
# *GESTIÓN DE PÓLIZAS DE SEGUROS: UN CASO PRÁCTICO DE BUSINESS INTELLIGENCE*

**UNIVERSIDAD CARLOS III DE MADRID**
**ESCUELA POLITÉCNICA SUPERIOR**

Ingeniería Informática
Proyecto Fin de Carrera

Autor: Silvia Rodríguez Mogollón
Tutor: Agapito I. Ledezma Espino

Diciembre, 2009

# Agradecimientos

En primer lugar quiero agradecer a Agapito Ledezma la ayuda que me ha prestado para realizar este proyecto que surgió de una idea que yo le presenté. Además, gracias por sus correcciones y su disponibilidad.

Debo agradecer a mi anterior empresa, GTBC por haberme dado el conocimiento necesario sobre Business Intelligence, que no sólo me han ayudado a hacer el proyecto fin de carrera, sino también a elegir mi futuro laboral. En especial quiero agradecerle a Rute Julio todo lo que me ha enseñado, haberme prestado los datos para el caso práctico de este proyecto, guiarme con el análisis y diseño, así como el programa utilizado para el proceso ETL, gracias Rute.

También quiero darles las gracias a mis padres por el esfuerzo que han hecho siempre para darme todas las facilidades para desarrollar mi carrera, y todo el cariño y apoyo que me han proporcionado para conseguir que llegue este momento. De igual forma, mis hermanos me han ayudado siempre y les agradezco el cariño y apoyo que suponen para mí. Y por supuesto al resto de mi familia, tíos y primos que siempre han estado ahí para apoyarme, en especial a mis abuelos que ya no están, y dónde estén estarán orgullos de lo que he conseguido gracias a su ayuda.

Quiero agradecer a mis compañeros, con los que he compartido muchos momentos, bueno y malos, agobios por prácticas que no acababan, exámenes que llegaban antes de tiempo, etc. Hemos estado muchos años juntos y nos hemos ayudado en todo lo que hemos podido, gracias Pili (mi gran compañera), Arturo, Fre, Bris, Javi, Jesús, María, Laura, Nieves, Alba, David, Diana, Laurita, y muchos más que me dejo en el tintero pero que también me acuerdo.

Y por último, pero no menos importantes a mis amigas, que no han ido conmigo a la universidad pero siempre han estado ahí para apoyarme, gracias Roci, Anita, Ana y Rosario.

# Índice de Contenidos

# Índice de Tablas

# Índice de Figuras

# Acrónimos

| ACRÓNIMO | DEFINICIÓN |
|---|---|
| BBDD | *Bases de Datos* |
| BI | *Business Intelligence* |
| BIR | *Business Intelligence Roadmap* |
| CASE | *Computer Aided Software Engineering* |
| DM | *Data Mart* |
| DSS | *Decision Suport System* |
| DW | *Data Warehouse* |
| ETL | *Extract, Transform & Load* |
| ODBC | *Open Database Connectivity* |
| ODS | *Operacional Data Store* |
| OLAP | *On Line Analytical Processing* |
| OLTP | *On Line Transacional Processing* |
| PFC | *Proyecto Fin de Carrera* |
| SI | *Sistema de Información* |
| SO | *Sistemas Operacionales* |

# Capítulo 1

## INTRODUCCIÓN

En este proyecto se van a tratar los sistemas *Business Intelligence* (BI). Para ello se describirá el origen de este y su estado actual. Se hablará de los Sistemas de Apoyo a la Decisión (DSS *Decision Support Systems*). Los DSS son herramientas BI enfocadas al análisis de los datos de una organización. Se hará ver la importancia de esos sistemas en las organizaciones y los beneficios que pueden aportar. Para comprender mejor este tipo de sistemas se hará una comparativa con los Sistemas de Información tradicionales, así como la metodología a seguir.

Cuando se desarrollan sistemas es muy importante saber qué pasos seguir y cómo hacerlo, y por esto se va a desarrollar la hoja de ruta para sistemas BI. Primero se hará una descripción teórica de cómo se debe hacer, y después se acompañará este conocimiento con un caso práctico. Este caso práctico consistirá en cubrir las carencias de una organización dedicada a las pólizas de seguros con la aplicación de BI. Durante este caso práctico se seguirán todos los pasos descritos previamente, así se podrá entender mejor el significado de cada uno de ellos, y cómo se debe hacer en la práctica.

La elección de BI como tema principal del proyecto se debe a su actual importancia e interés en las organizaciones. En estos momentos de crisis es vital que las organizaciones inviertan en nuevas tecnologías que les ayuden a seguir prosperando, y con *BI* pueden descubrir nuevas estrategias de negocio que les guíen hacia una nueva visión del mismo y a conseguir una mejor posición en el mercado al que pertenecen.

En la actualidad muchas empresas se muestran reticentes a invertir en este tipo de sistemas, lo ven como un gasto que no les aportará beneficios. Por este motivo, en este PFC se exponen claramente cuáles son los beneficios, y se demuestra que estas inversiones no sólo son un gasto para la organización que decide hacerlas.

Los sectores que más concienciados de los beneficios de este tipo de inversiones son Seguros, Banca y Telecomunicaciones. Casi todas las organizaciones de estos sectores incluyen aplicaciones BI como medio para incrementar su valor.

En el capítulo dos se describirán los objetivos de este Proyecto Fin de Carrera (PFC), dónde la principal finalidad es académica. Es importante que estos objetivos estén totalmente cubiertos al finalizar el trabajo realizado.

En el capítulo tres se detallarán todos los conocimientos necesarios del entorno que vamos a tratar (BI, DSS). Aquí se dejará muy clara la finalidad de este tipo de sistemas y sus peculiaridades con respecto a los sistemas clásicos, ya que de no ser así los siguientes capítulos no serían muy útiles.

Habrá un capítulo dedicado al desarrollo de un caso práctico, el capítulo cuatro, será bastante extenso y con él quedarán muy claros los pasos a seguir en un sistema de este entorno, qué se debe a hacer en cada uno de ellos y cómo. Este caso práctico se hará sobre una compañía de seguros no vida. La elección de este sector se debe a que es un ejemplo muy representativo de aplicación BI en las organizaciones. La mayoría de las grandes aseguradoras utilizan este tipo de sistemas para potenciar la captación de clientes, su satisfacción y mantenimiento, descubrir nuevos productos o servicios de gran impacto en el mercado, y más actividades que hacen que sus organizaciones ganen mucho valor.

Por último se incluirán capítulos para mostrar y analizar las conclusiones y futuras líneas. En esos apartados se demuestra que se han logrado los objetivos iniciales propuestos, se comentan las conclusiones obtenidas del proyecto y las posibles futuras líneas que se sugieren.

# Capítulo 2

## OBJETIVOS DEL PFC

En este proyecto se van a contemplar dos objetivos generales. Uno de ellos es estudiar el concepto de BI, para el cual será necesario que se cumplan los siguientes objetivos específicos:

- Describir los conceptos más importantes de estos sistemas.
- Describir los conocimientos necesarios para ser capaz de reconocer cuando es beneficioso utilizar este tipo de sistemas, y de desarrollarlos.
- Explicar de forma extensa la hoja de ruta a seguir en este tipo de sistemas.

El segundo objetivo general es desarrollar un caso práctico de aplicación de BI, el cual debe seguir todos los pasos establecidos como básicos para este tipo de sistemas. Este caso práctico será sobre una hipotética compañía de seguros, y para desarrollarlo con éxito es necesario que se cumplan los siguientes requisitos específicos:

- Proporcionar información sobre el ratio de la prima y el ratio del número de pólizas por dirección regional por año.

- Proporcionar información sobre el ratio de la prima y el ratio del número de pólizas por producto, sector y año.
- Estudiar la predicción del incremento de las primas de las pólizas por semestre y por trimestre, con la finalidad de evaluar los resultados y decidir si son aceptables o no.

Todos estos objetivos, tanto generales como específicos, deben cumplirse al finalizar el proyecto, por tanto es necesario que en los resultados y conclusiones así se refleje.

En todos los proyecto es muy importante la clara especificación de requisitos y describir objetivos específicos, dentro de uno general, hace que la consecución de éstos implique la del general al que pertenecen.

# Capítulo 3

# ESTADO DEL ARTE

En este capítulo se describen los sistemas basados en BI, dónde se incluirá su origen, la evolución que han sufrido y el estado actual en el que se encuentran. Se pondrá especial atención en las metodologías seguidas en este tipo de proyectos y cómo han evolucionado desde los sistemas de información clásicos.

A principios de los años noventa Howard Dresner (vicepresidente y director e investigaciones del Grupo Gartner) utilizó el término BI para agrupar las herramientas de búsqueda y reportes, herramientas DSS y las de procesamiento Analítico en Línea (*Online Analytical Processing, OLAP*), [6]. Las organizaciones que usan estas herramientas obtienen unos resultados que añaden ventaja competitiva a su negocio.

Según Dresner, *"el BI cambia la habilidad de la organización para entender el negocio, aprovecha la ventaja de nuevas oportunidades y, literalmente, cambia sus procesos de negocio, mejorando su competitividad y eficiencia. El BI es la habilidad en el acceso y análisis de la información de tal manera que incremente la posibilidad de realizar una mejor toma de decisiones en los negocios"* ([6]). En 2001 Dresner amplía estas definiciones en [7]: *"business intelligence es simplemente la habilidad de los usuarios finales para acceder y analizar tipos cuantitativos de información y ser capaz de actuar en consecuencia".*

## 3.1 ¿Qué es Business Intelligence?

Muchos autores hacen referencia a este término y dan su propia definición, pero todos se dirigen hacia el mismo punto, los sistemas BI incluyen las herramientas necesarias para dar soporte a la decisión, permitiendo el análisis y manipulación de información corporativa, con un acceso interactivo en tiempo real.

El permiso a los usuarios para acceder a una gran cantidad de datos hace incrementar su capacidad de decisión. Es muy importante obtener todo el conocimiento que posee una organización en sus datos. Sin las herramientas y tecnología adecuada esto resulta muy difícil. En los DSS es imprescindible disponer de todo el conocimiento acerca de la organización para poder tomar las mejores decisiones, que aporten ventaja competitiva a medio y largo plazo. Así pues, surge la necesidad de implantar BI en las organizaciones para poder conseguir estos objetivos.

El BI está más cerca de ser un producto que un sistema. Este es una arquitectura y una colección de sistemas operacionales integrados, como aplicaciones de soporte a la decisión y bases de datos, que proveen a la comunidad del negocio un acceso fácil a los datos del negocio.

Desde los inicios de la era de las computadoras, las organizaciones han cubierto sus necesidades de información desde sus sistemas operacionales. Los métodos usados para el acceso y tratamiento de los datos han evolucionado en el tiempo, pero todavía existen muchas organizaciones que utilizan datos no limpios e inconsistentes como apoyo para la toma de decisiones importantes. Además, en muchos casos, se toman decisiones basadas en el análisis y la relación de un gran volumen de datos no del todo fiables.

Los problemas de inconsistencia, falta de limpieza y fiabilidad en los datos normalmente son detectados por el equipo de dirección de una organización, porque es este quien solicita informes a sus colaboradores con datos claves de la compañía y el entorno competitivo. La recopilación de esta información puede ser imposible, estar incompleta o ser incongruente. Así pues, los altos directivos acabarán decidiendo con información de escasa confianza o incompleta.

Es muy importante tener acceso a todo el conocimiento existente, pero esto a veces resulta imposible debido a la estructura y arquitectura del modelo de datos. Para solucionar este problema se crean los *Data Warehouse* (DW), que se explicará con detalle más adelante. La creación de un DW es una de las tareas más importantes en

Sistemas BI, ya que depende totalmente de las necesidades de negocio, y a su vez, de él depende que puedan cubrir estas necesidades.

## 3.2 Data Warehouse y BI

La mayoría de las organizaciones realiza labores con el fin de conseguir información adecuada, pero estas acciones no son suficientes, porque la calidad de la información también depende del software y del hardware.

Con el objetivo de dar apoyo y solución a estos problemas surge el DW, que reúne y organiza grandes volúmenes de datos provenientes de las diversas unidades que contienen todos los datos. Además, asegura que los datos estén disponibles con la flexibilidad y velocidad necesarias. Con esto se puede decir que es una súper base de datos integrada.

Más adelante se comentarán las características, arquitectura, ventajas, etc. de los DW, pero antes se debe entender que la implantación de éste, no resuelve problemas por sí sólo, simplemente proporciona, los datos necesarios para tomar las mejores decisiones.

Véase en la Tabla 1 un contraste entre los datos almacenados en un sistema operacional y un depósito de datos DW.

| BBDD Operacional | Data Warehouse |
|---|---|
| Datos Operacionales | Datos del negocio |
| Orientado a la aplicación | Orientado al sujeto |
| Actual | Actual e histórico |
| Detallada | Detallada y más resumida |
| Cambia continuamente | Estable |

***Tabla 1.*** *Comparación BBDD frente a DW.*

El origen de los datos en un DW es, en la mayoría de los casos, el entorno operacional. Los datos de un DW están siempre transformados y separados físicamente de la aplicación de dónde proceden.

Los DW son transversales a toda la organización a la que pertenecen, y si esto no se cumple es difícil que se puedan cubrir perfectamente todas las necesidades. Los objetivos estratégicos generales de negocio son comunes a toda la organización, independientemente del departamento, área o sección. En una organización es normal encontrar diferentes sistemas según la actividad, función, procedimientos, etc., que realizan, pero todos ellos deben estar en perfecta armonía para conseguir los objetivos generales. Es por esto que todos los sistemas pertenecientes al conjunto deben tener visión sobre el DW. Algunos de estos sistemas alimentaran los datos del DW y otros los consultarán. Además, cuanto mayor sea la organización y mayor complejidad contenga su actividad más número de sistemas la formarán.

Los sistemas de información en las organizaciones se han dividido en cuatro niveles, que se detallan a continuación:

- Sistemas Estratégicos: estos sistemas están enfocados a facilitar la labor de la dirección, soportando la toma de decisiones. Proporcionan mejor información con el fin de que se tomen las mejores decisiones. Una característica de estos sistemas es que no se produce una utilización periódica, las peticiones al sistema no suelen ser predecibles. Dentro de este grupo destacan sistemas como “Sistemas de Información Gerencial”, “Sistemas de Información Ejecutivos”, “Sistemas de Simulación de Negocios (Sistemas Expertos o de Inteligencia Artificial)”.

- Sistemas Tácticos: en este nivel, los sistemas están diseñados para soportar actividades de coordinación y manejo de documentación, con el fin de facilitar las consultas realizadas sobre la información almacenada en el sistema. En esta capa de la organización se generan informes para facilitar el trabajo de los gestores intermedios de la organización y realizar una gestión independiente de la información.

- Sistemas Operacionales: estos sistemas cubren las necesidades de operaciones básicas de obtención masiva de datos y su tratamiento básico, según las tareas predefinidas. Estos sistemas están avanzando gracias a la evolución del mercado, sensores, autómatas, sistemas multimedia, bases de datos relacionales más avanzadas y DW. Los sistemas operacionales (SO)

son el pilar de toda organización y debido a esto y a su volumen han sido extendidos, revisados y mejorado considerablemente.

- Sistemas Interinstitucionales: este tipo de sistemas está recién instaurado. Forma parte del nivel más bajo de la organización. Surgen a partir de la necesidad creada al enfocarse las organizaciones desde un punto de vista mucho más global que obliga a las organizaciones a establecer una comunicación estrecha entre ellas y el mercado.

Cuando una organización tiene la necesidad de implantar un DW, debe tener conciencia de que éste estrechará la relación entre los sistemas operacionales y tácticos. No obstante, es imprescindible que existan procesos que integren perfectamente los sistemas de todos los niveles. La finalidad de crear un DW en una organización es poder obtener todo el conocimiento necesario para tomar las mejores decisiones y diseñar las mejores estrategias de negocio. Para que estas estrategias se puedan llevar a cabo con éxito, todos los niveles deben mirar hacia el mismo sitio. Es decir, en todos los sistemas tienen que estar bien definidos los objetivos, tanto globales como locales al nivel, y orientar los procesos a la consecución de los mismos, de tal forma que los procesos de integración sean transversales a la organización, como se muestra en la Figura 1.

La principal característica de un DW es que está orientado al tema. Los clásicos sistemas operacionales están orientados a la aplicación y funciones. De este modo, una organización financiera clasificaría los datos según aspectos como pueden ser préstamos, ahorros, tarjeta bancaria y depósito.

En un DW los datos son estructurados según los aspectos que son de interés en la organización. Por ejemplo, los temas para un fabricante pueden ser cliente, productos, proveedores y vendedores; para una entidad financiera cliente, vendedor, producto, actividad; para una universidad estudiantes, clases y profesores; para un hospital pueden ser pacientes, personal médico, medicamentos, etc. Así pues, el diseño de las bases de datos y la implementación de las mismas se ven afectados por la alineación del contexto de las áreas de los temas.

