Adquisición de los datos: Podemos diseñar una tarea que automáticamente se encargue de ello.
2. Análisis de las mediciones: Será realizar el almacenaje, recuperación, manipulación y estudio de los datos.
3. Presentación de los datos: Debemos generar tablas y gráficos que nos faciliten la comprensión de los resultados.

Las métricas a utilizar pueden consultarse a través de la lectura de la división 2503n de la norma ISO/IEC 25000, en concreto la Norma ISO/IEC 25030:2007 (Software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Quality requirements).

2. RIESGOS.

Cualquier riesgo potencial (sobre las metas u objetivos de la organización) causado por algún evento no planificado, se tiene que identificar, analizar y evaluar. Adquiere mayor importancia si vamos a tomar decisiones estratégicas en base al análisis de datos que nos proporciona la solución de BI.

El fundamento de la gestión de riesgos se basará en el ciclo de Deming: Plan, Do, Check, Act (Planifica, Ejecuta, Verifica y Corrige).

Nos basaremos en el proceso PO 9 (Planificar y organizar) de COBIT para el tratamiento de los riesgos. Para evaluar y administrar los riesgos de TI se deben establecer los principios que deben tenerse en cuenta para la gestión de los riesgos.

Se establece un marco que debe crearse y mantenerse un marco adecuado de trabajo para la gestión de riesgos. Dicho marco de trabajo debe documentar un nivel común y acordado de riesgos TI, estrategias para mitigarlos y riesgos que se acuerden aceptar como residuales.

Cualquier riesgo potencial sobre las metas u objetivos de la organización, causado por algún evento no planificado, se tiene que identificar, analizar y evaluar.

El marco de gestión de riesgos se completa con la implementación de los objetivos de control detallados a continuación sobre el proceso PO 9 de COBIT:

- PO 9.1 - Alineación de la administración o gestión de riesgos con el negocio.

Se debe integrar el gobierno, la administración del riesgo y el marco del control de TI, al marco de trabajo de la administración de riesgos de la organización.

- PO 9.2 – Establecimiento del contexto del riesgo.

Se debe establecer el contexto en el cual el marco de trabajo de la evaluación de riesgo se aplica para garantizar los resultados apropiados

- PO 9.3 – Evaluación de eventos.

Identificar aquellos eventos (amenazas y vulnerabilidades) con un impacto potencial sobre objetivos, retos u operaciones de la empresa, aspectos de negocio, regulatorios, legales, tecnológicos, de sociedad comercial, de recursos humanos y operaciones.

- PO 9.4 – Evaluación de riesgos TI.

Se debe evaluar de forma recurrente la posibilidad de impacto de todos los riesgos identificados, usando métodos cualitativos y cuantitativos.

- PO 9.5 – Respuesta a los riesgos.

Debemos identificar a los propietarios de los riesgos y a los dueños de los procesos afectados y elaborar y mantener respuestas a los riesgos que garanticen que los controles rentables y las medidas de seguridad mitigan y reducen la exposición a los riesgos de forma continua.

- PO 9.6 – Mantenimiento y monitoreo de un plan de acción de riesgos.

Debemos asignar prioridades y planear las actividades de control a todos los niveles para implantar las respuestas a los riesgos que sean necesarias, incluyendo costos y beneficios así como la responsabilidad de la ejecución.

3. Metodología de Desarrollo del SI

1. Kimball

La metodología ideal para la mayor parte de los proyectos de BI a la hora de crear el DW corporativo se corresponde con la diseñada por Ralph Kimball en su Ciclo de Vida.

El flujo general de implementación se corresponde como una sucesión de tareas a llevar a cabo como las que vienen en la siguiente figura:

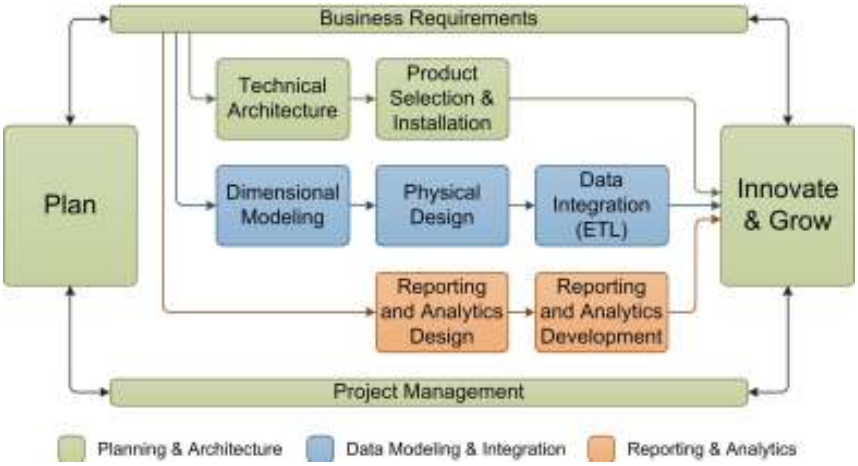

Figura 61. Metodología del ciclo de vida de Kimball.

La metodología cubre todas las posibilidades que pueden darse en un proyecto de desarrollo de un DW, pero no todas las tareas que se especifican deberán ser completadas, sino que se deben adecuar a la realidad de cada proyecto.

Las tareas se estructuran en tres grupos, que son:

1. Planificación y arquitectura (en verde en la figura) - Contiene las tareas de:
    - Planificación del proyecto (Plan).
    - Definición de los requerimientos del negocio (Business Requirements).
    - Diseño de la arquitectura técnica (Technical Architecrture).
    - Selección e instalación de Software (Product selection & installation).
    - Gestión del proyecto (Project Management).
    - Ciclo de innovación y evolución (Innovate & grow).

2. Modelado de datos e integración (en azul en la figura) – Contiene las tareas de:
    - Modelo dimensional (Dimensional modeling).
    - Diseño físico (Physical design).
    - Integración de datos - ETL (Data integration – ETL)
3. Análisis y reporte de información (en rojo en la figura) – Contiene las tareas de:
    - Diseño del análisis y reporte de la información (Reporting and analytics design).
    - Desarrollo del análisis y reporte de la información (Reporting and analytics development)

El enfoque de la metodología se conoce como *bottom-up* y quiere decir que es una metodología rápida que se basa en prototipos y experimientos, es decir, se trata de un modelo flexible que permite a la organización ir avanzando sobre versiones del producto poco a poco para corregir desde las primeras fases aquellas incoherencias que se detecten o revisar fallos de diseño o de toma de requisitos por lo que se puede ir más lejos con menores costes a medio-largo plazo.

El modelo de diseño del DW será construir Data marts independientes en cada ciclo de vida independiente e ir enlazando cada DM según vayan conformando el sistema completo y formar el DW corporativo. El almacenamiento de la información seguiría al de la siguiente figura que nos muestra las fuentes de datos, la consolidación de la información y la carga en la estructura de almacenamiento:

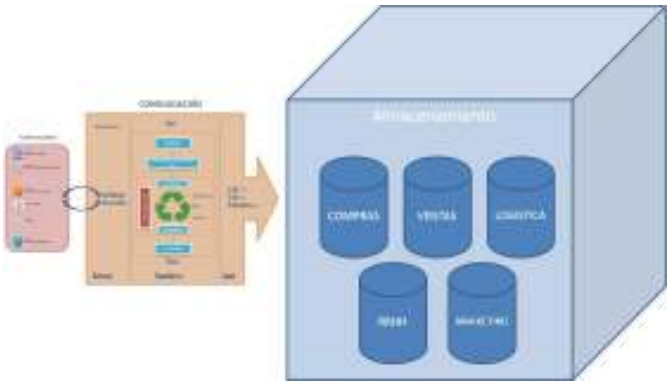

Figura 62. Modelo de almacenamiento de la filosofía de Kimball.

Las estrategias basadas en el flujo de información bottom-up se antojan potencialmente necesarias y suficientes porque se basan en el conocimiento de todas las variables que pueden afectar a los elementos del sistema.

Se puede consultar el detalle de la metodología en el libro “The Data Warehouse Lifecycle Toolkit” [14].

2. METODOLOGÍA DE GESTIÓN DEL SI

1. GESTIÓN DEL PROYECTO

Para la gestión del proyecto nos basaremos en la guía de buenas prácticas PMBoK asumiendo las siguientes cinco fases para definir el ciclo de vida del proecto.

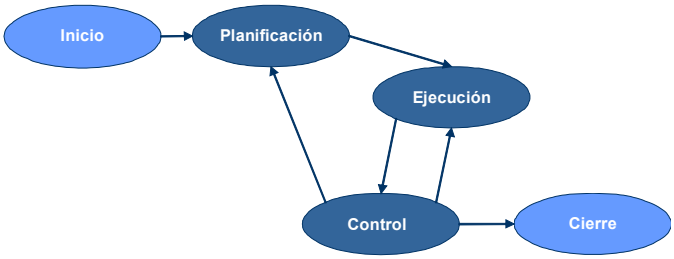

Figura 63. Fases de la gestión del proyecto.

Durante las 5 fases de gestión del proyecto necesitaremos realizar una serie de tareas que tienen que ver con una serie de áreas de conocimiento (se muestra el mapeo a continuación) que nos guíen a los largo de nuestra práctica de gestión sobre temas como la gestión de la integración, del alcance, de tiempos, de costes, de recursos humanos, de comunicaciones o de aprovisionamientos para nuestro caso.

Figura 64. Relación de áreas de conocimiento y fases de gestión del proyecto.

La gestión de riesgos y la gestión de recursos las asumimos a través de los apartado de las guidelines dedicadas a ellas, pero el resto encajan a la perfección con la guía de PMBoK.

Las fases del ciclo de vida la debemos de asumir según las siguientes indicaciones:

1. **Inicio**: Será importante para la integración del proyecto con el resto de la organización, por el hecho de que nos encargamos de abrir el proyecto e iniciar el proceso de definición de su alcance. Aquí debemos realizar el estudio de viabilidad del proyecto.
2. **Planificación**: Es una fase de mayor importancia de lo que parece, ya que aquí se deberá identificar de forma exacta cuál es el tipo de proyecto, consolidar las fechas del

mismo, hacer recopilación de documentación y realizar la planificación detallada del proyecto (dónde puede que el mismo deba dividirse en varios). Aquí es muy importante la gestión de recursos humanos que se aplique al mismo. Sin una buena planificación el resto de tareas pueden ser tediosas.

3. **Ejecución**: En base a la metodología de desarrollo aplicada se implementará el proyecto. Dado que se quiere integrar el proyecto en un proceso de mejora continua la adaptación del ciclo de vida del producto verá cómo se suceden iteraciones en el tiempo.
4. **Control**: En base a la política de seguimiento y calidad que se identifica se realizaran reuniones periódicas de seguimiento para comprobar la evolución del proyecto y poder abordar acciones correctores en tiempos óptimos.
5. **Cierre**: Una vez validada toda la documentación, y desplegado el sistemas en el entorno de producción, se procederá al cierre formal de la iteración del proyecto e integrando la primera piedra para volver a iniciar la iteración de mejora.
6. **Mantenimiento**: Durante el desarrollo de una iteración, puede que sean necesarias modificaciones sobre la versión de producción, con la incorporación de nuevas funcionalidades, con la modificación o actualización de las ya implantadas y con la corrección de errores detectados.

Es importante conocer la evolución de las aplicaciones a lo largo de las sucesivas versiones, con el objetivo de obtener un seguimiento de la evolución funcional y técnica a lo largo de su vida útil.

El escenario ideal es establecer el marco de buenas prácticas de ITIL para ese periodo de mantenimiento de la aplicación productiva.

GESTIÓN DEL PROYECTO Y GOBIERNO TI

Dado que no tratamos un proyecto de DW y BI como un ente aislado al resto de la organización, debemos involucrar la gestión del mismo junto con el gobierno de las TI. Por ello acoplaremos el proceso específica de COBIT denominado BAI01 – “Gestionar programas y proyectos” a nuestra gestión. Con el se abarca la dirección de programas (grupos de proyectos relacionados – puede ser la creación de un CMI, evolución de un DW, especificación de un nuevo DM, etc. – para administrarlos de forma coordinada para obtener beneficios y control que no se obtendrían se se gestionaran de forma individual) y del Portfolio (grupo de Programas y Proyectos), tal como se ilustra en el siguiente diagrama:

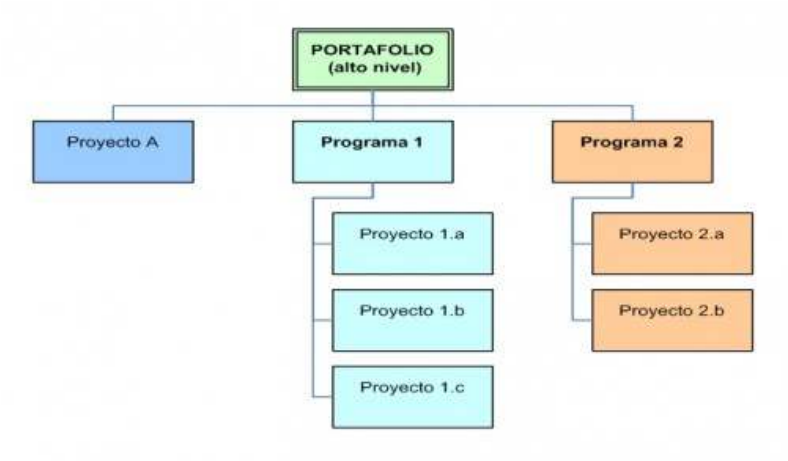

Figura 65. Gestión del portfolio, programas y proyectos.

Dado que PMBOK sólo se limita a la implementación y adaptación de procesos con buenas prácticas y a la gestión de los proyectos, la delimitación del alcance global con respecto al Portafolio, los programas de proyectos y los propios proyectos puede verse en la siguiente figura.

Figura 66. Alcance de la gestión del portfolio, los programas y los proyectos.

Se trata de ver el modelo de gestión como un conjunto de prácticas identificadas dónde se incluyas una lista de entradas requeridas, una serie de actividades que deben llevarse a cabo y que generen una serie de salidas esperadas a través del mapeo de PMBOK sin perder de vista el contexto de gobierno que propone el proceso de COBIT.

Figura 67. Modelo de gestión E-A-S.

2. GESTIÓN DE RECURSOS

La empresa deberá determinar y suministrar los recursos necesarios para:

- Realizar el trabajo de implementación y desarrollo y buscar la evolución continua en base a la mejora de capacitación, gracias a la experiencia del mismo recurso o la incorporación de nuevos recursos con las capacidades necesarias.
- Mantener una relación adecuada de recursos a lo largo del tiempo para ajustarse a la planificación del proyecto.
- Evitar que el desarrollo del sistema permanezca inmóvil por falta de capacitación.
- Mejorar la satisfacción del cliente cumpliendo con los requisitos.

Se deberá identificar el modo en que la organización determina los requisitos en términos de recursos. Los recursos incluyen no sólo el personal sino también las instalaciones, el material y los equipos y suministros. Se deberá realizar las siguientes tareas de gestión:

- Determinar la competencia necesaria para el personal
- Suministrar una capacitación que haga posible la satisfacción de estas necesidades
- Evaluar la eficacia de la capacitación (El sistema de evaluación debe alinearse a sus objetivos y planes de empresa.)
- Asegurarse de que el personal tome conciencia de la importancia de sus propias actividades y de cómo contribuye a los objetivos de calidad
- Mantener los datos de registro en apoyo de lo dicho anteriormente

Parece algo obvio, pero dadas las características de los desarrollos DW-BI, la empresa deberá determinar, suministrar y mantener las infraestructuras necesarias para lograr la conformidad a los requisitos de producto, incluyendo:

- Edificios, espacio de trabajo e instalaciones
- Equipos de proceso, hardware y software
- Servicios de soporte, como transporte o comunicaciones

3. MEJORA CONTINUA

Es necesario establecer niveles de madurez para los procesos que se han identificado a través de la gestión del proyecto. No se puede estancar un proyecto de BI en los niveles de calidad de una fecha del pasado sino que tiene que ir evolucionando a lo largo del tiempo. Por ello, con cierta periodicidad que se establezca en los comités de gobierno y gestión de proyectos se debe valorar que niveles de madurez se necesitan para los proyectos para focalizar esfuerzos en base a mejorar los que estén por debajo del nivel deseado.

Para el modelo de niveles de madurez se asume la constelación de desarrollo del modelo de mejora y evaluación de procesos CMMI (CMMI-DEV).

Nuestro modelo de madurez será el siguiente:

Figura 68. Modelo de madurez.

El marco propuesto tiene como finalidad que las aplicaciones de BI resuelvan las necesidades que motivaron su creación a lo largo del tiempo, y que su diseño y construcción faciliten la integración con las aplicaciones ya existentes (o futuras) además de adaptarse a los estándares corporativos, favoreciendo que los entornos de ejecución de las aplicaciones sean sencillos para los usuarios y, por tanto, mejora los ratios de productividad.

El enfoque que debe tomar la mejora continua del sistema BI nos ofrece el siguiente mapeo de niveles de madurez con áreas de proceso:

NIVEL		ENFOQUE	ÁREAS DE PROCESO
5	En optimización	Mejora continua del proceso	Innovación y despliegue organizativo Análisis causal
4	Gestionado cuantitativamente	Gestión cuantitativa	Performance de procesos organizativos Gestión cuantitativa de proyectos
3	Definido	Estandarización del proceso	Desarrollo de requisitos Solución técnica Integración de producto Verificación Validación Enfoque al proceso organizativo Definición del proceso organizativo Formación organizativa Gestión integrada del proyecto Gestión de riesgos Análisis y toma de decisiones
2	Gestionado	Gestión de proyectos básica	Gestión de requisitos Planificación del proyecto Seguimiento y control del proyecto Gestión de acuerdos con proveedores Medición y análisis Aseguramiento de la calidad Gestión de la configuración
1	Inicial	Sin áreas de proceso – ¡el trabajo se realiza de alguna manera!	

Figura 69. Enfoque de mejora en base al nivel establecido.

4. Aseguramiento de la Calidad del SI

1. Calidad del dato

Para poder estimar objetivamente si un dato es de calidad, se deberá incluir en cada caso si se cumplen las siguientes propiedades para aquellos datos que se consideren más críticos o sean de mayor uso. Todas las propiedades no serán necesarias de mapear para un mismo dato sino que deberá establecerse el nivel por campo de aplicación.

Las características son las siguientes ordenadas alfabéticamente:

a. Accesibilidad.

El grado en el que el dato puede ser accedido en un contexto específico de uso, particularmente por la gente que necesita el soporte de tecnología o una configuración especial porque tiene alguna indisponibilidad

b. Acceso seguro.

El hecho que el acceso a los datos pueda ser restringido y además mantenga la seguridad.

c. Cantidad apropiada de datos.

El hecho en que la calidad y el volumen habilitado para los datos es apropiado para la tarea actual.

d. Completitud.

El hecho que los datos son suficientemente amplios, profundos y están en el ámbito para la tarea actual.

e. Confidencialidad.

El grado en el que el dato tiene atributos que aseguran que éste es sólo accesible e interpretable por usuarios autorizados en un contexto específico de uso

f. Conformidad.

El grado en el que el dato tiene atributos que se adhieren a las normas, convenciones o regulaciones vigentes y reglas similares relacionadas con localidad de datos en un contexto específico de uso.

g. Consistencia.

El grado en el que el dato tiene atributos que son libres de contradicción y son coherentes con otros datos en un contexto específico de uso.

h. Credibilidad.

Es el grado en el que el dato tiene atributos que son considerados como verdaderos y creíbles por usuarios en un contexto específico de uso.

i. Disponibilidad.

Disponibilidad: El grado en el que el dato tiene atributos que le permiten ser recuperados por usuarios autorizados y/o aplicaciones en un contexto específico de uso.

Disponibilidad del sistema: Describe el porcentaje de tiempo que la fuente o el sistema de almacén de datos está disponible.

Disponibilidad transaccional: Describe el porcentaje de tiempo que la información en los almacenes de datos o fuentes está disponible debido a la ausencia de procesos de actualización de datos, los cuales bloquean la escritura en estos.

j. Eficiencia.

El grado en el que el dato tiene atributos que pueden ser procesados y proporciona los niveles esperados de desempeño al utilizar las cantidades y los tipos de recursos apropiados en un contexto específico de uso.

k. Entendibilidad.

El grado en el que el dato tiene atributos que le permiten ser leído e interpretado por usuarios, y es expresado en lenguajes apropiados, símbolos y unidades en un contexto específico de uso.

l. Exactitud.

El grado en el que el dato tiene atributos que representan correctamente el valor verdadero del atributo instanciado en un concepto o evento en un contexto específico de uso.

m. Facilidad de manipulación.

El grado en que el dato es fácil de manipular y añadir a tareas diferentes.

n. Grupo formado por la profundidad, actualidad y volatilidad.

Oportunidad: El hecho que la edad de los datos es apropiada para la tarea actual.

Actualidad: describe cuando la información fue ingresada en la fuente o/y en el almacén de datos.

Volatilidad: describe el período de tiempo en que la información es validad en el mundo real.

o. Interpretabilidad.

Interpretabilidad: El hecho que los datos son expandibles, adaptables y fácil adición para otras necesidades.

Interpretabilidad de los datos: Concierne con la descripción delos datos (es decir, disponer de datos de los sistemas heredados y datos externos, tablas de descripción de las bases de datos relacionales, llaves primarias y foráneas, alias, predefinidas, dominios, explicación de códigos de valores, etc.).

Interpretabilidad del modelo: Describe el grado para el que el almacén de datos es eficientemente modelado para ser repositorio de información. Es decir, lo más fácil en que se pueden realizar las consultas. (Esta interpretabilidad no se orienta tanto a los datos como al esquema).

p. Legalidad.

Dependiendo de las leyes en las que se mueva nuestra organización deberemos atender a los requisitos establecidos por las leyes vigentes, así, para el caso de

España se debe atender a la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal.

q. OBJETIVIDAD.

El hecho que los datos no tiene sesgos (sin prejuicios) e imparcialidades.

r. PORTABILIDAD

El grado en el que el dato tiene los atributos que le permiten ser instalado, substituido o movido de un sistema a otro conservando la calidad existente en un contexto específico de uso.

s. PRECISIÓN.

El grado en el que el dato tiene atributos que son exactos o que proporcionan la discriminación en un contexto específico de uso.

t. RECUPERABILIDAD.

El grado en el que el dato tiene atributos que le permiten mantener y conservar un nivel especificado de operaciones y calidad, aún en caso de falla, en un contexto específico de uso.

u. RELEVANCIA.

El hecho que los datos son aplicable y útiles para la tarea actual.

v. REPRESENTACIÓN CONCISA.

El hecho que los datos son representados compactamente sin ser imprecisos (es decir, breve en la representación, completa y al punto).

w. REPRESENTACIÓN CONSISTENTE.

El hecho que los datos son siempre presentados en el mismo formato y son compatibles con los datos previos.

x. REPUTACIÓN.

El hecho que los datos son verdaderos o considerados altamente creíbles en términos de las fuentes o contenidos de origen

y. TRAZABILIDAD.

El grado en el que el dato tiene atributos que proporcionan un rastro de auditoría al acceso a los datos y de cualquier cambio realizado a los datos en un contexto específico de uso.

z. Valor añadido.

El hecho que los datos son beneficiados y proporcionan ventajas en su propio uso

2. Evaluación y medición

Para el modelo de evaluación y medición usaremos la familia de normas de la ISO 25000, concretamente nos centraremos en la división de evaluación de calidad (**ISO/IEC 2504n**) que está formada por cuatro normas que comentaremos más adelante para que se deben usar.

Para la medición nos basaremos en las normas **ISO/IEC 2502n** que presentan aplicaciones de métricas para la calidad de software interna, externa y en uso.

El objetivo es crear un plan de evaluación y medición basándonos en directrices que aporta las guía de calidad. El plan de evaluación debe usarse durante todo el ciclo de vida del proyecto y será pieza clave para la toma de decisiones tácticas sobre el mismo.

En primera instancia debemos establecer los **requisitos de evaluación** (basándonos en la identificación del sistema y la calidad del dato, básico para medir y poder evaluar), seguidamente **especificar la evaluación** (usando la información de las guías que comentaremos adelante) y diseñar la **estrategia de evaluación** conjuntamente con la gestión del proyecto en su fase de seguimiento y control.

A continuación se desgrana el contenido de las guías de aplicación de la familia ISO 2504n.

- **ISO/IEC 25040 – Proceso de evaluación**.

Contiene los requisitos generales para la especificación y evaluación de la calidad del software y clarifica los conceptos generales →Lo usaremos para el establecimiento de la estrategia de evaluación de la calidad del sistema y establecer los requisitos para la aplicación del proceso.

- **ISO/IEC 25041 – Guías de evaluación para los desarrolladores, compradores y evaluadores**.

Contiene los requisitos y recomendaciones específicas para los desarrolladores, compradores y evaluadores → Lo usaremos para adaptar el modelo para los recursos.

- **ISO/IEC 25042 – Módulos de evaluación**.

Se define la estructura y el contenido de la documentación que se utiliza para describir un módulo de evaluación. Estos módulos de evaluación contienen la especificación del modelo de

calidad → Lo usaremos para adquirir las métricas que vayamos a usar para la medición de la calidad, se sustituirán las características de calidad del dato que se incluye en la norma por las depuradas para un DW de éste documento.

- **ISO/IEC 25045 – Módulos de evaluación de recuperabilidad**.

No la usaremos.

BLOQUE IV: APLICACIÓN PRÁCTICA: EL PROCESO ETLUSANDO MICROSOFT BIZTALK SERVER 2010 Y MICROSOFT SQL SERVER 2008. ADAPTACIÓN DEL MODELO DE CALIDAD (EN LOS DATOS Y EN EL PROCESO).

1. INTRODUCCIÓN.

El apartado de aplicación práctica va servir para acercar el modelo de calidad descrito en la memoria a posibles casos reales, dónde nos vamos a encontrar con una serie de tecnologías a usar para realizar el proyecto de BI. Es muy común que en las organizaciones se tengan contratos de licencias con proveedores y nos tengamos que amoldar al software, plataformas y licencias que figuren en esos contratos para realizar nuestro proyecto.

Dada la experiencia en el campo, es muy conocido el hecho de comprar varias licencias al mismo proveedor por el ahorro de coste aunque no se tenga claro que funcionalidad se puede sacar de las mismas, esto ocurre sobre todo en las grandes empresas y la Administración Pública.

Vamos a utilizar dos herramientas de la familia Microsoft para acercar el proceso ETL de generación de un DW corporativo para un sistema de BI.

Como se ha visto en el apartado de Componentes, herramientas y conceptos (4) y concretamente en “4 ETL: Extract, Transform and Load” hemos dividido no sólo en tres pasos, sino en cuatro el proceso ETL ya que en la práctica la mayoría de los inconvenientes que se tienen en el proceso de desarrollo de nuestro ETL se encuentra en la interoperabilidad de aplicaciones.

Para dar solución a la interoperabilidad de aplicaciones vamos a usar la herramienta de MicrosoftBizTalk Server 2010 que nos proporciona una gran capacidad de adaptadores para conectarnos a fuentes de información, una herramienta de administración fácil e intuitiva y se integra a la perfección con Microsoft SQL Server 2008, Sistema Gestor de Base de Datos Relacional que cuyo módulo SSIS, SQL Server Integration Services nos proporciona gran cantidad de utilidades para, sobretodo, llevar a cabo las tareas de transformación de datos.

Aunque SSIS nos permite conectar con una gran cantidad de fuentes externas, la plataforma de BizTalk nos permite desencapsular ese modelo de conexión a fuentes externas hacia personal de administración en lugar de perfiles de desarrollo dotando de una capacidad de configuración más fácil y completa desde su consola de administración.

Por otro lado BizTalk puede integrarse como módulo de ESB (Enterprise Service Bus), lo cual permitiría suscribir al sistema a aquellos servicios que necesitara desde una consola de administración sin necesidad de modificar paquetes de datos o crear clientes.

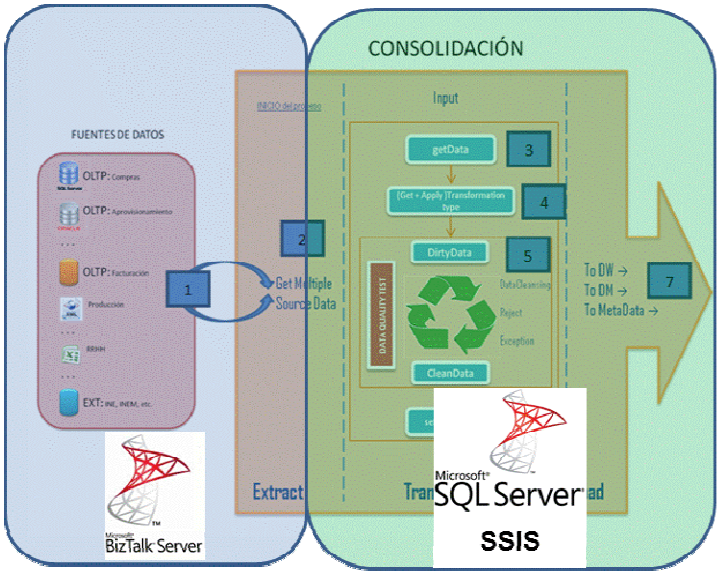

Figura 70. Mapeo de Herramientas sobre el proceso de consolidación del ETL.

2. INTEROPERABILIDAD: MS BIZTALK SERVER 2010.

1. ¿QUÉ ES BIZTALK?

Microsoft BizTalk Server o simplemente "BizTalk", es un servidor Gestión de procesos de negocio (BPM) que usa adaptadores diseñados para comunicarse con diferentes tipos de software (principalmente los más usados en las empresas de gran tamaño y otros tipos básicos de intercambio de información).

Biztalk permite automatizar e integrar los procesos de negocio configurando los diferentes adaptadores a través de una consola de administración independiente de un lenguaje de programación.

Actualmente se encuentra en su versión 2010 la cual se ha adaptado a las últimas necesidades de los intercambios B2B, B2C e incluye mejoras sobre su versión anterior (2006 R2).

Se puede ver Biztalk como un sistema de mensajería para arquitecturas empresariales, ya que usa los adaptadores para recuperar la información y convertirla en mensajes tipados por el desarrollador y facilitar su manejo pudiendo recibir cierta estructura de datos por diferentes canales (varias entradas) y generar una salida que esperamos (nuestro mensaje). Éste hecho permite conectar diferentes entidades de negocio con un repositorio central que se encargue de gestionar la información facilitando las tareas de interoperabilidad.

Además de lo comentado, nos permite crear ciertas reglas de negocio que se pueden configurar a alto nivel y permitan en base a la información contenida en los distintos mensajes

que se reciben en las entradas generar diferentes tipos de salidas sin necesidad de realizar subidas a producción de código o realizar actualizaciones en base de datos.

Los adaptadores de los que hablamos son interfaces específicas de un determinado sistema con Biztalk, por ejemplo, tenemos un adaptador para ficheros que nos permite transmitir a/desde el sistema de ficheros de una máquina, tenemos otro adaptador para transmitir datos a/desde una dirección HTTP, otro para transmitir datos a/desde una base de datos SQL Server, otro para comunicaciones SAP, etc.

2. MOTOR DE MENSAJERÍA.

Los adaptadores forman la entrada o salida de mensajes, pero el encargado de manejarlos es el motor de mensajería.

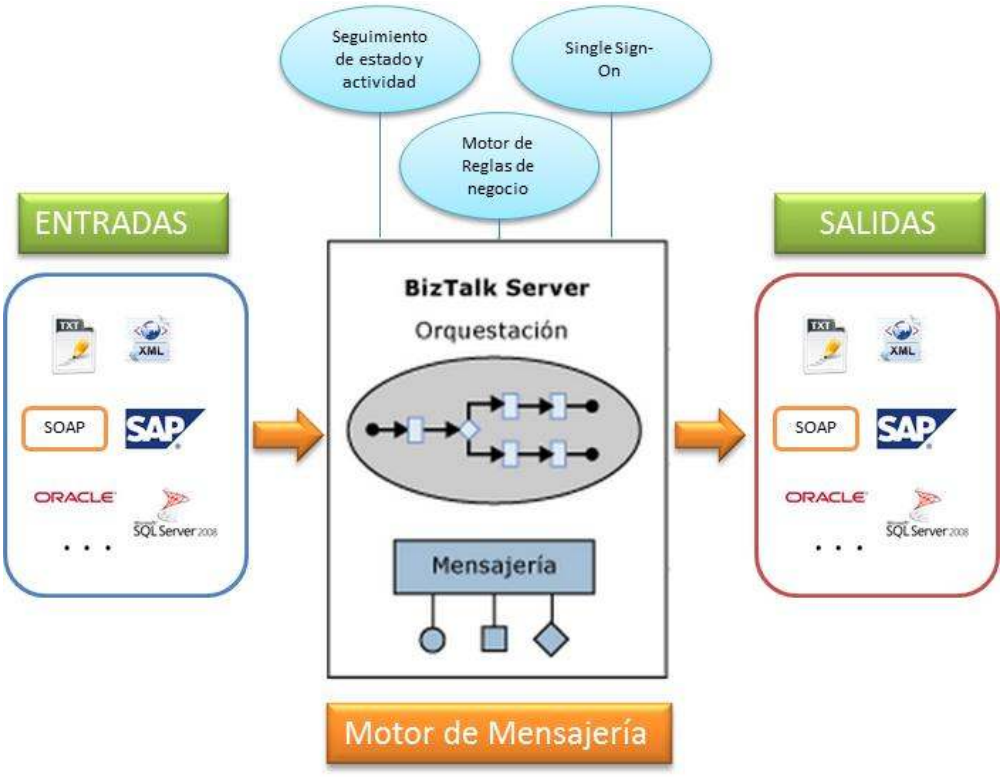

Figura 71. Motor de mensajería de Biztalk 2010.

Este motor cuenta con dos piezas principales:

- Un **componente de mensajería** que proporciona la capacidad de establecer una comunicación con otros tipos de software. Dado que depende en adaptadores para

distintos tipos de comunicación, el motor admite una amplia gama de protocolos y formatos de datos, incluidos los servicios web, entre otros.

- Un soporte para la creación y ejecución de procesos definidos gráficamente, las **orquestaciones**. Las orquestaciones, integradas sobre los componentes de mensajería del motor, implementan la lógica que dirige un proceso empresarial completo o parte de éste.

Junto con el motor pueden usarse, además, otros componentes de BizTalk, entre los que se incluyen los siguientes:

- Un **motor de reglas de negocios** que evalúa conjuntos complejos de reglas.
- Un concentrador de grupo que permite a los programadores y administradores supervisar y administrar el motor y las orquestaciones que éste ejecuta.
- Una función de **inicio de sesión único**(SSO) **empresarial** que proporciona la capacidad de asignar datos de autenticación entre Windows y otros sistemas ajenos.

El motor de mensajería permite crear procesos de negocio que abarquen varias aplicaciones al proporcionar dos elementos principales:

- Un modo de especificar e implementar la lógica que dirige el proceso empresarial.
- Un mecanismo para establecer comunicación entre las aplicaciones que utiliza el proceso empresarial.

El siguiente diagrama muestra los principales componentes del motor que afrontan estos dos problemas.

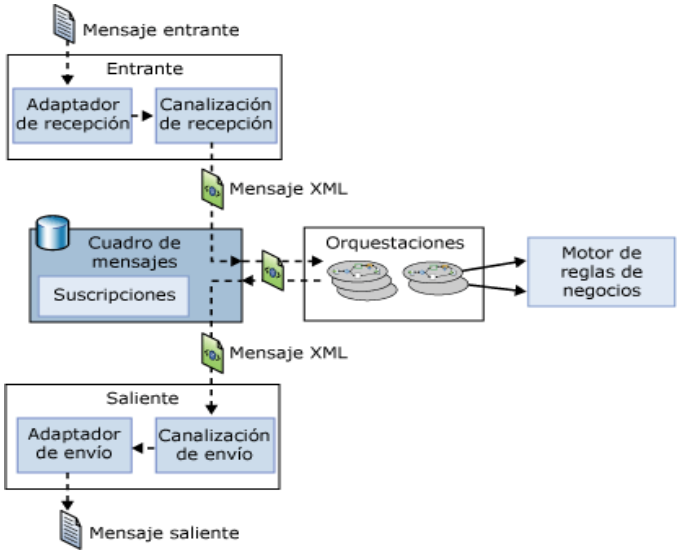

Figura 72. Estructura del motor de mensajería de Biztalk.

3. ADAPTADORES.

Un adaptador es un componente de software que permite enviar y recibir mensajes fácilmente con un mecanismo de entrega que cumple un estándar ampliamente reconocido, como SMTP,

POP3, FTP o Microsoft Message Queue Server (MSMQ). Según BizTalk Server ha ido evolucionando, ha aumentado la necesidad de adaptadores que habiliten con rapidez la conectividad con las aplicaciones y tecnologías más utilizadas.

Normalmente incluye gran cantidad de adaptadores que veremos a continuación, pero la escalabilidad de la arquitectura de Biztalk también es posible crear adaptadores personalizados para soluciones específicas.

Cada uno de los adaptadores nativos está asociado con una **ubicación de recepción** diseñada para escuchar mensajes de un transporte específico (tipo de transporte) en una dirección determinada.

Una vez recibido el mensaje en la ubicación de recepción, se transmite éste al adaptador. El adaptador adjunta la secuencia de datos al mensaje (normalmente en el cuerpo del mensaje), agrega los metadatos que pertenecen al extremo del que se recibieron los datos y envía el mensaje al motor de mensajería de BizTalk.

De manera predeterminada, cuando se ejecuta el Asistente para configuración de BizTalk, el asistente instala los adaptadores nativos y crea un controlador de adaptador con una configuración predeterminada para cada uno, pero gracias al **Explorador de BizTalk y la consola de administración de BizTalk Server**, se puede modificar la configuración predeterminada de los controladores de adaptadores y agregar, quitar y modificar los puertos de envío y las ubicaciones de recepción de los adaptadores sin necesidad alguna de realizar nuevos despliegues de la solución, por tanto, es una herramienta que puede operar el departamento de sistemas sin necesidad de los desarrolladores.

El uso de adaptadores simplifica notablemente la transferencia de mensajes ya que si se utiliza alguno de los transportes para los que hay un adaptador de BizTalk, el proceso de envío o recepción de mensajes es tan simple como configurar el adaptador correspondiente.

Obviamente no todos los adaptadores tienen las mismas propiedades y será necesario conocer como configurar cada uno por separado, si bien hay una serie de propiedades que pueden compartir como son el tipo de comunicación (unidireccional o bidireccional), la compatibilidad con transacciones, compatibilidad de recepción por orden (Id de mensaje o sobre un campo específico), si permite el uso de Single sign-On o el proceso dónde se aloja (Proceso levantado por Biztalk o por el Servidor Web Internet Information Services IIS).

A continuación vamos a comentar algunos de los tipos de adaptadores que pueden ser de mayor utilizad para la creación de una solución de BI en entornos empresariales.

A. FICHERO (FILE).

En la mayoría de las empresas existen aún hoy en día sistemas Legacy que generan ficheros de intercambio de datos con extensiones .TXT o .CVS, por ejemplo. El adaptador FILE o de

Fichero nos permite identificar una carpeta (ubicación de recepción) dónde se recogerán dichos ficheros y que el motor de mensajería, a través de orquestaciones o directamente con la creación de ubicaciones de envío se transmita de un lugar a otro.

Podremos configurar aspectos como su codificación, establecer cuál será el salto de línea y retorno de carro e incluso a través de orquestaciones, para un formato establecido poder generar objetos de información más complejos leyendo el fichero según un patrón introducido, todo configurado a través de un asistente. Éste será su uso más común, por ejemplo podemos tener definido el siguiente patrón:

- Carácter #: Indica nuevo registro.
- Carácter |: Indica nuevo campo.
- Orden de campos: Campo1 – Nombre, Campo 2 - Apellido 1, Campo 3 – Apellido 2.

Y podemos recibir en nuestra ubicación de recepción un fichero con el siguiente contenido:

```
#Pablo|Martín|Gutiérrez

#Pepe|Pérez|Pérez
```

Biztalk es capaz de generar dos objetos en memoria que contengan la siguiente información sin necesidad de programar nada:

Objeto 1: Nombre = Pablo

Apellido 1 = Martín

Apellido 2 = Gutiérrez

Objeto 2: Nombre = Pepe

Apellido 1 = Pérez

Apellido 2 = Pérez

Con los dos objetos podremos realizar las operaciones oportunas como puede ser llamar a un servicio web por cada objeto, realizar inserciones en una base de datos o simplemente generar un mensaje de salida predefinida.

La ventaja principal de éste adaptador es su facilidad de uso, pero no permite ninguna del resto de propiedades que hemos identificado.

B. FTP.

Similar al adaptador FILE existe el adaptador FTP, dónde la principalmente diferencia reside en que permite recuperar los ficheros alojador en un servidor de FTP. Por tanto, las opciones de configuración del adaptador de FTP son diferentes a las del fichero FILE ya que entre otras se

debe indicar la dirección del servidor FTP (p.ej. una dirección IP), establecer las credenciales que nos permitan acceder al servidor (si se requieren), etc.

Su principal ventaja es que sirve para conectar sistemas Legacy que en su día fueron segregados y securizados en servidores FTP mientras que el tipo FILE no lo permite además de permitir el uso de Single sign-On.

C. SOAP.

Para entornos distribuidos de los últimos años será el adaptador más usado ya que permite conectar a Biztalk con servicios web, es decir, se encarga de escuchar mensajes SOAP.

Además de usarse como ubicación de recepción, Biztalk tiene un asistente mediante el cual nos permite publicar ficheros XML como servicios Web .ASMX incluso orquestaciones, es decir, que una orquestación sea capaz de escuchar mensajes de entrada a través de una URL que establece el adaptador SOAP, éste será un método muy útil de crear servicios web aportando funcionalidad al mismo sin necesidad de manejar un lenguaje de programación sino a través del editor gráfico de orquestaciones que veremos más adelante.

Algunas de las propiedades que podemos usar para configurar el adaptador SOAP, serán:

- UseSoap12: Sirve para especificar si usará un proxy que admite protocolo SOAP 1.1 o SOAP 1.2 ,
- ClientConnectionTimeout: Sirve para especificar el timeout o tiempo máximo de espera para el establecimiento de una conexión con el servicio web.
- ClientCertificate: Sirve para especificar la huella digital de un certificado SSL.
- Username: Sirve para especificar el usuario que autenticará la conexión al servicio web.
- Password: sirve para especificar la contraseña que autenticará la conexión al servicio web.
- Etc.

El adaptador evidentemente permite la comunicación bidireccional de servicios web, también puede configurarse como unidireccional, es decir, como servicio web asíncrono. Además se permite el uso de single Sign-On.

D. WCF.

WCF (Windows Communication Foundation) es un marco de trabajo para la creación de aplicaciones orientadas a servicios de Microsoft. Es una alternativa a servicios web que usan protocolo SOAP que ha creado Microsoft y que muchos de sus productos usan como por ejemplo MS Sharepoint.

El adaptador WCF será muy importante para entornos que usen tecnologías Windows, debido a su gran variedad de posibilidades de configuración que permite conectar entre múltiples tipos de software.

Destacan los siguientes subtipos de adaptador:

- **WCF-BasicHttp**: Se comunica con servicios y clientes web basados en ASMX y con otros servicios compatibles con el Perfil básico de servicios web WS-I, versión 1.1 mediante HTTP o HTTPS. Permite Single Sign-On y comunicación bidireccional.
- **WCF-WSHttp**: Admite los estándares WS-* a través de transporte HTTP. Permite el uso de transacciones, Single Sign-On y comunicación bidireccional.
- **WCF-NetMsmq**: Admite las colas mediante la utilización de Microsoft Message Queue Server (MSMQ) como transporte. Permite transacciones, recepción de mensajes por orden, Single Sign-On y su comunicación es unidireccional.
- **WCF-NetTcp**: Admite los estándares WS-* a través de transporte TCP. Permite el uso de transacciones, Single Sign-On y comunicación bidireccional.

E. POP3 y SMTP.

Sirven para conectar cuentas de correo electrónico con Biztalk. Cada protocolo se configura siguiendo sus ventajas y desventajas, POP3 descargará todos los mensajes recibidos a la ubicación que se indique mientras que SMTP permite el envío de correos electrónicos.

Éste tipo de adaptador es muy útil para el proceso de ficheros que llegan de terceros con los que no se ha creado una interfaz automatizada sino que se dedican a adjuntar ficheros a correos electrónicos y usualmente se encarga un operario de integrar el fichero a la solución.

Un gran atajo de desarrollo sería combinar los adaptadores POP3 y SMTP con el de tipo FILE para que tras la recepción de un correo con datos adjuntos se procese el mensaje adjunto.

Para nuestro objeto de estudio puede ser de gran utilidad ya que muchos informes son enviados por correo electrónico cada cierto tiempo, podríamos generar una orquestación que tras recibir el correo electrónico (Adaptador POP3) verifique que existe el documento adjunto y su formato y en caso de que sea erróneo notificar mediante un correo electrónico de respuesta la situación al usuario emisor (Adaptador SMTP). En el caso de que sea correcto el envío alojar el documento adjunto en una ubicación de recepción configurada con un adaptador del tipo FILE.

F. SQL.

El adaptador SQL permite la comunicación directa entre una base de datos SQL Server y MS Biztalk. Se suele usar para sondear datos de una o varias tablas y transmitirlos como uno o varios mensajes XML a BizTalk que se usarán en procesos mayores.

También se puede usar el adaptador de SQL para mover grandes cantidades de datos a una base de datos de SQL Server o desde ella como parte de una solución de orquestación o mensajería de BizTalk, acción que no se recomienda ya que existen otras herramientas como SSIS que están destinadas a ello. Además, el adaptador de SQL permite insertar, actualizar y eliminar datos de tablas de SQL Server mediante updategrams de SQL o invocando a procedimientos almacenados.

El uso del adaptador será de tipo petición-respuesta y evidentemente permite el uso de transacciones.

G. SAP.

Como hemos indicado en el comienzo del apartado, Biztalk se integra con las soluciones de software empresarial más usadas, y sin duda una de ellas es SAP.

SAP es un conjunto de aplicaciones de negocios que provee integración de información, colaboración, funcionalidad específica de industria y opciones de crecimiento.

Habilita el intercambio de documentos intermedios (IDOC), BAPI y los mensajes de llamada a función remota (RFC) entre BizTalk Server y un sistema SAP R/3.

No viene por defecto instalado en Ms Biztalk Server, sino que se necesita instalar como un añadido.

Es compatible con las versiones SAP R/3 4.x y R/3 6.20 (empresas).

H. ORACLE.

Habilita la lectura y la escritura de información de y en la base de datos de un servidor Oracle a través de un adaptador ODBC para bases de datos de Oracle.

Para MS Biztalk 2010 viene por defecto definida su compatibilidad para las versiones Oracle 8i (8.1.6.0), 9i (9.2.0.1) o 10g de Oracle.

I. OTROS.

Existen otros adaptadores tanto que vienen preinstalados por defecto tanto de instalación como añadidos. Además hay adaptadores de terceros que implementan a través de Adaptadores personalizados aquellas carencias que se pueden deterctar en los adaptadores de Microsoft como puede ser el Adaptador por la versión 11g de oracle.

Algunos de ellos son:

- PeopleSoft Enterprise.
- JD Edwards OneWorld XE y JD Edwards EnterpriseOne.
- Aplicaciones Siebel eBusiness.
- TIBCO Rendezvous y TIBCO Enterprise Message Service.
- WebSphere MQ.
- HTTP(s).
- MQSeries.
- Etc.

4. APLICACIONES.

Hablando de forma puramente teórica, una **aplicación de BizTalk** es una agrupación lógica de elementos, denominados "artefactos", que se utiliza en una solución empresarial de BizTalk Server. Los artefactos son los adaptadores, las orquestaciones, las ubicaciones de recepción, etc.

Se utilizar para administrar un conjunto de artefactos como una unidad independiente y favorecer las tareas de administración y supervisión, sobre todo para entornos productivos dónde en un mismo servidor de Biztalk Server residen distintas soluciones de SW que aportan funcionalidades heterogéneas.

A. UBICACIONES DE RECEPCIÓN Y PUERTOS.

Algunos de los artefactos de Biztalk más importantes son las ubicaciones de recepción y los puertos tanto de envío como de recepción.

Las **ubicaciones de recepción** sirven para indicar un punto de acceso por el que llegarán mensajes de entrada, el tipo de mensaje de entrada dependerá del adaptador que se use para configurar la ubicación de recepción.

Estas ubicaciones de recepción suelen asociarse a **puertos de recepción** que iniciarán la funcionalidad diseñada para contestar al evento disparado por el mensaje de entrada. Los puertos de recepción son artefactos lógicos, el artefacto físico sería la ubicación de recepción, y es por ello que pueden asociarse varias ubicaciones de recepción a un solo puerto de recepción, es decir, podemos usar varias ubicaciones (con varios adaptadores, p.ej. FILE y FTP o dos de tipo FILE) con su dirección física (ya sea una ruta, una URL, etc.) pero que internamente inicie su operativa desde un puerto de recepción. Los puertos de recepción pueden ser tanto unidireccionales como bidireccionales y se suelen usar como punto de entrada de las orquestaciones para aportar el flujo de negocio.

Un **puerto de envió**, es una unidad lógica (no física) que utilizar Biztalk Server para enviar mensajes. Se utilizar para iniciar una acción de comunicación como puede ser llamar a otro

puerto, en este caso de recepción directamente o alojar un mensaje en una ubicación de recepción.

B. ORQUESTACIONES.

Las orquestaciones son componentes que pueden realizar suscripciones para recibir y enviar mensajes a través de Biztalk server. Mediante las orquestaciones podremos ejecutar diversas acciones sobre los mensajes entrantes como especificar árboles de decisión, ejecutar bucles, llamar a otras orquestaciones, consultar datos en una base de datos o aplicar una transformación a un mensaje de entrada para adecuarlo al mensaje de salida esperado.

Se suele tomar como entrada puertos de recepción, y en algunos casos hay puertos de envío como salida. Se puede configurar en tiempo de diseño la configuración de los puertos, pero es recomendable que se desarrolle la orquestación para que se realice la acción a posteriori desde la consola de administración de Biztalk Server y que sea el equipo de sistemas quién maneje la información de producción.

Microsoft Visual Studio nos proporciona un diseñador gráfico con el que realizar éstas acciones arrastrando formas desde el panel de herramientas y uniendo unas con otroas.

Podemos ver las diferentes funcionalidades que nos permite el diseñador de orquestaciones arrastrar al panel de diseño en el siguiente enlace http://msdn.microsoft.com/es-ES/library/aa577803(v=bts.10)

La apariencia de una orquestación puede ser como la siguiente:

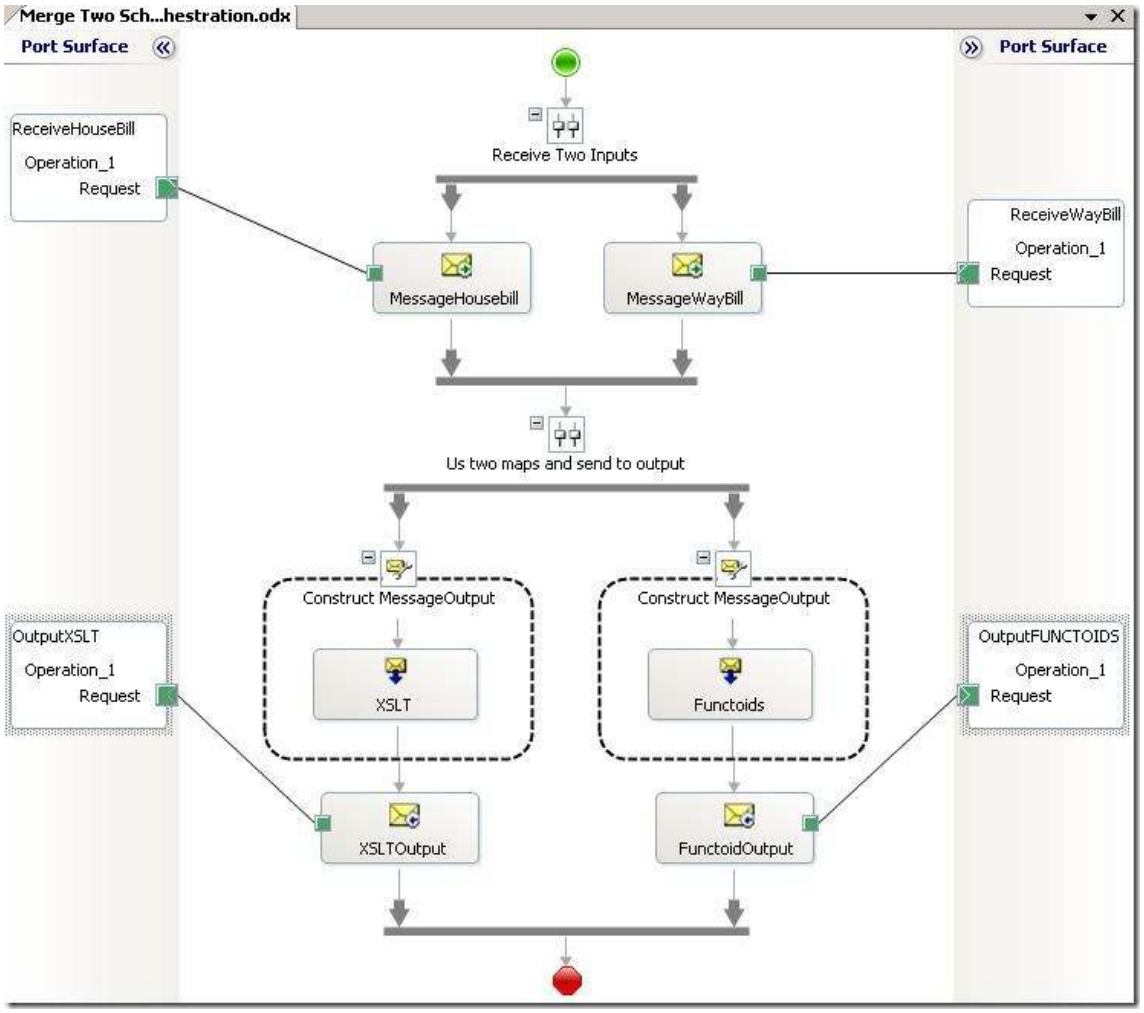

Figura 73. Ejemplo de orquestación.

a. PIPELINE.

El pipeline es un componente que proporciona una implementación del patrón de integración de canalizaciones y filtros. Durante la recepción y el envío de mensajes, pueden existir motivos para realizar ciertas transformaciones en los mensajes para poderlos procesar o que el destinatario los acepte.

Un ejemplo común es que podría necesitar transformar un archivo sin formato delimitado por comas en un archivo XML a fin de beneficiarse de algunas características de BizTalk Server como las asignaciones.

Cada pipeline tiene un conjunto de fases que se ejecutan por orden al ejecutar la canalización. Cada fase puede incluir cero o más componentes.

Existen dos tipos de pipelines que coinciden con los puertos en los que se ejecutan:

1. **De Envío**: Se ejecutan en puertos de envío y en la parte de respuesta de un puerto de recepción de solicitud-respuesta. Tienen como finalidad usarse en mensajes que se han suscrito y se están enviando fuera de BizTalk Server.

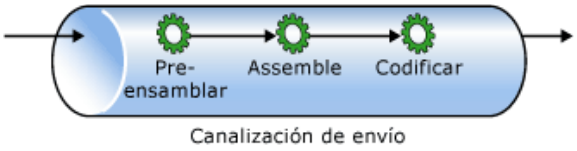

Figura 74. Pipeline de envío

   - **Preensamblar**: Realiza cualquier procesamiento de mensajes necesario antes de ensamblar el mensaje.
   - **Ensamblar**: Ensambla el mensaje y lo prepara para su transmisión realizando pasos como agregar sobres, convertir archivos XML en archivos sin formato o cualquier otra tarea complementaria a la fase de desensamblado en una canalización de recepción.
   - **Codificar**: Codifica o cifra el mensaje antes de la entrega.

2. **De Recepción**. Se ejecutan en ubicaciones de recepción y en la parte de respuesta de un puerto de envío de solicitud-respuesta. Tienen como finalidad transformar los mensajes.

Figura 75. Pipeline de recepción.

   - **Descodificar**: Descifra o descodifica los datos del mensaje.
   - **Desensamblar**: Desensambla un intercambio en mensajes más pequeños y analiza el contenido de los mensajes.
   - **Validar**: Valida los datos de mensajes, normalmente con un esquema.
   - **Resolver entidad**: Identifica la entidad del servidor BizTalk Server asociada con algún token de seguridad en el mensaje o el contexto del mensaje.

b. TRANSFORMACIONES.

Como se ha comentado anteriormente las transformaciones sirven para que en base a mensajes de entrada podamos crear mensajes de salida de distinta definición. Siempre serán mensajes XML los que se toman tanto en las entradas como en las salidas.

Las transformaciones operan dentro de orquestaciones y su uso es muy común tras recibir diversos mensajes de entrada para formar nuevos mensajes de salida.

Ya que usamos mensajes XML se usan transformaciones XSLT para crear los nuevos mensajes, y MS Visual Studio nos proporciona un editor gráfico que permite fácilmente crear esas hojas de transformación de un esquema origen (a la izquierda) a uno destino (a la derecha).

En el diseñador podremos arrastrar al panel una serie de funciones llamadas “functoids” que nos permiten realizar diversas operaciones como usar bucles, convertir tipos, acumular resultados, acceder a bases de datos, etc que se arrastran al centro del diseñador.

Podemos ver un ejemplo en la siguiente figura:

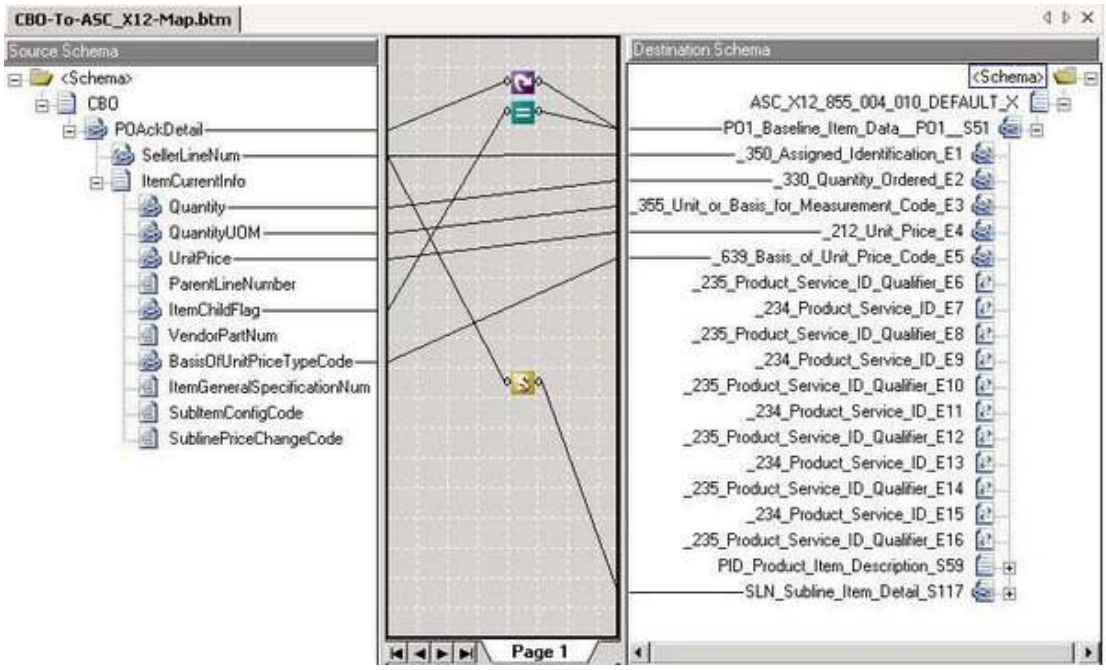

Figura 76. Ejemplo de transformación.

La lista de categorías de functoids se puede ver en el siguiente enlace http://msdn.microsoft.com/es-ES/library/aa546768(v=bts.10)

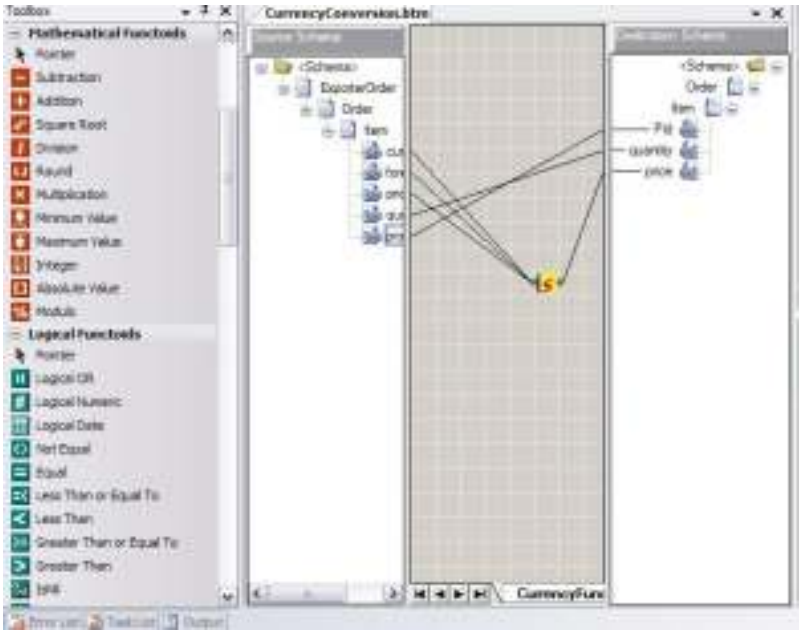

Figura 77. Ejemplo de transformación II.