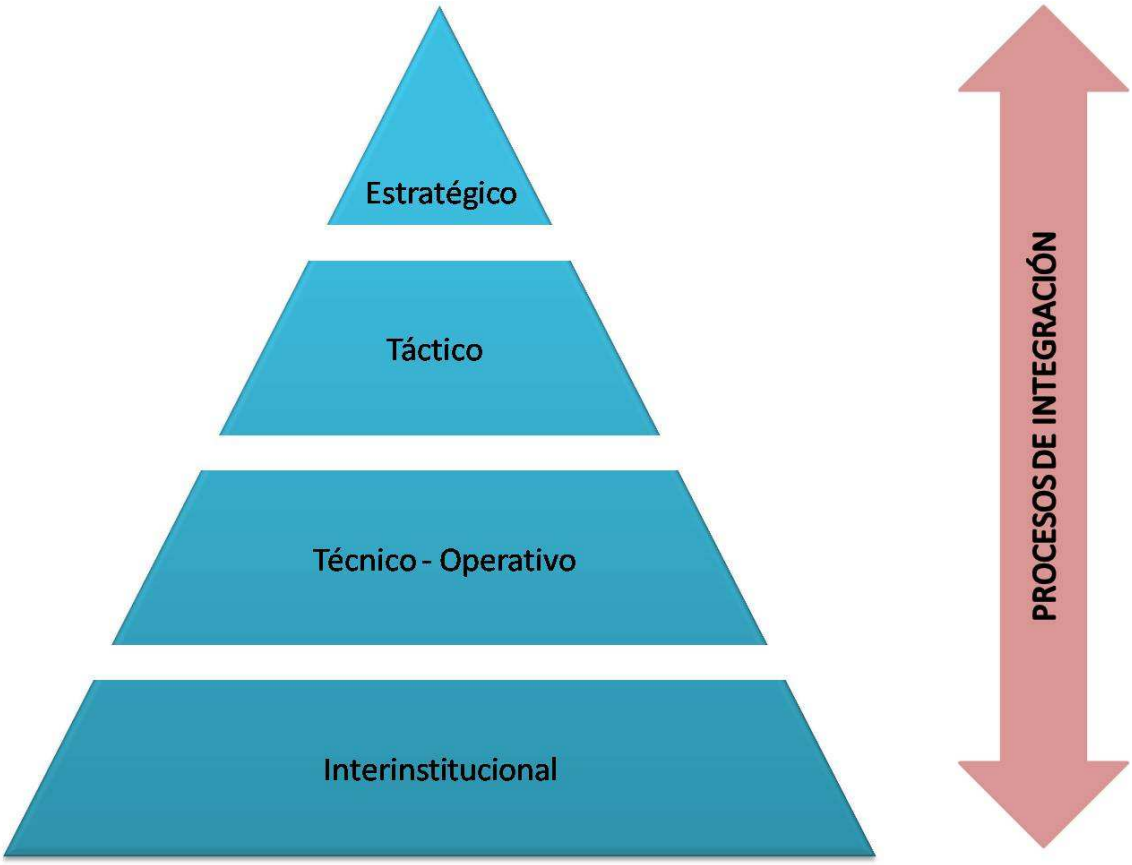



***Figura 1.*** *Pirámide Procesos de Integración.*

La principal diferencia entre estos dos enfoques radica en el contenido detallado de los datos. Un DW excluye la información que no usarán los sistemas de apoyo a la decisión. Por otra parte, los SO contienen toda la información necesaria para satisfacer inmediatamente los requisitos funcionales y de proceso, aunque esta no sea utilizada por los analistas de soporte de decisiones. Esta característica hace que el acceso y el entendimiento de los datos sean sencillos para los usuarios finales

Otro aspecto muy importante de los DW es que la información debe estar siempre integrada. La integración de datos se manifiesta en muchos aspectos, entre los cuales se encuentran los siguientes:

- Codificación: los diferentes diseñadores de aplicaciones, dentro de una misma organización, codifican un campo con distintas formas. Por ejemplo, uno codifica el campo GENERO como “M y F”, otro como “0 y 1”, otro como “X e Y”, etc. No es importante como llega el género al DW, pero debe llegar en un estado uniforme integrado, habiendo un único formato final en el DW.

- Medida de atributos: cada diseñador utiliza la medida de un atributo como mejor conviene a su aplicación, pero en un DW todas las medidas de un atributo deben llegar en las mismas unidades, independientemente de la fuente de la que proceda.

- Convenciones de nombrado: se refiere a un elemento por nombres diferentes en las distintas aplicaciones. En el DW se debe usar el mismo nombre siempre que se refiera al mismo elemento concreto.

- Fuentes múltiples: un elemento puede derivarse desde diferentes fuentes. El proceso de transformación debe asegurarse que el elemento se deriva de la fuente apropiada.

En un DW se almacenan históricos en los que el tiempo está implícito en la información que contiene. La información almacenada en un DW permite hacer análisis de tendencias. Así pues es necesario que se guarden los distintos valores que una variable ha tomado en el tiempo, lo que permitirá futuras comparaciones.

Por último, se debe destacar que un DW no es volátil. La información contenida en este almacén no puede ser modificada, es permanente. En un SO, si el valor de una variable se ve modificado se produce esta modificación en la BBDD, perdiendo así la información sobre qué valor tenía anteriormente. En un DW se realiza una nueva inserción en el almacén de datos, incluyendo el tiempo de validez del valor actual y anterior.

Existen varios tipos de arquitecturas para un DW dependiendo de las necesidades de la organización. Estos tipos son los siguientes:

- De una capa: este caso el DW se alimenta directamente de los sistemas operacionales. Se puede ver en la Figura 2.

- De dos capas: este caso el DW no se alimenta directamente de los sistemas operacionales. Se crea el *Staging Area*, que es un sistema intermedio entre el DW y el SO que contiene datos derivados del SO. Es la arquitectura más común, pues la carga del DW requiere operaciones dinámicas y pesadas, y éstas hacen que se sature el SO. Dentro de esta arquitectura hay dos modalidades, una centralizada y otra descentralizada, que se pueden observar en las Figuras 3 y 4 respectivamente. En la arquitectura centralizada, sólo existe un DW mientras que la descentralizada está formada por varios *Data Mart*, DM. Un DM es un subconjunto del DW que tiene

características similares al DW pero es de menos tamaño y está especializado en un área de negocio específica. Un DM tiene poco volumen de datos y las consultas son más rápidas y sencillas.

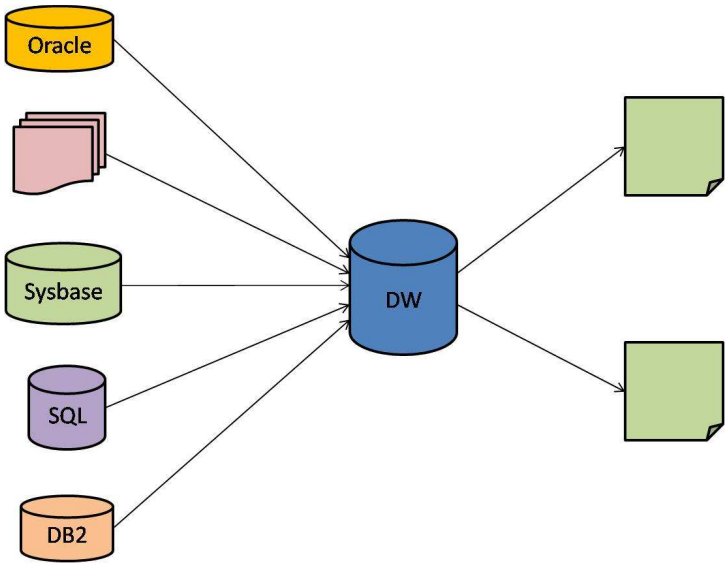



***Figura 2.*** *Arquitectura de un DW de una capa.*

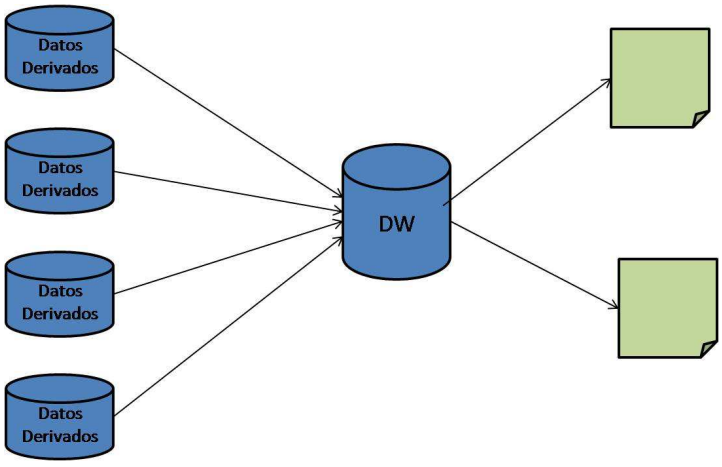



***Figura 3.*** *Arquitectura de un DW de dos capas, Centralizada.*

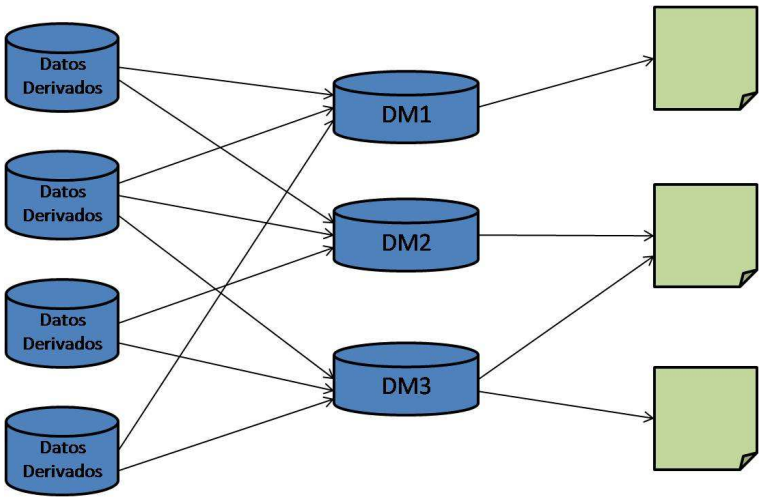



***Figura 4.*** *Arquitectura de un DW de dos capas, Descentralizada.*

- De tres o más capas: Esta arquitectura amplia la de dos capas centralizada. Añade DM departamentales y da la opción de utilizar o no ODS (*Operacional Data Store*). Este nuevo elemento forma parte del *Staging Area* y se usa para unificar datos. Sin embargo se debe tener en cuenta que sólo guarda los históricos temporalmente. Así pues este tipo de arquitectura tiene dos modalidades, sin ODS o con ODS, y se pueden observar en las Figuras 5 y 6 respectivamente.

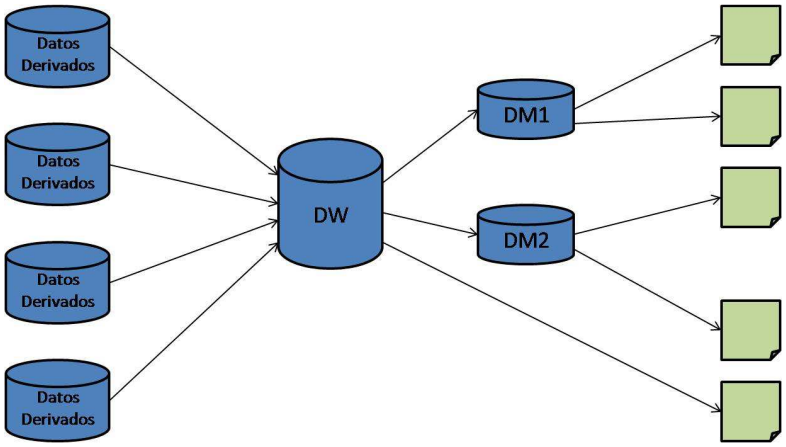



***Figura 5.*** *Arquitectura de un DW de tres o más capas sin ODS.*

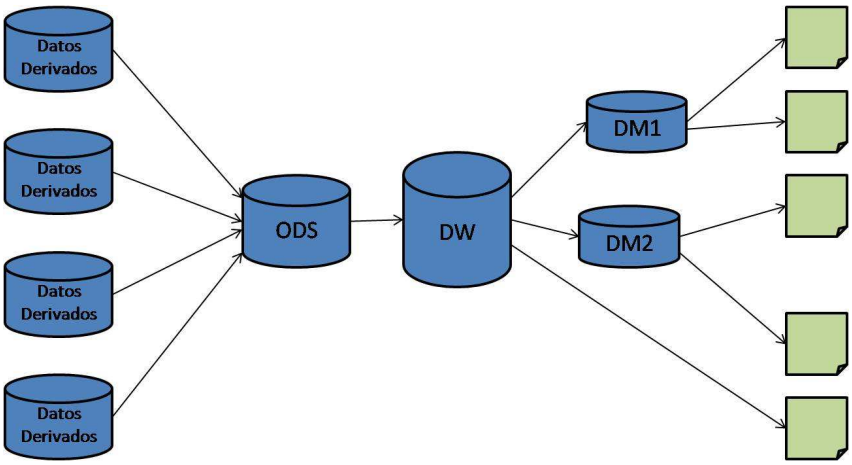



***Figura 6.*** *Arquitectura de un DW de tres o más capas con ODS.*

## 3.3 Elementos de Sistemas BI

Se debe tener en cuenta que los Sistemas BI de apoyo a la decisión no sólo están formados por un DW, también nos encontramos con más elementos necesarios para desarrollar y explotar los Sistemas BI. Sin las herramientas, conocimientos y habilidades adecuadas el Sistema no saldrá adelante aunque se haya creado un DW excelente. Es necesario que se incluyan los siguientes elementos:

- Aplicaciones BI: son las de más alto nivel, se encargan de presentar la información de forma personalizada a los usuarios, basándose en componentes de menor nivel dentro de la arquitectura BI. En este conjunto se incluyen las herramientas DSS de elaboración de reportes, OLAP, *data mining,* etc.

- Habilidades de acceso: contemplan las aplicaciones de interfaz, que permiten un acceso transparente a los datos almacenados.

- Manejo de datos: se define según los sistemas que almacenan la información físicamente. Son el pilar del BI, ya que todas las aplicaciones, sistemas, accederán a los datos. Aquí se incluye el DW, las herramientas para la modelación y construcción del propio DW.

Para comprender mejor el punto de actuación del BI dentro del flujo de información dentro de una organización, véase la Figura 7.

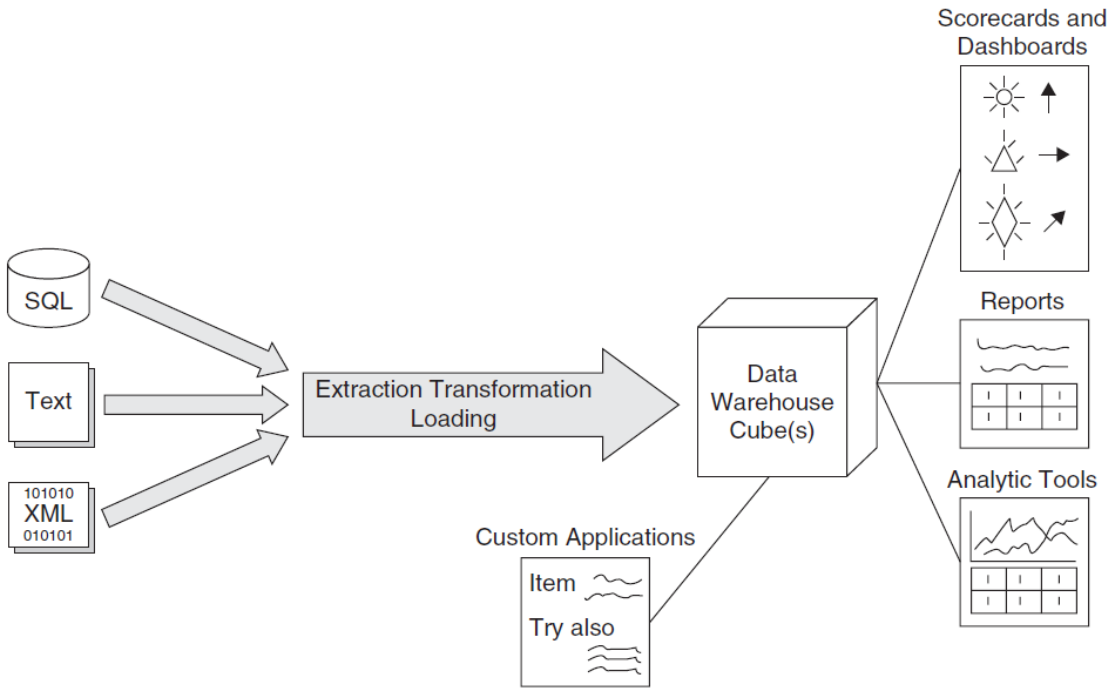



***Figura 7.*** *Flujo de Información dentro de una organización con Solución BI [22].*

Las iniciativas BI como soporte a la decisión son un esfuerzo caro. Los datos de negocio dispares deben ser extraídos y fusionados desde los sistemas de procesamiento transaccional en línea (OLTP, *On Line Transaccional Processing*), desde sistemas *batch* y desde orígenes de datos externos agrupados. Estas iniciativas, también llamadas nueva tecnología, consideran que se deben llevar a cabo otras tareas adicionales, como pueden ser especificación de roles y responsabilidades, y proporcionar aplicaciones de análisis y soporte a la decisión rápidas que mantengan una calidad aceptable.

## 3.4 Ciclo de Vida de Sistemas BI

Casi todos los tipos de proyectos de ingeniería pasan por 6 estados entre el inicio y la implementación del mismo. Estos estados son los siguientes:

1. Justificación: Evaluación de la necesidad de negocio que requiere el nuevo proyecto de ingeniería.

2. Planificación: Definir la estrategia y planes tácticos a seguir para desarrollar y llevar a cabo el proyecto con éxito.

3. Análisis de Negocio: Llevar a cabo un análisis detallado del problema de negocio u oportunidades de negocio para conseguir una comprensión sólida de los requisitos de negocio necesarios para llegar a una posible solución o producto.

4. Diseño: concebir un producto que resuelve el problema de negocio, o permite la oportunidad de negocio.

5. Construcción: construir el producto que ofrece una respuesta a la inversión, dentro de un periodo de tiempo.

6. Desarrollo: implementación o venta del producto acabado, y entonces medir su efectividad para determinar si la solución encuentra, excede o no llega al resultado esperado.

En la Figura 8 se puede observar como es el ciclo que sigue este tipo de metodologías iterativas.