5. CICLO DE VIDA DE UN MENSAJE EN BIZTALK SERVER

Una vez realizada la vista por encima de MS Biztalk, conviene aclarar cuál será realmente el funcionamiento de una aplicación y como fluirán los mensajes a través de ella. Por eso vamos a explicar brevemente en que consiste el ciclo de vida de un mensaje en Biztalk Server.

En la siguiente figura hemos detallado 7 pasos fundamentales por los que suele fluir un mensaje estándar de una solución de Biztalk.

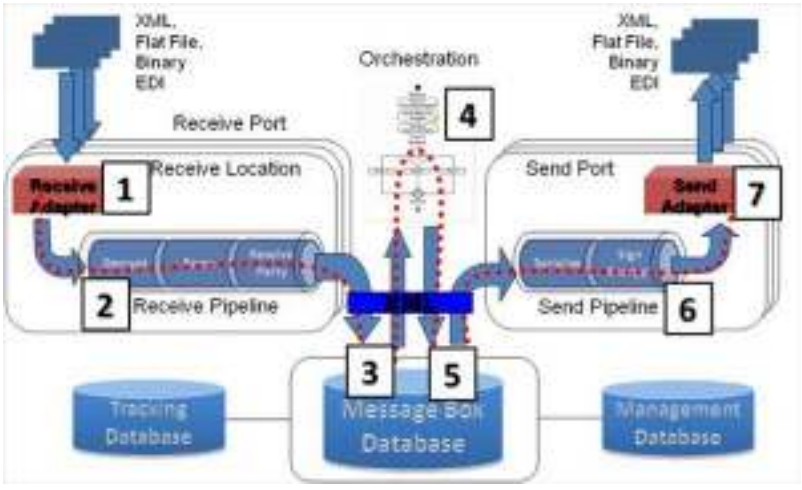

Figura 78. Ciclo de vida de un mensaje en Biztalk Server

1) **Recepción del mensaje**: Teniendo configuradas ubicaciones de recepción (receive location) se procede a la recepción con la ayuda del adaptador de recepción (receive

adapter de tipo XML, EDI, etc.) a recibir el mensaje de entrada en el puerto de recepción (receive port).

2) **Se ejecuta la transformación de recepción según el pipeline**: En caso de implementar alguno de los componentes de la pipeline de recepción se canalizará el mensaje según los mismos.
3) **Se inicia el motor de mensajería**: Gracias al motor de mensajería se almacena el mensaje y se obtiene la lista de orquestaciones suscritas a su recepción, que serán las que se inicien desde aquí.
4) **Ejecución de la orquestación**: Se ejecuta la funcionalidad de la orquestación.
5) **El motor de mensajería procede con la transmisión del mensaje**: Con la salida que provee la orquestación se procede a recopilar en el motor de mensajería los destinatarios a los que enviar el mensaje de salida.
6) **Se ejecuta la transformación de envío según el pipeline**:En caso de implementar alguno de los componentes de la pipeline de envío se canalizará el mensaje según los mismos.
7) **Se envía el mensaje**: a través de puertos de envío (send port) y gracias a los adaptadores de envío oportunos (send adapter de tipo XML, etc.) se procede al envío de los mensajes.

Para llevar más cerca el contenido teórico vamos a trasladar éstos puntos teóricos a un caso práctico. Imaginamos que tenemos un sistema Legacy que genera ficheros .ZIP con un listado de los clientes que hacen compras hoy, para incluir la información de nuestro fichero en el proceso y que sea de utilidad para una solución de BI podríamos realizar lo siguiente.

Con MS Biztalk cargaríamos el fichero en la base de datos de Stagging para que se encargue el ETL de llevar los datos a nuestro DW. Simplemente necesitamos trasladar la información del fichero de texto en las tablas oportunas, para ello el flujo de nuestro mensaje podría ser el siguiente, imaginamos que hemos creado una orquestación con un puerto de entrada y uno de salida:

1) **Recepción del mensaje**: Tenemos configurada una ruta, por ejemplo, D:\Compras\2012\08\ dónde se alojan los ficheros diariamente. Se debe configurar una ubicación de recepción que apunte a nuestra ruta y configurar un adaptador de recepción de tipo FILE. Asociamos el puerto de entrada definido en nuestra orquestación con la ubicación de recepción para que el motor de mensajería pueda asociar la ubicación con la orquestación.
2) **Pipeline de entrada**:Por ejemplo descomprimimos el fichero .ZIP y sabemos que obtenemos ficheros planos .TXT con los que operar en nuestra orquestación.
3) **Se inicia el motor de mensajería**:asociando a nuestra orquestación los ficheross .TXT como entrada.

4) **Ejecución de la orquestación**:Podemos comprobar datos, obtener números de cuenta, realizar operaciones de acumulación para la nueva BBDD, etc. y generar un mensaje nuevo que será del tipo que recibe el adaptador SQL para insertar datos.
5) **Pipeline de salida**:Puede que no necesitemos realizar ninguna acción.
6) **El motor de mensajería asocia el mensaje al destinatario**.
7) **Se envía el mensaje**:Al ser un adaptador de SQL el mensaje se transformará por el adaptador en una sentencia INSERT de base de datos sobre el destino que se haya configurado. Para ello se necesita configurar el puerto de envío para que acepte mensajes del tipo especificado y configurar el adaptador para que sepa dónde tiene que insertar, se le den permisos, etc.

3. FLUJO ETL: MS SQL SERVER 2008.

1. ¿QUÉ ES SSIS?

SSIS son las siglas en inglés de SQL Server Integration Services, es una herramienta diseñada para simplificar los proyectos de ETL, siendo uno de sus objetivos reducir al mínimo las necesidades de intervención manual cuándo se realizan estas operaciones.

Para nuestros objetivos se debe integrar la herramienta SSIS como motor de soluciones ETL. Siguiendo una estructura en la que gracias a MS Biztalk se resuelvan los problemas de interoperabilidad con sistemas externos y nos proporcionan la información en un formato común o incluso en la base de datos de stagging, usaremos esa entrada como punto de inicio del flujo ETL.

La recuperación de información será una tarea básica y gracias a las posibilidades de SSIS podremos realizar las transformaciones necesarias para cargar los datos en la base de datos OLAP destino.

La arquitectura de SSIS gira en torno a dos motores que son los que realizan las operaciones configuradas. Estos son:

2. ARQUITECTURA DE SSIS

1. El motor de flujo de control.

Se encarga de interpretar una serie de operaciones llamadas tareas (tasks), que se enlazan entre sí y realizan funciones tales como preparar información, realizar flujos de datos, ejecutar scripts o enviar correos electrónicos.

El flujo de control se define mediante un editor gráfico contenido dentro de la herramienta llamado “SQL Server Business Intelligence Development Studio (BIDS)”. Además de las tareas, el flujo puede incluir contenedores (containers) que sirven para agrupar varias tareas en unsola unidad lógica. Como resultado de un flujo de control obtendremos un diagrama de cajas enlazadas por flechas.

Figura 79. Flujo de control de SSIS.

Dentro del flujo de control tendremos paquetes (Package). Son la unidad de trabajo de SSIS y por tanto sobre ellos podemos guardarlos, recuperarlos y ejecutarlos. En ellos se almacena el flujo de control, las conexiones, etc.

Los paquetes tienen la extensión .dtsx como archivo físico o pueden guardarse directamente embebidos en la base de datos msdb de SSIS.

Figura 80. Paquete de SSIS

2. El motor de flujo de datos.

Utiliza orígenes y destinos de datos (tales como tablas de SQL Server, hojas Excel o archivos de texto plano) y mueve datos entre uno o varios orígenes y destinos. Además permite realizar transformaciones en los datos entre la recuperación desde el origen y la carga en el destino.

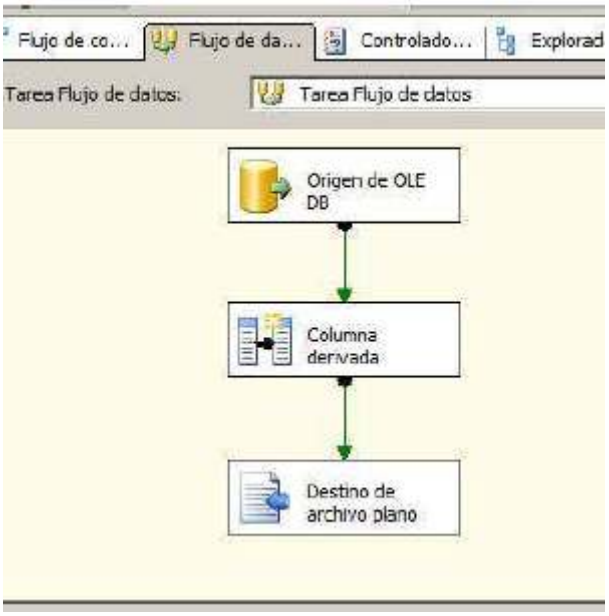

Figura 81. Data Flow de SSIS.

Al igual que el flujo de control, se diseña gráficamente en BIDS y forma un diagrama de cajas o bloques unidos por flechas. Entre la caja de origen y la caja de destino se podrán incorporar múltiples tipos de cajas que facilitarán las tareas de transformación de datos.

Conexiones de SSIS.

El tercer elemento de mayor importancia en una solución de SSIS son las conexiones. Los paquetes de SSIS utilizan conexiones para realizar las diferentes tareas de acceso a datos. Básicamente, hay que usar una conexión cada vez que se leen o graban datos en algún origen o destino de datos.

Entre las operaciones más comunes que se suelen realizar con las conexiones de SSIS podemos enunciar las siguientes:

- Conectar con almacenes de datos (origen o destino) tales como archivos de texto, archivos XML, archivos Excel o bases de datos.
- Conectar con bases de datos relacionales para realizar búsquedas con o sin filtros.
- Conectar con bases de datos relacionales para ejecutar instrucciones SQL (SELECT, DELETE, INSERT o UPDATE) o ejecutar procedimientos almacenados.
- Conectar con SQL Server para realizar tareas de transferencia y mantenimiento, tales como copias de seguridad de bases de datos o regeneración de índices.
- Escribir entradas en archivos XML o archivos de texto e incluso en tablas de SQL Server.
- Crear tablas temporales en SQL Server para usar durante las transformaciones.
- Conectar con bases de datos y proyectos de Analysis Services para tener acceso a modelos de minería de datos, procesar cubos OLAP y dimensiones o ejecutar código de DDL.
- Especificar carpetas y archivos existentes o crear carpetas y archivos nuevos para utilizar en tareas y bucles For Each.
- Aunque se recomienda usar para ello MS BizTalk Server podemos, conectar con colas de mensajes y servidores de Windows Management Instrumentation (WMI), objetos de administración de SQL Server (SMO), web o servidores de correo.

Para realizar estas conexiones, SSIS utiliza administradores de conexión (connection manager) que forman la representación lógica de una conexión. En tiempo de diseño, se establecen las

propiedades de un administrador de conexión para que describan la conexión física que crea SSIS cuando se ejecuta el paquete.

Como es obvio, estas conexiones pueden ser parametrizadas para ser configuradas en los diferentes entornos en los que se ejecute el paquete, por ejemplo de desarrollo o de explotación.

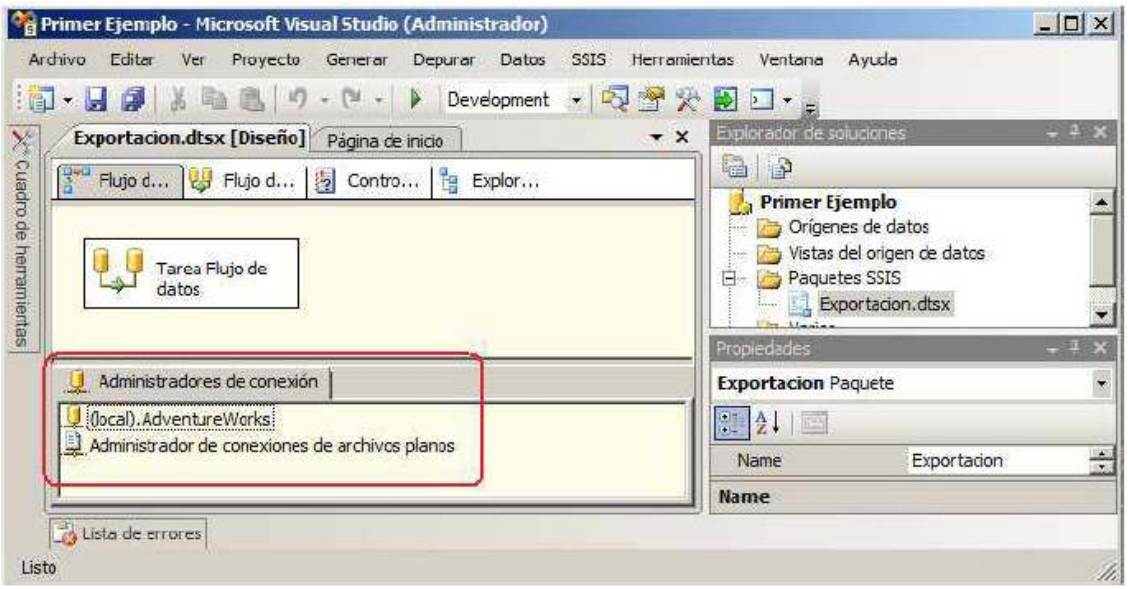

Figura 82. Conexiones de SSIS.

En la siguiente tabla podemos ver los tipos de conexión a los que nos permite conectar SQL Server por defecto. Existen paquetes de actualización de SQL Server que nos permiten utilizar más tipos de conexión, aunque como se ha mencionado en la memoria, se debe dejar los problemas de interoperabilidad para que los resuelva MS BizTalk Server y utilizar desde SSIS las conexiones más comunes que serán, ficheros XML o bases de datos.

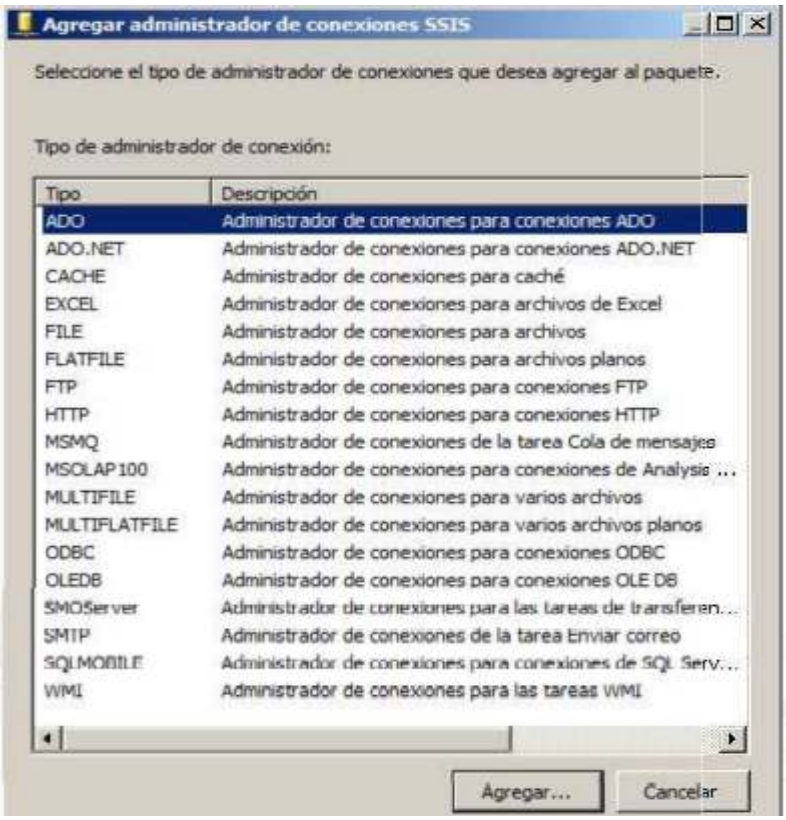

Figura 83. Conexiones de SSIS – tipos

Para realizar los ejemplos se ha usado la base de datos de prueba AdventureWorks que puede descargarse directamente desde codeplex http://sqlserversamples.codeplex.com/ o http://msftdbprodsamples.codeplex.com/ y su documentación es la siguiente http://msdn.microsoft.com/es-es/library/ms124659(v=sql.100).aspx

No obstante, para nuestros objetivos, lo ideal es manejar ficheros XML o leer de una base de datos SQL Server dónde nuestro componente de interoperabilidad (MS Biztalk Server) haya alojado la información que recupera de las diversas fuentes.

3. DESARROLLO DE PAQUETES SSIS.

A. ORÍGENES DE DATOS.

Un Origen de Datos (Data source) es una referencia, en tiempo real, a una conexión al almacén de datos. No es obligatorio usarlos, pero pueden facilitar el desarrollo de los proyectos, ya que múltiples administradores de conexiones de uno o varios paquetes pueden usar el mismo origen de datos. Proporciona varias ventajas:

- El origen de datos está disponible para todo el proyecto. Se define una sola vez pero puede configurarse desde los administradores de conexiones para que se use en varios paquetes del proyecto.
- Cuando se crea un origen de datos un administrador de conexión se copia pero el administrador de conexiones no requiere que continúe existiendo el origen de datos para funcionar.
- Si se necesita reconfigurar la conexión (p.ej. para cambios de entorno, etc.) basta con cambiar el origen de datos y la nueva conexión se propaga a todos los administradores de conexión.

Los orígenes de datos se definen desde el proyecto de Integration Services en BIDS.

Figura 84. SSIS. Orígenes de datos.

Existe otro elemento relacionado con los orígenes de datos que conviene conocer, son las Vistas del Origen de Datos (Data source View), que representan un subconjunto de datos obtenido a partir del origen de datos. Al igual que los orígenes de datos, las vistas se definen desde el proyecto de Integration Services de BIDS.

Figura 85. SSIS. Vistas del Origen de datos

Una vez vayamos a crear una vista de orígenes de datos, se ejecuta un asistente que nos ayuda a crear vistas. Nos preguntará por el origen de datos a partir del que se va a crear la vista y después nos deja elegir las tablas que formarán parte de la vista.

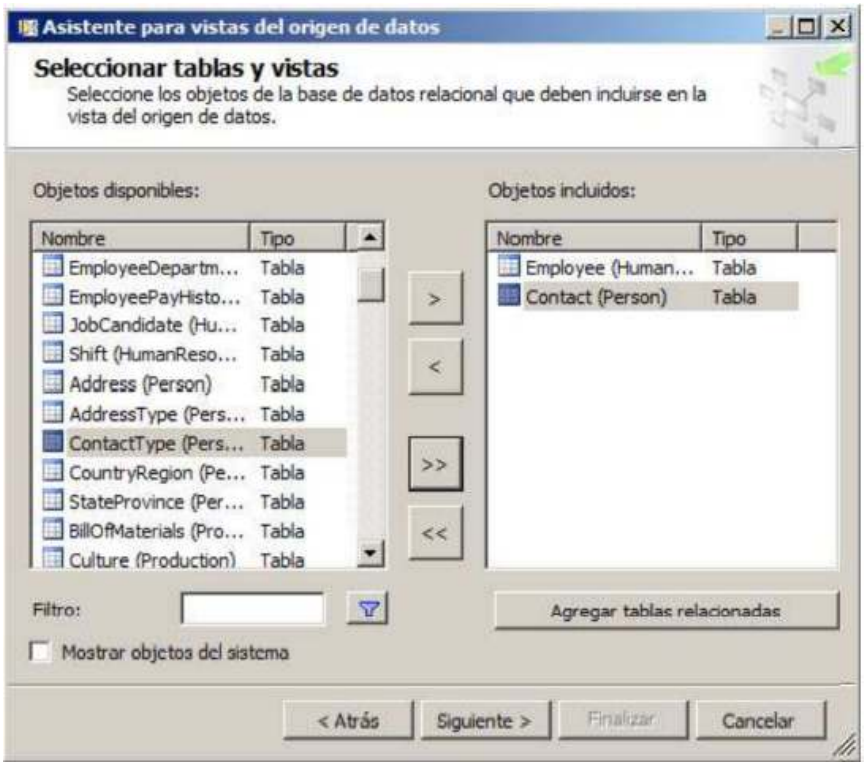

Figura 86. SSIS. Crear Vistas del Origen de datos I

Terminado el asistente, nos genera la vista y nos abre un editor desde el que podemos ver y modificar las tablas y sus relaciones. Muy parecido al diseñador de bases de datos de MS SQL Server para la versión estándar.

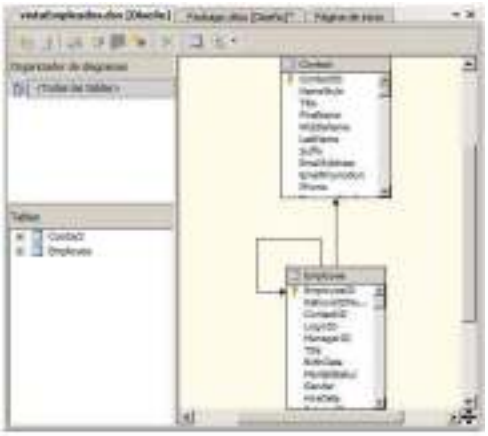

Figura 87. SSIS. Crear Vistas del Origen de datos II

Al igual que hemos comentado con los orígenes de datos, no es obligatorio usar las vistas de orígenes de datos pero nos proporcionan ventajas como por ejemplo poder reutilizarlas desde los distintos paquetes que formen parte del proyecto.

B. FLUJOS DE CONTROL

Como ya hemos visto en el apartado que define SSIS, un paquete consiste en un flujo de control y (opcionalmente) en uno o varios flujos de trabajo. Aquí será dónde se desarrolla el proceso de negocio de nuestro ETL, lo ideal es especificar tres divisiones de ejecución, una para la extracción de datos, otra para el juego de transformaciones y una última para la carga en la base de datos destino.

Existen tres tipos diferentes de elementos de flujo de control.

1) **Contenedores**: Permiten estructurar el contenido de los paquetes.
2) **Tareas**: Proporcionan la funcionalidad del paquete.
3) **Restricciones de precedencia**: Conectan los contenedores y las tareas, ordenando el flujo de control.

Vamos a ver un ejemplo para comprender el funcionamiento real de un paquete a través del flujo de control. El siguiente diagrama muestra un flujo de control con un contenedor y tres tareas. Dos de las tareas se definen en el nivel de paquete y una de ellas se define dentro del contenedor.

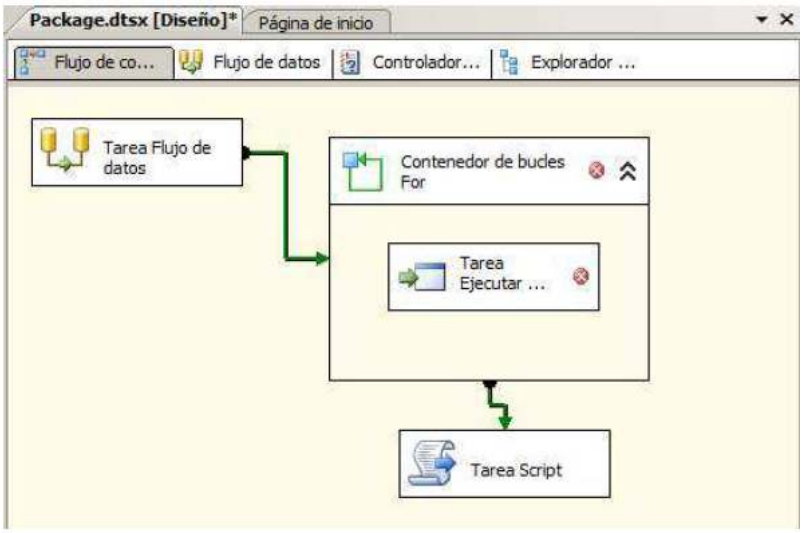

Figura 88. SSIS - Ejemplo de paquete.

Como se puede comprobar, aparece una estructura de cajas conectadas por flechas que indican las restricciones de precedencia.

La primera tarea es un flujo de datos, que por ejemplo puede recuperar datos a través de una consulta SQL y los ofrece a la siguiente tarea.

La segunda tarea nos indica que existen tareas dónde se pueden anidar otras y ejecutar de forma anidada, en éste caso tenemos un contenedor de bucles for, que ejecutará el número de veces correspondiente la tarea contenida en el contenedor. En este caso sería Ejecutar un paquete.

La tercera tarea nos indica que se ejecutaría un SCRIPT. La tarea Script proporciona código para realizar funciones que no están disponibles en las tareas integradas ni en las transformaciones proporcionadas por SQL Server Integration Services. Sirve para trabajos que se deben realizar una sola vez en un paquete (o una vez por objeto enumerado), en lugar de una vez por fila de datos.

A. Contenedores.

Los contenedores permiten estructurar el contenido de los paquetes y organizar el comportamiento de las tareas que hay en el flujo de control.

Lo situación habitual será contener cada división conceptual en un contenedor, aportando mayor importancia a la de transformación de datos. En condiciones normales tendremos un contenedor para alojar todas las transformaciones que necesitemos realizar para preparar los datos para su inserción en la base de datos OLAP.

SSIS incluye los siguientes tipos de contenedor para agrupar tareas e implementar flujos de control repetidos.

1) **El contenedor de bucles Foreach**: Enumera una colección y repite su flujo de control para cada miembro de la colección.
2) **El contenedor de bucles For**: Repite su flujo de control hasta que una expresión especificada adquiere el valor *False*.
3) **El contenedor de secuencias**: Sirve para agrupar un subconjunto del flujo de control dentro de un contenedor, permitiendo administrar las tareas que contiene como una unidad.

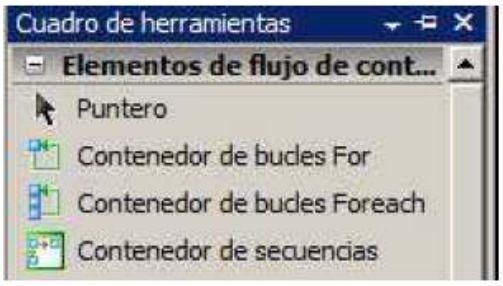

Figura 89. SSIS – Contenedores del flujo de control

B. Tareas

Para saber cómo implementar el flujo ETL debemos conocer qué tareas nos proporciona SSIS y cómo usarlas. Las tareas realizan el trabajo de los paquetes, hay una gran selección de tareas (tasks) accesibles desde el cuadro de herramientas de BIDS, se pueden arrastrar al flujo de control los siguientes **tipos de tareas**:

1) **Tarea Flujo de datos**

   Es la tarea que ejecuta flujos de datos para extraer datos, aplicar transformaciones de nivel de columna y cargar datos. Ya hemos hablado de ella y veremos en profundidad más adelante.

2) **Tareas de preparación de datos**

   Estas tareas llevan a cabo los procesos siguientes:

   - Copiar archivos y directorios.
   - Descargar archivos y datos.
   - Ejecutar servicios web.
   - Aplicar operaciones a documentos XML.
   - Generar perfiles de los datos para facilitar las tareas de limpieza.

3) **Tareas de flujo de trabajo**

   Tareas que se comunican con otros procesos. Sirven para:

   - Ejecutar paquetes.
   - Ejecutar programas o archivos por lotes.
   - Enviar y recibir mensajes entre paquetes.
   - Enviar mensajes de correo electrónico.
   - Leer datos de Instrumental de administración de Windows (WMI)
   - Detectar eventos de WMI.

4) **Tareas de SQL Server**

   Tareas de acceso, copia, inserción, eliminación y modificación de objetos y datos de SQL Server.

5) **Tareas de scripting**

   Tareas que amplían la funcionalidad de los paquetes mediante scripts.

6) **Tareas de Analysis Services**

   Tareas de creación, modificación, eliminación y procesamiento de objetos de Analysis Services.

7) **Tareas de mantenimiento**

   Tareas que realizan funciones administrativas como crear copias de seguridad y reducir bases de datos de SQL Server, volver a generar y reorganizar índices, y ejecutar trabajos del Agente SQL Server.

8) **Tareas personalizadas**

   Además, también se pueden crear tareas personalizadas mediante un lenguaje de programación compatible con COM, como Visual Basic, o un lenguaje de programación .NET, como C#. Se puede crear y registrar una interfaz de usuario para la tarea y así tener acceso a ella en BIDS.

En el cuadro de herramientas de BIDS tenemos 25 tareas distintas que se pueden usar en los flujos de control. Algunas de ellas las veremos más adelante pero podemos consultar la documentación oficial de Microsoft a través de la siguiente URL.

http://msdn.microsoft.com/es-es/library/ms139892(v=sql.105).aspx

C. Perfiles y calidad de datos.

Presentamos en éste apartado una de las tareas de SSIS, que se denomina ***Tarea de generación de perfiles de datos*** (Data profiler task). La razón por la que se dedicamos especial atención en comparación con el resto de tareas, es porque no suele ser una de las más explicadas en los diferentes tutoriales y porque forma parte esencial de los objetivos de ésta memoria.

La tarea de generación de perfiles de datos calcula diversos perfiles que le ayudan a familiarizarse con un origen de datos y a identificar en los datos problemas que deban corregirse. Puede utilizar la tarea de generación de perfiles de datos dentro de un paquete de SSIS para generar perfiles de datos que están almacenados en SQL Server e identificar posibles problemas de calidad de los datos.

Esta tarea proporciona un método sencillo y efectivo para analizar los datos contenidos en nuestras tablas y vistas, típicamente con el fin de determinar su calidad antes de construir una solución de ETL que los consuma.

Con ésta tarea se pueden lanzar diversos “perfiles” contra una base de datos de SQL Server 2000 o posterior, usando una conexión ADO .Net. Para ejecutar un paquete que contiene la tarea de generación de perfiles de datos, debe utilizar una cuenta que tenga permisos de lectura/escritura, incluidos los permisos CREATE TABLE, en la base de datos de tempdb.

El resultado de estos perfiles se graba en un archivo XML, que podemos procesar desde nuestras propias aplicaciones o visualizarlo mediante una herramienta específica denominada

Visor de perfil de datos (data profiler viewer), ubicada en *"…\100\DTS\Binn\DataProfilerViewer.exe"* por debajo del directorio de instalación de SQL Server.

Podemos consultar la documentación en línea de Microsoft a través de la siguiente URL.

http://msdn.microsoft.com/es-es/library/bb895263(v=sql.105)

Hay dos tipos de perfiles de datos.

1. Perfiles que analizan una sola columna.

    - **Distribución de longitudes de datos**: Notifica las diferentes longitudes de valores de cadena existentes en la columna seleccionada y el porcentaje de filas de la tabla que representa cada longitud.
    - Este perfil le ayuda a identificar problemas en los datos, como los valores no válidos. Por ejemplo, genera un perfil de una columna de códigos postales de España que deberían ser de cinco dígitos y detecta valores diferentes.

    - **Patrones de columna:**Notifica un conjunto de expresiones regulares que cubren el porcentaje de valores especificado en una columna de cadenas. Este perfil le ayuda a identificar problemas con los datos, como las cadenas no válidas. Este perfil también puede sugerir expresiones regulares que se pueden usar en el futuro para validar los valores nuevos. Por ejemplo, un perfil del patrón de una columna de códigos postales de Estados Unidos podría generar las expresiones regulares: \d{5}-\d{4}, \d{5} y \d{9}. Si aparecen otras expresiones regulares, es posible que los datos contengan valores no válidos o que tengan un formato incorrecto.