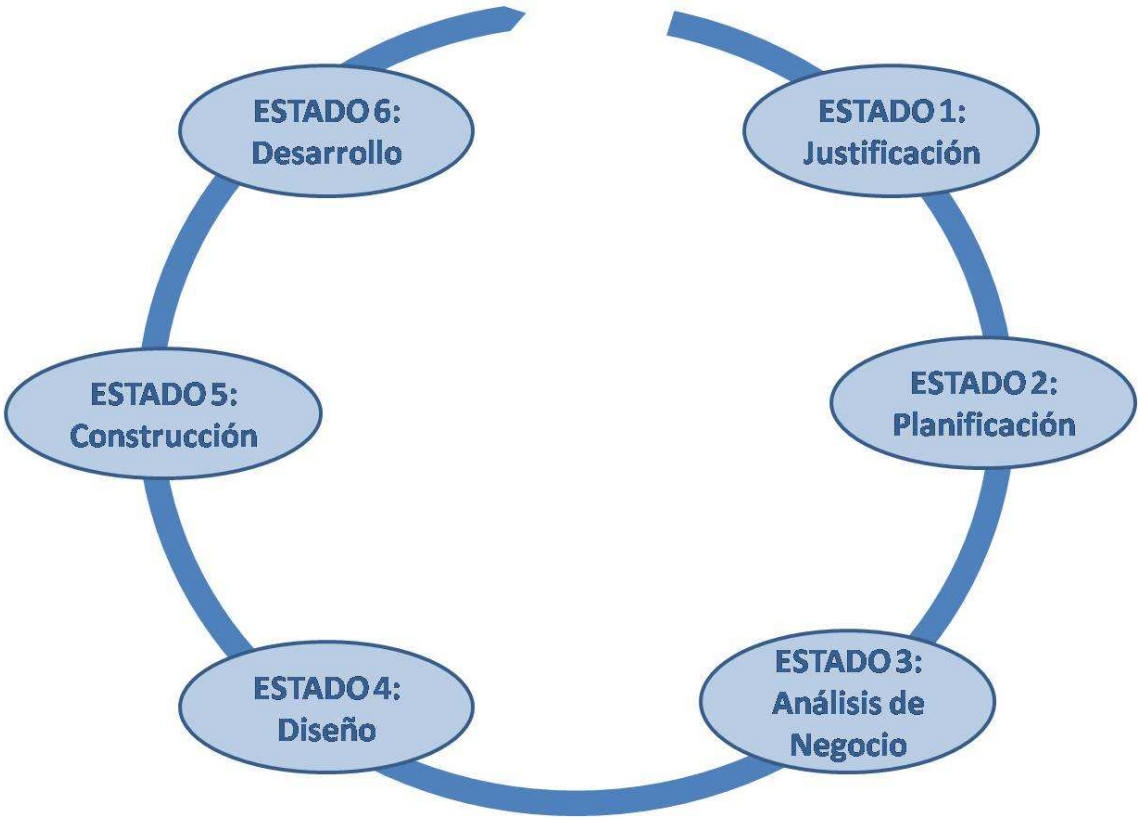



***Figura 8.*** *Estados de metodología iterativa BI.*

Como muestra la flecha de la Figura 8, los procesos son iterativos. Una vez completado, el producto está continuamente siendo mejorado, basándose en la información que proporciona la comunidad de negocio que utiliza el producto. En cada iteración produce una nueva versión como evolución y madurez del producto.

Las prácticas del pasado para desarrollos de sistemas son inadecuadas e inapropiadas para sistemas de BI. En el pasado, los sistemas no eran diseñados o construidos con integración de inteligencia. Cada sistema tenía un inicio y un final, y sólo eran diseñados para resolver un problema aislado de un grupo de gente dentro de una línea de negocio. No estaban adaptados para integrar iniciativas BI, debido a que las prácticas no incluían actividades transversales a la organización, necesarias para sostener un entorno de amplias iniciativas de soporte a la decisión. Las actividades transversales a la organización no sólo eran consideradas innecesarias sino que se consideraba que se reducía el avance de los proyectos.

Para desarrollar sistemas no integrados las metodologías en cascada era suficiente. Con ello se proveía de bastante orientación en la planificación, construcción e implementación de sistemas aislados. Sin embargo, estas metodologías tradicionales no cubren la planificación estratégica, el análisis de negocio transversal a la organización o la evaluación de las nuevas tecnologías con cada proyecto. Además, estas metodologías típicamente comienzan con una necesidad funcional, y concentran el diseño y desarrollo en ella. Finalmente acaban en el mantenimiento, como se puede observar en la Figura 9.

A continuación se listan las diferencias entre los sistemas aislados y las aplicaciones BI:

- Las aplicaciones BI están mucho más dirigidas a oportunidades que a necesidades de negocio.
- Las aplicaciones BI implementan estrategias transversales en vez de silos departamentales como apoyo a la decisión.
- Los requisitos de BI para apoyo a la decisión están más orientados a información estratégica que a funcionalidad y operaciones.
- El análisis de los proyectos BI enfatizan más el análisis de negocio que los análisis de sistema. El análisis es la actividad más importante cuando se desarrolla un entorno de BI como soporte a la decisión.

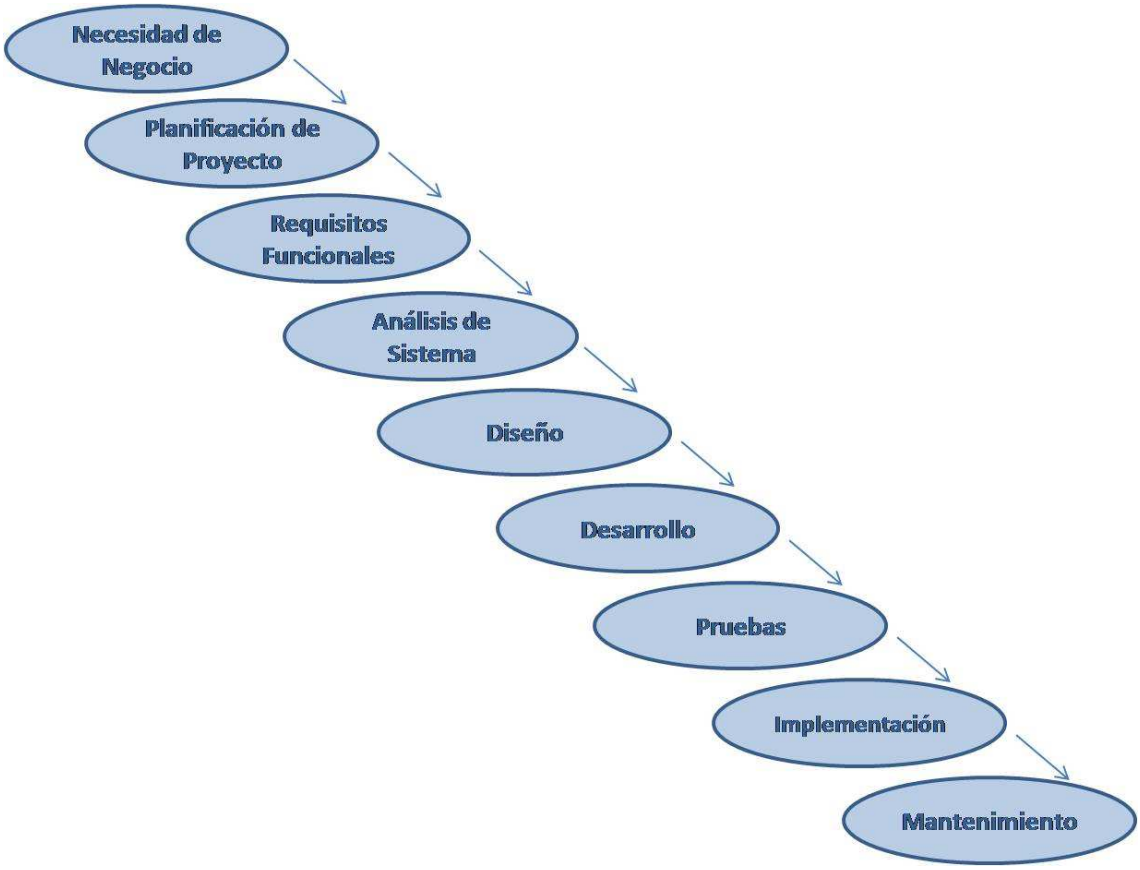



***Figura 9.*** *Metodología convencional de desarrollo de Sistemas de Información.*

Con la expansión del *e-business* se ha incrementado la demanda para integración transversal en la organización. Pero esto no se refiere simplemente a migrar antiguos sistemas sobre diferentes plataformas basadas en iniciativas de aplicación de integración, sino la consolidación, integración e integridad de la información, perfecta funcionalidad del negocio y procesos alineados en la organización.

Trasladar la perspectiva de una organización desde un punto de vista de sistemas aislados a sistemas transversales requiere cambios organizacionales, incluyendo un cambio cultural. Una iniciativa que muestra esto de forma muy visual es la gestión de la relación con los clientes (CRM, Customer Relationship Management). Si las compañías los usasen, se verían reducidos considerablemente sus esfuerzos a la hora de construir aplicaciones BI de soporte a la decisión.

## 3.5 Business Intelligence Roadmap

En términos generales de BI, el ***Business Intelligence Roadmap (BIR)*** especifica el camino y la dirección que deben seguir las aplicaciones, estructuras, herramientas y personas que intervienen en un proyecto de este tipo. Este BIR es, en primer lugar, una guía del ciclo de vida de un proyecto para desarrollar aplicaciones de soporte a la decisión utilizando datos estructurados.

El 60% de los proyecto de BI son abandonados o acaban fallando por una inadecuada planificación, por falta de tareas, por entregas fuera de plazo, por una mala gestión del proyecto, por una ausencia de requisitos de negocio o por una mala calidad en las entregas. Los gestores necesitan saber qué hacer y no hacer en implementaciones de BI, basándose en experiencias fiables.

En el BIR se definen las actividades necesarias en un proyecto BI para mantener la integración de la infraestructura del entorno al que pertenecen. Las infraestructuras técnicas y no técnicas son las competencias esenciales para la alineación organizacional. Además, se deben definir los roles y responsabilidades asignados a cada persona del equipo para cada paso del desarrollo.

BIR describe 16 pasos a seguir en un proyecto de BI, los cuales se encuentran repartidos dentro de los seis estados que se han descrito antes como necesarios para crear un proyecto de ingeniería con éxito. A continuación se van a describir cada uno de ellos dentro de su estado correspondiente.

### *Estado 1:* Justificación

Durante este estado se debe evaluar la necesidad de negocio para saber si se requiere un nuevo proyecto de ingeniería. Dentro de este estado sólo se contempla un paso a seguir, como se puede observar en la Figura 10.


***Figura 10.*** *BIR - Estado 1: Justificación.*

**Paso 1: Evaluación Caso de Negocio**

La mayoría de los proyectos suponen un gran coste, así pues es necesario justificar este coste mostrando el balance entre el coste de la inversión y los beneficios conseguidos. Los beneficios generados por un sistema BI de apoyo a la decisión son muchos, beneficios tangibles, como incrementar el volumen de ventas, y no tangibles, como mejorar la reputación de la organización. Beneficios de este tipo son difíciles de cuantificar en términos de valor monetario. Así pues se debe hacer una lista detallada de beneficios en forma de medida, contra el coste de la implementación del sistema BI. Para justificar la iniciativa se deben relacionar estos beneficios detallados con los problemas de negocio específicos de la organización y con los objetivos estratégicos del negocio.

La justificación siempre debe estar promovida por el negocio, no por la tecnología. Se debe comenzar la justificación identificando los objetivos estratégicos de negocio de la organización y definir claramente la dirección del negocio.

Hay que buscar la información necesaria, investigar qué información será requerida, si estará disponible y será accesible. Se deben identificar los datos dónde reside esta información, y hacer un estudio de estos datos. Se comprobarán los tipos y la calidad de estos datos.

Hacer un análisis de costes-beneficios es imprescindible para garantizar la viabilidad del proyecto. Incluyendo los beneficios intangibles que se han comentado antes podemos hacer una clasificación de éstos:

- Incremento ingresos.

- Incremento beneficio.
- Mejora la satisfacción de los clientes.
- Incremento ahorro.
- Ganancia de mercado compartido.

Otro factor que se deben tener en cuenta en la justificación son los riesgos. Los riesgos son un factor o condición que pueden poner en peligro un proyecto. Para detectar estos riesgos, y su grado de peligro se deben tener en cuenta los siguientes variables:

- La tecnología usada para implementar el proyecto.
- La complejidad de las capacidades y procesos que se implementaran.
- La integración de varios componentes y datos.
- La organización y su soporte financiero y moral.
- El equipo de proyecto y sus habilidades, actitudes y grados de responsabilidad.
- Inversión financiera.

Se debe evaluar cada variable y su nivel de riesgo.

Como resumen se describen las distintas actividades que se deben realizar para completar la justificación:

1. Determinar las necesidades de negocio.
2. Evaluar las soluciones de sistemas de apoyo a la decisión actuales.
3. Valorar la fuente operacional y los procedimientos.
4. Evaluar a los competidores de iniciativas BI.
5. Determinar los objetivos de la aplicación BI.
6. Proponer una solución BI.
7. Realizar un análisis de costes-beneficios.
8. Hacer una evaluación de riesgos.

9. Escribir el resultado de la evaluación.

Como resultado de realizar todas estas actividades se debe obtener las metas estratégicas de negocio de la organización, objetivos de la aplicación BI propuesta, exposición de la necesidad de negocio (oportunidad o problema), explicación de cómo la aplicación BI satisface la necesidad, consecuencias de no satisfacer la necesidad de negocio y no conseguir la solución propuesta, análisis de resultado de costes-beneficios y recomendaciones para los procesos y procedimientos de los sistemas operaciones del negocio.

Por último, se deben definir los roles y responsabilidades de cada persona implicada para cada actividad. Estos son: representante de negocio, *Sponsor*, analista de calidad de datos, gestor de proyecto y expertos de cada materia.

### *Estado 2: Planificación*

La planificación de un proyecto consiste en definir la estrategia y planes tácticos a seguir para desarrollar y llevar a cabo el proyecto con éxito. En este estado se diferencian dos pasos a seguir, evaluación de la infraestructura y la planificación del proyecto, y como se muestra en la Figura 11 deben llevarse a cabo de forma secuencial.

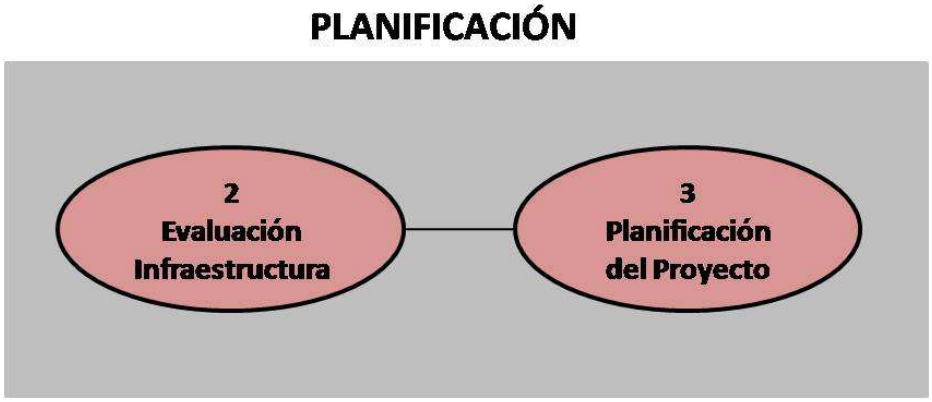



***Figura 11.*** *BIR - Estado 2: Planificación.*

**Paso 2: Evaluación Infraestructura**

La infraestructura de una organización está formada por dos grandes bloques, la infraestructura técnica (*hardware, middleware* y sistemas de gestión de BBDD, SGBD) y la no técnica (estándares, *metadata*, reglas y políticas de negocio).

- **Infraestructura Técnica.**

En primer lugar se debe elegir exhaustivamente una plataforma para la aplicación, que garantice la mejor respuesta para la recuperación y acceso a los datos. Lo que implica una buena elección del *hardware, middleware* y SGBD.

Respecto a la plataforma de *hardware*, se debe comprobar que tiene suficiente escalabilidad y capacidad. En muchas organizaciones aparece el caos, tienen una variedad de hardware muy amplia y desorganizada, debido a que tienen un enorme portfolio de software dispar, y muchos empleados que poseen pocas herramientas de soporte a los sistemas existentes, se tienen muchos componentes, poco relacionados y utilizados. Para resolver este caos es necesario aplicar entornos de BI, teniendo en cuenta los siguientes puntos a la hora de elegir el hardware:

- La nueva plataforma debe mantener la existente configuración de *hardware*.
- El SGBD de la nueva plataforma debe responder bien al acceso y crecimiento de la BBDD. La escalabilidad es muy importante.
- La selección de la plataforma está delimitada por la necesidad de interoperabilidad entre varias plataformas hardware si fuese requerido.
- El coste y el retorno de inversión de los anteriores puntos serán factores controlados.

El *hardware* debe tener suficiente capacidad para manipular accesos complejos y requisitos de análisis sobre grandes volúmenes de datos. Esta plataforma debe soportar periódicos cambios en el volumen de datos, modificaciones frecuentes, grandes modelos de acceso a los datos, gran número de consultas e informes, numerosos usuarios, bastantes herramientas trabajando sobre la BBDD y gran número de sistemas operacionales que alimentan la BBDD.

El *middleware* se refiere a sistemas software ejecutándose a la vez. Este es una capa entre los programas de aplicación y los SO, entre los cuales actúa como puente para la integración de estos sistemas, en entornos de redes de trabajo con muchos nodos.