    - **Relación de NULLs en la columna:** Notifica el porcentaje de valores nulos en la columna seleccionada. Este perfil permite identificar problemas con los datos, como una proporción inesperadamente alta de valores nulos en una columna. Por ejemplo, si al generar un perfil de una columna de códigos postales se detecta un porcentaje excesivamente alto de códigos que faltan.

    - **Estadísticas de la columna:**Notifica estadísticas, como los valores mínimo, máximo, medio y la desviación estándar para las columnas numéricas, y los valores mínimo y máximo para las columnas *datetime*.

Este perfil le ayuda a identificar problemas existentes en los datos, como las fechas no válidas. Por ejemplo, si al generar un perfil de una columna de fechas históricas descubre una fecha máxima futura.

- **Distribución de los valores de la columna**: Notifica todos los valores distintos existentes en la columna seleccionada y el porcentaje de filas de la tabla que representa cada valor. También puede notificar los valores existentes en un número de filas de la tabla que supera cierto porcentaje. Este perfil le ayuda a identificar problemas con los datos, como un número incorrecto de valores distintos en una columna. Por ejemplo, si al generar un perfil de una columna que se supone que contiene los estados de Estados Unidos detecta más de 50 valores distintos.

2. Perfiles que analizan múltiples columnas en múltiples tablas.

- **Candidato para la clave:**Notifica si una columna o un conjunto de columnas es una clave, o una clave aproximada, para la tabla seleccionada. Este perfil le ayuda a identificar problemas con los datos, como por ejemplo, valores duplicados en una posible columna de clave.

- **Dependencia funcional:** Notifica hasta qué punto los valores de una columna (la columna dependiente) dependen de los valores de otra columna o de un conjunto de columnas (la columna determinante). Este perfil le ayuda a identificar problemas con los datos, como valores no válidos. Por ejemplo, al generar un perfil de la dependencia entre una columna que contiene códigos postales de España y una columna que contiene provincias de España. El mismo código postal debería tener siempre el mismo código de provincia, pero el perfil detecta incumplimientos de esta dependencia.

- **Inclusión de valor**: Calcula la superposición existente entre los valores de dos columnas o conjuntos de columnas. Este perfil puede determinar si una columna o un conjunto de columnas resulta adecuado para actuar como una clave externa entre las tablas seleccionadas. Este perfil le ayuda a identificar problemas con los datos, como valores no válidos. Por ejemplo, puede generar el perfil de una columna IdProducto de una tabla Ventas y detectar que dicha columna contiene valores que no se encuentran en la columna IdProducto de la tabla Productos.

Para **configurar la tarea de generación de perfiles de datos** se siguen los siguientes pasos:

- Se crea un administrador de conexión de ADO. Net que vaya a parar a la base de datos que queremos analizar.
- Se crea otro administrador de conexión para salvar los resultados en un archivo XML.
- Se arrastra la tarea de generación de perfiles de datos al diseñador de flujo de control en BIDS.
- Se editan las propiedades de la tarea, seleccionando los perfiles deseados.
  La tarea de generación de perfiles de datos tiene estas prácticas opciones de configuración:
  - **Columnas de carácter comodín**: mientras se configura una solicitud de generación de perfil, la tarea acepta el carácter comodín (*) en lugar de un nombre de columna. Esto simplifica la configuración y permite descubrir con facilidad las características de los datos poco familiares. Cuando se ejecuta la tarea, ésta genera perfiles para cada columna con un tipo de datos adecuado.
  - **Perfil rápido**: puede seleccionar un perfil rápido para configurar la tarea rápidamente. Un perfil rápido genera perfiles para una tabla o una vista mediante todos los perfiles y valores de configuración predeterminados.
- Se termina de configurar la tarea especificando los administradores de conexión que antes hemos configurado.
- Finalmente ya se puede ejecutar el paquete y examinar los resultados con el Visor de perfil de datos.

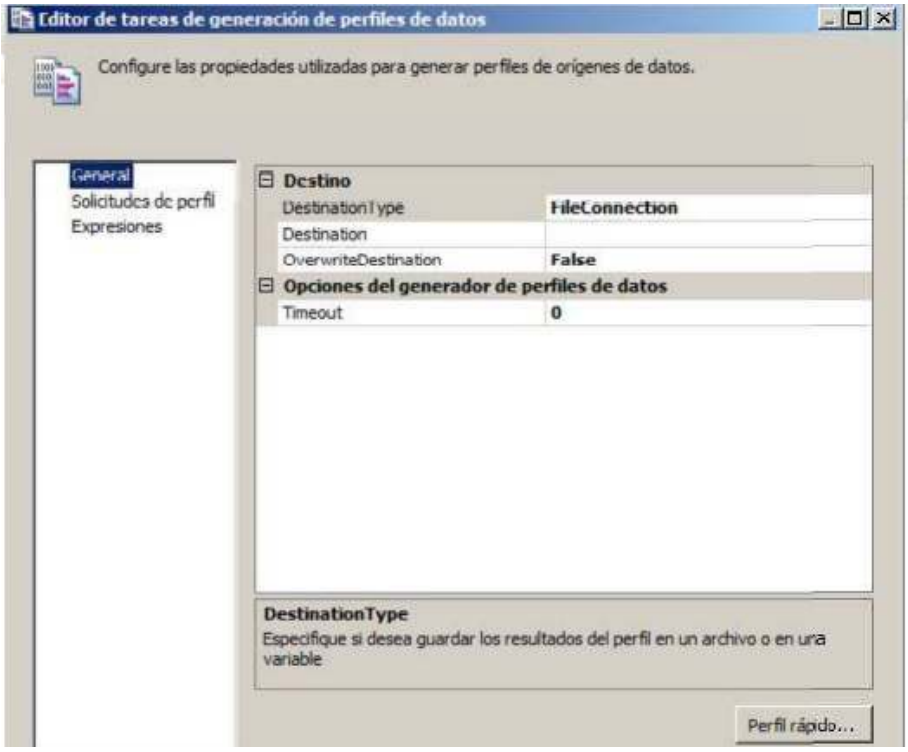

Figura 90. SSIS – Perfiles.

Se debe tener en cuenta que hay unos requisitos previos que se necesitan poder superar para que un perfil sea válido. Un perfil no es válido si no selecciona tablas y columnas que no están vacías, y las columnas contienen tipos de datos que son válidos para el perfil. Estos requisitos pueden ser de dos tipos:

3. Tipos de datos válidos.

Algunos de los perfiles disponibles solo tienen sentido para ciertos tipos de datos. Por ejemplo, no tiene sentido calcular un perfil de patrón de columnas para una columna que contiene valores numéricos o datetime. Por consiguiente, este tipo de perfil no es válido.

Perfil	Tipos de datos válidos*
ColumnStatisticsProfile	Columnas de tipo numérico o **datetime** (no se calcula **mean** ni **stddev** para columnas de tipo **datetime**)
ColumnNullRatioProfile	Todas las columnas**
ColumnValueDistributionProfile	Columnas de tipo **integer**, **char** y **datetime**
ColumnLengthDistributionProfile	Columnas de tipo **char**
ColumnPatternProfile	Columnas de tipo **char**
CandidateKeyProfile	Columnas de tipo **integer**, **char** y **datetime**
FunctionalDependencyProfile	Columnas de tipo **integer**, **char** y **datetime**
InclusionProfile	Columnas de tipo **integer**, **char** y **datetime**

Siendo:

* En la tabla anterior de tipos de datos válidos, los tipos **integer**, **char**, **datetime** y **numeric** incluyen los tipos de datos específicos siguientes:

- Los tipos enteros incluyen **bit**, **tinyint**, **smallint**, **int** y **bigint**.
- Los tipos de caracteres incluyen **char**, **nchar**, **varchar** y **nvarchar,** pero no incluyen **varchar(max)** ni **nvarchar(max)**.
- Los tipos de fecha y hora incluyen **datetime**, **smalldatetime** y **timestamp**.
- Los tipos numéricos incluyen los tipos **integer** (excepto **bit**), **money**, **smallmoney**, **decimal**, **float**, **real** y **numeric**.

** No se admiten los tipos **image**, **text**, **xml**, **udt** y **variant** para los perfiles distintos del perfilde proporción de columnas nulas.

4. **Tablas y columnas válidas.**

Si la tabla o la columna están vacías, la tarea de generación de perfiles de datos realiza las acciones siguientes:

- Cuando la tabla o la vista seleccionada esté vacía, la tarea de generación de perfiles de datos no calculará ningún perfil.
- Cuando todos los valores de la columna seleccionada sean NULL, la tarea de generación de perfiles de datos solo calculará el perfil de proporción de columnas nulas. La tarea no calculará el perfil de distribución de longitud de columnas, el perfil de patrón de columnas, el perfil de estadísticas de columnas ni el perfil de distribución de valores de columna.

4. IMPLEMENTAR LOS COMPONENTES DEL FLUJO DE DATOS

Ya hemos hablado con anterioridad de los flujos de datos e incluso hemos visto algún ejemplo. Es flujo de datos es una de las tareas que pueden incluirse en un flujo de control, e internamente se descompone en bloques de tres grandes tipos que pueden asociarse a cada sigla del ETL:

5. ***Orígenes***: Correspondería a la ***Extracción***.
6. ***Transformaciones***.
7. ***Destinos***: Correspondería a la ***Carga***.

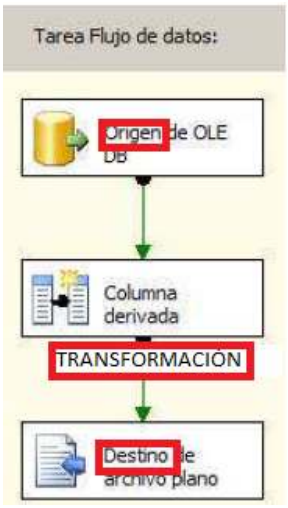

Figura 91. SSIS – Flujo de datos.

Ya hemos visto en apartados iniciales lo básico de los **orígenes de datos**, éstos se encargan de extraer datos desde la fuente de datos (p.ej. tablas o vistas de base de datos, arhivos planos o XML, hojas de Excel, etc.). Normalmente en soluciones empresariales combinadas con MS Biztalk Server lo ideal es recuperar de la base de datos de Stagging o zona dónde se alojen los XML de salida como origen de información, esta llegara a la zona de transformación directamente y se cargará tras el conjunto de transformaciones a la base de datos OLAP de destino.

Las **transformaciones** sirven para modificar, resumir y limpiar datos y se adecúa a la perfección con el proceso de transformación descubierto en el flujo ETL del proceso de desarrollo de un DW para una solución de BI. En éste apartado pondremos algo más de énfasis para ver que métodos aporta SSIS para realizar nuestras tareas de Data CleanSign y de transformación de los datos del Staging Area hacia las bases de datos OLAP.

Los **destinos** realizan la carga de datos en tablas o archivos, pueden cargar directamente los datos en el destino OLAP o generar estructuras limpias que se vayan a cargar con otro proceso fuera de éste flujo de datos (p.ej. a través de una tarea de Script etc.).

Además, parecido a las restricciones de precedencia del flujo de control, SSIS proporciona rutas que conectan la salida de un componente con la entrada de otro componente. Las rutas definen la secuencia de los componentes y se establecen en BIDS conectando flechas entre los distintos bloques que representan los orígenes, transformaciones y destinos.

A. ORÍGENES

Como hemos mencionado antes, en SSIS, los orígenes de datos sirven para extraer de las diferentes fuentes de datos ese conjunto de datos con el que iniciar el proceso ETL o con el que presentar datos al inicio de una tarea de flujo de datos. Dentro del diseñador de flujo de datos tenemos una serie de tareas en el cuadro de herramientas que podemos utilizar para extraer los datos:

Figura 92. SSIS - Flujo de datos - Orígenes de datos.

Como podemos ver se puede extraer directamente datos de los siguientes orígenes:

- **Origen de ADO. Net**: Para cualquier base de datos de la que exista un proveedor ADO. Net.
- **Origen de Archivo plano**: Para archivos .txt y similar.
- **Origen de Archivo sin formato**: Se deberá indicar como acceder al contenido configurando más propiedades que para un archivo plano.
- **Origen de Excel**: Para hojas de Excel.
- **Origen de OLE DB**: Para cualquier base de datos que permita acceso OLE DB.
- **Origen XML**: Optimizado para acceder a archivos XML.

Existen tareas en el flujo de control que permiten acceder a más tipos de origen de datos, que podemos usar por ejemplo para abrir un archivo de Access y guardar su salida en un archivo XML del que leer en el flujo de datos o en una tabla de SQL Server.

El origen del flujo de datos normalmente tiene una salida llamada **salida normal**. La salida normal contiene columnas de salida, que son columnas que el origen agrega al flujo de datos. La salida normal hace referencia a las columnas externas (las que existen en los datos que se leen). Los metadatos de las columnas externas incluyen información tal como el nombre, el tipo de datos o la longitud de la columna de origen.

Además, podemos establecer una salida llamada **salida de error**. Una salida de error para un origen contiene las mismas columnas que la salida normal y dos columnas adicionales que proporcionan información sobre los errores.

El modelo de objetos de SSIS no restringe la cantidad de salidas normales y las salidas de error que pueden tener los orígenes. Los orígenes incluidos con SSIS, con la excepción del componente **script**, tienen una salida normal y muchos de los orígenes tienen también una salida de error. Los orígenes personalizados se pueden codificar de forma que implementen carias salidas normales y salidas de error.

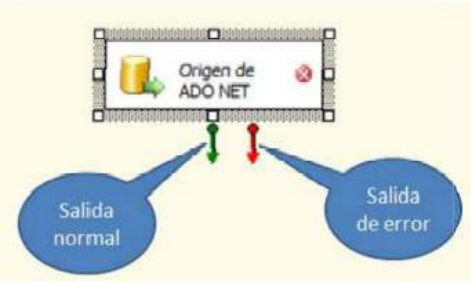

Figura 93. SSIS - Flujo de datos - Orígenes de datos – Salidas.

Todas las columnas de salida están disponibles como columnas de entrada para el siguiente componente del flujo de datos, que será una tarea de transformación de datos o directamente un destino.

B. TRANSFORMACIONES.

Las capacidades de las transformaciones de datos varían mucho de una transformación a otra. Las transformaciones pueden realizar tareas tales como actualizar, resumir, limpiar, mezclar o distribuir datos.

Se encuentran en el cuadro de herramientas en un apartado llamado tareas de flujo de datos.

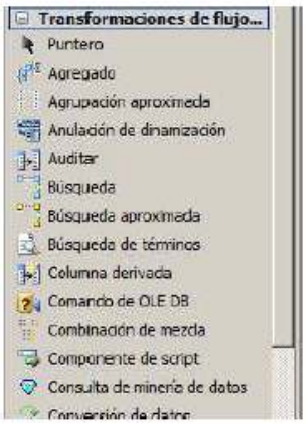

Figura 94. SSIS - Flujo de datos – Transformaciones.

Las **entradas y salidas de una transformación**definen las columnas de datos de entrada y salida.

Según la operación realizada con los datos, algunas transformaciones tienen una sola entrada y varias salidas, mientras que otras pueden tener varias entradas y una sola salida.

Las transformaciones, al igual que los orígenes de datos, también pueden contener **salidas de error**, que proporcionan información sobre el error que se ha producido junto con los datos en los que se ha producido un error.

El modelo de objetos de SSIS no restringe la **cantidad** de entradas, salidas normales y salidas de error que pueden contener las transformaciones. Se pueden crear transformaciones personalizadas que implementan cualquier combinación de múltiples entradas, salidas normales y salidas de error. Ésta será la acción más común en la creación de flujos ETL para el desarrollo de DW de un tamaño serio.

La **entrada** de una transformación se define como una o más columnas de entrada. Algunas transformaciones de SSIS también pueden hacer referencia a columnas externas como entrada. Por ejemplo, la entradas de la transformación comando de OLE DB incluye columnas externas.

Una columna de **salida**es una columna que la transformación agrega al flujo de datos. Tanto las salidas normales como las salidas de error contienen columnas de salida. Estas columnas de salida a su vez funcionan como columnas de entrada para el siguiente componente en el flujo de datos, ya sea otra transformación o un destino.

Las siguientes **transformaciones**son las que consideramos más importantes y tienen propiedades que se pueden actualizar a través de expresiones de propiedad

a. TRANSFORMACIÓN DIVISIÓN CONDICIONAL (CONDITIONAL SPLIT)

Permite dirigir filas de datos a salidasdiferentes en función del contenido de los datos. La implementación es **similar a una estructura de decisión CASE**de un lenguaje de programación. Evalúa expresiones y, en función de los resultados, dirige la fila de datos a la salida especificada.

También proporciona una salida predeterminada, de modo que si una fila no coincide con ninguna expresión, se dirige a la salida predeterminada.

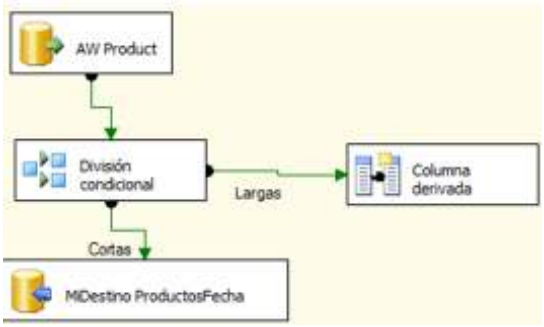

Figura 95. SSIS - Flujo de datos – Transformaciones - División condicional.

Se puede configurar de las maneras siguientes:

- Proporcionando una expresión cuya evaluación devuelva un valor booleano para cada condición que desee usar en la transformación.
- Especificando el orden de evaluación de las condiciones. El orden es importante, ya que una fila se envía a la salida correspondiente a la primera condición que dé como resultado True.
- Especificando la salida predeterminada para la transformación. La transformación requiere que se especifique una salida predeterminada.

Cada fila de entrada sólo se puede enviar a una salida, la correspondiente a la primera condición que resulte ser verdadera. Por ejemplo, las siguientes condiciones dirigen las filas de

la columna **Nombre** que empiecen por la letra **A** a una salida, las filas que empiecen por la letra **B** a una salida diferente y todas las demás filas a la salida predeterminada.

- Salida 1

  SUBSTRING(**Nombre**,1,1) == "**A**"

- Salida 2

  SUBSTRING(**Nombre**,1,1) == "**B**"

SSIS incluye ***funciones y operadores*** que se pueden utilizar para crear expresiones que evalúen los datos de entrada y dirijan los datos de salida. Los operadores pueden verse en la siguiente URL.

http://msdn.microsoft.com/es-es/library/ms141232(v=sql.100).aspx

La transformación incluye la propiedad personalizada **FriendlyExpression**. Esta propiedad se puede actualizar a través de una expresión de propiedad, al cargar el paquete. Una expresión de propiedad es una expresión asignada a una propiedad para permitir la actualización dinámica de la propiedad en tiempo de ejecución

Esta transformación tiene una entrada, una o más salidas, y una salida de error.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL.

http://msdn.microsoft.com/es-es/library/ms137886(v=sql.100).aspx

b. TRANSFORMACIÓN COLUMNA DERIVADA (DERIVED COLUMN)

Crea nuevos valores de columna aplicando expresiones a las columnas de entrada de la transformación.

Una expresión puede contener cualquier combinación de variables, funciones, operadores y columnas de la entrada de transformación.

El resultado puede agregarse como una nueva columna o insertarse en una columna existente como un valor de reemplazo. Se pueden definir varias columnas derivadas, y cualquier variable o columna de entrada puede aparecer en varias expresiones.

Se puede utilizar esta transformación para realizar las siguientes tareas:

- **Concatenar datos de distintas columnas en una columna derivada**. Por ejemplo, puede combinar valores de las columnas Nombre y Apellidos en una sola columna

derivada, denominada NombreCompleto, mediante la expresión Nombre + " " + Apellidos.

- **Extraer caracteres de datos de cadena mediante funciones como SUBSTRING y después almacenar el resultado en una columna derivada**. Por ejemplo, puede extraer de la columna Nombre la inicial del nombre de una persona mediante la expresión SUBSTRING(Nombre,1,1).
- **Aplicar funciones matemáticas a datos numéricos y almacenar el resultado en una columna derivada**. Por ejemplo, puede cambiar la longitud y la precisión de una columna numérica, IVA, a un número con dos cifras decimales mediante la expresión ROUND(IVA, 2).
- **Crear expresiones que comparen columnas de entrada y variables**. Por ejemplo, puede comparar la variable Version con los datos de la columna ProductoVersion y, en función del resultado de la comparación, usar el valor de Version o ProductoVersion mediante la expresión ProductoVersion == @Version? ProductVersion : @Version.
- **Extraer partes de un valor datetime**. Por ejemplo, puede utilizar las funciones GETDATE y DATEPART para extraer el año actual mediante la expresión DATEPART("year",GETDATE()).

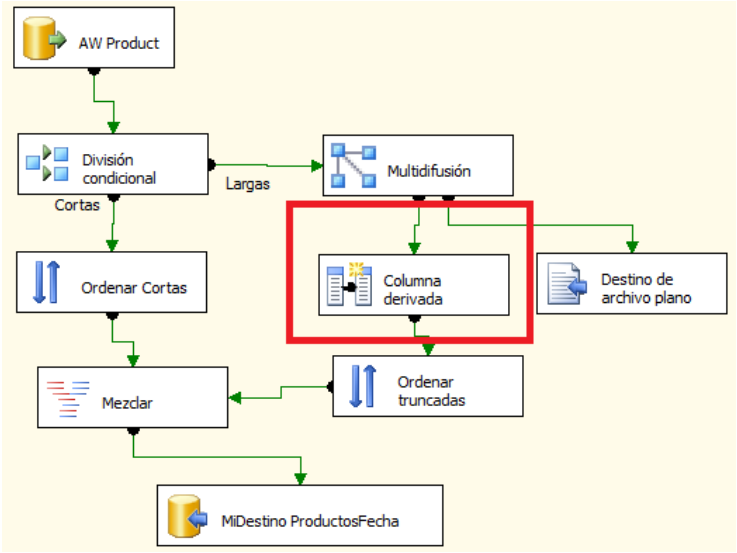

Figura 96. SSIS - Flujo de datos - Transformación columna derivada

Se puede configurar la transformación Columna derivada de las maneras siguientes:

- Proporcionando una expresión para cada columna de entrada o nueva columna que se vaya a modificar. Si una expresión hace referencia a una columna de entrada sobrescrita por la transformación Columna derivada, la expresión utiliza el valor original de la columna, no el valor derivado.

- Si agrega resultados a columnas nuevas y el tipo de datos es string, especifique una página de códigos. Para obtener más información, vea Comparar datos de cadena.

La transformación incluye la propiedad personalizada **FriendlyExpression**. Esta propiedad se puede actualizar a través de una expresión de propiedad, al cargar el paquete

Esta transformación tiene una entrada, una salida normal y una salida de error.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms141069(v=sql.100).aspx

c. TRANSFORMACIÓN AGRUPACIÓN APROXIMADA (FUZZY GROUPING)

Realiza tareas de **limpieza de datos**, identificando filas de datos que probablemente se van a duplicar y seleccionando una fila de datos canónica para utilizarla en la normalización de los datos.

Requiere una conexión a una instancia de SQL Server para crear las tablas temporales de SQL Server que necesita el algoritmo de la transformación para realizar su trabajo. La conexión debe establecerla un usuario que tenga permiso para crear tablas en la base de datos.

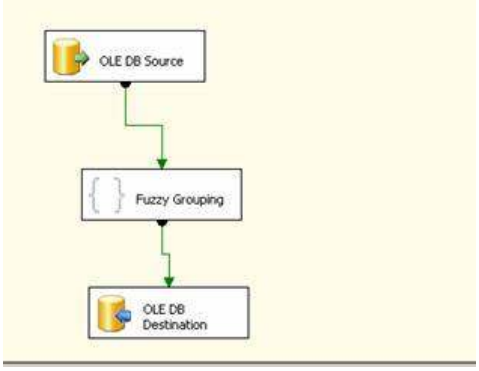

Figura 97. SSIS - Transformaciones - Agrupación aproximada.

Para configurar la transformación, debe seleccionar las columnas de entrada que desee utilizar para identificar duplicados y el tipo de coincidencia, aproximada o exacta, para cada columna.

Una **coincidencia exacta** garantiza que sólo se agruparán las filas de la columna que tengan valores idénticos.

La **salida** de la transformación incluye todas las columnas de entrada, una o más columnas con datos normalizados, y una columna que contiene la puntuación de similitud.

La puntuación es un valor decimal entre 0 y 1. La fila canónica tiene una puntuación de 1. Las otras filas de la agrupación aproximada tienen puntuaciones que indican su nivel de coincidencia con la fila canónica.

Cuanto más se acerque el resultado a 1, mayor será la coincidencia entre la fila y la fila canónica.

Si la agrupación aproximada contiene filas que son duplicados exactos de la fila canónica, dichas filas también tienen una puntuación de 1.

La transformación no quita las filas duplicadas. Se agrupan creando una clave que relaciona la fila canónica con las filas similares.

La transformación produce una fila de salida por cada fila de entrada, con las siguientes columnas adicionales:

- **_key_in**, una columna que identifica de forma única cada fila.
- **_key_out**,una columna que identifica un grupo de filas duplicadas. La columna _key_out tiene el valor de la columna _key_in en la fila de datos canónica. Las filas con el mismo valor en _key_out forman parte del mismo grupo. El valor _key_outpara un grupo corresponde al valor de _key_in en la fila de datos canónica.
- **_score**,un valor entre 0 y 1 que indica la similitud entre la fila de entrada y la fila canónica.

Éstos son los nombres de columna predeterminados y se puede configurar la transformación para que utilice otros. La salida también proporciona una puntuación de similitud para cada columna que participa en una agrupación aproximada.

La transformación Agrupación aproximada incluye dos **características para personalizar la agrupación**que realiza:

1. **Delimitadores de token**: proporciona un conjunto predeterminado de delimitadores que se utilizan para convertir los datos en tokens, pero puede agregar delimitadores nuevos que mejoren la conversión en tokens de los datos.

Umbral de similitud: indica cuanto de estricta es la transformación a la hora de identificar duplicados. Los umbrales de similitud se pueden establecer en el nivel de componente y de columna.

Se puede personalizar la agrupación que lleva a cabo la transformación, estableciendo las propiedades de las columnas en la entrada de la transformación. Por ejemplo, la propiedad **FuzzyComparisonFlags** especifica cómo compara la transformación los datos de cadena en una columna y la propiedad **ExactFuzzy**especifica si la transformación realiza una coincidencia aproximada o exacta.

La cantidad de memoria que utiliza la transformación se puede configurar al establecer la propiedad personalizada **MaxMemoryUsage**.

Esta transformación tiene una entrada y una salida. No admite una salida de error.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms141764(v=sql.100).aspx

d. TRANSFORMACIÓN BÚSQUEDA APROXIMADA (FUZZY LOOKUP)

Realiza tareas de **limpieza de datos** como normalizar datos, corregir datos y proporcionar valores que faltan.

Difiere de la transformación búsqueda simple en su uso de coincidencia aproximada. Utiliza una **combinación de igualdad** para localizar los registros que coinciden en la tabla de referencia. Devuelve una coincidencia exacta o ninguna coincidencia de la tabla de referencia. En cambio, la transformación búsqueda aproximada utiliza la coincidencia aproximada para devolver una o más coincidencias similares de la tabla de referencia.

La transformación búsqueda aproximada suele ir después de una transformación búsqueda en un flujo de datos de paquete. Primero, la transformación Búsqueda intenta encontrar una coincidencia exacta. Si no lo consigue, proporciona coincidencias similares de la tabla de referencia.

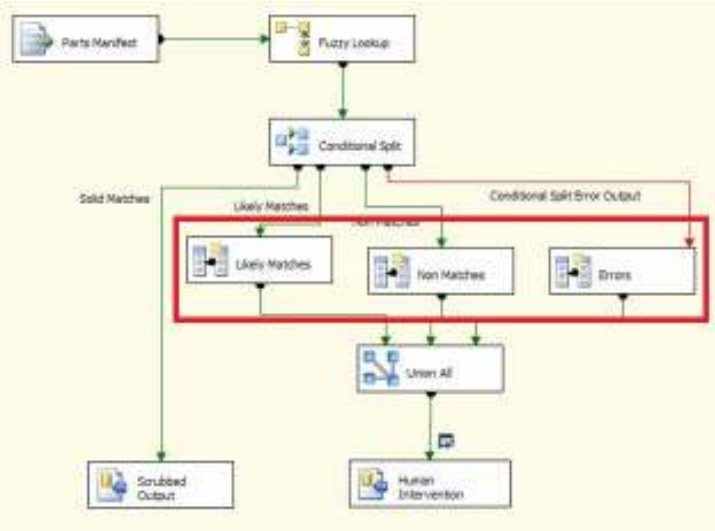

Figura 98. SSIS - Transformaciones - Búsqueda aproximada.

Necesita acceso a un origen de datos de referencia que contenga los valores que se utilizan para limpiar y ampliar los datos de entrada.

El origen de datos de referencia debe ser una tabla de una base de datos de SQL Server 2000 o posterior.

La coincidencia entre el valor en una columna de entrada y el valor en la tabla de referencia puede ser exacta o aproximada. Sin embargo, la transformación requiere que se configure al menos una coincidencia de columna para la coincidencia aproximada. Para buscar únicamente coincidencias exactas, hay que usar la transformación búsqueda simple.

Esta transformación tiene una entrada y una salida.

Se puede personalizar la transformación especificando la cantidad máxima de memoria, el algoritmo de comparación de filas y el almacenamiento en caché de índices y tablas de referencia que utiliza la transformación

Incluye tres características para personalizar la búsqueda realiza:

1. El número máximo de coincidencias que se van a devolver por fila de entrada.
2. Delimitadores de token.
3. Umbrales de similitud.

Especificar un número máximo de coincidencias no garantiza que la transformación devuelva ese número de coincidencias; solo garantiza que la transformación devolverá, como máximo, ese número de coincidencias.

Las columnas de salida de transformación incluyen las columnas de entrada marcadas como **columnas de paso**, las columnas seleccionadas en la tabla de búsqueda y las siguientes columnas adicionales:

- **_Similarity**: una columna que describe la similitud entre los valores de las columnas de entrada y de referencia.
- **_Confidence**: una columna que describe la calidad de la coincidencia.

La transformación utiliza la conexión a la base de datos de SQL Server para crear las tablas temporales que utiliza el algoritmo de coincidencia aproximada.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms137786(v=sql.100).aspx

e. Transformación Comando de OLE DB (Ole Db Command)

Ejecuta una instrucción SQL para cada fila en un flujo de datos. Por ejemplo, puede ejecutar una instrucción SQL que inserte, actualice o elimine filas en una tabla de base de datos.

Se puede configurar de las maneras siguientes:

- Proporcionar la instrucción SQL que la transformación ejecuta para cada fila.
- Especifica la cantidad de segundos que tienen que transcurrir antes de que la instrucción SQL agote el tiempo de espera.
- Especificar la página de códigos predeterminada.

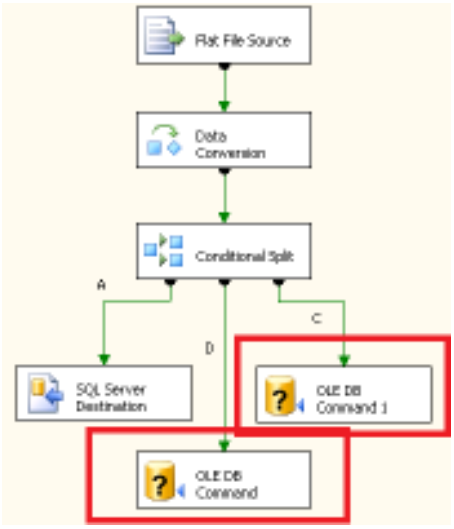

Figura 99. SSIS - Flujo de datos - Transformación comando OLE DB.

Normalmente, la instrucción SQL incluye **parámetros**. Los valores de parámetro se almacenan en columnas externas en la entrada de transformación y al asignar una columna de entrada a una columna externa se asigna una columna de entrada a un parámetro.

Por ejemplo, para buscar filas en la tabla DimProduct según el valor en su columna ProductKey y luego eliminarlas, puede asignar la columna externa denominada Param_0 a la columna de entrada denominada ProductKey y, a continuación, ejecutar la instrucción SQL DELETE FROM DimProduct WHERE ProductKey = ?.

La transformación proporciona los nombres de parámetro y no puede modificarlos. Los nombres de parámetro son Param_0, Param_1 y así sucesivamente.

La transformación incluye la propiedad personalizada **SQLCommand**. Esta propiedad se puede actualizar a través de una expresión de propiedad, al cargar el paquete.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms141138(v=sql.100).aspx

f. Transformación Muestreo de porcentaje (Percentage Sampling)

Crea un conjunto de datos de muestra seleccionando un porcentaje de las filas de entrada de la transformación. El conjunto de datos de muestra es una selección aleatoria de filas de la entrada de transformación, de forma que la muestra resultante sea representativa de la entrada.

Es especialmente útil para la **minería de datos**. Se puede dividir de forma aleatoria un conjunto de datos en dos conjuntos de datos: uno para el entrenamiento del modelo de minería de datos y otro para probar el modelo.

También es útil para crear conjuntos de **datos de ejemplo de desarrollo de paquetes**. Si aplica la transformación a un flujo de datos, puede reducir uniformemente el tamaño de los datos conservando sus características. El paquete de prueba podrá ejecutarse más rápido porque utilizará un conjunto de datos pequeño, pero representativo.

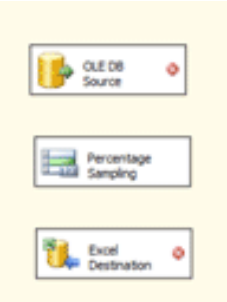

Figura 100. SSIS - Flujo de datos - Transformación muestreo de porcentaje

Se puede especificar un valor de inicialización de muestreo para modificar el comportamiento del generador de números aleatorios utilizado por la transformación para seleccionar filas.

Esta transformación es similar a la **transformaciónmuestreo de fila**, que crea a conjunto de datos de ejemplo seleccionando un número especificado de filas de entrada.

Incluye la propiedad personalizada **SamplingValue** que se puede actualizar a través de una expresión de propiedad al cargar el paquete.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms139864(v=sql.100).aspx

g. TRANSFORMACIÓN ORDENAR (SORT)

Ordena los datos de entrada en orden ascendente o descendente, y copia los datos ordenados a la salida de la transformación.

Puede aplicar varias ordenaciones a una entrada dónde cada ordenación se identifica mediante un número que determina el criterio de ordenación.

La columna con el número más bajo se ordenará primero, la columna con el segundo número más bajo se ordena a continuación, etc.

Un número positivo indica que la ordenación es ascendente y un número negativo indica que la ordenación es descendente.

Las columnas que no se están ordenadas tienen un criterio de ordenación de 0. Iincluye un conjunto de opciones de comparación para definir cómo controlará la transformación los datos de cadena de una columna. Se pueden ver a través de la siguiente URL.

http://msdn.microsoft.com/es-es/library/ms141038(v=sql.100).aspx

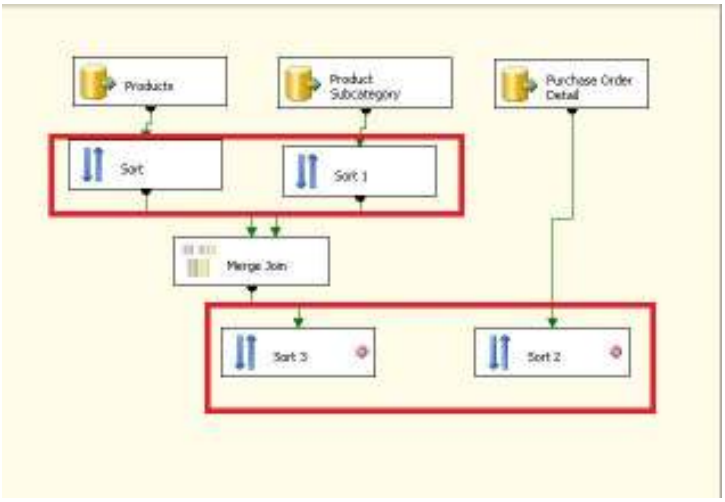

Figura 101. SSIS - Transformaciones – Ordenar.

La transformación también puede **quitar filas duplicadas** como parte de la ordenación. Las filas duplicadas son filas con los mismos criterios de ordenación. El valor del criterio de ordenación se genera a partir de las opciones de comparación de cadenas usadas, lo que implica que cadenas literales diferentes pueden tener los mismos criterios de ordenación. La transformación identifica filas en las columnas de entrada que tienen valores distintos pero un mismo criterio de ordenación que los duplicados.

El grueso de opciones de configuración puede consultarse en la documentación en línea de Microsoft a través de la siguiente URL

http://msdn.microsoft.com/es-es/library/ms140182(v=sql.100).aspx

h. OTRAS TRANSFORMACIONES

- **Aggregate**: aplica funciones de agregado (sum, min, max, avg, etc.) a valores de columnas y copia los resultados a la salida de la transformación. Para ello dispone de una cláusula group by que permite decidir los grupos que componen el agregado.
- **Audit**: permite agregar al flujo de datos columnas con información sobre el entorno de ejecución del paquete, como el usuario, el nombre del paquete, el equipo que lo ejecuta, la fecha y hora de ejecución, etc.
- **Caché Transform**: escribe los datos que vienen en el flujo de datos en un administrador de conexión de caché, permitiendo almacenar estos datos en un archivo .raw, que pueden ser utilizados por otros componentes como Lookup.
- **Character Map**: aplica funciones de cadenas sobre los datos de tipo carácter.
- **Copy Column**: crea columnas nuevas que son una copia de las columnas indicadas.
- **Data Conversion**: convierte de un tipo de datos a otro creando una nueva columna con el resultado.

- **Data Mining Query**: realiza consultas de predicción en modelos de minería de datos, utilizando para ello el lenguaje DMX (Data Mining eXpressions).
- **Export Column**: permiten leer datos del flujo de datos e insertar cada uno de ellos a un archivo. Por ejemplo, puedo extraer las imágenes de los artículos de la base de datos y guardarlas todas en una carpeta.
- **Lookup**: busca valores en una tabla de refencia con una coincidencia exacta.
- **Merge**: permite combinar conjuntos de datos ordenados, obteniendo como resultado el conjunto de filas de las diferentes entradas.
- **Merge Join**: combina conjuntos de datos ordenados, obteniendo por cada uno de ellos una fila con el conjunto de columnas de ambas entradas. Permite realizar operaciones que en SQL hacemos con las cláusulas INNER JOIN, LEFT/RIGHT OUTER JOIN y FULL OUTER JOIN.
- **Multicast**: distribuye una copia del conjunto de datos que recibe por cada una de las salidas que creemos.
- **Pivot y Unpivot**: permite pivotar datos entre filas y columnas. Por ejemplo, si tenemos las columnas país, año e importe, podemos hacer que mediante la transformación Pivot nos devuelva una matriz de dos dimensiones, con una fila por país, una columna por año, y en el cruce de éstas se mostrará el importe. La transformación Unpivot hace el proceso inverso.
- **Row Count**: cuenta las filas que pasan a través de ella y almacena el resultado en una variable.
- **Row Sampling**: igual a Percentage Sampling, pero en vez de devolver un porcentaje de filas sobre el total, devuelve el número de filas indicado sobre el total.
- **Script Component**: permite extraer, transformar o cargar datos mediante código personalizado escrito en VB.NET o C#. Es muy útil cuando tenemos que realizar, por ejemplo, un cálculo que no nos lo permite hacer ninguna de las transformaciones existentes.
- **Slowly Changing Dimension**: coordina la inserción y modificación en una tabla de dimensiones, aplicando los diversos tipos de cambios descritos anteriormente en el apartado dedicado a Slowly Changing Dimensions (SCD), así como la gestión de miembros inferidos (inferred members), también vista anteriormente. Contiene un asistente que nos guía paso a paso en la implementación de esta casuística.
- **Term Extraction**: permite extraer términos de un texto. Funciona sólo con textos en inglés, ya que sólo tiene un diccionario lingüístico para este idioma.
- **Term Lookup**: busca términos en una tabla de referencia y cuenta los términos extraídos de dicho texto. Esta transformación resulta útil para crear una lista personalizada de palabras basada en el texto de entrada, que incluye estadísticas de frecuencia de aparición de palabras.

- **Union All**: combina varios conjuntos de datos de entrada en uno sólo. Es la función opuesta a Multicast.

C. DESTINOS.

Un destino es aquél componente del flujo de datos que escribe el propio flujo a un almacén de datos específico o crea un conjunto de datos almacenado en la memoria para que sea otro componente más específico el encargado de cargar los datos en el destino.

Podemos ver que los diferentes destinos aparecen en el cuadro de herramientas separados en un conjunto:

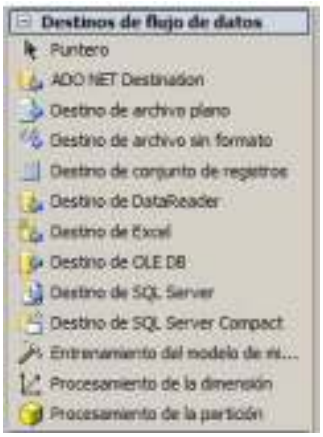

Figura 102. SSIS - Flujo de datos – Destinos.

Un destino de SSIS debe tener al menos una entrada asociada. La entrada contiene columnas de entrada, que proceden de otro componente de datos. Las columnas de entrada se asignan a columnas en el destino. Se puede hacer desde un mapa de columnas origen a columnas destino como se puede ver en la siguiente figura:

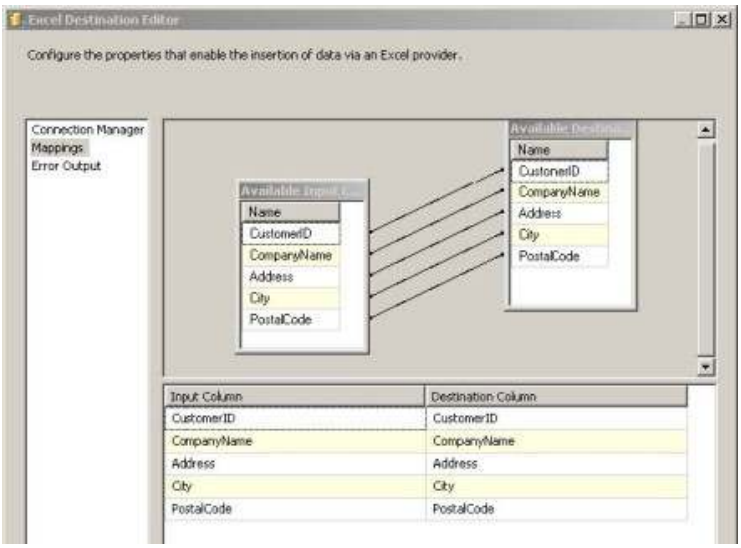

Figura 103. SSIS – Mapa de destino.

Muchos destinos también se suelen configurar con **salida de error**. La salida de error de un destino contiene columnas de salida, que normalmente contienen información de los errores que se producen mientras se escriben datos en el almacén de datos de destino dado que los errores pueden producirse por múltiples motivos. Por ejemplo, se produce un error si una columna trae un valor NULL y se intenta grabar en una tabla que no admite NULLS en esa columna o si se recupera un date cadena y se intenta grabar en una columna datetime.

El modelo de objetos de SSIS no limita la cantidad de entradas regulares y salidas de error que los destinos pueden tener. Además se pueden crear destinos personalizados que implementan varias entradas y salidas de error, que será lo más normal en soluciones ETL de un tamaño serio.

Conviene tener en cuenta que no siempre el destino se encargará de la carga de datos en la base de datos OLAP, será muy común que presente los datos en tablas de SQL Server y se utilicen tareas de SCRIPT para realizar esa carga si no usamos base de datos OLAP de MS SQL Server.

PRESUPUESTO

Universidad Carlos III de Madrid

PRESUPUESTO DE PROYECTO

1.- Autor:	Pablo Martín Gutiérrez
2.- Departamento:	Departamento de Informática
3.- Descripción del Proyecto:	Establecimiento de un Marco de Calidad para sistemas Data
Titulo	Data Warehouse: Marco de Calidad.
Duración (meses)	15
Tasa de costes Indirectos:	20%

4.- Presupuesto total del Proyecto (valores en Euros):

48853,2825 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	Categoría	Dedicación (hombres mes) [a]	Coste hombre mes	Coste (Euro)	Firma de conformidad
Martín Gutiérrez, Pablo	Ingeniero	1	2.694,39	2.694,39	
Personas mes		1	Total	2.694,39	

[a] 1 Persona mes = 131,25 horas. Máximo anual de dedicación de 12 personas mes (1575 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable [d]
Ordenador personal	700,00	100,00	15,00	60,00	175,00
Memoria Flash USB.	24,00	100,00	15,00	60,00	6,00
Disco duro externo.	90,00	100,00	15,00	60,00	22,50
Licencia Microsoft Office 2007 Professional	0,00	100,00	15,00	60,00	0,00
Licencia SW Laboratorios	0,00	100,00	15,00	60,00	0,00
[12] W. H. Inmon: "Building the Data Warehouse.". (John Wiley & Sons Inc., 1992, 1ª Ed.).	0,00	75,00	15,00	60,00	0,00
[14] R. Kimball: "The Data Warehouse Lifecycle Toolkit". (John Wiley & Sons Inc., 1998, 1ª Ed.).	0,00	75,00	15,00	60,00	0,00
[16] R. Kimball: "The Data Warehouse ETL Toolkit (2nd Edition)". (John Wiley & Sons Inc., 2008, 2ª Ed.)	0,00	75,00	15,00	60,00	0,00
[21] ISO, I.O.o.S., ISO/IEC FDIS 25012 Software engineering - Software Product - Quality Requirements and Evaluation (SQuaRE) - Data quality model. 2008.	66,60	75,00	15,00	60,00	12,49
[80] ISO, I.O.o.S., ISO/IEC 90003:2004 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software, 2004.	134,86	75,00	15,00	60,00	25,29
[81] ISO, I.O.o.S., ISO/IEC 25000:2005 Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 2005.	121,54	75,00	15,00	60,00	22,79

[82] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition.	39,50	75,00	15,00	60,00	7,41
Papelería	100,00	50,00	15,00	60,00	12,50
Infraestructura ADSL	45,00	100,00	15,00	60,00	11,25
				Total	295,21875

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
--	--	--
	Total	0

OTROS COSTES DIRECTOS DEL PROYECTO[e)]

Descripción	Empresa	Costes imputable
--	--	--
	Total	0

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales (Euros)
Personal	40.415,85
Amortización	295,22
Subcontratación de tareas	0,00
Costes de funcionamiento	0,00
Costes Indirectos	8.142,21
Total	48853,28

CONSLUSIONES

El propósito principal de la memoria residía ***en crear "una guía de actuación basada en un marco de calidad que nos permita crear sistemas Data Warehouse con éxito"***,por ello hemos centrado los dos primeros bloques de la memoria en presentar el conocimiento más adecuado para enfocar la propia guía en el bloque III de la memoria:

"Guidelinespara el desarrollo de un Data Warehouse de Calidad en un Sistema BI. Calidad en el Dato, Calidad en el Proceso".

Con la definición de las *guidelines* además hemos visto cómo afrontar basándonos en un marco de calidad las dos corrientes que nos preocupaban al principio de la memoria que no solían verse **expuestas en un mismo documento**, como son las **calidad en el dato o la calidad en el proceso**.Se ha creado el documento para centrarnos en los procesos que incluye todo el ciclo de vida de un proyecto que incluye un DW y hacemos especial hincapié en cómo medir y evaluar la calidad del dato a través de la aportación de las características que un dato de calidad es susceptible de tener.

No ha sido fácil hacer el mapeo de que procesospueden ser más representativos o indispensables para adaptar a la memoria, sobre todo**por la diversidad de filosofías y la limitada documentación adaptada**, por ello en base a la lectura de muchos ensayos de calidad en los DW y revisión de normas, estándares de calidad y guías de buenas prácticas hemos logrado extraer una serie de puntos en común que han conformado la estructura final de las guidelines explicando en cada caso de dónde hemos extraído el mejor mapeo de procesos a un marco de calidad o en el caso de los datos, haciendo directamente el estudio sobre el estado del arte.

Las *guidelines* constituyen **la primera piedra**para poder atacar con grandes posibilidades de éxito el desarrollo de un sistema de BI que incluya un DW o varios DM ya que proponemos como metodología de desarrollo la filosofía de Kimball que se centra en primera instancia en los DM para llegar a formar el DW corporativo. Por ello damos respuesta a las preguntas (y otras muchas más) que surgían en los objetivos, como son:

- ¿Quién tiene que tomar las principales decisiones sobre el DW?
- ¿Cómo gestionar el proyecto del DW?
- ¿Cómo saber si el DW es de calidad? Nos referimos en ésta caso a Calidad en el dato y saber cómo evaluarla.
- Saber cómo abarcar todos los procesos que forman parte del DW en base a unas pautas o directrices que nos garanticen la calidad de los procesos.
- Etc.

Otro aspecto que no se ha incluido en las diretrices, pero que ha resultado de gran interés son los **factores críticos de éxito**que hemos ido identificando a lo largo del proceso de desarrollo del documento. Éste apartado nos servirá como guía extra de apoyo para afrontar con garantías nuestros objetivos y a pesar de ser un documento alejado de marcos de calidad resulta de gran utilidad.

Por último queríamos **acercar las *guidelines* al mundo real**, a situaciones en las que tenemos que afrontar desarrollos de DW con herramientas específicas o no podemos usar el SW que prefiramos. Por ello se ha centrado el apartado práctico en crear dos guías de uso, con diferente nivel de detalle dado el perfil al que va destinado, de herramientas muy útiles para nuestro cometido de la familia de Microsoft.

Además se centra la solución el componente más crítico a la hora de conformar el DW, el proceso ETL, **usando MS Biztalk Server para eliminar problemas de interoperabilidad**y dejar las tareas en manos de administradores de sistemas y el módulo **SSIS de SQL Server 2008**para que el desarrollador afronte con grandes facilidades las tareas del **ETL**.

Este proceso **me ha servido para saber cómo explotar las virtudes de ambas herramientas,** ya que antes a pesar de conocerlas, no veía muy bien cuándo utilizar una en detrimento de la otra o como acoplar correctamente ambas en una arquitectura definida. Conocer como funciona internamente el motor de flujo de datos de SSIS permite a los desarrolladores grandes facilidades a la hora de implementar las soluciones.

TRABAJO FUTURO

La primera extensión de futuro sobre la memoria, sería ***la aplicación de las guidelines en un proyecto real*** (incluido otro PFC) dónde sirvan las guidelines como marco base para el desarrollo del proyecto.

Dado su estructura, propone la definición de una serie de documentos para la identificación del proyecto, gestión de los proceos y una metodología de desarrollo muy claro que permite guiar al equipo encargado de la realización de proyecto de cómo hacer las tareas.

Otra extensión del trabajo, que se ha creído fuera del ámbito de la memoria, es ***identificar dentro del apartado de medición y evaluación de la calidad de las guidelines unos una serie de fichas base***dónde se aporte qué propiedades de calidad se mapearían mejor con qué datos y qué métricas usar en cada caso basándonos en nuestras propiedades de calidad del dato y las métricas de la serie ISO 2504n que indicamos.

Con éste trabajo, se puede ***crear una pequeña aplicación***que en base a la catalogación de según los parámetros que indiquemos nos muestre unas fichas u otras para medir y evaluar la calidad.

Por último, me gustaría que la aplicación de las posibilidades que ofrecen las guidelines se estudie en base a dos corrientes:

1) ***Licencias Open Source***. Podría ser muy buena práctica encontrar soluciones Open Source que emulen las capacidades de las herramientas que hemos visto para acometer el proceso ETL.
2) ***Licencias de desarrollo de Windows***. Gracias a la nueva política de licencias de Windows tenemos versiones gratuitas de la mayoría de sus herramientas de desarrollo, por lo que podrías ser recomendable, extender el proceso de implementación del desarrollo ETL con el uso de sus versiones gratuitas o de evaluación creando soluciones que fácilmente pudieran ser del mundo real.

GLOSARIO

BI	*Business Intelligence o Inteligencia de Negocio.*
DW	*Data Warehouse o Almacén de Datos Central.*
DM	*Data Mart.*
BBDD	*Base de Datos.*
OLTP	*On-Line Transaction Processing o Procesamiento en línea de transacciones.*
OLAP	*On-Line Analytical Processing o Procesamiento analítico de transacciones.*
ISO	*International Organization for Standardization u Organización Internacional de Estandarización.*
IEC	*International Electrotechnical Commission o Comisión Internacional Electrotécnica.*
IEEE	*Institute of Electrical and Electronics Engineers o Instituto de Ingenieros Electricos y Electrónicos.*
PMBOK	*Project Management Body of Knowledge.*
COBIT	*Control Objectives for Information and related Technology.*
ITIL	*Information Technology Infrastructure Library.*
CMMI	*Capability Maturity Model integration*
ETL	*Extract, Transform and Load o Extracción, transformación y carga.*
DSS	*Decision Support System o Sistema de Soporte a Decisiones*
EIS	*Executive Information System o Sistema de Información Ejecutiva.*
SQUARE	*Software product Quality Requirements and Evaluation: ISO/IEC 25000.*
SPICE	*Software Process Improvement Capability Determination: ISO 15504.*
DWQ	*Data Warehouse Quality Project.*

SI	*Sistema de Información.*
FTP	*File Transfer Protocol.*
SOAP	*Simple Object Access Protocol*
SQL	*Structured Query Language*
SSIS	*SQL Server Integration Services*
MS	*Microsoft*
WCF	*Windows Communication Foundation*
POP3	*Post Office Protocol*
SMTP	*Simple Mail Transfer Protocol*
IMAP4	*Internet Message Access Protocol*
SAP	*SW de Gestión de la Compañía SAP AG.*
ORACLE	*SW de Gestión de Bases de datos de la compañía ORACLE.*
BIDS	*Business Intelligence Development Studio*
VS	*Microsoft Visual Studio 2010*
.NET	*Microsoft .Net Framework*
HTTP	*Hypertext Transfer Protocol.*
HTTPS	*Hypertext Transfer Protocol que usa el protocolo SSL (Secure Sockets Layer).*
MQSeries	*Message Queue Series de la compañía IBM*

REFERENCIAS

[1] http://www.sinnexus.com/business_intelligence/index.aspx. Accedido en Enero 2011.

[2] http://www.ibermatica.com/ibermatica/businessintelligence2. Accedido en Enero 2011.

[3] http://es.wikipedia.org/wiki/Inteligencia_empresarial. Accedido en Enero 2011.

[4] Atos Consulting, Business Intelligence. Madrid, Noviembre de 2010.

[5] Ibermática, Business Intelligence – El conocimiento compartido, Septiembre de 2007.

[6] http://en.wikipedia.org/wiki/Data_model#Database_model. Accedido en Enero de 2011.

[7] http://es.wikipedia.org/wiki/Base_de_datos. Accedido en Enero de 2011.

[8] http://en.wikipedia.org/wiki/Relational_model. Accedido en Enero de 2011.

[9] Atos Consulting, Seminario Data Warehouse. Madrid, enero de 2009.

[10] http://www.sinnexus.com/business_intelligence/Data Warehouse.aspx. Accedido en Febrero de 2011.

[11] Data Warehouse (Almacenes de Datos). Base de Datos 1. Casales Cabrera María Evelia. Maestría en ciencias e ingenierías de la computación, 2009-1. http://hp.fciencias.unam.mx/~alg/bd/dwh.pdf Accedido Enero de 2011.

[12] W. H. Inmon: *"Building the Data Warehouse."*. (John Wiley & Sons Inc., 1992, 1ª Ed.).

[13] R. Kimball: *"The Data Warehouse Toolkit"*. (John Wiley & Sons Inc., 1996, 1ª Ed.).

[14] R. Kimball: *"The Data Warehouse Lifecycle Toolkit"*. (John Wiley & Sons Inc., 1998, 1ª Ed.).

[15] R. Kimball: *"The Data Warehouse ETL Toolkit"*. (John Wiley & Sons Inc., 2004, 1ª Ed.)

[16] R. Kimball: *"The Data Warehouse ETL Toolkit (2nd Edition)"*. (John Wiley & Sons Inc., 2008, 2ª Ed.)

[17] SAS Institute: "SAS Warehousing Methodology", (SAS Institute white paper, 2001)

[18] http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos. Accedido en Abril de 2011.

[19] http://es.wikipedia.org/wiki/Middleware. Accedido en Marzo de 2011.

[20] ISO, I.O.o.S., *ISO/IEC 9126:2001 Information Technology - Software Product Quality*. 2001.

[21] ISO, I.O.o.S., *ISO/IEC FDIS 25012 Software engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Data quality model*. 2008.

[22] ISO, I.O.o.S., *ISO/IEC 9001:2008 Quality Managements Systems - Requirements* 2008.

[23] http://es.wikipedia.org/wiki/ISO/IEC_25000. Accedido en Enero de 2012.

[24] Matthias Jarke, Y.V., *Data Warehouse Quality: A review of the DWQ Project*, in *Conference of Information Quality*. 1997: Massachusetts Institute of Technology, Cambridge.

[25]Matthias Jarke, M.a.J., Christoph Quix, and Panos Vassiliadis, *Architecture and quality in Data Warehouses: an extended repository approach*, in *Science Direct*. 1999.

[26] Richard Y. Wang, D.M.S., *Beyond accuracy: what data quality means to data consumers.* Journal of Management Information Systems, 1996.**12**(4):

[27] Leo L. Pipino, Y.W.L., and Richard Y. Wang, *Data Quality Assessment*, in *COMMUNICATIONS OF THE ACM*. 2002.

[28]Rudra, A.Y., E., *Key Issues in Achieving Data Quality and Consistency in Data Warehousing*
among Large Organizations in Australia, in *Proceedings of the 32nd Annual Hawaii International Conference on* 1999, System Sciences, 1999. HICSS-32.: Hawaii, EEUU.

[29] Rudra, A.Y., E., *Issues in user perceptions of data quality and satisfaction in using a data warehouse-an Australian experience*, in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*. 2000: Hawaii United State.

[30] Leitheiser, R.L., *Data Quality in Health Care Data Warehouse Environments*, in *Proceedings of the 34th Hawaii International Conference on System Sciences - 2001*. 2001.

[31] U.M. Fayyad. Tutorial report.Summer school of DM. Monash Uni Australia.

[32] http://www.ati.es/novatica/2011/211/Nv211-Digital.pdf. Revista Novática número 211. Accedido en Marzo de 2012

[33] B. Azvine, Z. Cui, D. D. Nauck. Towards realtime business intelligence. BTTechnology Juornal Vol 23 No 3 July 2005.

[34] K.R. Quinn. Establishing a culture of Measurement, a practical guide to BI. White paper Information Builders 2003.

[35] J. Becker, L. vilkov, C. Brelage. Multidimensional Knowledge Spaces for Strategic Management-Experiences at a Leading Manufacturer of Construction and Mining Equipment.DEXA Workshops 2004.

[36] A. Countain, P. Finnegan, D. Sammon. Towards a framework for evaluating investments in data warehousing. Inf. Syst. J.

[37] M. Gibson, D. Arnott, I. Jagielska. Evaluating the intangible Benefits of Business Intelligence: Review & Research Agenda.IFIP TC8/WG8.3 International Conference, 2004.

[38] A. Faulkner, A. MacGillivray. A business lens on Business Intelligence – 12 tips for success.ODTUG 2001.

[39] L. T. Moss. Ten Mistakes to avoid for data warehouse projects managers. TDWI'S best of Business Intelligence Vol 3.

[40] T. Chenoweth, K. corral, H. Demirkan. Seven key interventions for data warehouse success. Commun, ACM 49 (1).

[41] M. D. Solomon. Ensuring a successful data warehouse initiative. ISM Journal winter 2005.

[42] D. Briggs, D. Arnott. Decision Support Systems Failure: An Evolutionary Perspective (Working Paper. No 2002/01). Melbourne, Australia: Decission Supprot systems Laboratory, Monash University.

[43] D. Briggs. Acritical Review of Literature on Data Warehouse Systems Success/Failure (Working paper No 2004/01). Melbourne, Australia: Decission Supprot systems Laboratory, Monash University.

[44] B. H. Wixom, H. J. Watson. An empirical investigation of the factors affecting data warehouse success.MIS Quarieriy Vol 25 No.1, Marzo 2001.

[45] B. H. Wixom, H. J. Watson. The current state os Business Intelligence.IEEE Computer (COMPUTER) 40 (9).

[46] D. Sammon, P. Finnegan, The Ten commandaments od data Warehousing. The DATA BASE for Advances in Information Systems.Vol 31.No. 4.

[47] R. Weir, T. Peng, J. M. Kerridge. Best Practise for Implementing a Data Warehouse: Areview for strategic alignment. DMDW 2003.

[48] R. S. Abdullaev, I. S. Ko. A Study on successful Business Intelligence systems in Practice.JCIT 2 (2).

[49] I.S. Ko, R. S. Abdullaev. A Study on the Aspects os successful Business Intelligence System Development, Computational Science – ICCS 2007.

[50] W. Yeoh , J. Gao, A. Koronios. Towards Critical Success Factor Framework for Implementing Business Intelligence systems: A Delphi Study in Engeneering Asset Management Organizations. CONFENIS 2007.

[51]http://es.wikipedia.org/wiki/OLTP. Accedido en Mayo de 2012.

[52]http://es.wikipedia.org/wiki/Gestor_transaccional. Accedido en Mayo de 2012.