Un tipo de middleware son los *gateways*. Éstos son requeridos para conectar diferentes arquitecturas de redes de trabajo, escritorios de ordenadores, clientes remotos o servidores. Los *Gateway* pueden ser de punto a punto, para dar acceso de forma universal, para usar SQL o basados en aplicaciones de programación de interfaces (APIS), ODBCs.

Los SGBDs deben adaptarse a los grandes cambios de tamaño que sufren las BBDD. Para seleccionar este gestor se debe tener en cuenta:

- El grado de paralelismo en el tratamiento de consultas y cargas de datos simultáneas.
- Inteligencia en el tratamiento y optimización de modelos de datos dimensionales.
- Escalabilidad de la BBDD.
- Integración con Internet.
- Disponibilidad de esquemas de índices avanzados.
- Duplicar en plataformas heterogéneas.
- Número de operaciones que son atendidas.

Para hacer la mejor selección de la infraestructura se deben realizar las siguientes actividades:

1. Evaluar la plataforma existente.
2. Evaluar y seleccionar nuevos productos.
3. Escribir un informe con la evaluación de la infraestructura técnica.
4. Expandir la plataforma actual.

Las actividades 1 y 2 pueden ser realizadas simultáneamente. Las personas que participan en estas actividades son arquitectos de infraestructuras BI y administradores de BBDD.

Si no se hace una buena selección de las infraestructuras técnicas, éstas pueden quedarse obsoletas en muy poco tiempo.

- **Infraestructura No Técnica.**

Este tipo de infraestructuras es un factor crítico de éxito. Sin una infraestructura transversal a la organización, las aplicaciones BI solo contribuirían al caos en el puente entre aplicaciones y BBDD.

Antiguamente, se utilizaba el lema “divide y vencerás” para solucionar los problemas de negocio. Esto implica dividir el problema en pequeños problemas, y resolverlos separadamente. Así es más sencillo, pero se pueden crear muchos silos de información y son necesarias muchas más conexiones y se pierden relaciones departamentales, lo que implica una pérdida de conocimiento.

Esta infraestructura es necesaria para prevenir la fragmentación del entorno BI de soporte a la decisión. La creación de esta infraestructura implica tareas transversales a la organización, entre las que se encuentran:

- Hacer un extenso análisis de negocio implicando a las personas del negocio.
- Adoptar un sistema de revisión del soporte y evaluación de las actividades de análisis del negocio.
- Resolver conflictos en la definición de datos y dominios.
- Estandarizar nombres y valores de los datos.
- Crear reuniones regulares.
- Crear una arquitectura de datos no redundante y consolidada en el tiempo.
- Crear un repositorio de *metadata*.
- Hacer un inventario de los datos de origen.
- Crear y gestionar una S*taging Area* de expansión central.

La arquitectura está formada por varios modelos referidos a funciones de negocio, procesos de negocio y datos de negocio. Cada modelo de arquitectura

se complementa con el soporte *metadata*, como definiciones de estándares, reglas de negocio y políticas.

Estos modelos deben estar documentados para evitar el abuso, malversación o la recreación redundante de procesos únicos o de datos sobre objetos de negocio. Esto puede hacer que se pierda visión transversal de la organización.

Una arquitectura completamente documentada debe incluir al menos estos componentes:

- Modelo funcional de negocio: representa la descomposición jerárquica de una naturaleza organizacional de negocio (qué hace la organización). Con este modelo se organiza o reorganiza la estructura de una organización en sus líneas de negocio.
- Modelo de procesos de negocio: describe los procesos implementados para las funciones de negocio.
- Modelo de datos de negocio: es a nivel lógico. Indica qué objetos de datos participan en una actividad de negocio, qué relaciones existen entre estos objetos, los elementos de datos cargados sobre estos objetos y qué reglas de negocio gobiernan estos objetos.
- Inventario de aplicación: considera la implementación física de los componentes, cómo funciones, procesos y datos. Muestra en qué parte de la arquitectura física residen, dentro de la infraestructura técnica.
- Repositorio de *metadata*: contiene los detalles descriptivos de los modelos. La *metadata* de negocio es recolectada durante el análisis de negocio, y los técnicos los utilizan durante el diseño y la construcción. Es una herramienta esencial de navegación. Algunos componentes de *metadata* son: nombre de columnas, dominio de columnas, nombre de tablas, nombre de programas, nombre de reporte, etc.

Dentro de una organización, todas las aplicaciones de BI tienen que seguir el mismo estándar. Hay varias categorías de estándares, como pueden ser enfoque de desarrollo, nomenclatura y abreviaciones, captura de *metadata*, modelación lógica de datos, calidad de datos, pruebas, reconciliación, seguridad, acuerdos a nivel de servicios, políticas y procedimientos.

Las actividades dentro de este paso necesitan ser realizadas linealmente. Estas actividades son:

1. Evaluar la efectividad de los componentes actuales de infraestructura no técnica.
2. Escribir las conclusiones de la evaluación de la infraestructura no técnica.
3. Mejorar la infraestructura no técnica.

Las personas que intervienen en estas actividades son el arquitecto de infraestructuras BI, administrador de datos, analista de calidad de datos y administrador de metadatos.

**Paso 3: Planificación del Proyecto**

Los proyectos de BI no son como otros proyectos con una colección finita y estática de requisitos. Estos proyectos incluyen nuevas tareas, cambios en roles y responsabilidades, etc.

Al describir las actividades de gestión de proyecto se debe responder a las siguientes preguntas:

- ¿Qué será entregado?
- ¿Cuándo será hecho?
- ¿Cuánto costará?
- ¿Quién lo hará?

La planificación del proyecto incluye la creación de una guía, anteproyecto, que defina finalidad y objetivos, ámbito, riesgos, constantes, asunciones, procedimientos de control de cambios y gestión de procedimientos.

La planificación de un proyecto no es un proceso que se hace en un momento determinado sino que necesita ser ajustado constantemente. En primer lugar se deben realizar las siguientes acciones para preparar la planificación del proyecto:

1. Hacer una descomposición del trabajo, listando las actividades, tareas y sub tareas.

2. Estimar el esfuerzo en horas para estas actividades, tareas y sub tareas.
3. Asignar recursos a las actividades, tareas y sub tarea.
4. Determinar las dependencias entre tareas.
5. Determinar las dependencias entre recursos.
6. Determinar la trayectoria crítica basada en las dependencias.
7. Crear el plan detallado.

Para conseguir una planificación completa del proyecto BI es necesario llevar a cabo las siguientes actividades.

1. Determinar los requisitos de proyecto.
2. Determinar las condiciones de ficheros fuente y bases de datos.
3. Determinar o revisar las estimaciones de coste.
4. Revisar la evaluación de los riesgos.
5. Identificar los factores críticos de éxito.
6. Preparar el anteproyecto.
7. Crear un plan de proyecto de alto nivel.
8. Presentación del proyecto.

Algunas de estas actividades pueden ser realizadas en paralelo, pero la mayoría deben ser llevadas a cabo secuencialmente.

Las personas que intervienen en estas actividades son diseñador de la aplicación, representante del negocio, administrador de datos, analista de calidad de datos, administrador de BBDD, diseñador de procesos ETL ( Extract, Transform & Load), administrador meta datos, gestor de proyecto y experto en la materia.

### *Estado 3: Análisis de Negocio*

Se debe llevar a cabo un análisis detallado del problema de negocio u oportunidades de negocio para conseguir una comprensión sólida de los requisitos de negocio necesarios para llegar a una posible solución o producto. Dentro de este estado existen cuatro pasos a seguir, el paso cuatro debe realizarse antes que los demás pero el resto pueden ser realizados en paralelo como se muestra en la Figura 12.

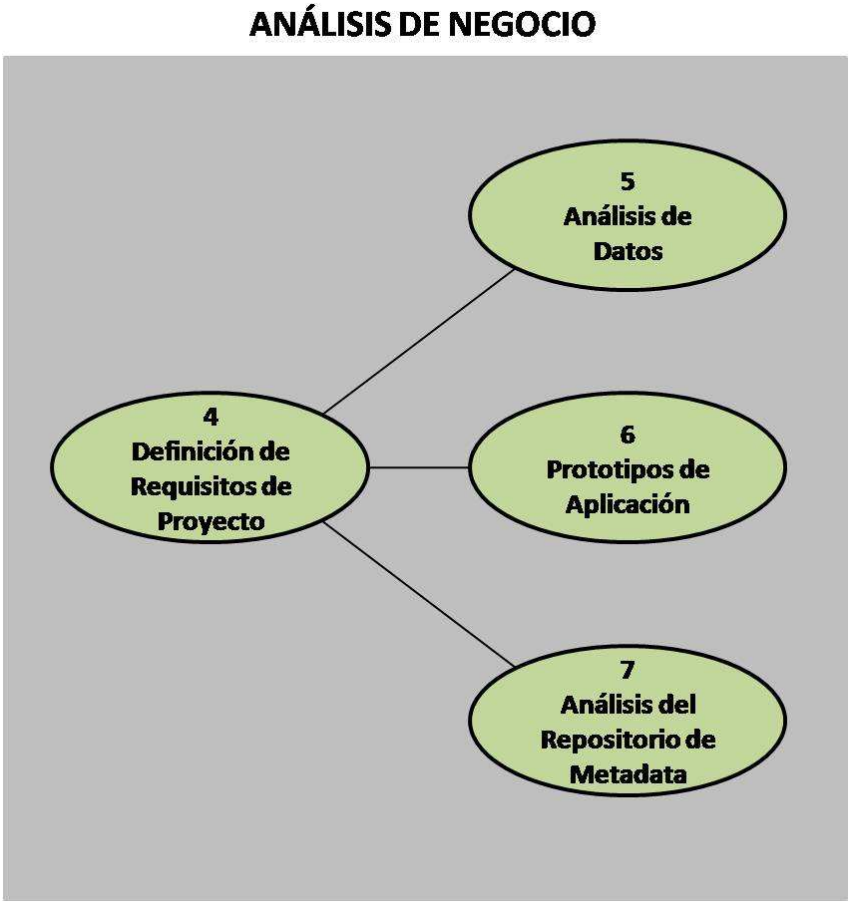



***Figura 12.*** *BIR - Estado 3: Análisis de Negocio.*

#### Paso 4: Definición de Requisitos de Proyecto

Los requisitos se dividen en dos grupos, los requisitos de alto nivel generales del negocio, identificados a partir de la iniciativa BI, y que son revisados periódicamente, y los requisitos específicos del proyecto, concentrados en los detalles de los resultados esperados de cada aplicación BI relacionada. Para hacer más clara su distinción se ha incluido la Tabla 2 en este apartado en vez de en el anexo.

| | REQUISITOS GENERALES DE NEGOCIO | REQUISITOS ESPECÍFICOS DE NEGOCIO |
|---|---|---|
| **PROPÓSITO** | *Determinan necesidades generales del negocio de la organización, para el entorno del soporte a la decisión BI.* | *Define las funciones y datos específicos que deben estar desarrollados al finalizar el proyecto BI.* |
| **INTERLOCUTORES** | *Ejecutivos de negocio,*<br>*Gestores de TI.*<br>*Personal TI.*<br>*Gestores de la línea de negocio.*<br>*Expertos en la materia.* | *Patrocinador del negocio.*<br>*Representante del negocio.*<br>*Usuarios potenciales.*<br>*Stakeholders.*<br>*Expertos en la materia.* |
| **RESULTADO** | *Reporte de requisitos de negocio.* | *Documento de requisitos de aplicación.* |
| **CONTENIDO DEL RESULTADO** | • *Descubrimientos.*<br>• *Soluciones.*<br>• *Oportunidades.*<br>• *Recomendaciones.*<br>• *Pasos siguientes.* | • *Requisitos funcionales.*<br>• *Requisitos de datos.*<br>• *Requisitos de seguridad.*<br>• *Requisitos de modelos.*<br>• *Requisitos de viabilidad.* |

***Tabla 2.*** *Comparación de Requisitos Generales con Específicos de Negocio.*

Para obtener los requisitos es necesario que haya un equipo entrevistador, que invierta todo el tiempo necesario para clarificar todos los requisitos. Este equipo debe hacer entrevistas a los interlocutores y estudiar su entorno. Es bueno que los entrevistadores utilicen técnicas gráficas en el documento de requisitos de aplicación, como diagramas de causa-efecto, diagramas entidad-relación, modelos de esquema en estrella, diagramas de descomposición funcional, flujos de datos, etc. Estos

diagramas son muy representativos, y facilitan la verificación, por parte de los interlocutores, de los requisitos obtenidos por los entrevistadores.

Antes de llevar a cabo las entrevistas es necesario hacer una preparación, teniendo en cuenta las siguientes consideraciones:

- Equipo entrevistador: es preferible que la persona que toma notas no sea la misma que conduce la entrevista, aunque es difícil.
- Interlocutores: en una entrevista los interlocutores pueden ser uno o varios, es preferible que sea por pares, así se pueden rebatir y obtener más información, pero estos deben tener responsabilidades similares. Esto a veces también puede ser contraproducente, pueden ser deshonestos. Así pues lo ideal es un balance entre entrevistas con un interlocutor y con varios.
- Documentarse: los entrevistadores deben obtener información de otros medios antes de acudir a las entrevistas. Se pueden consultar documentos existentes, reportes, *web sites*, incluidas las *web sites* de los competidores. Esto les ayudará a una mejor visión de la industria, de los procesos de negocio y los acrónimos y terminología de la organización.
- Cuestionarios: es un buen elemento para basarse en la entrevistas, además puede ser enviado antes de éstas a los interlocutores, así ellos podrán prepararlas, o incluso completarlas y enviarlas.
- Planificación de la entrevista: no planificar más de 4 entrevistas de 1 hora por día. Es importante que después de cada entrevista se clarifiquen las notas tomadas, esto debe hacerse el mismo día de la entrevista.

Hay algunas buenas prácticas que garantizarán la efectividad de estas entrevistas, como son:

- Dirigir la primera entrevista a los requisitos básicos, generales necesarios para resolver el problema específico de negocio.
- Prepararles una guía para obtener la mayor información de ellos, sino puede que no sepan contar todo lo que se quiera saber.
- Estar preparados para escuchar y resolver conflictos y prioridades.

- Tomar notas mientras trascurre la entrevista, revisarlas con el interlocutor el mismo día de su realización y dejarlas a su alcance para poder consultarlas en cualquier momento.
- Se pueden grabar las entrevistas, así no se perderá detalle. Para ello se debe pedir permiso a los interlocutores.
- Tan pronto como sea posible hacer un documento con las notas, enviarlo al interlocutor y a todas las personar que intervienen, y solicitar modificaciones o inclusiones de notas.

Como en todos los pasos, para el éxito es necesario realizar una serie de actividades:

1. Definir los requisitos de la infraestructura técnica.
2. Definir los requisitos de la infraestructura no técnica.
3. Definir los requisitos de reportes.
4. Definir los requisitos de los datos origen.
5. Revisar el alcance del proyecto.
6. Expandir el modelo lógico de datos.
7. Definir los niveles de servicios preliminares.
8. Escribir el documento de requisitos de aplicación.

Las personas que intervienen en este paso son los desarrolladores de aplicaciones, representantes de negocio, administradores de datos, analistas de calidad de datos, administrador de *metadata* y expertos de la materia.

**Paso 5: Análisis de Datos**

Las aplicaciones BI, a diferencia de las tradicionales, hacen un análisis de datos centrado en el negocio, no en el sistema. Un punto básico es que los datos estén disponibles de forma transversal para todos los departamentos.

El análisis de los datos es diferente del análisis del sistema. El análisis de datos está centrado en comprender y corregir las discrepancias existentes en los datos de

negocio, independientemente del diseño de los sistemas o métodos de implementación.

Para realizar un análisis riguroso son necesarios dos métodos complementarios:

1. Modelado lógico de datos de alto nivel para la integración y consistencia.

La técnica más efectiva para descubrir y documentar la integración transversal de la organización son los modelos entidad-relación, pero en general son favorables todos los modelos lógicos de datos para las especificaciones de proyecto y los modelos lógicos de datos de la empresa.

En las sesiones de modelado participan los administradores de datos, representantes de negocio, expertos en la materia, analistas y desarrolladores de sistemas, administradores de BBDD y técnicos de TI.

En los modelos lógicos es necesario que se establezca un estándar de meta datos, dónde se incluyan nombre de los datos, su definición, relaciones, identificadores, tipos, longitud, dominio, reglas, políticas y autoridades sobre los datos.

2. Análisis de orígenes de datos de bajo nivel para la estandarización y calidad.

El análisis de datos no acaba con la realización de los modelos lógicos, porque a veces los orígenes de datos no siguen las reglas y políticas de negocio capturadas durante las sesiones de modelado. Los problemas con los datos y las violaciones de las reglas de negocio pueden no ser descubiertas hasta que se implementan los procesos ETL.

Para resolver este problema es necesario aplicar reglas durante la distribución de los datos, reglas de conversión de datos técnicos (tipos, longitudes, etc.), reglas de dominios de datos, reglas de integridad de datos.

Además, otros aspectos muy importantes para garantizar la calidad de los datos son su limpieza, integridad y conciliación con la comunidad del negocio. Las personas del negocio deben implicarse y ser responsables de la calidad de los datos, no sólo implica a los técnicos de TI.

También es muy importante definir bien los procesos de selección de los datos origen. En ellos hay que tener en cuenta los puntos clave, como son

garantizar la integridad de los datos, su precisión, la exactitud y veracidad, la fiabilidad y el formato.

Se debe tener especial cuidado en la limpieza de los datos, y utilizar sistemas que nos ayuden a detectar sobre qué datos se puede hacer y de qué manera.

Las actividades que hay que realizar en este paso son:

1. Análisis de los orígenes de datos externos.
2. Refinar los modelos lógicos de datos.
3. Analizar la calidad de los datos de origen.
4. Expandir el entorno de los modelos lógicos de datos.
5. Resolver las discrepancias de los datos.
6. Escribir las especificaciones de limpieza de datos.

Las personas que intervienen en estas actividades son el representante de negocio, administrador de datos, analista de calidad de datos, diseñador de procesos ETL, administrador de *metadata*, *Stakeholders* y expertos en la materia.

**Paso 6: Prototipos de aplicación**

Los prototipos pueden ser un método efectivo para validar los requisitos y encontrar partes que faltan y discrepancias. Con ellos también se pueden ver el camino de acceso para los requisitos, las capacidades de la tecnología BI y el acceso y análisis de una parte de su aplicación BI. Si se puede, es recomendable hacer el prototipo sobre los requisitos reales del negocio, así la evaluación será mucho más óptima.

Para realizar el prototipo es necesario conocer las buenas prácticas, como son:

- Limitar el dominio: hacer pequeños prototipos ayuda a centrarse en una parte de todos los requisitos.
- Entendimiento cercano de los requisitos de BBDD: ayudar a los administradores a entender el camino seguido por los datos.
- Elegir los datos correctos: intentar seleccionar datos limpios para el prototipo.

- Validar la usabilidad de las herramientas de acceso y análisis.
- Involucrar a la gente de negocio.

Cuando se lleva a cabo la creación de un prototipo se debe tener en consideración el equipo, el gestor, el entorno, el resultado, métodos de resultado (interfaces de usuario gráficas), integración de los datos y criterios de éxito.

Existen varias técnicas de prototipos que se pueden utilizar, la elección de una de ellas depende de las necesidades. En la Tabla 3 se muestran algunas de ellas, y se ha incluido en este apartado en vez de en el anexo por su importancia para la comprensión del propósito de los prototipos:

| | ***PROPÓSITO*** | ***IMPLICACIONES*** |
|---|---|---|
| ***Show & Tell*** | *Anular costes, ganancias de la gente del negocio, ganancias de soporte al negocio y seguridad aportada por la aplicación BI.* | *Aclarar la limitación funcional a la gente de negocio, que funcionamiento está y no está incluido en la aplicación.* |
| ***Mock-Up*** | *Comprender los requisitos de aplicación, actividades de negocio e iniciar funciones del sistema.* | *Poner atención en las interfaces y usar lenguajes de programación menos sofisticados para construirlo más rápido.* |
| ***Proof-of-concept*** | *Explorar los riesgos desconocidos y de implementación, y decidir si proceder en conjunto o no.* | *Estar delimitado al alcance, no construir interfaces de aplicación y construir sólo funcionalidad para tomar o no una decisión.* |
| ***Visual-Desing*** | *Entender el diseño de las interfaces visuales y especificaciones de diseños para las interfaces.* | *Si el prototipo se usará como parte de la aplicación final, usar el mismo lenguaje de programación, sino usar otro diferente al que se usará en la aplicación final.* |

| | PROPÓSITO | IMPLICACIONES |
|---|---|---|
| ***Demo*** | *Trasladar la visión de la aplicación de BI a la gente de negocio o grupos externos, comprobar la viabilidad del mercado del entorno de la aplicación y probar o demostrar la usabilidad del acceso propuesto y la parte de análisis del prototipo.* | *Indicar qué porcentaje de la aplicación está representada y ser realista con las expectativas.* |
| ***Operacional*** | *Crear un piloto casi total de una funcionalidad concreta, mostrando su acceso y análisis. Obtener feedback con los verdaderos usuarios.* | *Indicar qué porcentaje de la aplicación está representada y ser realista con las expectativas. Crear en el lenguaje final, y crear código de calidad.* |

***Tabla 3.*** *Tipos de prototipos, propósitos e implicaciones.*

Al igual que con el proyecto se hace un documento de anteproyecto, con el prototipo se debe hacer lo mismo. En este caso se especificará el principal propósito del prototipo, los objetivos, la lista de la gente de negocio implicada, los datos, las plataformas software y hardware, las medidas de éxito y una interfaz de aplicación.

Como en todos los pasos es necesario realizar una serie de actividades:

1. Analizar los requisitos de acceso.
2. Determinar el alcance del prototipo.
3. Seleccionar las herramientas para la creación.
4. Preparar el anteproyecto.
5. Diseñar los reportes y las consultas.
6. Construir el prototipo.
7. Demostración del prototipo.

Las personas que intervienen en este paso son el desarrollador de la aplicación, representante de negocio, administrador de la BBDD, *Stakeholders*, expertos en la materia y el *Web Máster*.

**Paso 7: Análisis del Repositorio de *Metadata***

El repositorio de *metadata* es una BBDD, pero no una BBDD ordinaria. Ésta no diseña la carga de los datos de negocio para una aplicación, sino que diseña la carga de la información contextual sobre los datos de negocio, como por ejemplo significado y contenido, políticas de gobierno, atributos técnicos, especificaciones de transformación y programas que manipulan los datos. La información contextual existe de forma inherente a cada organización, esté documentada o no. Lo más usual en las organizaciones es que no esté documentado, y en esos casos las organizaciones crean sus propias reglas de negocio, con sus datos y procesos redundantes, sin darse cuenta de que lo que necesitan ya está creado.

La *metadata* describe una organización en términos de sus actividades de negocio y sus objetos sobre los que actúan estas actividades. La gran importancia de la *metadata*, en las aplicaciones BI de apoyo a la decisión, reside en ayuda a hacer una metamorfosis de los datos de negocio a información, obteniendo conocimiento. La *metadata* se puede ver como un componente semántico dentro del entorno del soporte a la decisión BI, proveyendo al negocio el contexto dónde los datos son usados.

Se pueden distinguir dos categorías de *metadata*, de negocio y técnica,

Los metadatos siempre han formado parte del sistema operacional, en documentos de sistemas, registros de *layout*, catálogos de BBDD, secciones de declaración de datos en los programas. En el entorno BI la *metadata* toma un papel más importante. Ayudan a la gente del negocio a localizar manejar, comprender y usar los datos generados por la aplicación BI. Así pues son vistos como una herramienta de navegación.

En primer lugar se debe hacer una clara estandarización de los datos, para garantizar que los conceptos son unívocos para todas las personas involucradas en el negocio. También es recomendable agrupar los componentes de *metadata* para hacer más sencilla su identificación y comprensión:

- Propiedades: los datos y la aplicación propiedad de la organización.

- Descripción de características: nombre, definición, tipo y longitud, dominio y notas.
- Reglas y políticas: relaciones, reglas y políticas de negocio, seguridad, líneas de limpieza de los datos, aplicabilidad y programación (tiempos de acción).
- Características físicas: origen de datos, localización física, transformaciones, derivaciones, agregaciones y sumatorios, volumen y crecimiento.

Para crear el repositorio de metadatos es necesario realizar las siguientes actividades:

1. Analizar los requisitos del repositorio de metadatos.
2. Analizar los requisitos de interface para el repositorio.
3. Analizar los requisitos de acceso y reporte del repositorio.
4. Crear el modelo lógico de metadatos.
5. Crear los meta-meta datos (describen en detalle los componentes *metadata* requeridos).

Los roles que deben participar en estas actividades son el administrador de los datos, administrador de *metadata* y experto en la materia.

### *Estado 4: Diseño*

El diseño consiste en concebir un producto que resuelve el problema de negocio, o permite la oportunidad de negocio. En este estado se deben seguir tres pasos, dos que tienen que ejecutarse de forma secuencial y otro que puede hacerse de forma paralela como se muestra en la Figura 13.

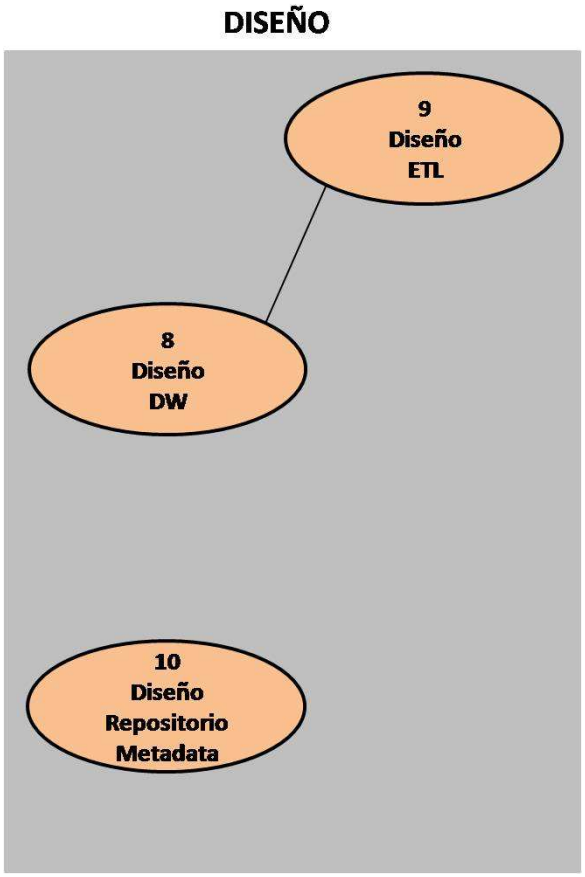



***Figura 13.*** *BIR - Estado 4: Diseño.*

**Paso 8: Diseño del DW.**

Los requisitos de agregación y de datos de los sistemas BI han introducido un nuevo tipo de diseño de BBDD y un nuevo modo de carga de datos. Este nuevo esquema de diseño de BBDD multidimensionales junto con las nuevas tecnologías BI almacena la información en cubos con dimensiones, los DW. Los administradores y desarrolladores deben aprender las nuevas técnicas de diseño y la nueva forma de trabajar con los DW.

El acceso a los datos se puede hacer de la forma convencional, mediante SQL, o de forma multidimensional con herramientas OLAP.

En la Tabla 4 se muestran las diferencias entre una BBDD convencional y un DW, que no se ha incluido en el anexo por su importancia en la comprensión.

| BBDD OPERACIONAL | DW |
|---|---|
| *Generado hacia la eliminación de redundancia, coordinación de modificaciones y repetición del mismo tipo de operaciones muchas veces cada día.* | *Generado para soportar un gran rango de consultas y reportes, que pueden variar de un analista a otro según el departamento. Todas las consultas y reportes puede que no se hagan en el mismo día, incluso tampoco periódicamente.* |
| *Muchas de las transacciones requieren tiempo de respuesta mayor a segundos.* | *El tiempo de respuesta es muy importante, los tiempos de respuesta son mínimos.* |
| *Fuertes normalizaciones para soportar modificaciones consistentes y mantenimiento de la integridad referencial.* | *Des normalización para proveer rápidas respuestas de un gran rango y gran cantidad de datos. Los datos que normalmente son consultados juntos se almacenan juntos.* |
| *Cargas muy pequeñas de datos derivados. Los datos son derivados dinámicamente cuando se necesita.* | *Grandes cargas de datos derivados. Esto acorta el tiempo de las consultas y reportes.* |
| *Los datos históricos no se guardan, son archivados.* | *Se almacenan gran cantidad de datos históricos y detalles de éstos.* |

***Tabla 4.*** *Comparación de bases de datos frente a Data Warehouse.*

El diseño del DW incluye las siguientes consideraciones:

- El DW se diseña para simplificar, no por eficiencia en la carga y mantenimiento.
- La eliminación o minimización de la redundancia de los datos no es un objetivo.
- Los datos son cargados de la misma forma en que vayan a ser accedidos por los usuarios.

- El diseño está dirigido al acceso y usabilidad.
- Un diseño normalizado no es intuitivo, y puede confundir a una persona de negocio.
- Los datos siempre provienen de una ocurrencia interna o externa, o son derivados.

Existen varios tipos de modelos de diseño lógico, éstos son:

- Esquema en Estrella: la tabla de hechos (datos derivados) están en el centro del diseño, todas las tablas de dimensiones se relacionan con ella.
- Esquema Copo de Nieve: puede haber tablas de dimensiones que no estén relacionadas con la tabla de hechos.

Además se deben tener en cuenta las siguientes cuestiones: el lugar físico, particionamiento, clasificación, indexación, reorganización, *backup* y recuperación de los datos y el paralelismo en la ejecución de consultas.

Para llevar a cabo el diseño con éxito es necesario realizar las siguientes actividades:

1. Revisar los requisitos de acceso a los datos.
2. Determinar los requisitos de agregación y resumen.
3. Diseñar el DW.
4. Diseñar las estructuras físicas del DW.
5. Construir el DW.
6. Diseñar los procedimientos de mantenimiento del DW.
7. Preparar la monitorización y refinamiento del diseño del DW.
8. Preparar la monitorización y refinamiento del diseño de consultas.

Los roles que intervienen en esas actividades son el diseñador de aplicación, administrador de datos y de BBDD, y diseñador de procesos ETL.

**Paso 9: Diseño ETL.**

Los datos de origen para las aplicaciones BI provienen de varias plataformas, que son gestionadas por una variedad de sistemas operacionales y aplicaciones. El propósito de los procesos ETL es unir los datos de estas plataformas heterogéneas y transformar a un formato estándar para el DW en los sistemas de apoyo a la decisión.

Existen varias estrategias de implementación dependiendo de la combinación del DW. La más utilizada es el entorno *Data Mart*. Para la correcta implementación de la estrategia es necesario que se construya un puerto para el entorno, que sea común para todos los orígenes de datos, y sea integrado y reconciliado. Es primordial que todas las transformaciones sean comunes para todos los datos origen, que sólo se llevarán a cabo una vez, y los datos se conciliarán volviendo a acceder a las BBDD origen.

El proceso ETL comienza con la preparación de los datos para un formato común, su conciliación y limpieza. Estos procesos requieren 3 subprocesos:

1. Carga Inicial: primero se mapean los elementos de datos que serán cargados, seleccionando el elemento más apropiado en la BBDD destino (*target*), que será el más similar en nombre, definición, tamaño, longitud y funcionalidad. Después se crean los procesos de transformación de los datos.

2. Carga del Histórico: carga datos estáticos que provienen de dispositivos de carga fuera de línea. Los ficheros que contienen los históricos suelen ser varios y con diferencias importantes. Esta tarea consiste en conciliar todos los históricos y unificarlos.

3. Carga Incremental: una vez realizadas las cargas anteriores se debe diseñar otra que haga las cargas periódicas (mensuales, semanales o diarias) para incluir los cambios. Este proceso se puede hacer extrayendo todos los registros o sólo los cambios. La extracción de todos los registros suele ser una opción poco viable por el gran volumen de datos.

Desde el punto de vista operacional, lo más sencillo sería duplicar todos los contenidos, y que los procesos ETL trabajen sobre ellos. Pero esto no es una buena solución para los diseñadores ETL, pues ellos no necesitan todos los datos. Para ellos lo mejor sería realizar la extracción directa, cortando, filtrando y limpiando los datos de una vez. Pero esto puede suponer un problema en muchas organizaciones, las aplicaciones BI impactan sobre el sistema operacional, que puede quedar suspendido

durante varias horas para el resto de personal. Así pues se deben realizar procesos que provoquen el menor trastorno al sistema operacional, en la medida de lo posible.

El 80% del trabajo ETL son transformaciones, y el resto son extracciones y cargas. Así pues las transformaciones ocuparán la mayoría del esfuerzo. Durante estos procesos es normal encontrarse con muchos problemas. Se pueden tener registros con claves primarias inconsistentes, que normalmente en los DW son identificadas por otras claves, ya que puede que estos registros provengan de distintos ficheros y que en cada uno se identifique de una manera. Estos registros deben ser consolidados o transformados. También se pueden encontrar inconsistencias en los valores de los datos, formatos diferentes para un mismo elemento de datos, valores de datos incorrectos, homónimos y sinónimos.

Una vez realizados los procesos ETL, el administrador de BBDD y el analista de calidad de datos deben diseñar un diagrama de flujo para estos procesos. La finalidad de este diagrama es mostrar de forma clara las dependencias entre todas las extracciones, utilidades de ordenación y fusión, transformaciones, ficheros o tablas temporales, procesos de errores y secuencias de carga.

Se debe conocer el término *Staging Area,* es el área dónde corren los procesos ETL, que puede ser centralizada o descentralizada. Esta hace que los procesos ETL sean menos complicados y se coordinen más fácilmente, pues sólo deben acceder a ella en vez de a todos los orígenes de datos.

Durante el paso de diseñar los procesos ETL se deben realizar las siguientes actividades:

1. Crear un documento con el mapeo del origen al destino.
2. Probar las funciones de la herramienta ETL.
3. Diseñar el flujo de los procesos ETL.
4. Diseñar los programas ETL.
5. Crear el *Staging Area*.

Las personas implicadas en estas actividades son el analista de calidad de datos, administrador de BBDD, diseñador ETL y experto en la materia.

**Paso 10: Diseño del Repositorio *Metadata*.**

La mayoría de los administradores usan productos de diccionarios genéricos de datos, el resto intentan crear sus propios diccionarios de datos (*metadata*) o añadir nuevos componentes a diccionarios ya existentes. Se debe intentar tener metadatos personalizados para cada organización, pues su información no es común a otras. Crear el repositorio de *metadata* no es fácil, y nos encontramos con muchos silos, pero se debe poner empeño para resolverlos.

Son muchas las herramientas que requieren componentes de *metadata*:

- Herramientas CASE (*Computer Aided Software Engineering*): necesitan la *metadata* para los modelos lógicos.
- Diccionarios de SGBD: estructuras como BBDD, tablas, columnas, índices, etc.
- Herramientas ETL: para los mapeos y las especificaciones de transformación.
- Herramientas de limpieza de datos: para dominios y reglas usadas.
- Herramientas OLAP: para consultas, reportes, etc.
- Herramientas de *Data Mining*: para los modelos analíticos y algoritmos que utilizan.