[53]http://Data Warehouse4u.info/OLTP-vs-OLAP.html. Accedido en Mayo de 2012.

[54]http://es.wikipedia.org/wiki/OLAP. Accedido en Mayo de 2012.

[55]http://blog.multiplexion.net/data-cleansing-la-informacion-si-puede-ser-fiable/ . Accedido en Junio de 2012.

[56]http://es.wikipedia.org/wiki/Extract,_transform_and_load .Accedido en Junio de 2012.

[57]F. García Merayo y E. Luna Ramírez. El proceso datawarehousing y los metadatos. Conciencia tecnológica, diciembre, número 015. IT Aguascalientes, México.

[58] J.M. Franco. EDS-Institut Prométhéus. El data Warehouse.El data Mining. Eyrolles, 1997.

[59]http://churriwifi.wordpress.com/2009/11/24/2-3-eis-executive-information-system/

[60]http://es.wikipedia.org/wiki/Cuadro_de_mando_integral. Accedido en Junio de 2012

[61] http://es.wikipedia.org/wiki/EIS. Accedido en Junio de 2012.

[62] http://es.wikipedia.org/wiki/DSS. Accedido en Junio de 2012.

[63]https://www.agpd.es/portalwebAGPD/canalresponsable/obligaciones/calidad_de_datos/index-ides-idphp.php . Accedido en Junio de 2012.

[64] Boehm, B., A spiral model of software development and enhancement. IEEE Computer, 1988. 21(5): p. 61-72.

[65]http://es.wikipedia.org/wiki/ISO/IEC_15504. Accedido en Mayo de 2012

[66]http://www.iso15504.es/. Accedido en Mayo de 2012.

[67]http://www.it360.es/iso15504.php. Accedido en Julio de 2012.

[68]http://www.iso15504.es/index.php/la-norma-iso-15504-spice.html. Accedido en Julio de 2012.

[69]http://es.wikipedia.org/wiki/ISO/IEC_12207. Accedido en Junio de 2012.

[70]http://www.iso15504.es/index.php/modelo-procesos-iso122072008.html. Accedido en Julio de 2012.

[71]IEEE Standard for Software Quality Assurance Plans.IEEE Computer Society, 2002.

[72]http://www.luiscorral.webs.com/sqa_plan.pdf . Accedido en Julio de 2012.

[73]http://es.wikipedia.org/wiki/PMBOK. Accedido en Junio de 2012.

[74] Atos Consulting, PMBOK, Junio de 2011.

[75]Atos Consulting, CMMI, Junio de 2011.

[76]http://es.wikipedia.org/wiki/Objetivos_de_control_para_la_informaci%C3%B3n_y_tecnolog%C3%ADas_relacionadas. Accedido en Julio de 2012.

[77] http://msdn.microsoft.com/en-us/library/aa578560.aspx Accedido en Agosto de 2012

[78] http://msdn.microsoft.com/en-us/library/aa561521 Accedido en Julio de 2012

[79] http://msdn.microsoft.com/en-us/library/ms141026.aspx Accedido en Julio de 2012

[80] ISO, I.O.o.S., *ISO/IEC 90003:2004 Software engineering -- Guidelines for the application of ISO 9001:2000 to computer software,* 2004.

[81] ISO, I.O.o.S.,ISO/IEC 25000:2005 Software Engineering -- Software product Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE, 2005.

[82] Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) - Fourth Edition.

ANEXOS

1. BUSINESS INTELLIGENCE DEVELOPMENT STUDIO (BIDS)

BIDS o Business Intelligence Development Studio es la herramienta principal que se utilize para crear los paquetes de SSIS. Una vez instalado el módulo de SSIS para SQL Server 2008, aparece en la lista de programas que se anidan desde la herramienta.

BIDS incluye un conjunto de ventanas para todas las fases del desarrollo y la administración de proyectos. Por ejemplo, Business Intelligence Development Studio incluye ventanas que permiten administrar varios proyectos como una unidad, así como ver y modificar las propiedades de objetos en proyectos. Estas ventanas están disponibles para todos los tipos de proyecto en Business Intelligence Development Studio.

El diagrama siguiente muestra las ventanas de Business Intelligence Development Studio con la configuración predeterminada.

Figura 104. BIDS

BIDS consta de cuatro ventanas principales:

1. **Explorador de soluciones**
2. **Ventana Propiedades**
3. **Ventana Diseñador**
4. **Cuadro de herramientas**

Otras ventanas incluidas en BIDS Studio permiten ver resultados de búsqueda y obtener información sobre mensajes de error e información que generan los diseñadores o depuradores del proyecto. El Explorador de servidores enumera conexiones de base de datos; el Explorador de objetos muestra los símbolos que están disponibles para utilizarlos en un proyecto; la Lista de tareas presenta las tareas de programación definidas por el usuario; y la Lista de errores proporciona descripciones detalladas de errores.

Explorador de soluciones

Los diferentes proyectos en una solución se pueden administrar desde una sola ventana: Explorador de soluciones. El Explorador de soluciones presenta la solución activa como un contenedor lógico para uno o varios proyectos, e incluye todos los elementos asociados a ellos. Desde esta vista, puede abrir directamente elementos de un proyecto para modificarlos o realizar otras tareas de administración. Debido a que los diferentes tipos de proyecto almacenan los elementos de diferente manera, la estructura de carpetas del Explorador de soluciones no refleja necesariamente el almacenamiento físico real de los elementos enumerados en la solución.

En el Explorador de soluciones se pueden crear soluciones vacías y, a continuación, agregar proyectos nuevos o existentes a la solución. Si crea un proyecto sin antes crear una solución, BIDS crea automáticamente la solución. Cuando la solución incluye proyectos, las tres vistas incluyen nodos para objetos específicos del proyecto. Por ejemplo, el proyecto de Analysis Services incluye un nodo Dimensiones, el proyecto de Integration Services incluye un nodo Paquetes y el proyecto de modelos de informes incluye un nodo Informes.

Para obtener acceso al Explorador de soluciones, haga clic en Explorador de soluciones en el menú Ver.

Ventana Propiedades

La ventana Propiedades enumera las propiedades de un objeto. Use esta ventana para ver y cambiar las propiedades de objetos, como los paquetes, que se abren en editores y diseñadores. También puede usar la ventana Propiedades para editar y ver las propiedades de archivos, proyectos y soluciones.

Los campos de la ventana Propiedades tienen incrustados distintos tipos de controles que se abren al hacer clic en ellos. El tipo de control de edición depende de la propiedad concreta. Entre estos campos de edición se incluyen cuadros de edición, listas desplegables y vínculos a cuadros de diálogo personalizados. Las propiedades que aparecen atenuadas son de solo lectura.

Para obtener acceso a la ventana Propiedades, haga clic en Ventana de propiedades en el menú Ver.

Ventana Cuadro de herramientas

La ventana Cuadro de herramientas muestra varios elementos que se utilizan en los proyectos de Business Intelligence. Las fichas y los elementos disponibles en el cuadro de herramientas varían en función del diseñador o editor que se esté utilizando.

La ventana Cuadro de herramientas muestra siempre la ficha General y puede que también muestre fichas como Elementos de flujo de control, Tareas de mantenimiento, Orígenes de flujo de datos o Elementos de informe.

Algunos diseñadores y editores no utilizan elementos del cuadro de herramientas. En ese caso, el cuadro de herramientas solamente contiene la ficha General.

Para obtener acceso al cuadro de herramientas, haga clic en Cuadro de herramientas en el menú Ver.

Ventana Diseñador

La ventana Diseñador es la ventana de herramienta donde se crean o modifican objetos de Business Intelligence. El diseñador proporciona una vista de código y una vista de diseño de un objeto. Cuando se abre un objeto en un proyecto, el objeto se abre en un diseñador especializado en esta ventana. Por ejemplo, si abre una vista de origen de datos en uno de los proyectos de Business Intelligence, la ventana del diseñador se abre con el Diseñador de vista de origen de datos.

La ventana Diseñador no está disponible hasta que se agrega un proyecto a una solución y se abre un objeto en ese proyecto.

También se puede acceder a BIDS desde SQL Server Management Studio (SSMS). Desde el explorador de objetos de SSMS, se puede elegir en la opción “conectar” la opción de “Integration Services…”.

Figura 105. BIDS – Acceso desde SSMS

Nos aparecerá la ventana de solicitud de credenciales típica que cuándo nos conectamos al servidor de base de datos pero una vez validadas las credenciales y abierta la conexión, el explorador de objetos de SSMS nos da acceso a los paquetes de SSIS.

Figura 106. BIDS – Conexión desde SSMS.

Accediendo desde SSMS a BIDS, podremos ejecutar el asistente para importar o exportar paquetes, ejecutarlos, eliminarlos, etc.

Figura 107. BIDS – Acciones para paquetes.

2. ASISTENTE PARA IMPORTAR Y EXPORTAR EN SSIS.

Como puede verse en el Anexo de BIDS, se puede acceder al asistente de importación o de exportación de paquetes desde el explorador de objetos de BIDS o también directamente desde la herramienta abierta a través de visual studio.

El acceso desde visual studio será desda Proyecto, “Asistente para importación y exportación de SSIS…”.

Figura 108. SSIS – Acceso al asistente de importación – exportación.

Veremos un ejemplo de exportación de datos, por lo que para ello una vez iniciado el asistente debemos seleccionar la opción “Exportar Datos…”. Nos aparece la pantalla de presentación típica de Microsoft y seguidamente ya iniciaríamos realmente el asistente de exportación de datos, para seleccionar nuestra fuente u origen de datos.

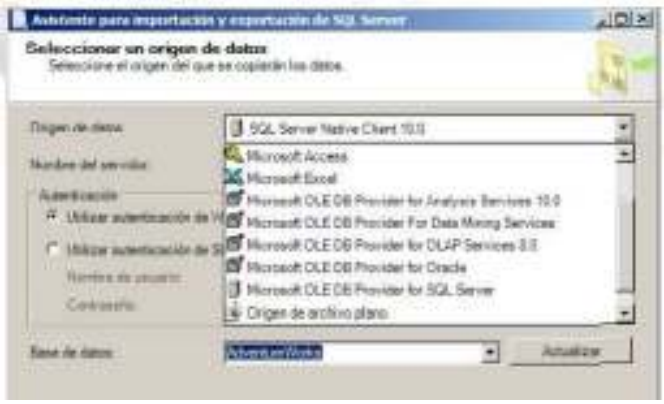

Figura 109. Asistente de exportación de datos. Selección de fuentes de datos.

Como hemos visto en la memoria, hay multiples conexiones o fuentes de datos y por tando aparecerán muchas de ellas en el asistente. Podemos comprobar que aparecen para MS Access, MS Excel, Archivos Planos, etc. Si elegimos como origen “SQL NAtive Client”, y rellenamos la ventana de credenciales

correspondiente (será la ventana típica de MS SQL Server), podremos avanzar hasta la selección de nuestro destino. Si por ejemplo elegimos un fichero plano deberíamos de configurar la siguiente pantalla:

Figura 110. Asistente de exportación de datos. Selección de destino de datos

Como hemos elegido por destino un fichero plano se nos indica que configuremos opciones relacionadas con ficheros de datos, como el nombre del archivo destino (ruta), la configuración regional o el formato que deseamos proporcionar para separar los campos, registros, etc.

Si el destino por ejemplo fuera un proveedor OLE DB de Microsoft para SQL Server o SQL Server Native Client nos aparecerían opciones típicas de bases de datos como el nombre del servidor de la base de datos, el modo de autenticación para acceder al destino, el nombre de la base de datos destino, etc.

Si como destino eligiéramos un archivo de Excel, deberíamos indicar la ruta del archivo o la versión de Excel, etc.

Es decir, dependiendo del tipo de origen y tipo de fuente de datos que usemos en el proceso de exportación o importación de datos tendremos que configurar el asistente para indicar como se realizan dichas tareas.

Se puede ver la lista de opciones a configurar a través del siguiente enlace a MSDN (Microsoft Software Developer Network): http://msdn.microsoft.com/es-es/library/ms178430.aspx

Continuando con el ejemplo, tras configurar el destino de datos pasamos a la pantalla de selección de datos, que nos indica si deseamos copiar los datos de una o varias tablas o por el contrario especificamos una consulta para extraerlos.

En caso de indicar que queremos seleccionar directamente los datos, se abre un editor de consultas SQL dónde podremos crear o copiar nuestra consulta (incluso analizar si la sintaxis es correcta) o para la opción

de tablas directamente se abre el listado de tablas de la base de datos y podremos seleccionar las indicadas a través de un lista de checkbox.

Avanzando llegaremos al paso dónde configuraremos el formato específico para el destino, que en nuestro caso es un archivo plano y podremos configurar como separar las filas o columnas. Visualizar como quedará la exportación (sobre una colección reducida de datos) e incluso editar las asignaciones o la consulta/tablas/s origen de los datos.

Figura 111. Asistente de exportación de datos. Configurar destino.

Una vez finalizado el asistente, éste tendrá toda la información necesaria para crear el paquete de exportación. Cuando el asistente se lanza desde SSMS, aparece una pantalla que nos pregunta si queremos ejecutar el proceso de exportación inmediatamente o guardarlo como paquete de SSIS. La opción recomendada es la última para poder abrir el paquete desde BIDS y reutilizarlo en futuras exportaciones, no debemos olvidar que para integrar una exportación en un flujo ETL, en el proceso de extracción se usarán varios paquetes para cada fuente de datos y por tanto se necesita tener controlados y localizados los paquetes de extracción. El resultado de nuestra finalización del asistente, desde BIDS será:

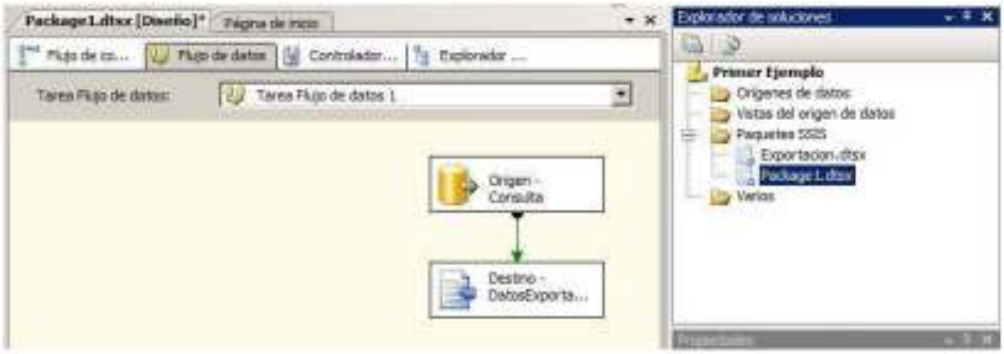

Figura 112. SSIS - Exportacion de Datos. Paquete Final

El paquete generado es muy sencillo y únicamente tiene dos componentes en el flujo de datos, el origen y el destino.

Para configurar rápidamente exportaciones de datos es algo muy cómodo, pero para utilizar en una solución de ETL real, será necesario aplicar varias transformaciones a los datos entre ambas cajas.

3. ASISTENTE PARA CONFIGURAR PAQUETES EN SSIS.

En tiempo de desarrollo, se introducen una serie de datos en la configuración del paquete que posiblemente luego hayan de ser diferentes cuándo el paquete se instale sobre uno o varios servidores, ya sea de pruebas o de producción. Por ejemplo, puede que se necesite configurar rutas que en desarrollo sean del tipo “C: \Pruebas” pero que en producción no tengamos acceso a la misma carpeta y haya una específica para ello, “E:\Export”, o que se esté leyendo desde un servidor SQL de prueba mientas desarrollamos y necesitamos conectar con otros servidores de producción al cambiar de entorno.

Para realizar automáticamente la configuración en tiempo de despliegue, se utiliza lo que se denomina “configuración del paquete” (Package Configuration). Se puede acceder al asistente de configuración de paquetes de SSIS desde BIDS a través del menú SSIS y “Configuraciones de paquetes…”.

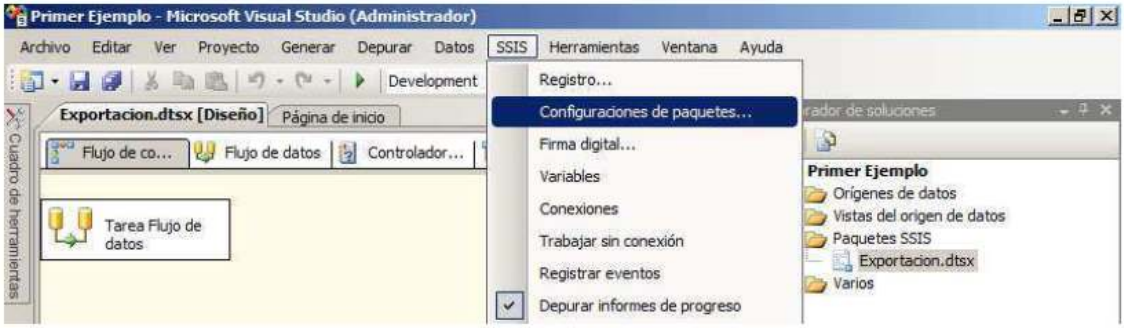

Figura 113. SSIS. Configuración de paquetes.

La configuración del asistente es sencilla, en primer lugar nos pregunta si queremos habilitar las configuraciones de paquetes. Tendremos que habilitar la opción ya que es de lo que se trata.

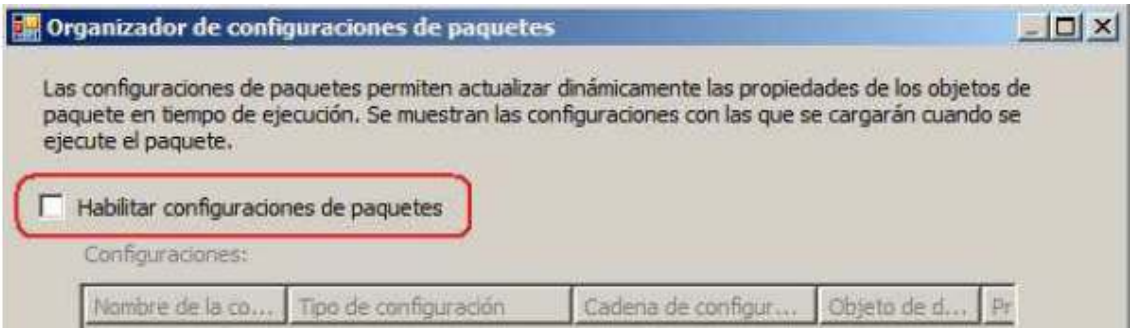

Figura 114. SSIS. Configuración de paquetes II.

Una vez habilitada la configuración podremos comenzar el proceso y se activará la zona cenrtal de la ventana, en la que podemos pulsar el botón “agregar” para ir añadiendo a la pantalla los distintos parámetros que deseemos configurar a posteriori dentro de nuestro paquete.

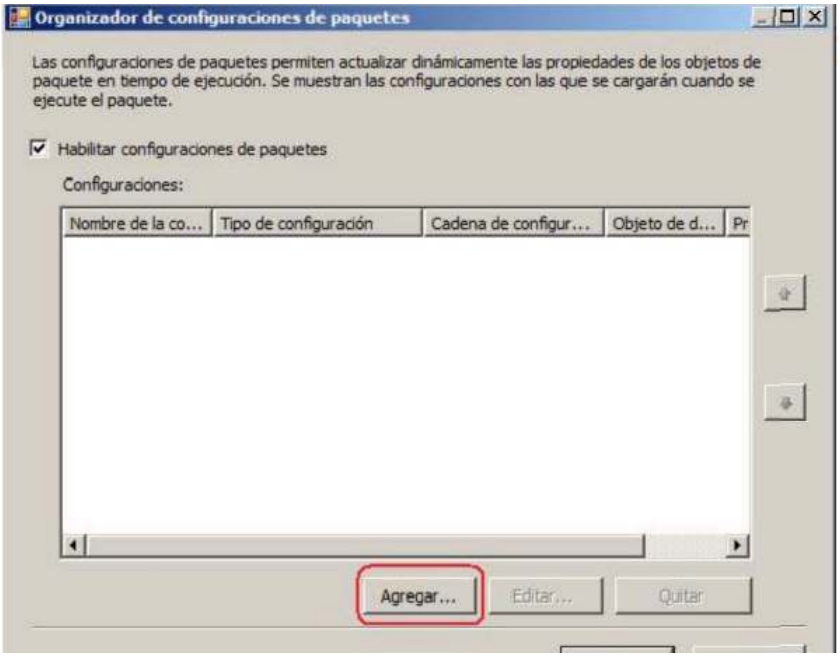

Figura 115. SSIS. Configuración de paquetes III

Cuándo pulsemos el botón, aparecerá el asistente que nos guiará para que aportemos la información necesaria para configurar cada parámetro. La primera pantalla pregunta desde dónde queremos leer, en tiempo de despliegue, los datos que se van a cambiar en el paquete. Se puede usar un archivo XML (archivo que no aparece por defecto, sino que se debe crear a mano y copiar al servidor de destino antes de instalar el paquete), una variable de entorno, una entrada del registro de Windows, una variable o una tabla de sql server. Se recomienda usar un archivo XML.

Según el valor seleccionado en el desplegable, la pantalla cambia para pedir la información oportuna. Por ejemplo, si seleccionamos un archivo XML nos pregunta el nombre del archivo.

Figura 116. SSIS. Configuración de paquetes IV

A continuación se nos pregunta cuál es el parámetro del paquete que se desea configurar. Se muestra una ventana con un árbol en el que se van desplegando todos los datos del paquete para que seleccionemos el que necesitemos.

Figura 117. SSIS. Configuración de paquetes V

Escogida la propiedad pertinente, haciendo click en siguiente se nos pregunta el nombre que queremos darle a la configuración que acabamos de crear. Éste dato es importante si vamos a proceder a desplegar el paquete en distintos servidores para que sepamos cuál aplicar, por lo que el nombre recomendado suele ser el nombre del servidor destino o similar.

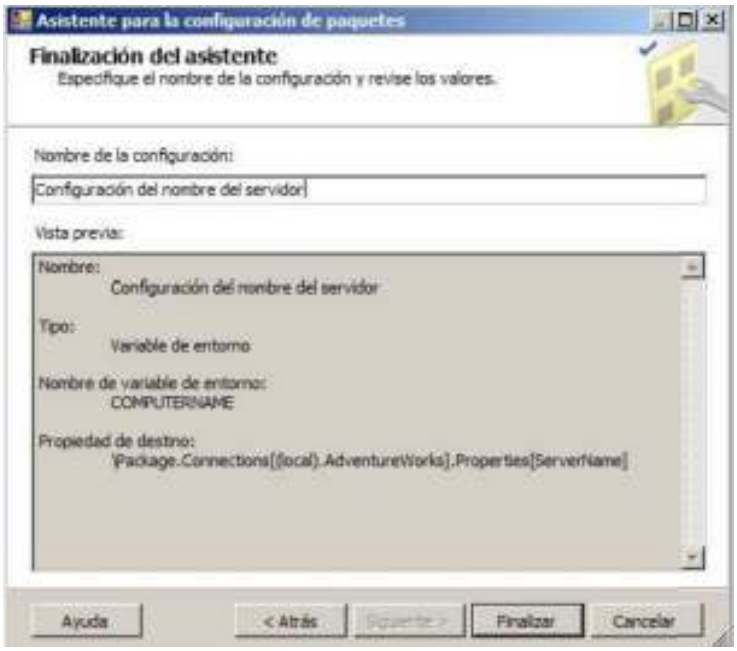

Figura 118. SSIS. Configuración de paquetes VI

Una vez terminada la configuración, volveríamos a la ventana de configuraciones, en la que podemos seguir añadiendo líneas hasta terminar la configuración de nuestro paquete.

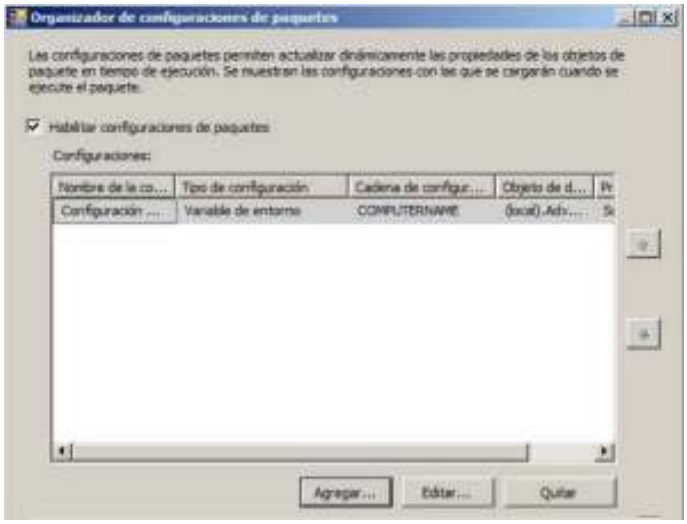

Figura 119. SSIS. Configuración de paquetes VII

4. ASISTENTE PARA INSTALAR PAQUETES EN SSIS.

Una vez terminado nuestro desarrollo de paquetes **.dtsx**,para instalarlos sobre un servidor de producción podemos generar lo que se denomina “utilidad de implementación o de despliegue” (Deployment Utility). Se genera automáticamente un archivo terminado en la extensión **.SSISDeploymentManifest**, que deberemos copiar a la máquina destino junto con el archivo .dtsx y cualquier otro archivo dependiente de éste (archivos de configuración, etc.). Estos archivos se copian automáticamente a la carpeta seleccionada al configurar BIDS para que genere la utilidad de implementación.

Para que se genere el archivo .SSISDeploymentManifest, tenemos que activar la opción de generar la utilidad de implementación. Para ello debemos ir a las **propiedades**de la solución y establecer en **True** la opción **CreateDeploymentUtility**.

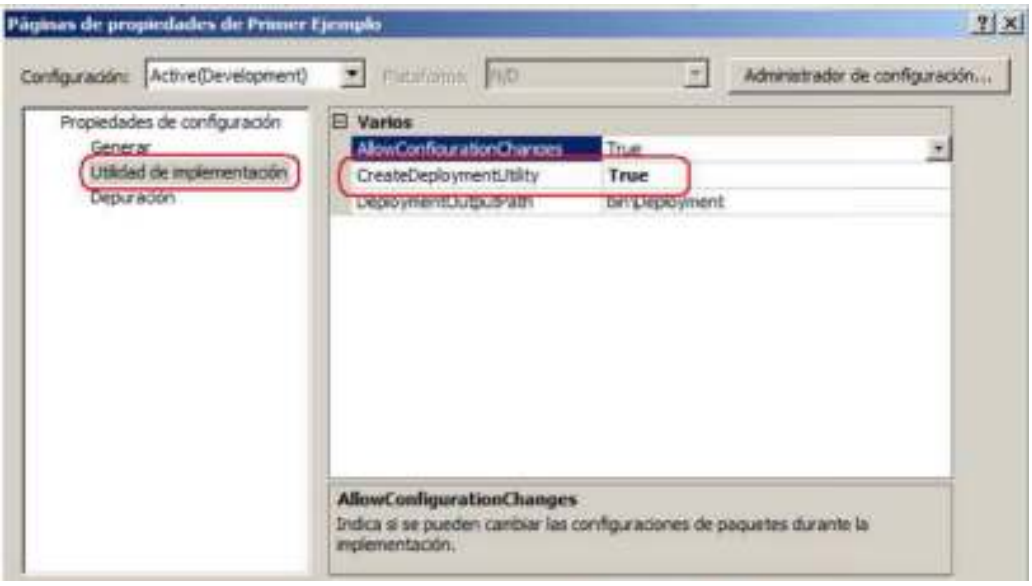

Figura 120. SSIS. Instalación de paquetes

Al aplicar ésta configuración, al seleccionar en BIDS la opción de “Generar”, se creará en la carpeta indicada en la pantalla (bin\Deployment por defecto) un fichero con el nombre de la solución y la extensión .SSISDeploymentManifest, además de los correspondientes .dtsx y sus dependencias. Los archivos de ésta carpeta serán los que usemos para realizar el despliegue en producción.

Una vez copiados los archivos en el servidor destino, al ejecutar el .SSISDeploymentManifest se activa el asistente para instalar paquetes que nos guiará durante el proceso de instalación como podemos ver en las siguientes capturas de pantalla:

Figura 121. SSIS. Instalación de paquetes II

Podemos elegir dónde implementar el paquete, si en el sistema de archivos del sistema o en el propio SQL Server. La opción a elegir suele depender de la política de la organización.

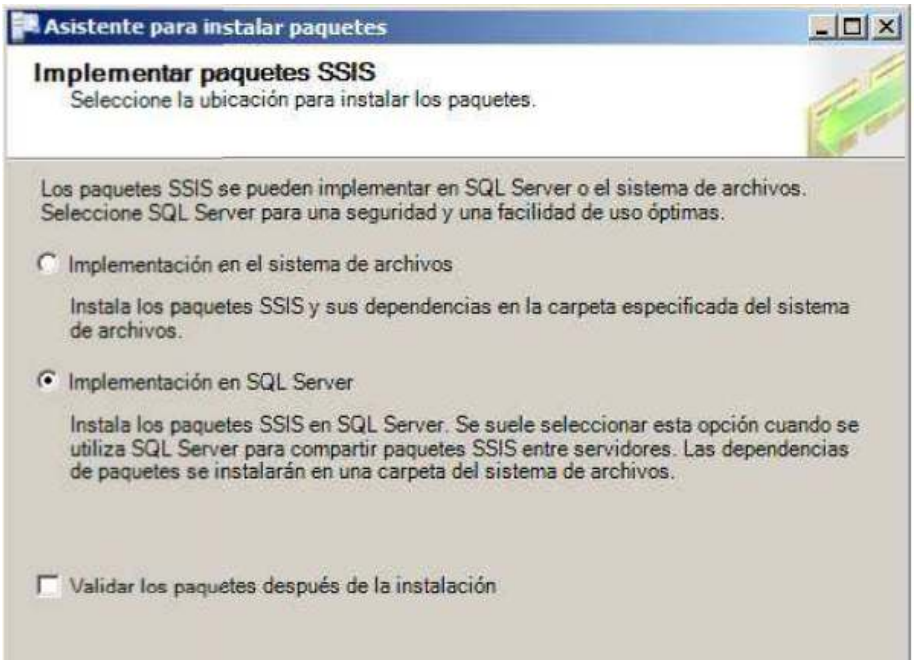

Figura 122. SSIS. Instalación de paquetes III

Si elegimos el sistema de ficheros nos pedirá la ruta destino, si elegimos en SQL Server debemos elegir el servidor de SQL Server que usaremos como destino, especificar las credenciales para poder alojar el paquete en el servidor y la ruta de acceso al paquete.