Existen varias posibles soluciones para el repositorio de *metadata*. A continuación se describen las posibles soluciones:

- *Repositorio Metadata Centralizado:* Es la solución más común y fácil de implementar porque sólo hay una BBDD y una aplicación de mantenimiento. Esta BBDD no necesita ser coordinada con otras, lo que implica una fácil gestión. Esta solución incluye todos los requisitos de *metadata*, las aplicaciones y herramientas de acceso se diseñan orientadas al cliente, los reportes y las funciones de ayuda se diseñan exactamente como se desea y los técnicos tienen total control sobre el diseño y funcionalidad del repositorio. Pero también tiene desventajas, es necesario estar constantemente manteniéndolo, las aplicaciones de acceso también deben tener mantenimiento, hay que mejorarla porque la funcionalidad cambia y aumenta.
- *Repositorio Metadata Descentralizado:* Se crean multitud de repositorios, cada uno en una BBDD en una localización diferente. Se tiene un SGBD

común para todos para poder mantener la consistencia. Con esta solución se permite que varios usuarios mantengan y gestionen su propio repositorio, son más pequeños y fáciles de usar, cada uno puede tener su propio modelo, los reportes pueden ser caracterizados para cada repositorio individual y el gestor único hace que el nombre y localización de cada repositorio sea transparente para el usuario. A su vez, esta solución hace difícil controlar la redundancia y mantener la consistencia, puede haber problemas de sincronización entre las distintas plataformas, se puede incrementar demasiado la comunicación y es más difícil de aprender y utilizar.

- *Solución MetdData Distribuida con XML:* Esta es la solución más prometedora, pero a su vez la más difícil de implementar. Es una mezcla de las dos anteriores, hay varios diccionarios de datos incluidos en cada herramienta (CASE, ETL, OLAP, etc.) de donde se crea un XML que se alimenta de todos ellos, y a su vez es gestionado por un SGBD único, que accede a todas las BBDD de las distintas herramientas. Esta solución incluye en el XML etiquetas y tipos estándares, la *metadata* no se duplica, la localización es transparente para el usuario, la *metadata* y los datos de negocio pueden ser asociados y transmitidos a la vez. Pero la selección inicial de toda la *metadata* es un proceso manual muy laborioso, el SGBD y los vendedores de herramientas deben seguir los mismos estándares y soportar XML.

Para hacer el diseño del repositorio de *metadata* se pueden usar modelos entidad-relación y/o modelos orientados a objetos.

Las actividades que se deben realizar para el diseño del repositorio son las siguientes:

1. Diseñar la BBDD para el repositorio de *metadata*.
2. Instalar y probar el producto creado o adquirido para el repositorio.
3. Diseñar los procesos de migración de *metadata*.
4. Diseñar la aplicación *metadata*.

Las personas que intervienen en estas actividades son el arquitecto de infraestructura BI, administrador de datos y administrador de *metadata*.

### *Estado 5: Construcción*

En este estado se debe construir el producto que ofrece una respuesta a la inversión, dentro de un periodo de tiempo. En la construcción se deben seguir cuatro pasos que pueden ser llevados a cabo de forma paralela como se muestra en la Figura 14.

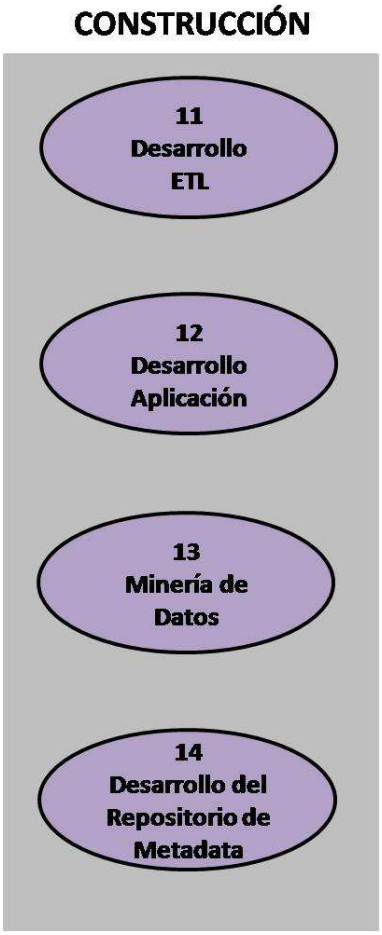



***Figura 14.*** *BIR - Estado 5: Construcción.*

**Paso 11: Desarrollo ETL.**

Las reglas técnicas y de negocio definidas y acumuladas durante los pasos de planificación son requeridas por las transformaciones de los datos, y son reflejadas en este paso.

Para la transformación de los datos en primer lugar se debe realizar la limpieza de éstos, luego los sumatorios, derivaciones y agregaciones pertinentes, y por último la integración.

Un aspecto muy importante en el desarrollo de procesos ETL es la reconciliación, que es hacer que los datos origen y destino estén en concordancia. Para verificar esta concordancia es bueno calcular contadores de registros, dominios y cantidades, con estos valores resulta fácil contrastar la información.

Una vez se llega a este paso es muy recomendable hacer revisiones, para verificar el correcto seguimiento del proyecto. Hay una técnica de revisión por parejas, que consiste en reunirse con una pareja y mostrarles el desarrollo de una parte del trabajo, y que ellos lo verifiquen o discutan. Esto es muy constructivo, y si fuese con más personas sería más complicado y menos productivo. También aportan mucho conocimiento las sesiones de *brainstorming*.

Además de estas revisiones es preciso realizar pruebas. El mayor problema es que normalmente se hace una batería de pruebas muy pobre. Así pues se van a nombrar los distintos grupos de pruebas que se deben hacer para validar perfectamente:

- Pruebas Unitarias: no se prueba el flujo, sino módulos pequeños y scripts. Se prueba la compilación, funcionalidad y edición.
- Pruebas de Integración: se prueba el programa completo, para observar las interacciones y flujos.
- Pruebas de Regresión: son las más largas, comprueban que las modificaciones en los programas ETL existentes no producen errores inesperados.
- Pruebas de Rendimiento: comprueban los tiempos de ejecución, conviene usar herramientas de simulación que representen situaciones reales.
- Pruebas de Calidad y Seguridad: se deben cumplir las normas de la organización para poner una aplicación en producción.
- Pruebas de Aceptación: antes de realizarlas se deben establecer los indicadores y valores de aceptación.

Es habitual que las pruebas no se realicen, que se hagan incorrectamente, en desorden o no se les de la importancia necesaria. Por esto es muy importante realizar un plan formal de pruebas. Cada plan debe especificar su propósito, esquema, casos de prueba y fichero de registro de pruebas, esto hará que se lleven a cabo correctamente y la revisión de estas no sea imposible.

Como todos los pasos es necesario realizar una serie de actividades:

1. Construir y procesar las pruebas unitarias.
2. Construir y procesar las pruebas de integración y regresión.
3. Construir y procesar las pruebas de rendimiento.
4. Construir y procesar las pruebas de seguridad y calidad.
5. Construir y procesar las pruebas de aceptación.

Los roles que intervienen en este paso son el representante del negocio, administrador de BBDD, diseñadores y desarrolladores ETL, experto en la materia y equipo de pruebas.

**Paso 12: Desarrollo Aplicación.**

La principal razón de la iniciativa de sistemas BI de apoyo a la decisión es proveer de un acceso rápido y sencillo a los datos para analizar el negocio. Es por ello que surgen las herramientas OLAP, y se convierten en el mayor componente de estos sistemas.

La definición aceptada generalmente es que OLAP se refiere a tecnología de procesamiento analítico en línea que crea nueva información de negocio a través de robustas transformaciones y cálculos ejecutados sobre datos existentes. El foco está en procesos analíticos especializados, en aspectos multidimensionales y la posibilidad de navegar por las dimensiones del DW.

Las características de estas herramientas son que presentan una vista multidimensional, permiten sumatorios y agregaciones, capacidad de análisis y consultas interactivas, da soporte al análisis de negocio, navegar por las dimensiones que pertenecen a jerarquías, capacidades de modelación analítica. También soportan modelos funcionales para el análisis de tendencias en los datos y visualiza los datos en forma de gráficos y tablas.

La principal diferencia de estas herramientas con las convencionales es la forma de presentar la información. Los indicadores o hechos son normalmente presentados en un formato multidimensional, columnas en una tabla de hechos o celdas en un

cubo. Estas columnas o celdas contienen datos numéricos pre calculados que son relacionadas con un objeto de negocio del área al que pertenece.

Todas estas características de estas herramientas hacen que sean capaces de realizar análisis muy complejos, como son información de clientes, planificación financiera y marketing.

Otro aspecto importante de estas herramientas es que analizan indicadores y hechos sobre la perspectiva de múltiples variables o características. Estas variables normalmente describen objetos de negocio o dimensiones.

Conceptualmente la arquitectura OLAP consiste en 3 componentes funcionales, éstos son:

- *Presentación de Servicios*: los datos necesitan ser presentados en un formato que entienda la gente de negocio, para realizar propuestas, decidir cuánto gastar, definir niveles de investigación, etc.
- *Servicios OLAP*: ofrecen servicios de consultas, reportes y análisis, que a su vez están interrelacionados, son interactivos e iterativos.
- *Servicios de BBDD*: las arquitecturas OLAP soportan dos tipos de BBDD, relacionales (accesibles con herramientas ROLAP) y multidimensionales (accesibles con herramientas MOLAP).

Las actividades que se realizan en este paso son las siguientes:

1. Determinar los requisitos del proyecto final.
2. Diseñar los programas de aplicación.
3. Construir y realizar pruebas unitarias de los programa de aplicación.
4. Hacer las pruebas de programas de aplicación.
5. Proveer acceso a los datos y pruebas de análisis.

Las personas que intervienen en estas actividades son los diseñadores y desarrolladores de aplicación, representante del negocio, administrador de BBDD, experto en la materia equipo de pruebas, desarrolladores *Web* y *Web Máster*.

**Paso 13: Minería de Datos.**

Muchas organizaciones tienen acumuladas cantidades masivas de datos en sus sistemas operacionales. Estos datos constituyen un potencial origen de valor de información de negocio que puede ser analizada. Con los modelos analíticos generados se pueden encontrar patrones en los datos. Los patrones permiten usar la información como ventaja competitiva en el mercado. Esto da a los gestores y ejecutivos del negocio la información que ellos necesitan para actuar, incrementar beneficios, reducir costes, crear estrategias de productos innovadoras, y expandirse en el mercado compartido.

La minería de datos no es algo que se pueda comprar o adquirir fácilmente, requiere construir una aplicación BI de apoyo a la decisión, especificar una aplicación de minería de datos y usar una herramienta para ello. La aplicación puede usar una combinación sofisticada de clasificación y componentes avanzados como inteligencia artificial, reconocimiento de patrones, BBDD, estadística tradicional, gráficas para representar relaciones ocultas y patrones encontrados en los datos de la organización.

La minería de datos es el análisis de los datos que intenta descubrir “joyas” de información oculta en la gran cantidad de datos que han sido capturados en el curso normal de ejecución del negocio. Esta técnica de análisis tiene diferencias con los análisis convencionales estadísticos que se pueden ver en la Tabla 5.

La minería de datos encuentra respuestas a cuestiones que los responsables de tomar decisiones no saben contestar. Por eso es un componente importante de BI junto con los sistemas de información ejecutiva, herramientas de consulta y reportes, herramientas estadísticas y las nuevas herramientas OLAP.

Las BBDD de entornos BI son el origen de datos más popular para la minería de datos. Estas contienen una riqueza de datos internos que han sido garantizados y consolidados por el negocio, limitados, validados y limpiados por los procesos ETL. Además, pueden contener datos externos valiosos, como regulaciones, información demográfica o geográfica.

| ANÁLISIS ESTADÍSTICO | MINERÍA DE DATOS |
|---|---|
| *Normalmente comienza con una hipótesis.* | *No requiere una hipótesis.* |
| *Tienen que desarrollar sus propias ecuaciones para seguir la hipótesis.* | *Los algoritmos pueden desarrollar ecuaciones automáticamente.* |
| *Sólo usan datos numéricos.* | *Pueden usar diferentes tipos de datos.* |
| *Pueden encontrar y filtrar datos inválidos durante el análisis.* | *Depende de la limpieza y buena documentación de los datos.* |
| *Los resultados no son fáciles de interpretar. Un estadístico debe estar implicado en el análisis de los resultados, y hacérselos llegar a los ejecutivos y gestores de negocio.* | *Interpretan sus propios resultados, y hacen llegar estos resultados a los ejecutivos y gestores de negocio.* |

***Tabla 5.*** *Comparación Análisis Estadístico y Minería de Datos.*

Las técnicas de minería de datos son implementaciones específicas de algoritmos. A continuación se describen las cinco técnicas más comunes:

- Descubrimiento de Asociaciones: se usa para identificar el comportamiento de un determinado evento o proceso. Se descubren asociaciones entre eventos simples mediante reglas de tipo SI x ENTONCES y.
- Descubrimiento de Patrones Secuenciales: es parecido a las asociaciones, descubre enlaces entre eventos, pero en este caso están relacionados, condicionados por el tiempo.
- Clasificación: es la más común. Clasifica según comportamientos o atributos de grupos predeterminados.
- Agrupación: *Clustering*. Se usa para crear agrupaciones dentro de los datos, es igual que la clasificación pero los grupos aun no han sido definidos.
- Predicción: consiste en predecir eventos o valores de datos futuros.

Las operaciones más comunes de minería de datos son los modelos predictivos y de clasificación, el análisis relacional y la segmentación de BBDD.

Todas estas técnicas y operaciones son aplicadas en la gestión de mercado, para la detección de fraude, gestión de riesgos, servicios financieros, distribución, etc.

Las actividades que se realizan durante este paso son:

1. Plantear el problema de negocio.
2. Reunir los datos.
3. Consolidar y limpiar los datos.
4. Preparar los datos.
5. Construir el modelo de datos analítico.
6. Interpretar los resultados de minería de datos.
7. Validar externamente los resultados.
8. Monitorizar el modelo analítico de datos en el tiempo.

Los roles que intervienen en estas actividades son el representante de negocio, experto en minería de datos, administrador de BBDD y experto en la materia.

**Paso 14: Desarrollo del Repositorio *Metadata*.**

Para navegar más eficientemente a través del entorno BI, la gente de negocio necesita tener acceso al repositorio de *metadata*. Solo hay dos opciones, comprarlo o crearlo. Si un repositorio es implementado debe ser mantenido y expandido a lo largo del tiempo. También tiene que ser cargado y modificado durante cada ciclo del proceso ETL, con cargas estadísticas, métricas de datos fiables, contadores de datos rechazados y las razones de estos rechazos.

Cargar el repositorio normalmente no es un esfuerzo manual. Este recibe su *metadata* de diferentes orígenes de *metadata*. Estos orígenes pueden ser ficheros de procesamientos de palabras, hojas de cálculo, herramientas CASE, diccionarios internos de SGBD, herramientas ETL, herramientas OLAP y herramientas de minería de datos. Si se produce algún cambio en la *metadata* de alguno de estos orígenes, el administrador debe informar antes de hacerlo efectivo, así pues este debe colaborar con los administradores de todos estos componentes que participan como orígenes.

Cada repositorio de *metadata* debe tener dos interfaces, una herramienta de interfaz que acepte la *metadata* desde otras herramientas, y otra interfaz de acceso que interactúe con la gente del negocio y técnicos.

Desarrollar un repositorio de *metadata* es tan complicado como cualquier otra aplicación, así pues debe seguir las mismas guías, especialmente en las pruebas. Los gestores de negocio y de tecnologías de la información demandan calidad en sus aplicaciones, pero no quieren gastar mucho tiempo en hacer pruebas. Ellos no saben muy bien cuánto tiempo es necesario, y designan poco tiempo para realizar las pruebas. Se debe programar suficiente tiempo para hacer las pruebas del repositorio de *metadata*, e incluirlo en el plan del proyecto.

De todos los tipos de pruebas para el desarrollo ETL, vistas en el paso 11, en este caso solo son necesarias cuatro pruebas, unitarias, de integración, de regresión y de aceptación.

Se debe preparar la puesta en producción del repositorio antes de tener todo el código y las pruebas realizadas. Para ellos hay que preparar todos los elementos que intervienen, como son la plataforma del servidor, producción del SGBD, librerías de programas y consultas, seguridad, manuales y guías de instrucciones y formación del repositorio de meta datos para la gente de negocio y técnicos.

Las actividades que hay que realizar en este paso son:

1. Construir la BBDD del repositorio de *metadata*.
2. Construir el proceso de migración de *metadata* y las pruebas unitarias.
3. Construir la aplicación de *metadata* y las pruebas unitarias.
4. Probar el repositorio de datos, los programas o funciones de productos.
5. Preparar el repositorio para producción.
6. Proveer formación del repositorio.

Los roles que intervienen en este paso son el representante de negocio, administrador de BBDD, administrador de meta datos, desarrolladores del repositorio de meta datos y equipo de pruebas.

### *Estado 6: Desarrollo*

En este estado se debe implementar y evaluar el producto acabado, y entonces medir su efectividad para determinar si la solución encuentra, excede o no llega al resultado esperado. Este estado consta de dos pasos que deben llevarse a cabo de forma secuencial, como puede observarse en la Figura 15.

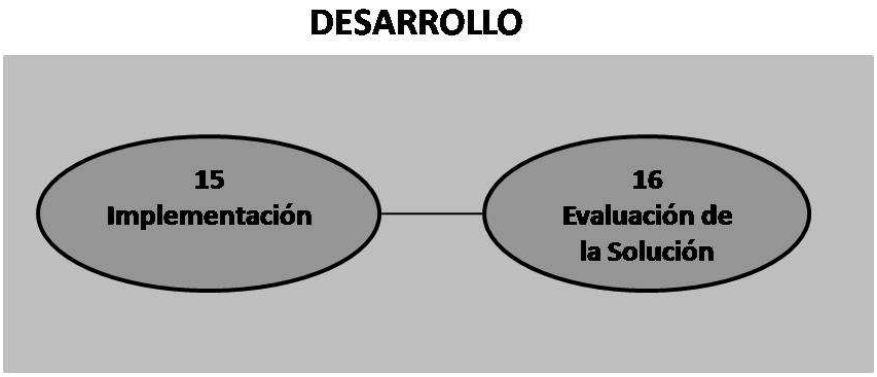



***Figura 15.*** *BIR - Estado 6: Desarrollo.*

**Paso 15: Implementación.**

Una vez que la aplicación está construida y probada está lista para ser implementada en el entorno de producción. Esto se puede hacer de formas, todo de una vez, de forma tradicional, o incrementalmente.