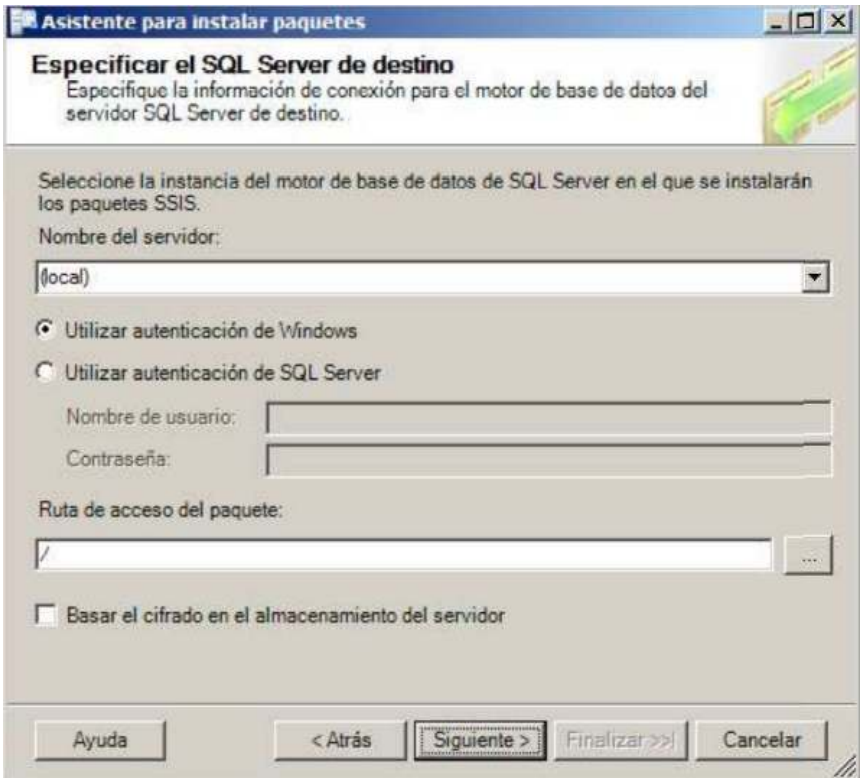

Figura 123. SSIS. Instalación de paquetes IV

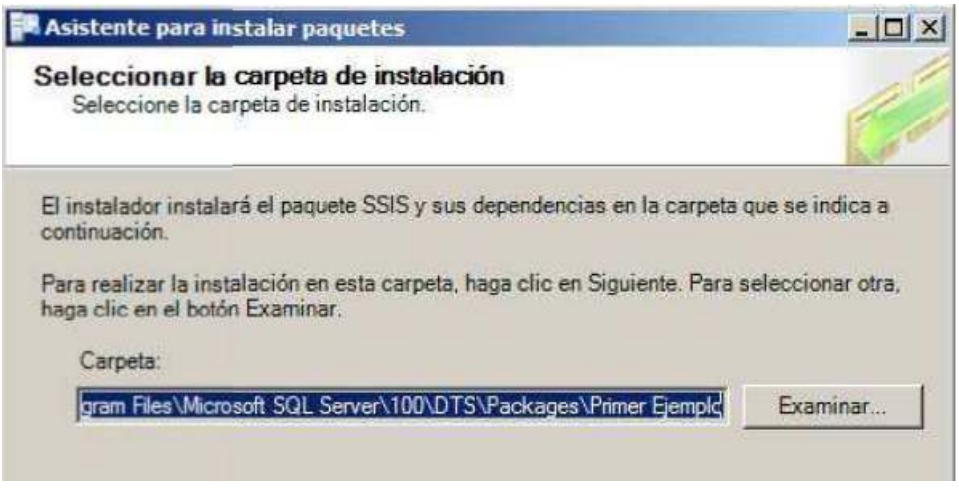

Figura 124. SSIS. Instalación de paquetes V

Una vez especificado todo, se procedería a instalar el paquete y se mostraría una ventana informativa indicando el resultado de la instalación.

5. UTILIDAD DE EJECUCIÓN DE PAQUETES EN SSIS.

Tenemos una herramienta muy útil, ***dtexecui.exe***, que permite seleccionar una paquete y ejecutarlo directamente.

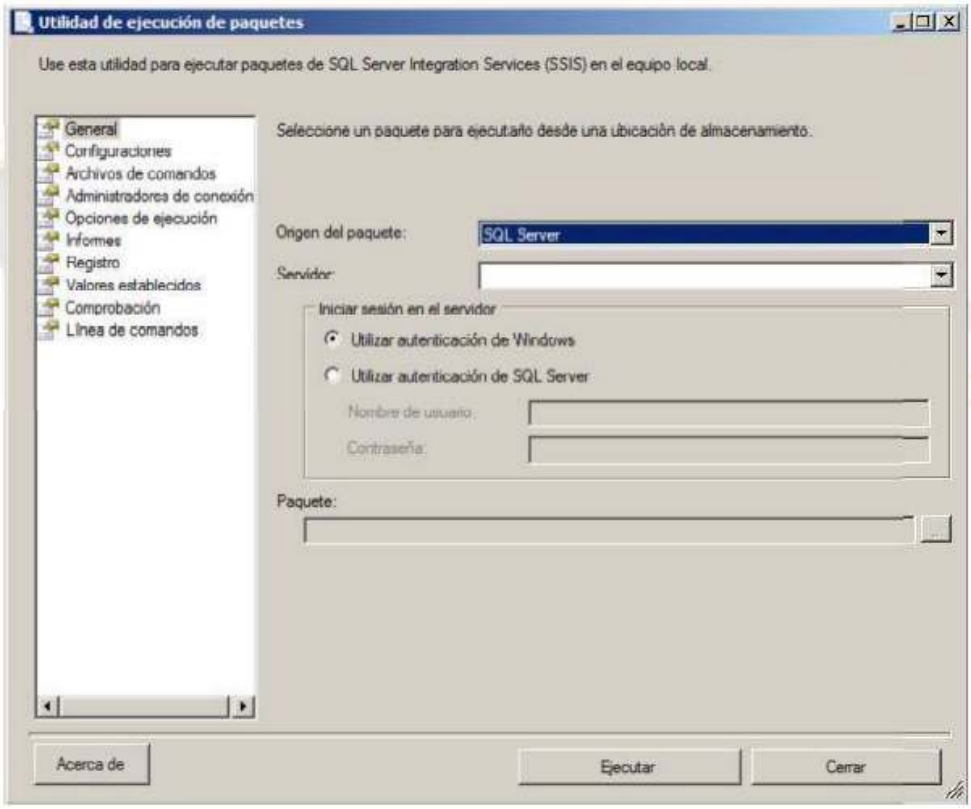

Figura 125. SSIS - Utilidad de ejecución de paquetes.

El asistente nos pregunta la ubicación del paquete a ejecutar (será dentro de SQL Server o del sistema de archivos). Además nos permite configurarlo si es necesario, establecer opciones de registro e informes. Haciendo click en el botón de *Ejecutar* podemos lanzar la ejecución del paquete.

Entre las diversas opciones más o menos evidentes que se presentan en las pestañas de la izquierda, resulta de interés la última, ***"Línea de comandos"***, que nos presenta la siguiente pantalla:

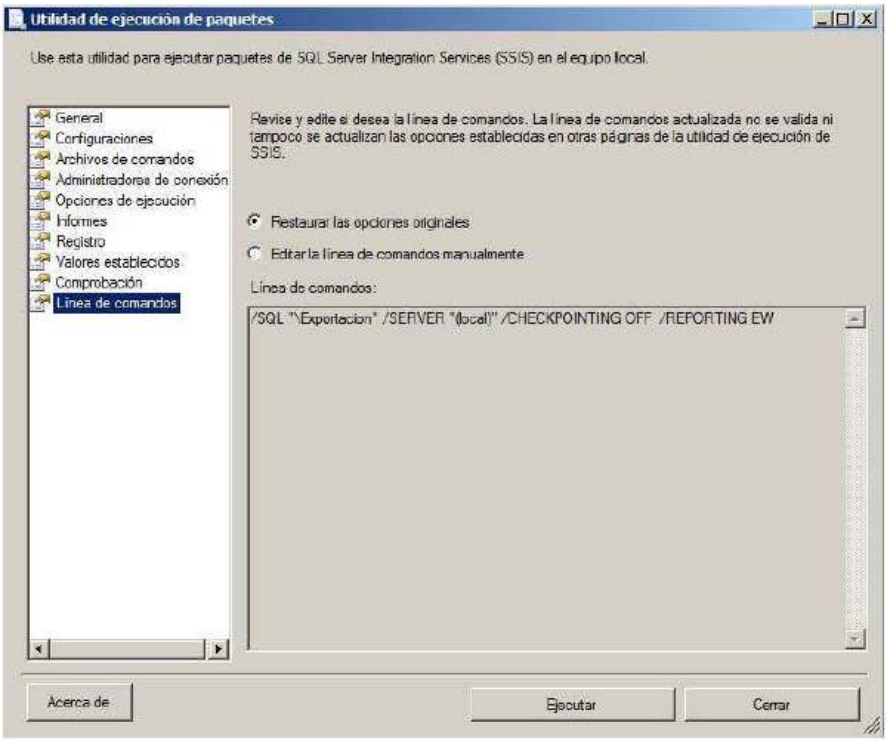

Figura 126. SSIS - Utilidad de ejecución de paquetes. Línea de comandos.

Desde ésta pantalla podemos copiar las línea de comentas y usarla con la utilidad de línea de comandos ***dtexec.exe***, que nos permitirá lanzar el paquete .dtsx desde línea de comandos sin necesidad de utilizar ninguna ventana gráfica. Es muy útil utilizar el comando ***"dtexec.exe" + " " + "línea de comandos"*** para planificar la ejecución de paquetes en entornos de explotación o producción, dónde no podemos tener acceso a ellos y se debe encapsular la ejecución por ejemplo en archivos .bat o .cmd, etc.

6. HERRAMIENTAS DE LÍNEA DE COMANDOS DE SSIS

Hemos mencionado en el anexo 5, Utilidad de ejecución de paquetes en SSIS, la herramienta ***dtexec.exe*** que sirve para lanzar paquetes desde línea de comandos y es muy útil para ejecutar paquetes (ya sea de forma planificada o excepcional) en entornos productivos. Podemos ver la documentación en línea de SQL Server dónde explica en profundidad el uso de la herramienta, puede verse en la siguiente URL.

http://msdn.microsoft.com/es-es/library/hh231187

A destacar tenemos la posibilidad de utilizar otra herramienta de comandos, ***dtutil.exe***, que se utiliza para administrar paquetes.

Dtutil puede importar y exportar paquetes, firmarlos, verificarlos o borrarlos. Estas acciones se pueden realizar sobre cualquier paquete SSIS almacenado en una de estas tres ubicaciones:

1) Una base de datos de MS SQL Server o /SQL.
2) Un almacén de paquetes de SSIS o /DTS.
3) El sistema de archivos o /FILE.

La sintaxis a utilizar para el comando es la siguiente:

"dtutil" + " " + "/opción [valor]" + " " + "/opción [valor]" + " " + "/opción [valor]" + " "…

La opciones que indicamos pueden ir en cualquier orden, las más importantes son:

- **/C: Copiar**→dtutil /SQL srcPackage /C DTS;destFolder\destPackage
- **/Del: Borrar**→dtutil /SQL paquete/DEL serviría para borrar "paquete" almacenado en SQL Server.
- **/Ex: Comprobar la existencia de un paquete** →Para comprobar que se ha desplegado correctamente un paquete en producción.
- **/H: Ayuda.**
- **/M: Mover un paquete a otra ubicación.**

Estas son sólo las opciones más importantes, la documentación en línea de SQL Server enumera y explica en profundidad las 26 opciones posibles, puede verse en la siguiente URL.

http://msdn.microsoft.com/es-es/library/ms162820.aspx

7. VARIABLES DE SSIS.

Las variables permiten almacenar información desde algún punto de un paquete de SSIS para luego recuperarla en otro punto. Con ésta técnica se permite la comunicación entre los objetos de un paquete y entre paquetes primarios y secundarios. Además las variables son accesibles en expresiones o en scripts.

Las variables pueden ser de dos tipos:

1) Variables del sistema.
2) Variables definidas por el usuario.

Al crear un paquete y agregar un contenedor o una tarea a un paquete, o crear un controlador de eventos, SSIS incluye un conjunto de **variables del sistema**para el contenedor. Las variables del sistema contienen información útil sobre el paquete, el contenedor, la tarea o el controlador de eventos. Por ejemplo, en tiempo de ejecución, la variable del sistema **MachineName**contiene el nombre del equipo en el que se ejecuta el paquete y la variable **StartTime**, la hora a la que se empezó a ejecutar el paquete. Las variables del sistema son de sólo lectura.

Como hemos identificado, también se pueden crear **variables definidas por el usuario**y utilizarlas en los paquetes. Las variables definidas por el usuario se pueden utilizar de muchas formas en SSIS:

- En scripts.
- En expresiones de las restricciones de precedencia.
- En el contenedor de Bucles For.
- En la transformación de columna derivada.
- En la transformación de división condicional.
- En las expresiones de propiedad que actualizan los valores de las propiedades.
- Etc.

Uno de sus usos más comunes es usar las variables en las expresiones. El generador de expresiones nos ofrece tanto las variables del sistema como las variables del usuario para crear las expresiones.

Las variables se utilizan en las expresiones escribiendo su nombre precedido por el símbolo @. Por ejemplo, la siguiente expresión accede a la variable Tabla:

```
SELECT * FROM @Tabla
```

Se pueden ver las variables que tenemos definidas en el paquee desde el menú SSIS > Variables.

Figura 127. SSIS – Acceder a las variables

Por ejemplo, seleccionando esta opción y creando una variable llamada Archivo, podemos ver el ámbito en el que se define y que hay una serie de iconos que nos permiten añadir, borrar o mostrar las variables.

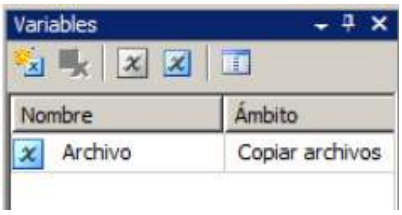

Figura 128. SSIS – Variables.

8. AUDITORÍA, REGISTROS Y GESTIÓN DE EVENTOS DE SSIS.

Se puede configurar un paquete SSIS para que escriba un **log** cuando se produzcan eventos en tiempo de ejecución. De esta forma podemos auditar la ejecución del paquete o de las partes del mismo en las que hayamos habilitado el registro. Ésta herramienta es especialmente importante para entornos de producción dónde no se puede depurar los paquetes mediante puntos de interrupción.

A nivel de tarea o de contenedor, se puede modificar la propiedad **LoggingMode** para indicar si el log está habilitado,deshabilitado o si se hereda desde su contenedor (UseParentSetting) que será el valor predeterminado, es decir cada tarea hereda de su contenedor y , a su vez, cada contenedor hereda de quien lo contenga.

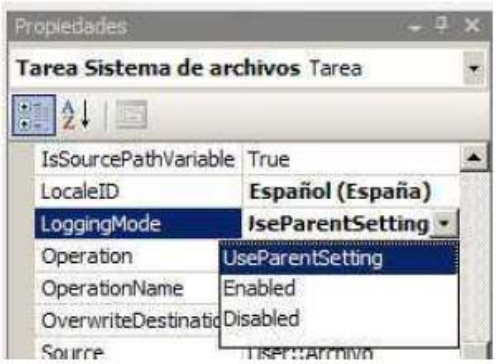

Figura 129. SSIS – Opciones de logging

Para habilitar y establecer el modeo de escritura de los registros, se debe seleccionar la opción de **Registro** en el menú de SSIS.

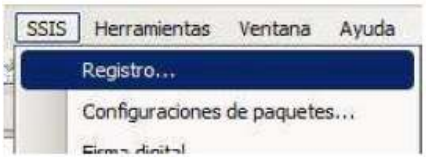

Figura 130. SSIS - Habilitar logging.

Se abrirá una pantalla en la se selecciona en un árbol el objeto a cuyo nivel al que queremos establecer los registros. Aparecerán en gris aquellos objetos que están marcados como UseParentSetting, indicándonos que hay que seleccionar una opción más arriba en el árbol.

Figura 131. SSIS - Configurar logging de SSIS.

En la parte derecha, se selecciona el **tipo de proveedor**, que nos permite determinar si queremos escribir el registro de log en:

- Un archivo de texto.
- El registro de eventos de Windows.
- Un archivo XML.
- Una tabla de SQL Server.
- En SQL Server Profiler.

Según el proveedor de destino que seleccionemos, se nos pide configurar las opciones oportunas. Desde la pestaña de "***Detalles***" se seleccionan cuáles son los eventos que queremos registrar

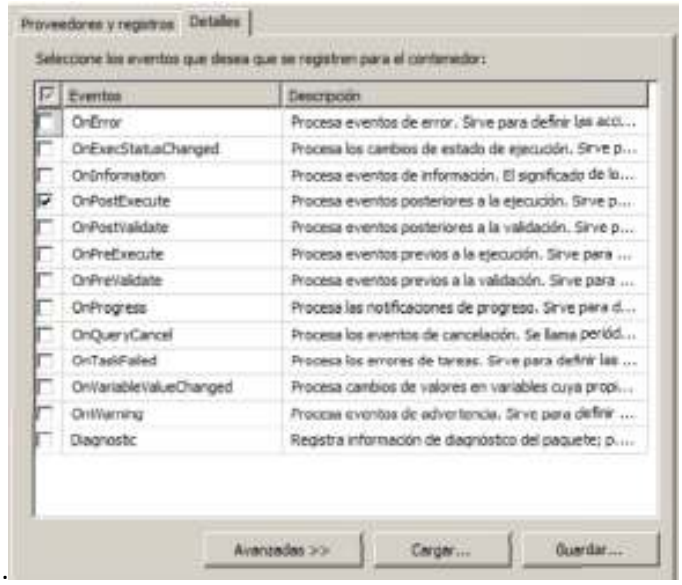

Figura 132. SSIS - Detalles logging de SSIS.

Una vez activado el registro, los eventos correspondientes se graban en el destino seleccionado al ejecutar el paquete.

También podemos ver los eventos en el propio BIDS sobre la marcha, mientras se están produciendo, en caso de que estemos ejecutando el paquete en un entorno de desarrollo. Para ello, se debe seleccionar la opción de "***Registrar Eventos***" en el menú de SSIS.

Figura 133. SSIS - Acceder a los eventos en caliente.

Una vez seleccionada ésta opción se abrirá una ventana que va mostrando los eventos según se van ejecutando las tareas del paquete.

Figura 134. SSIS - Captura de los eventos en caliente.

Como acabamos de ver se producen eventos en tiempo de ejecución que provienen de los contenedores y tareas. Nosotros podemos crear controladores de eventos que respondan a éstos ejecutando un flujo de trabajo. Por ejemplo, se puede crear un controlador de eventos que envíe un mensaje de correo electrónico cuando una tarea genera un error.

Un **controlador de eventos** es similar a un paquete. Al igual que un paquete, un controlador de eventos puede contener variables, e incluye un flujo de control y opcionalmente flujos de datos.

Se pueden generar controladores de eventos para paquetes, contenedores o tareas.

Para crear controladores de eventos se utiliza la pestaña controladores de eventos en el diseñador de BIDS.

Figura 135. SSIS -Crear controladores de eventos.

Una vez seleccionado el objeto sobre el que queremos conectar controladores de eventos, y el evento concreto que queremos controlar, la pestaña se transforma al aspecto de una superficie de diseño similar a la del flujo de control. Sobre esta superficie podemos arrastrar contenedores y tareas y configurarlos en la misma forma que los hacíamos en la pestaña del flujo de control.

Figura 136. SSIS -Configurar controladores de eventos.

9. EXTENSIONES DE SSIS MEDIANTE CÓDIGO .NET.

Una gran herramienta de SSIS es el hecho de que aparte de la funcionalidad que viene ya incorporada con las tareas y transformaciones propias de SSIS, podemos añadir prestaciones adicionales mediante código escrito por nosotros mismos en alguno de los lenguajes de programación de .Net (Visual Basic o C#).

Lógicamente, no se trata de arrastrar y configurar cajas en una superficie de diseño sino que se trata de puro código escrito en un lenguaje de programación.

En el flujo de control podemos añadir una tarea que se denomina **tarea Script** (Script Task).

Figura 137. SSIS -Tarea script.

Cuando editamos las propiedades de la tarea, nos permite abrir una ventana de Visual Studio (Editor de Microsoft para código .Net y el que se usa para abrir BIDS) desde la que podemos escribir código de .Net que se ejecutará cuando el flujo de control pase por ésta tarea.

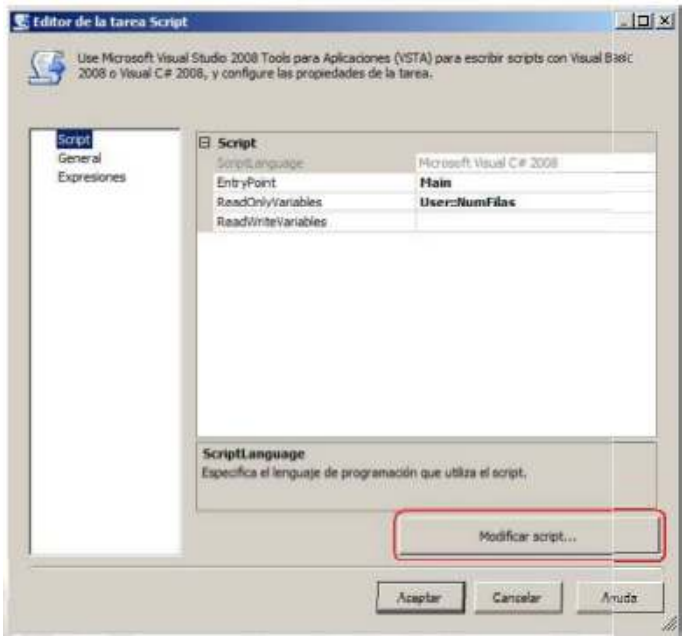

Figura 138. SSIS -Modificar la tarea script.

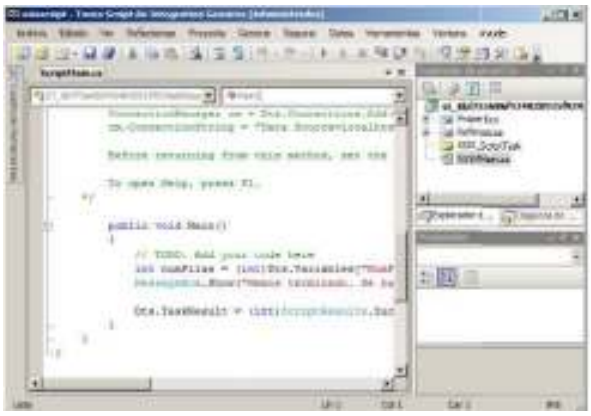

Figura 139. SSIS -Ventana de Visual Studio

Además, desde la ventana de **Flujo de datos**, tenemos disponible en la zona de transformaciones el **componente de script**.

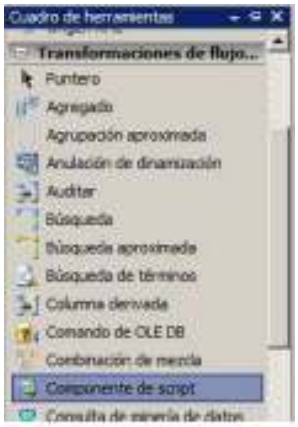

Figura 140. SSIS -Componente de script.

Aunque se encuentra en la categoría de transformaciones, en realidad este componente también se puede utilizar como origen o destino del flujo de datos. Al arrastrarlo sobre la superficie de diseño, aparece una ventana que nos pregunta qué tipo de uso va a tomar:

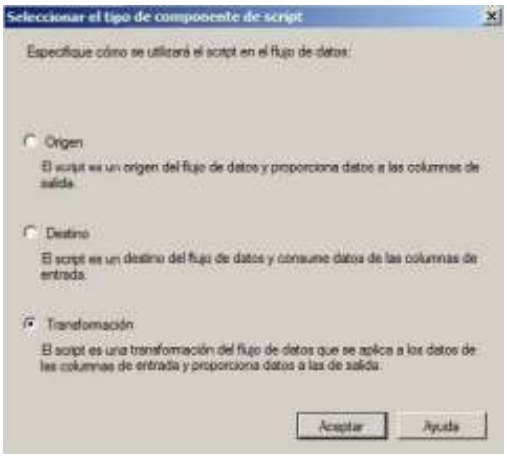

Figura 141. SSIS -Usos del componente de script

La ventana de edición de propiedades tiene el botón "***Editar Script***" que abre una ventana de Visual Studio, de manera similar a la que se ha visto en la tarea de Script.

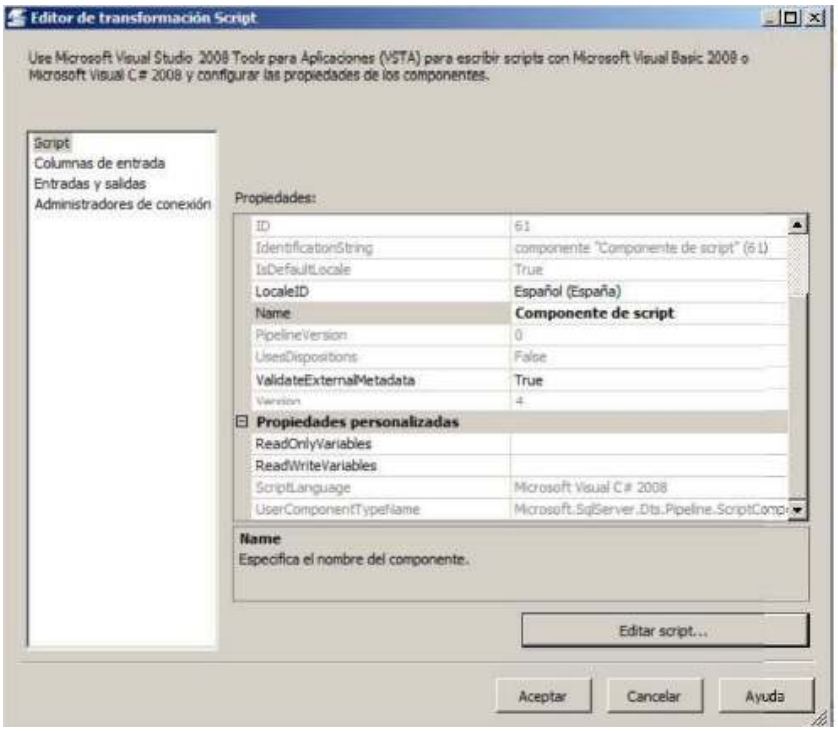

Figura 142. SSIS -Editar componente de script

Sobre la ventana que aparece, se escribe el código que determina el comportamiento del componente. La ventana es del tipo:

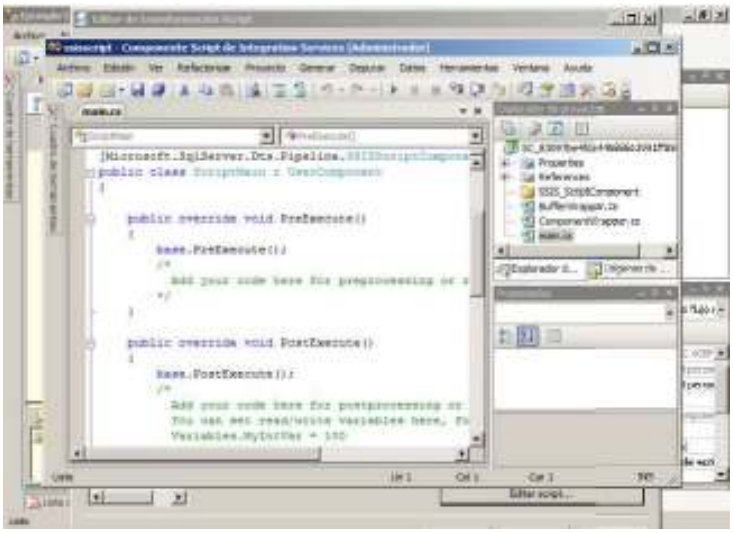

Figura 143. SSIS -Editar componente de script en Visual Studio

10. SSIS: RESTRICCIONES DE PRECEDENCIA.

Las restricciones de precedencia conectan contenedores y tareas de paquetes en un flujo de control ordenado. En el diseñador de BIDS, se representan mediante flechas que conectan las tareas y contenedores.

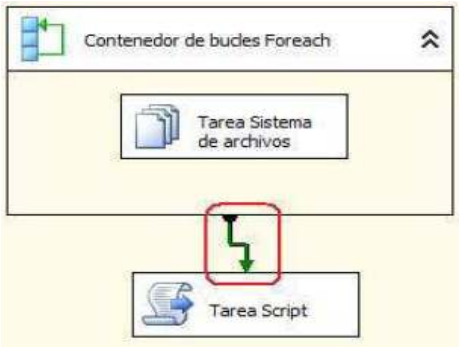

Figura 144. SSIS - Restricciones de precedencia

Haciendo click con el botón derecho del ratón sobre la flecha, se puede seleccionar la opción "***Editar***" en el menú de contexto. Esto nos permite cambiar las propiedades de la restricción de precedencia y configurarla a nuestro gusto.

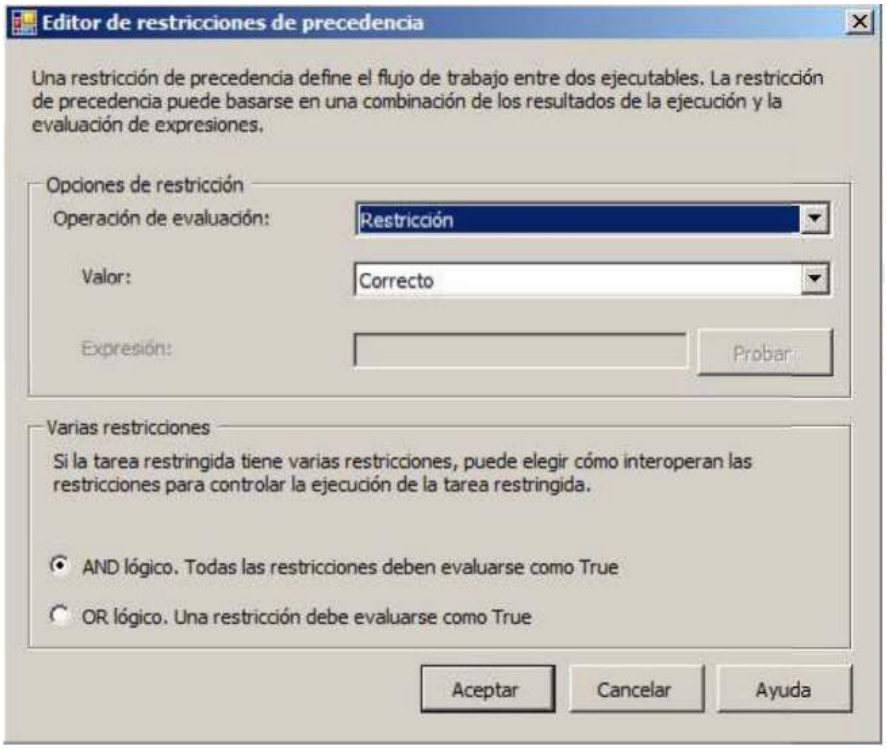

Figura 145. SSIS - Editar restricciones de precedencia.