El método incremental reduce el riesgo de la exposición de los defectos potenciales en la aplicación BI de toda la organización. Esto da la oportunidad de mostrar informalmente los conceptos BI y las herramientas BI a la gente de negocio que no estaba directamente implicada en el proyecto.

Se sugiere empezar por un pequeño grupo de personas de negocio, que consisten en los principales usuarios, los trabajadores sin conocimiento de la tecnología y los analistas de negocio. Se debe tratar a la gente de negocio como clientes, manteniendo su cuidado, implementar el sistema dándoles formación sobre él, y apoyo continuado. Hay que aprovechar el acercamiento de los usuarios a la aplicación para probarla, y así ver si hay que modificar algo. Por último se deben trasladar estas acciones al resto de equipos en la organización.

La seguridad debe estar probada durante la primera implantación. La seguridad se pasa por alto o se le da una atención superficial. Las medidas de seguridad deben

ser informales para algunos de los datos, pero no para todos. Esta requiere haber sido implementada a través de varias características de seguridad de los SGBD y de las herramientas de acceso y análisis usadas por la aplicación BI. La solución de imponer seguridad a nivel de tabla puede no tener la suficiente granularidad, se puede conseguir haciendo una partición de la tabla a nivel lógico y físico, y así restringir el acceso sólo al distribuidor apropiado según sea tabla de hechos o tabla de dimensión. Otra alternativa puede ser mejorar la *metadata* con definiciones de parámetros de los datos, así se puede controlar el acceso a los datos dependiendo de la identidad del distribuidor, con programas lógicos. Esta medida de seguridad será tan buena como sea el programa que lo controla.

Se debe tener cuidado con los paquetes de seguridad que se adquieren, el personal se puede frustrar al tener que usar muchos identificadores y contraseñas diferentes, y que caducan en momentos diferentes.

Implementar las medidas de seguridad en un sistema descentralizado es más difícil que en uno centralizado, pues es mucho más fácil controlar un punto de acceso que múltiples. Pero en un entorno BI mantener todos los datos en un lugar central no es siempre posible o deseado. Para este tipo de entornos distribuidos, primero se deben buscar los puntos finales en la arquitectura de la red de trabajo y los caminos que los enlazan, se puede hacer un dibujo de la arquitectura física. Después se deben determinar los caminos de conectividad usados para llegar a los datos. Y por último comparar estos caminos con las medidas de seguridad. Es recomendable hacer una matriz con todos los caminos de conectividad cruzados con todos los paquetes de seguridad que se poseen, y ver si existe esa seguridad.

También es muy importante la seguridad para el acceso a Internet. Se debe controlar la autenticación de los usuarios, su autorización, si podrán acceder a unos registros y a otros no, y el cifrado de los datos para su transmisión por la red.

Se debe hacer un *back-up* correcto y apropiado, para garantizar la permanencia de los datos frente a posibles riesgos de pérdida.

Otro aspecto que se debe considerar es el crecimiento considerable de los datos, éstos duplicarán su volumen en dos años. Pero no sólo crecen los datos, sino la usabilidad, el número de los usuarios que acceden al sistema, y el hardware. Hay que tener en cuenta y prever todos estos aspectos para garantizar la consistencia permanente.

Las actividades que hay que realizar en este paso son:

1. Planear la implementación.
2. Preparar el entorno de producción.
3. Instalar todos los componentes de la aplicación BI.
4. Poner el esquema en producción.
5. Cargar las BBDD de los repositorios *metadata,* el *Stagging Area* y el DW.
6. Preparar el soporte en curso.

Las personas que intervienen en este paso son los desarrolladores y diseñadores de la aplicación, expertos en minería de datos, administrador de BBDD, desarrolladores y diseñadores ETL, administrador de *metadata*, desarrolladores del repositorio, desarrolladores *Web* y *Web Máster*.

**Paso 16: Evaluación de la Solución.**

La construcción de un sistema BI es un proceso que nunca acaba, a diferencia de muchos sistemas operacionales. Los objetivos y necesidades cambian, por esta razón el sistema BI también.

Es difícil que la primera solución sea la mejor, normalmente hacen falta varias iteraciones para que sea la más adecuada. Muchas organizaciones no aceptan este hecho cuando el sistema se desarrolla por la misma organización, sin embargo cuando es comprado a un vendedor sí se acepta. En ese caso la solución también necesitará pasar por varias iteraciones antes de ser una buena solución, y además su precio será mayor al no hacerse de forma interna en la organización.

Las organizaciones deben concienciarse de que este tipo de sistemas BI son así, independientemente de quién lo lleve a cabo. Para ello se puede seguir una guía con aspectos que se deben tener en cuenta, entre los que están:

- Las ampliaciones deben ser entregadas cada tres o seis meses, menos la primera que puede tomar más tiempo.
- Los entregables deben ser pequeños y manejables.
- Las pequeñas soluciones en conjunto formarán la aplicación BI completa, una de ellas no puede ser igual a la solución completa.

- La primera solución BI sólo debe contener lo básico.
- Los gestores de negocio deben aceptar las soluciones parciales.
- Nada es impuesto específicamente, todo es negociable.
- La infraestructura, técnica y no técnica, debe ser robusta.
- La *metadata* deben ser una parte integral de cada pequeña solución.
- Diseños, programas y herramientas deben ser flexibles para soportar ocasionales rediseños.
- Los nuevos requisitos deben ser tratados y priorizados.
- Pequeños errores o defectos son direccionados bajo estrictos procedimientos de control de cambios durante el desarrollo de cada parte de la solución completa.
- Grandes errores o defectos son aplazados a otra nueva iteración, eliminando la función o datos asociados con el problema.

Después de la implementación del proyecto BI se debe hacer una revisión, para ver si la aplicación funciona perfectamente o tiene problemas. Es bueno hacer esta revisión en conjunto con otros equipos de otros proyectos, así se puede aprender más, ya que se compartirán conocimientos.

Algunos tópicos a revisar son si la planificación ha sido seguida correctamente, el presupuesto ha sido realista y se ha cumplido. También analizar si se está satisfecho con la habilidad de negociación, el personal, si se ha perdido gente clave en el proyecto, si la habilidad y formación del personal ha sido adecuada, la consecución del desarrollo, contratos, consultores y vendedores.

Se debe organizar una sesión de revisión después de la implementación. Para ello hay que preparar la reunión, decidir cuándo programarla, cuánto tiempo durará, dónde se hará, quién asistirá y qué se discutirá durante la sesión. La sesión debe estar muy estructurada y seguir un procedimiento predeterminado, dónde se indique el flujo que tendrá, por qué participantes pasará.

Las actividades de este paso son:

1. Preparar la revisión después de la implementación.

2. Organizar la reunión para la revisión.

3. Conducir la reunión.

4. Seguir la reunión según lo planificado.

Los roles que intervienen en estas actividades son el desarrollador y diseñador de la aplicación, arquitecto de infraestructuras BI, representante de negocio, patrocinador del negocio, administrador de los datos, experto en minería de datos, analista de calidad de los datos, administrador de BBDD, desarrolladores ETL, revisor (que no ha participado en la construcción del proyecto, sólo en la revisión), administrador de meta datos, gestor del proyecto, escribiente (no participa en el proyecto, solo documenta la sesión de revisión), *Stakeholders*, experto en la materia y *Web Máster*.

En la Figura 16 se muestra un diagrama de todos los pasos que se deben realizar en cada estado del desarrollo y sus dependencias, así se podrá ver qué actividades pueden ser realizadas en paralelo y cuáles no.

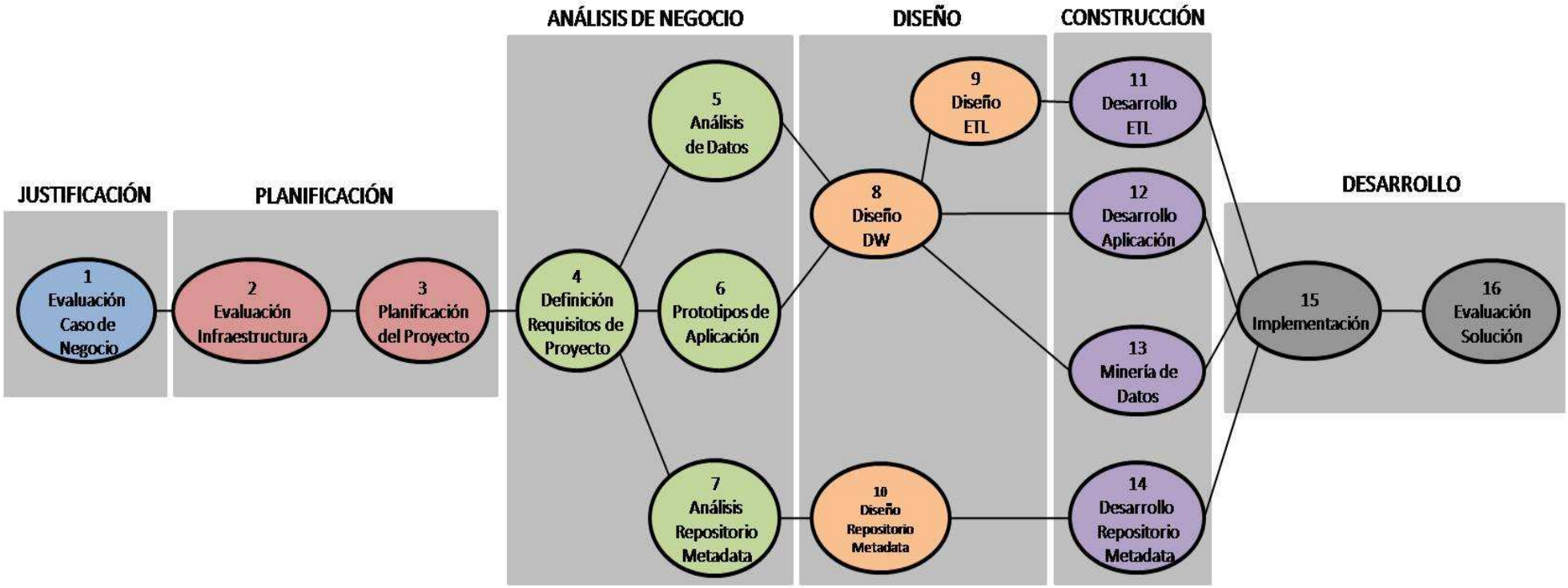



***Figura 16.*** *Dependencias de pasos en el desarrollo del BIR.*

## 3.6 Herramientas de Business Intelligence

En el ambiente BI intervienen muchos componentes, y hacen falta procesos para poder conectar unos con otros, acceder a ellos y mostrar la información. En todos estos procesos intervienen herramientas ETL, OLAP, de análisis de datos, de presentación de informes y bases de datos. Se van a comentar algunas de estas herramientas, principalmente las de código abierto, pero no todas las que existen en el mercado, ya que hay muchas y están continuamente apareciendo y despareciendo. En la Figura 17 se muestra una valoración del panorama actual del BI comercial, aunque este puede ser diferente dependiendo de la fuente dónde se consulte, pues una valoración siempre es algo subjetiva. En este caso se ha obtenido de [4].


***Figura 17.*** *Panorama actual BI Comercial.*

A la hora de elegir una herramienta es necesario analizar todos los aspectos, las prestaciones que ofrece y las que realmente se necesitan en un proyecto particular. Muchas veces se compran licencias de herramientas con muchas prestaciones que en el proyecto donde se pretenden usar no se requieren. También se debe tener en cuenta el precio de la licencia, en algunos casos quizás sea preferible utilizar software

libre en vez de herramientas comerciales. Con el fin de conocer mejor este tipo de herramientas de software libre se van a ir comentando en los siguientes apartados algunas de las existentes actualmente.

### 3.7.1 Herramientas de tratamiento de Datos (ETL)

Estas herramientas son las encargadas de realizar los procesos de extracción, transformación y carga ETL. Estos procesos se pueden realizar directamente sobre los sistemas operacionales o sobre el *Stagging Area*. En el caso práctico de este proyecto se ha utilizado la herramienta *Power Center* de Informatica, pero existen muchas más, y a continuación se comentan algunas de ellas:

- *Clover:* es un entorno de código abierto basado en java, para datos estructurados, capaz de funcionar como aplicación independiente o estar incluida en otra aplicación.
- *Enhydra Octopus:* sólo soporta fuentes de datos con controladores JDBC e incluye también controladores especiales que permiten la conectividad con archivos CSV, XML, MS-SQL y archivos propietarios.
- *Kettle:* conocido actualmente como *Pentaho Data Integration*, un proyecto belga de código abierto que ha sido adoptado por la plataforma Petaho BI. Uno de sus objetivos es que el proceso de ETL sea fácil de generar, mantener y desplegar.
- *Talend:* es una herramienta de código abierto. Su interfaz gráfica de usuario está basada completamente en eclipse RCP. Incluye numerosos componentes para procesos de modelado de negocios.

Estas son las más usadas, pero hay muchas más, como por ejemplo Paquel ETL, CpluSQL, JetStream, OpenDigger.

### 3.7.2 Desarrollos OLAP

Las herramientas OLAP (*On-Line Analitical Processing*) se encargan de analizar y explotar los conjuntos de datos almacenados en el DW. En el caso práctico de este proyecto se ha utilizado la herramienta comercial *Busines Objetcs*, recientemente adquirida por *SAS*. Algunas de las herramientas disponibles en el mercado son las siguientes:

- *Mondrian:* es uno de los componentes más antiguos de BI código abierto. También se usa como el motor OLAP en otros softwares de fuente abierta de OLAP y *BI Suite*. Actualmente forma parte del proyecto *Pentaho*. Es un motor ROLAP desarrollado en Java y se encarga de recibir y responder a consultas dimensionales en lenguaje MDX. Para acceder a las funcionalidades de Mondrian se requiere una aplicación cliente.

- *CubeDesigner:* es un entorno gráfico que permite diseñar un documento XML que representa un cubo o hipercubo para analizar la información almacenada. Este esquema es interpretado por Mondrian para obtener la información de las consultas de MDX.

- *JPivot:* es uno de los posibles clientes para Mondrian, hace de *front-end* sobre el motor de Mondrian. Es una librería de Java Server Pages (JSP) personalizados que soporta XMLA. No utiliza las *API's* de Mondrian directamente, implementa su propio modelo OLAP.

- *JRubik:* es un cliente OLAP realizado en *Java/Swing* y basado en los componentes de JPivot. Es capaz de conectar fuentes OLAP basadas en Mondrian.

- *Jedox PALO (JPalo):* es un servidor de código abierto de bases de datos multidimensional capaz de centralizar y administrar casi un número infinito de hojas de cálculo. El sistema opera en tiempo real.

- *OLAP4J (OnLine Analitical Processing for Java):* es una API para el entorno Java 2 EE, que soporta la creación, almacenamiento y administración de datos para una aplicación OLAP. Hyperion, IBM y Oracle iniciaron su desarrollo con la intención de que fuera un equivalente a la conexión JDBC pero específica para OLAP.

### 3.7.3 Entornos de Desarrollo para Dashboards

Estos entornos se encargan de crear y manejar cuadros de mando. Automatizan de forma eficaz actividades tales como: uso de repositorios de documentos compartidos, gestión de flujos de trabajo e información, notificaciones al grupo, foros de discusión diferidos e interactivos, votaciones electrónicas, reuniones a través de la red con videoconferencia, gestión de agendas compartidas, etc. Actualmente la mayoría de los portales, corporativos y públicos, soportan la creación de servicios para grupos y comunidades.

El rol de un portal es la integración de aplicaciones a nivel de interfaz de usuario, que disponga de un acceso personalizado a tales aplicaciones o servicios y que permita su uso tanto en entornos de Internet como intranet. Los portales pueden ser de primera generación, los descritos hasta ahora, y de segunda generación, que además incluyen *portlets*. Un *portlet* es una mini-aplicación web interactiva que devuelve fragmentos de *markup (lenguaje de marcado o marcas, HTML, XML)*.

Algunas de las plataformas incluidas dentro de este marco son:

- *Jboss:* es un servidor de aplicaciones J2EE de código abierto implementado en Java, por lo que puede ser usado en cualquier sistema operativo que lo soporte.
- *Jboss Portal:* es una plataforma de código abierto para albergar y servir un interfaz de portales Web, publicando y gestionando el contenido así como adaptando el aspecto de la presentación.
- *JetSpeed:* es un portal de información empresarial de tipo abierto, escrito completamente bajo la licencia O*penSource* Apache en Java y XML. Los *portlets* individuales pueden ser agregados para crear una página. Cada *portlet* es una aplicación independiente actuando JetSpeed como centro.

### 3.7.4 Bases de Datos

Los sistemas gestores de bases de datos más conocidos y utilizados son *Oracle* y *MySQL*, aunque existen otros. Este tipo de sistemas son más clásicos y comúnmente utilizados, no sólo en entornos BI, por lo que no se va a entrar más en detalle.

### 3.7.5 Soluciones Integradas

Estos nuevos desarrollos, conocidos como *Appliance*, pretenden quitar complejidad en la implementación de un DW. Un DW *Appliance* es un conjunto integrado de servidores, discos de almacenamiento, Sistemas Operativos, Bases de Datos y Software, preinstalados y preparado para montar y hacer funcionar un DW. La diferencia con las soluciones completas es que en este caso no se necesitan otros complementos, mientras que en la soluciones completas sí, como por ejemplo el motor de BBDD.