Una restricción de precedencia vincula dos ejecutables (o tareas) que serán:

1) El ejecutable o tarea de **precedencia**: Que se ejecuta siempre en primer lugar.
2) El ejecutable o tarea **restringido**: Que se puede ejecutar siempre o dependiendo del resultado de la ejecución de la tarea de precedencia.

En el gráfico se ve que el ejecutable de precedencia apunta hacia el restringido. Se pueden configurar las restricciones de la siguiente manera:

- **Especificar una operación de evaluación**.
  La restricción de precedencia usa un valor de restricción, una expresión, ambas ellas o una de las dos para determinar si se ejecuta la tarea restringida.

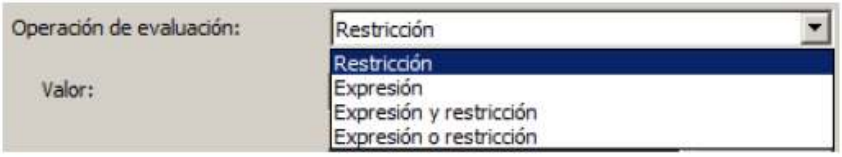

Figura 146. SSIS - Configurar restricciones de precedencia I

- **Depender del resultado de ejecución**.
  Si la restricción de precedencia usa un resultado de ejecución, puede especificar el resultado de ejecución para que sea correcto, de error o de conclusión del paquete.

Figura 147. SSIS - Configurar restricciones de precedencia II.

- **Usar una expresión**.
  Si la restricción de precedencia usa un resultado de evaluación, puede proporcionar una expresión que se evalúa como un valor booleano.

Figura 148. SSIS - Configurar restricciones de precedencia III

- **Especificar si la restricción se evalúa individualmente o forma parte de un conjunto.**

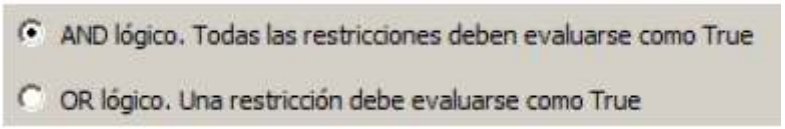

Figura 149. SSIS - Configurar restricciones de precedencia IV

En muchos casos no es necesario configurar las restricciones, simplemente nos limitamos a conectar las flechas verdes en el diseñador de BIDS, con lo que ya queda configurada la tarea de destino para ejecutarse si el resultado de la ejecución de la tarea anterior es correcto.

Este es el comportamiento predeterminado y el que normalmente se necesita, pero también podemos utilizar paquetes de contingencia que serán ejecutados cuándo ciertas tareas se hayan ejecutado con error para por ejemplo devolver la marca de procesado a una fase inicial o revertir acciones (más adelante veremos las transacciones). También se pueden usar estas tareas que se ejecutan con error para notificar de ello, por ejemplo, a través de correo electrónico a los responsables de la ejecución.

11. SSIS: PUNTOS DE COMPROBACIÓN O CHECKPOINTS

Gracias a los checkpoints o puntos de control, SSIS puede reiniciar un paquete en el que se haya producido un error desde el punto exacto dónde se produce el error. La existencia de los puntos de control nos permite ejecutar el paquete desde el error en vez de volver a ejecutar todo el proceso.

Si se configura un paquete para que utilice puntos de comprobación, la información relacionada con la ejecución del paquete se escribe en un **archivo de punto de comprobación**.

Cuando se vuelve a ejecutar el paquete después de haber ocurrido el error, se utiliza el archivo de punto de comprobación para reiniciar el paquete desde el punto en el que ocurrió el error.

Si el paquete se ejecuta correctamente, el archivo de punto de comprobación se elimina y se vuelve a crear la siguiente vez que se ejecuta ese paquete.

La ventaja de los puntos de comprobación es que nos pueden evitar repetir operaciones costosas que ya estaban completadas, en caso de que falle una de las operaciones posteriores. Algunos ejemplo de utilidad serían los siguientes:

- Evitar que se repita la descarga y la carga de archivos grandes. Por ejemplo, un paquete que descarga varios archivos grandes mediante una tarea FTP para cada descarga, se puede reiniciar cuando la descarga de uno de ellos genera un error, de forma que no haya que volver a empezar desde cero y volver a descargar archivos ya descargados.
- Evitar repetir la carga de grandes cantidades de datos.
- Evitar repetir la agregación de valores.

Para configurar los checkpoints o puntos de comprobación, en BIDS se hace click con el botón derecho del reatón sobre cualquier zona del fondo del área de diseño del flujo de trabajo. En el menú de contexto elegimos "***Propiedades***" y en la ventana de propiedades se configuran los valores referentes a los checkpoints.

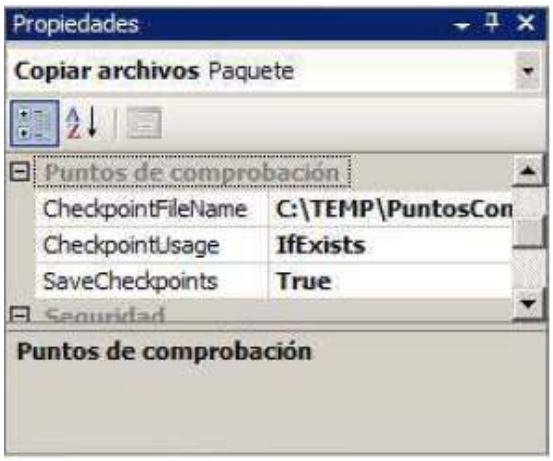

Figura 150. SSIS - Checkpoints I.

Además, hay que establecer el valor de **True** para la propiedad ***FailPackageOnFailure*** de todos los contenedores del paquete que se desea que funcionen como puntos de reinicio.

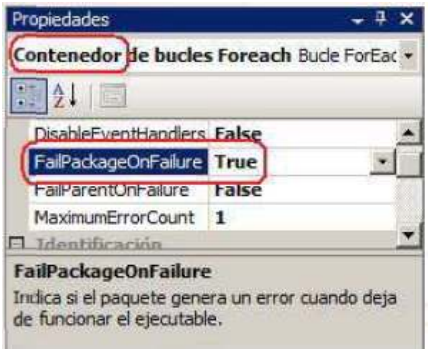

Figura 151. SSIS - Checkpoints II.

Para comprobar cómo funcionan los puntos de reinicio, se puede utilizar la propiedad ***ForceExecutionResult*** de las tareas o contenedores. Si se pone el valor **Failure** se ocasiona un fallo de la tarea o contenedor y de esa forma podemos probar cual es el comportamiento que sucede al reiniciar el paquete.

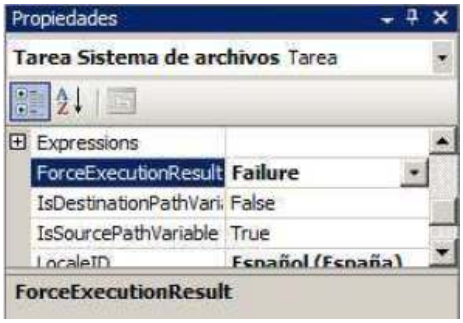

Figura 152. SSIS - Checkpoints III.

12. SSIS: DEPURACIÓN DEL FLUJO DE CONTROL.

Para facilitar la depuración del flujo de control, se pueden colocar puntos de interrupción en el flujo de control para que se detenga el proceso cuando se cumpla una condición determinada y podamos seguir el progreso del mismo.

Para poner un punto de interrupción desde el diseñador de SSIS, se acude a la ficha Flujo de control y se hace click con el botón derecho. En el menú de contexto se elige la opción "***Editar puntos de interrupción***". También se puede realizar la misma operación sobre un contenedor.

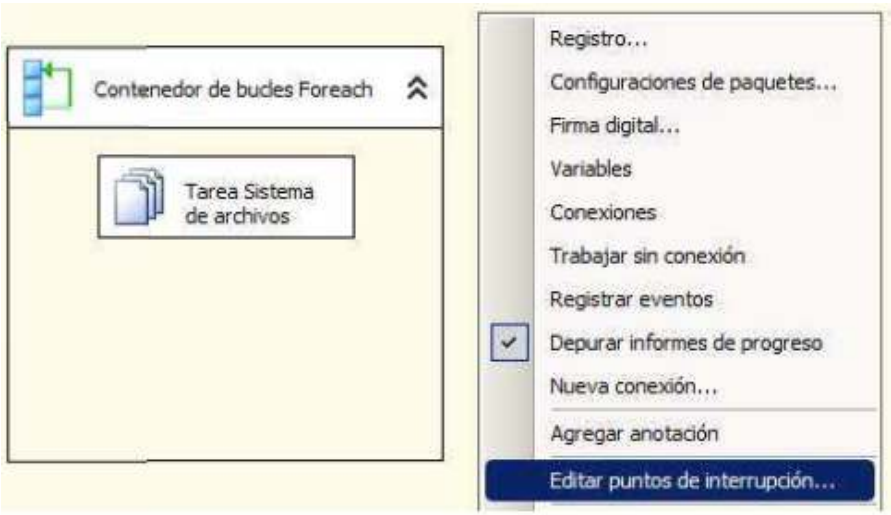

Figura 153. SSIS – Depuración del flujo de control.

En la ventana que aparece se pueden establecer las condiciones bajo las que se producirá la interrupción. Tenemos que darnos cuenta de que se puede indicar un recuento, por ejemplo, establecer un número n para parar la ejecución cada n iteraciones de un bucle.

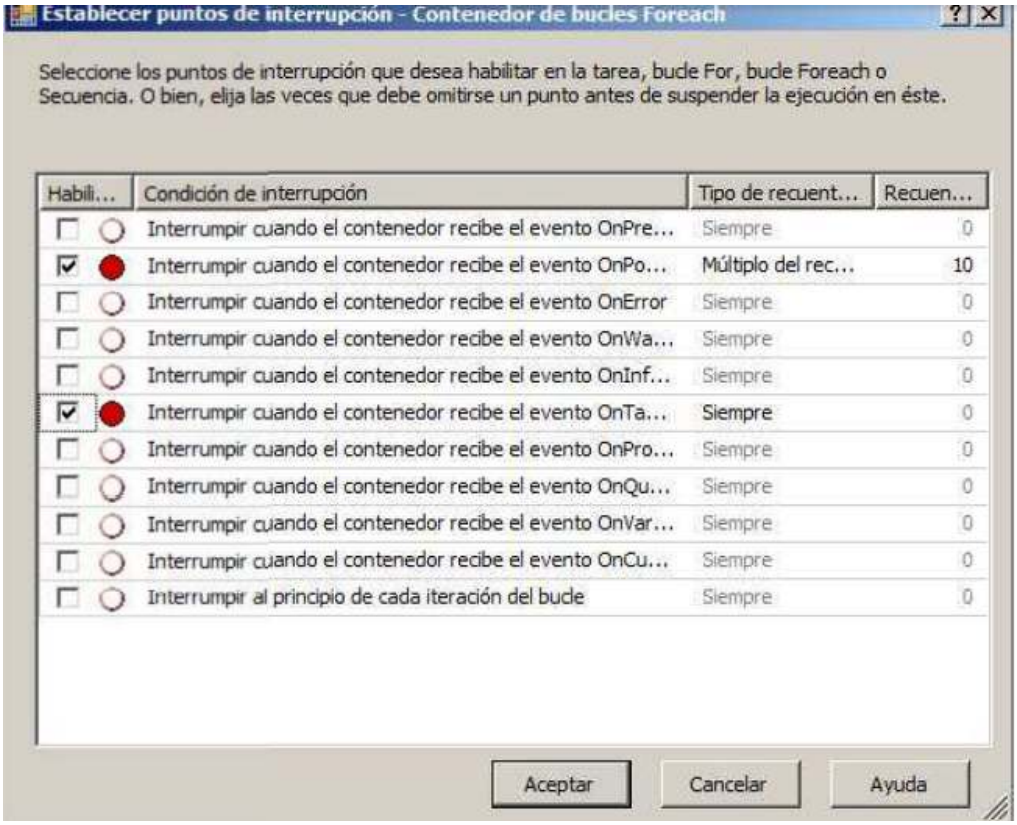

Figura 154. SSIS - Puntos de interrupción.

13. SSIS: TRANSACCIONES.

Los paquetes pueden utilizar transacciones para agrupar las acciones de base de datos realizadas por las tareas en unidades atómicas y así mantener la integridad de los datos.

Como todas las acciones de la base de datos que forman parte de una transacción se confirman o se revierten juntas, el uso adecuado de las transacciones nos permite garantizar que los datos permanecen en un estado coherente.

Estos son algunos de los objetivos para los que pueden resultar útiles las transacciones:

- Agrupar los resultados de varias tareas en una sola transacción para asegurar la coherencia de las actualizaciones.
- Garantizar la actualización coherente en varios servidores de base de datos.
- Garantizar las actualizaciones en un entorno asíncrono. Por ejemplo, un paquete podría utilizar una tarea de *Cola de Mensajes* para leer y eliminar un mensaje que contenga el nombre de un archivo que se desea cargar. Si se produce un error en la tarea que carga el archivo, al revertir la transacción se descartan los cambios realizados en la base de datos y se vuelve a colocar el mensaje en la cola.

Todos los tipos de contenedor de SSIS, incluyendo el propio paquete, se pueden configurar para que utilicen transacciones. Hay tres opciones para configurar transacciones que son:

1) ***NotSupported.***La tarea no participaría en la transacción.
2) ***Supported.***La tarea participaría en la transacción en caso de que haya sido previamente iniciada por otro componente. Es la opción por defecto.
3) ***Required***. La tarea obliga a iniciar una transacción en caso de que no esté aún iniciada.

Se configuran desde la ventana de propiedades:

Figura 155. SSIS – Transacciones.

Puede verse que también se puede establecer el nivel de aislamiento de la transacción. Los niveles disponibles corresponden con los que admite la directiva SET TRANSACTION ISOLATION LEVEL de Transact-SQL.Para que el paquete utilice transacciones, hay que configurar al menos un componente con el valor de **Required**. A partir de ahí, los sucesivos componentes configurados como **Supported** formarán parte de la transacción.

14. CONFIGURAR LOS COMPONENTES DEL FLUJO DE DATOS DE SSIS

Acabamos de ver a grandes rasgos los componentes del flujo de datos y un poco más en detalle algunas transformaciones importantes, pero vamos a detallar más la operativa natural que se realiza al configurar un flujo de datos.

Los componentes de datos se pueden configurar a nivel de componente, en los niveles de entrada, salida y salida de error y en el nivel de columnas.

1) En el nivel de componente, se configuran propiedades que son comunes para todos los componentes y las propiedades personalizadas del componente.

Figura 156. SSIS - Flujo de datos - Configuración a nivel de componente.

2) En los niveles de entrada, salida y salida de error, se configurar las propiedades comunes de entradas, salidas y salida de error.

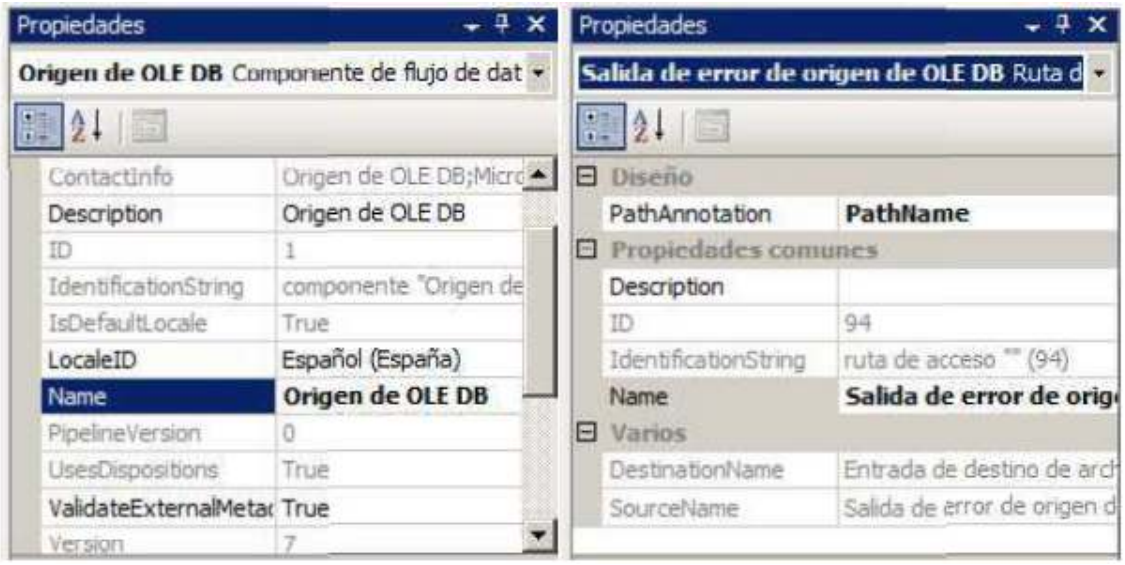

Figura 157. SSIS - Flujo de datos - Configuración a nivel de E, S y Serror.

3) En el nivel de columna se establecen las propiedades que son comunes a todas las columnas, además de cualquier propiedad personalizada que el componente proporciones para las columnas. Si el componente admite la adición de columnas de salida, puede agregar columnas a las salidas.

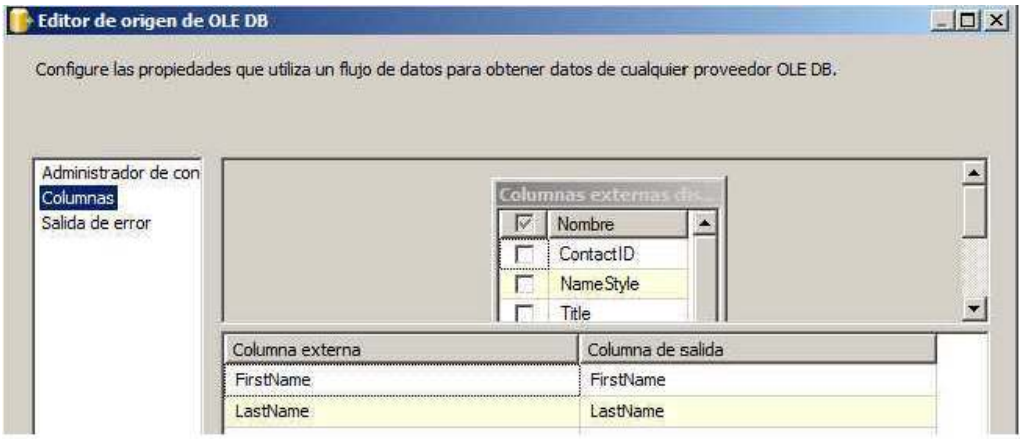

Figura 158. SSIS - Flujo de datos - Configuración a nivel de columna

Se pueden establecer las propiedades a través del diseñador de SSIS mediante cuadros de diálogo proporcionados para cada tipo de elemento o mendiante la ventana de "Propiedades" o en el cuadro de diálogo de "Editor Avanzado".

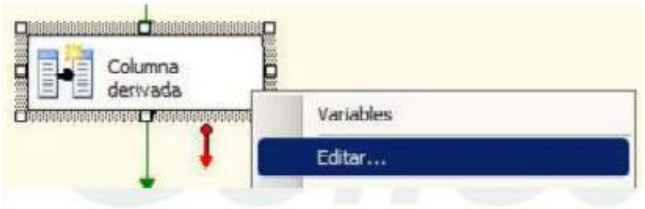

Figura 159. SSIS - Flujo de datos - Configuración a nivel de columna II

15. DEPURAR EL FLUJO DE DATOS DE SSIS.

Cuándo hablábamos del flujo de control indicábamos que nuestra principal herramienta para depurarlo era la introducción de puntos de interrupción y checkpoints ya que nos permiten seguir el flujo de la ejecución desde un componente dado. En el caso de los flujos de datos la depuración cambia, ya que el flujo no es la ejecución del paquete sino los datos a través de la tarea, que van pasando de un componente a otro. Por lo tanto, lo que nos interesa conocer a la hora de realizar la depuración es cuáles son los datos que están fluyendo a través de cada una de las salidas y entradas de cada componente del flujo de datos. Esta capacidad se consigue usando los **visores de datos**.

Para agregar un visor de datos a un flujo de datos, se hace click en el botón derecho del ratón en una ruta de acceso (flecha) entre dos componentes del flujo de datos y se selecciona la opción “Visores de datos”.

Figura 160. SSIS - Flujo de datos - Visores de datos.

Una vez seleccionada la opción, se muestra el editor de rutas de datos en el que tenemos que pulsar la opción “Agregar” en la pestaña de visires de datos para insertarlo.

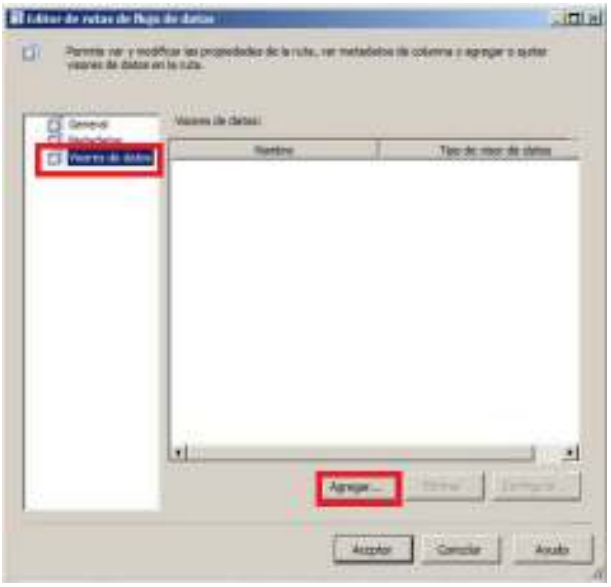

Figura 161. SSIS - Flujo de datos - Agregar visores de datos.

Una vez agregado el primer visor de datos aparecerá un cuadro de diálogo que nos permite configurar el tipo de visor que vamos a usar, establecer un nombre al mismo, etc.

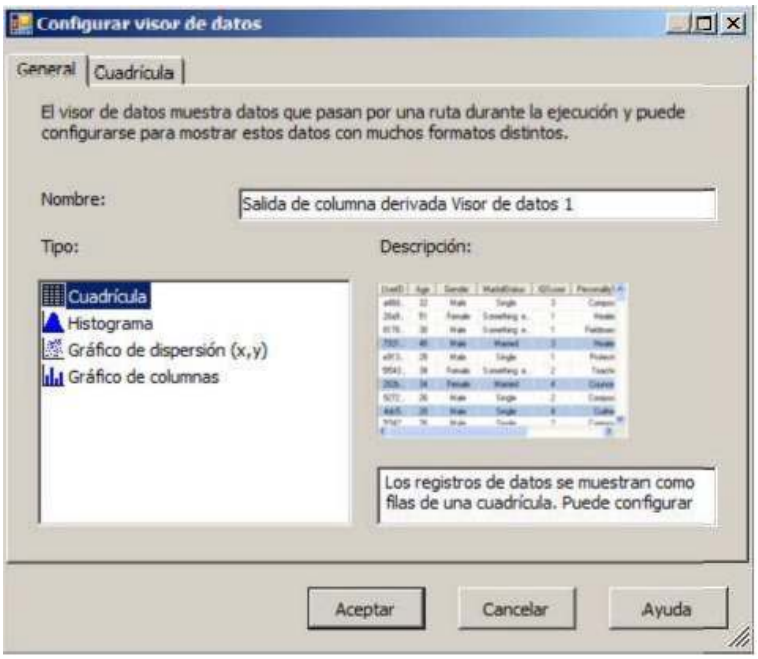

Figura 162. SSIS - Flujo de datos - Configurar visores de datos.

Seleccionando el tipo de visor y configurándolo, aparecerá una marca en la ruta de acceso o flecha entre los dos componentes seleccionada.

Figura 163. SSIS - Flujo de datos - Icono visor de datos

Por ejemplo, si hubiéramos configurado un visor de datos de tipo “cuadrícula”, el visor nos captura una rejilla con los datos que circulan por la ruta seleccionada. El resto de visores suelen mostrar información estadística que nos ayuda en tareas de calidad de datos.

Si ejecutamos el paquete, cuándo el flujo siga por la tarea de flujo de datos y concretamente cuándo pase por esa flecha aparecerá en el visor la información que deseamos, dónde podremos indicar que continúe el flujo o copiar los datos seleccionados para verificarlos.

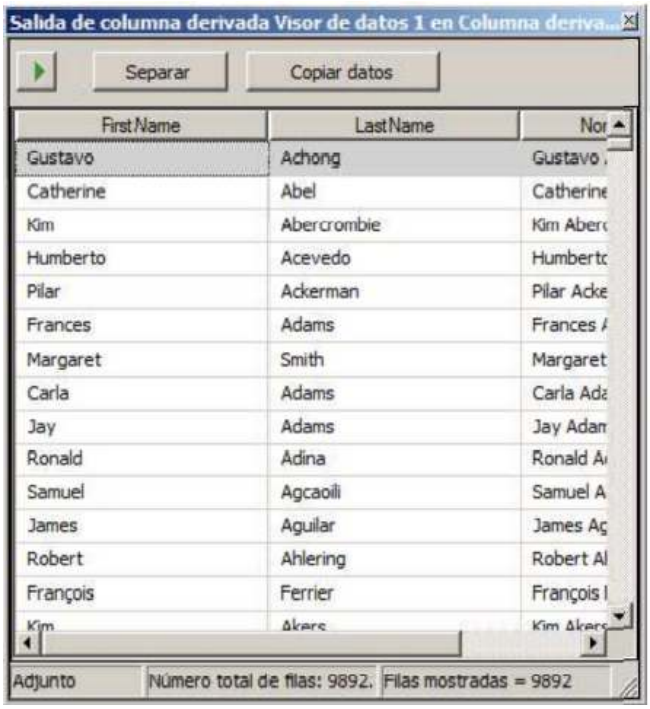

Figura 164. SSIS - Flujo de datos - Visor de datos de cuadrícula.

Gracias a ésta pantalla, podemos saber cuáles son exactamente los datos que están circulando por cada una de las rutas de acceso dibujadas en nuestro diagrama de flujo de datos y depurar el comportamiento no deseado.

16. EXPRESIONES EN SSIS

Durante las distintas fases hemos hablado de las expresiones que se pueden usar en las transformaciones, vamos a ver más en profundidad que son.

Una expresión es una combinación de símbolos (identificadores, literales, funciones y operadores) que realiza una serie de operaciones y devuelve un único valor.

Las expresiones más simples pueden consistir en una sola constante, variable o función. Otras veces, las expresiones pueden ser más complejas, con varios operadores y funciones, y hacer referencia a varias columnas y variables.

En SSIS, se pueden utilizar expresiones para definir condiciones para las instrucciones CASE, crear y actualizar valores de las columnas de datos, asignar valores a variables, actualizar o dar valor a propiedades en tiempo de ejecución, definir restricciones de precedencia y responder a las comparaciones que utiliza el contenedor de bucles for.

Estos son alguno de ejemplos de situaciones en los que podríamos usar expresiones:

- La transformación **división condicional** implementa una estructura de decisión basada en expresiones para dirigir filas de datos a diferentes destinos. El resultado de evaluar las expresiones usadas en una transformación de división condicional debe ser true o false (p.ej. Column1 > Column2).
- La transformación **columna derivada** utiliza valores creados mediante expresiones para llenar nuevas columnas en un flujo de datos o para actualizar columnas existentes. (p.ej. “nombre” + “ “ + “apellidos”).
- Las **variables** (podemos verlas en los anexos) utilizan una expresión para establecer su valor. (p.ej. VAR1 = GETDATE()).
- Las **restricciones de precedencia** pueden usar expresiones para especificar las condiciones que determinan si se ejecuta la tarea o el contenedor restringido de un paquete. El resultado de evaluar las expresiones usadas en una restricción de precedencia debe ser true o false. (p.ej. @A > @B determina la condición de restricción).
- El **contenedor de bucles For** puede usar expresiones para egnerar las instrucciones de inicialización, evaluación e incremento utilizadas por la estructura de bucle. (p.ej. @Counter = 1).
- Etc.

Las expresiones también se pueden utilizar para actualizar los valores de las propiedades de paquetes, contenedores de los bucles For y ForEach, tareas, administradores de conexión, proveedores de registro y enumeradores ForEach. Por ejemplo, con una expresión se puede asignar la cadena “Localhost.AdventureWorks” a la propiedad ConnectionName de la tarea **Ejecutar SQL**, detalle muy importante cuándo tenemos que cambiar conexiones de paquetes en un entorno de desarrollo a entornos de explotación o producción.

Las expresiones se bana en un lenguaje de expresiones y en el evaluador de expresiones. El evaluador de expresiones analiza la expresión y determina si sigue las reglas del lenguaje de expresiones.

El generador de expresiones está disponible en los cuadros de diálogo “Editor” de la transformación de división condicional, de la transformación columna derivada. El generador de expresiones es una herramienta gráfica que permite generar de una forma muy sencilla expresiones. Presenta una serie de carpetas que contienen elementos específicos del paquete y otras carpetas que contienen las funciones, conversiones de tipo y operadores que proporciona el lenguaje de expresiones.

Los elementos específicos del paquete incluyen variables del sistema y variables definidas por el usuario. Por ejemplo, este es el aspecto del generador de expresiones que aparece al configurar la transformación de **columna derivada**:

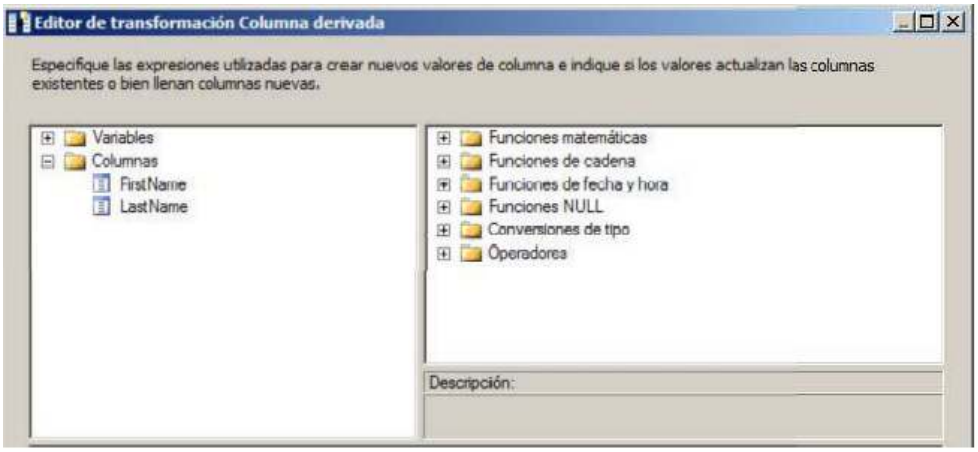

Figura 165. SSIS – Expresiones.

El editor se usa pinchando en los distintos elementos de la izquierda o en las funciones y operadores que se despliegan a la derecha, después se tiene que arrastrar al cuadro de texto en el que se está construyendo la expresión. Adicionalmente podemos escribir directamente la expresión.