Se ha empezado a aplicar el término a soluciones en las que las combinaciones priman sobre todos los componentes de Software. Este nuevo desarrollo es capaz de reducir el coste total, mejorar el rendimiento, reducir el tiempo dedicado a la administración, mejorar la disponibilidad del sistema ante caídas, garantizar la escalabilidad en rendimiento y en capacidad, y un rápido retorno de la inversión.

Dentro de estos nuevos desarrollos hay varios tipos.

- *Native data warehouse appliance:* tanto el hardware como el software están estrechamente integrados en una sólo plataforma, no se pueden licenciar ni utilizar de forma separada, individualmente. Algunas de las herramientas de este tipo son *DATAllegro, Netezz y Teradata.*

- *Software data warehouse appliance:* en este caso, bases de datos relacionales, de código abierto o comercial, son optimizadas para su uso en entornos DataWarehouse, pudiéndose utilizar en diferentes configuraciones de hardware. Algunos ejemplos son *Greenplum, Sybase* e *Ingres.*

- *Packaged data warehouse appliance:* donde software y hardware comercial es configurado para funcionar como una única plataforma y que es comercializado por un único vendedor. Además se instala y mantiene

en un único sistema. Ejemplos de este tipo son: *HP Neoview, IBM Balance Warehouse* y *Sun/Greenplum.*

- *Data management appliance:* obtiene los datos de forma intensiva desde un servidor. Estos sistemas pueden implicar procesos operacionales, analíticos o de almacenamiento. Algunos ejemplos de estos son *ParAccel* y *Dataupia.*

### 3.7.6 Soluciones Completas

El proceso de elegir una plataforma y una solución de BI es un proceso que consume mucho tiempo y dinero. Este coste se puede reducir utilizando soluciones de código abierto (*open source*) y/o soluciones completas, con las cuales se pueden llevar a cabo proyectos BI completos. Las soluciones completas cubren todas las necesidades del proyecto, ETL, OLAP, *Dashboard* y BBDD. Algunas de estas soluciones de código abierto son:

- *Pentaho BI:* es una plataforma de BI orientada a la solución y centrada en procesos. Esta solución está compuesta por *Mondrian* (servidor de OLAP), *JFreeReport* (diseñador de informes), *Kettle* (integración de datos, ETL), *Pentaho* (plataforma de inteligencia empresarial) y *WEKA*(data mining).

- *Jasper Intelligence:* desarrollado por Jasper, sus componentes son *JasperServer, JasperReports, JasperDecisions, JasperAnalysis* y *JasperETL.*

- *BIRT:* no se puede llamar suite completa a este proyecto, ya que *Birt Report Designer* es un entorno de desarrollo basado en ECLIPSE, que genera reportes con un diseño de salida en XML, lo cual facilita la interacción con otras interfaces de salida como así también provee una gran capacidad para generar gráficos.

- *SpagoBI:* está desarrollado en Java, pretende ser una solución completa de BI que incluye desde la extracción a la minería. Utiliza componentes muy similares a los de Pentaho.

# Capítulo 4

# CASO PRÁCTICO: GESTIÓN DE PÓLIZAS

## 4.1 Introducción

Este caso práctico consiste en la construcción de un sistema BI siguiendo la metodología que se explicó en el estado del arte. Se irán desarrollando todos los pasos necesarios para conseguir los objetivos propuestos.

Con este caso práctico se afianzarán los conceptos descritos en el estado de arte. Se siguen todos los pasos que indica la metodología descrita con la finalidad de conseguir los siguientes objetivos:

- Proporcionar información sobre el ratio de la prima y el ratio del número de pólizas por dirección regional por año.
- Proporcionar información sobre el ratio de la prima y el ratio del número de pólizas por producto, sector y año.
- Hacer un estudio para la predicción del incremento de las primas de las pólizas por semestre y por trimestre, con la finalidad de evaluar los resultados y decidir si son aceptable o no.

Para poder cumplir estos objetivos será necesario hacer un modelado dimensional orientado a ellos, a partir del cual se pueda obtener la información requerida. Se crea un DW que dará soporte a los informes que se exigen. Además, se generan los datos correspondientes para poder hacer el estudio de predicción, que se realiza con Minería de Datos, usando la herramienta Weka [21].

## 4.2 Aplicación Business Intelligence Roadmap (BIR)

Anteriormente, en el estado del arte, se explicó el BIR, y ahora se ve desde el punto de vista práctico. Se describe cada paso aplicado al caso práctico que se ha desarrollando en este PFC, con el objetivo de clarificar qué se debe hacer en cada uno de ellos. Se debe tener en cuenta que todos los pasos no han sido desarrollados, pues hay algunos que quedan fuera del alcance del PFC.

## 4.3 Paso 1: Evaluación Caso de Negocio

En este apartado se debe hacer un estudio de las necesidades de negocio que llevan a la implementación de este proyecto, incluyendo el presupuesto detallado. Pero en este caso sólo se describen necesidades por las que se llevará a cabo, sin incluir el presupuesto ni acciones financieras, ya que esto está fuera del alcance del PFC.

La Compañía de seguros Pepito S.L. se encarga de la realización de pólizas de seguros de los denominados “no vida”. Sus clientes son tanto particulares como empresas que quieren asegurar alguno de sus bienes con la compañía.

En la Figura 18 se muestra y explica el ciclo de vida de una póliza, y las características de gestión de éstas dentro de la política de Pepito S.L..

En la Figura 18 se puede observar que por medio de los diferentes canales (centro emisión, oficina, *Web* o *Call Center*) una póliza puede ser creada o modificada a lo largo del tiempo, hasta que se cancela o se acaba el periodo de vigencia sin renovación posterior. Por cada modificación se crea un nuevo registro en el acta de modificaciones de la póliza con los datos vigentes de la póliza, que son lo que se analizarán posteriormente para la creación de los informes.

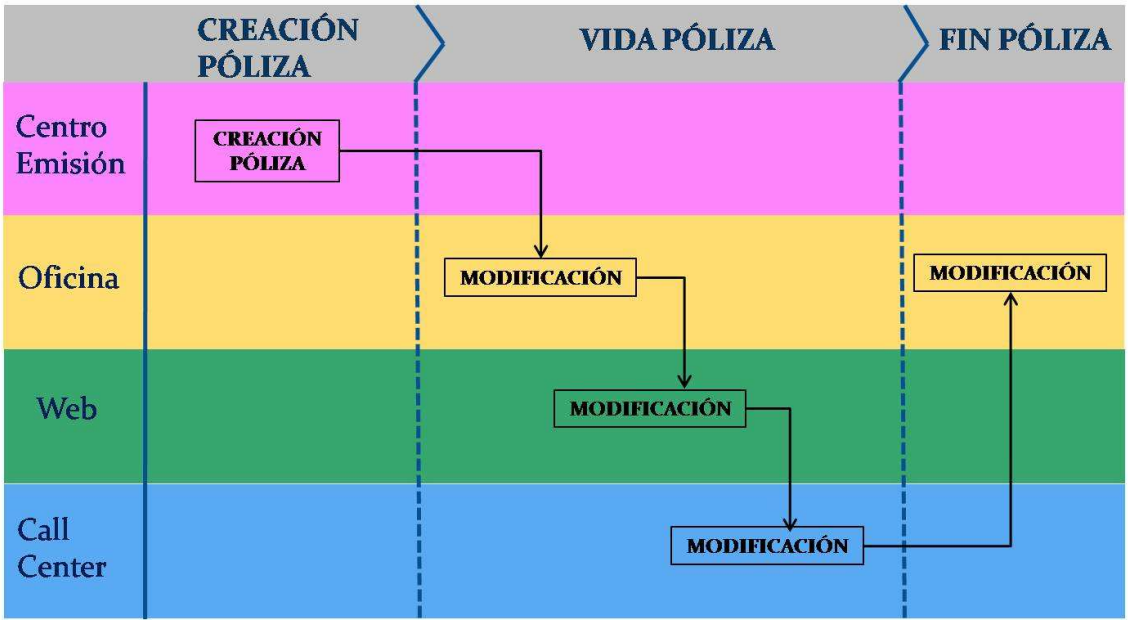



***Figura 18.*** *Ciclo de Vida de una Póliza de Seguros.*

Por cada modificación se hace una nueva inserción en el registro del acta, no se hacen actualizaciones en los mismos, por lo que quedan registrados todos los movimientos que han sufrido las actas.

Se han hecho los informes con respecto a los productos y a los comerciales:

1. Informes basados en sector de productos:

   Los productos no cambian de un año al otro, puede cambiar condiciones, precio, tomador. Sus valores sufren un cambio lento en el tiempo. Se estudia la relación de factores determinados de las pólizas vigentes con respecto a los del año anterior.

   ```
   Prima total = Prima comercial + Prima impuestos
   ```

   Los campos que hay que calcular a partir de campos dados de la base de datos se calcularan en el momento de crear el DW en los procesos ETL, añadiéndolos posteriormente al *DW*. Estos campos han de tenerse en cuenta en la fase de diseño del DW.

2. Informes basados en dirección regional:

   La compañía de seguros realiza la demarcación territorial por una estructura comercial de tres niveles, el informe se obtiene a partir del nivel más alto, Dirección Regional.

Anualmente la compañía de seguros Pepito S.L. estudia la situación de su estructura regional, siendo posibles la reestructuración de agentes a otras demarcaciones dependiendo de los cambios que estas hayan sufrido. Cada agente posee una cartera de clientes, la cual les pertenece a ellos. Por lo que cada vez que un agente cambia de Dirección o Coordinación, se lleva su cartera con él.

Cuando se reestructuran las coordinaciones se sopesan los movimientos que ha habido a lo largo del tiempo recolocando los agentes dentro de estas, nivelando las nuevas necesidades que puedan surgir en las distintas demarcaciones territoriales. Estas se suelen hacer de la forma menos traumática posible para los agentes comerciales, apoyándoles para que estén contentos con Pepito S.L. y de forma totalmente transparente para el cliente final.

Cada vez que se emite una nueva acta (suplemento) al registro se le añade los códigos del actual agente, coordinación y dirección, etc. Este registro no es nunca modificado. Cuando la póliza sufra cualquier otra alteración se volverá a añadir un nuevo registro a la base de datos, quedando así un histórico de movimientos de cada póliza.

Es importante recalcar que sobre esta tabla no se realizan actualizaciones, como se ha comentado anteriormente cada registro no se modifica una vez añadido.

Por otro lado, la compañía desea realizar un estudio para determinar si es posible predecir el incremento que sufrirán las pólizas por trimestre y semestre.

## 4.4 Paso 2: Evaluación de la Infraestructura

Como se explicó con anterioridad, en este paso se debe describir con total claridad la infraestructura técnica y no técnica que será requerida para el proyecto.

En este caso no es necesario porque solo es un caso de prueba, pero en otros proyectos reales de este tipo sí sería necesario. Se deberían evaluar la plataforma *hardware*, *middleware*, sistemas de gestión de BBDD, así como la infraestructura no técnica como estándares, reglas y políticas de negocio.

Si la plataforma existente no cumple con los requisitos mínimos exigidos es aquí cuando se deben seleccionar nuevos productos que sí los cumplan y expandir la plataforma actual.

## 4.5 Paso 3: Planificación del Proyecto

Para este caso práctico no ha sido necesario hacer una planificación detallada, como sí lo es en proyectos reales en este dominio.

Dentro de la planificación se debe dejar claro qué será entregado, cuándo se hará, cuánto costará y quién lo hará.

En primer lugar se debe hacer una descomposición del trabajo que será realizado en actividades y tareas. Habrá que estimar el esfuerzo en horas para estas actividades y tareas, además de asignarles recursos. También es importante determinar las dependencias entre actividades, tarea y entre recursos, ya que así se podrá trazar la trayectoria crítica basada en estas dependencias. Y por último hay que crear el plan detallado.

Este paso no se ha llevado a cabo una planificación detallada debido a que se excedía la dimensión del PFC. La planificación se ha hecho a grandes rasgos y no tiene importancia que sea comentada con mayor extensión.

## 4.6 Paso 4: Definición de Requisitos de Proyecto

La definición de requisitos debe ser exhaustiva y muy clara, ya que de ella dependerá la solución que se cree. La solución final de un proyecto tiene que cubrir por completo las necesidades de la organización, y esto no será posible si los requisitos no están bien definidos.

Existen dos tipos de requisitos, los generales y los específicos del negocio. Para hacer la captación de éstos hay muchos métodos que se pueden llevar a cabo, como son entrevistas, cuestionarios, observaciones, etc.

Los requisitos tienen que ser revisados a lo largo del proyecto, ya que las necesidades del negocio pueden cambiar durante el proyecto, por lo que los requisitos también deben cambiar.

En este caso práctico no se ha hecho una definición de requisitos, únicamente se deben cumplir los objetivos definidos, pues no es una solución para ninguna organización.

## 4.7 Paso 5: Análisis de Datos

En este apartado se ha hecho un análisis de los datos origen. Describiendo el modelo que siguen, tipos, relaciones, limitaciones, etc.

En primer lugar se debe decir que los datos que componen el sistema operacional están en un archivo Excel, y se van a pasar a una BBDD Access para crear el *Staging Area*. Y será a esta *Staging Area* dónde accederán los procesos ETL que cargarán el Data *Warehouse*. Este es el único origen de datos que existe.

Los datos contenidos en el archivo Excel han sido facilitados por una persona externa a la universidad muy implicada en el sector. Esta persona ha dado total autorización para su utilización, sabiendo que los datos no son reales.

Para comprender qué datos contienen las tablas se muestra la Tabla 6, que describe cada tabla del sistema operacional. Esta tabla no se ha podido incluir en el anexo porque es necesario que el usuario conozca las tablas que se irán comentando y utilizando a lo largo de todo el capítulo.

| ENTIDAD/TABLA | DESCRIPCIÓN |
|---|---|
| **A3000030 (Datos Generales de Pólizas)** | En esta tabla además de ser la tabla operacional donde se registran todos los movimientos de las pólizas están los tipos de renovación, las primas, el capital asegurado y las fechas de inicio del suplemento. |
| **A3000060 (Datos Generales de Terceros)** | Tabla de registros de los Datos Personales de cada tercero así como su tipo (asegurado, tomador, empleado, agente). |
| **A1000100 (Datos Direcciones Regionales)** | Es la tabla del nivel 1 de la estructura comercial que por ahora es una tabla con las Comunidades Autónomas. |
| **A1000101 (Datos Coord. Regionales)** | Es la tabla del nivel 2 de la estructura comercial que por ahora es una tabla con las Provincias. |

| ENTIDAD/TABLA | DESCRIPCIÓN |
|---|---|
| **A1000102 (Datos Agentes)** | Tabla de registros de los movimientos de los agentes que llama a la tabla A3000060 para los Datos Personales y dice si el agente permanece en su Coordinación Regional o Dirección Regional o ha renovado en ella. |
| **A1800001 (Datos Centro de Coste)** | Es la tabla de los Canales de Información. |
| **A1800002 (Datos de Localidades)** | Es la tabla con las Localidades |
| **A1800003 (Datos de Municipios)** | Es la tabla con los Municipios |
| **A1800004 (Datos de Provincias)** | Es la tabla con las Provincias. |
| **A1800005 (Datos de Regiones Autónomas)** | Es la tabla con las Comunidades Autónomas. |
| **A1800012 (Datos de Códigos Postales)** | Tabla con los Códigos Postales dependiendo de la localidad. |
| **A1700000 (Datos de Garantías de Producto)** | Tabla con los tipos de Garantías. |
| **A1700001 (Datos de Productos)** | Tabla con la descripción de los productos. |
| **A1700002 (Datos de Sectores de Producto)** | Tabla con la descripción de los sectores de productos. |
| **A1700004 (Datos de Tipo de Suplemento)** | Tabla con la descripción de los tipos de suplementos. |
| **A1700005 (Datos de Sub-tipo de Sup.)** | Tabla con la descripción de los sub-tipos de suplementos. |

***Tabla 6.*** *Descripción de Tablas del Sistema Operacional.*

Estas son todas las tablas que forman parte del modelo entidad/relación del sistema operacional, pero no todas ellas son utilizadas para la creación del DW. Las tablas que no se usan son:

- A1700000 (Datos de Garantías de Producto)
- A1800012 (Datos de Códigos Postales)
- A1800002 (Datos de Localidades)
- A1800003 (Datos de Municipios)
- A1800004 (Datos de Provincias)
- A1800005 (Datos de Regiones Autónomas)

El motivo por el que estas tablas no se usan es que cuando se crea un DW sólo se deben incluir las dimensiones que serán utilizadas, necesarias. En este caso se ha creado un *Data Mart* del DW para cubrir las necesidades del negocio, crear los informes solicitados. No obstante, en un futuro se pueden crear nuevos *Data Mart* que necesiten nuevas dimensiones que requieran obtener datos de estas tablas que ahora no son utilizadas. Es importante tener claro que en un DW siempre se pueden crear nuevas dimensiones y nuevas tablas de hechos, pero las que ya estén creadas no pueden ser modificadas. Es por esto que hay que poner especial interés en definir correctamente qué columnas tendrá cada dimensión y tener una visión futura sobre qué podrá ser necesario más adelante que ahora no lo es.

También se ha podido comprobar que los datos están limpios, y será más adelante, durante el diseño de la nueva BBDD y el diseño ETL dónde se tengan en cuenta los campos vacíos o erróneos, y cómo se tratan, siempre y cuando sea necesario hacerlo.

Para comprender mejor el modelo se va a representar gráficamente en la Figura 19, así se pueden apreciar las relaciones existentes entre las diferentes tablas:

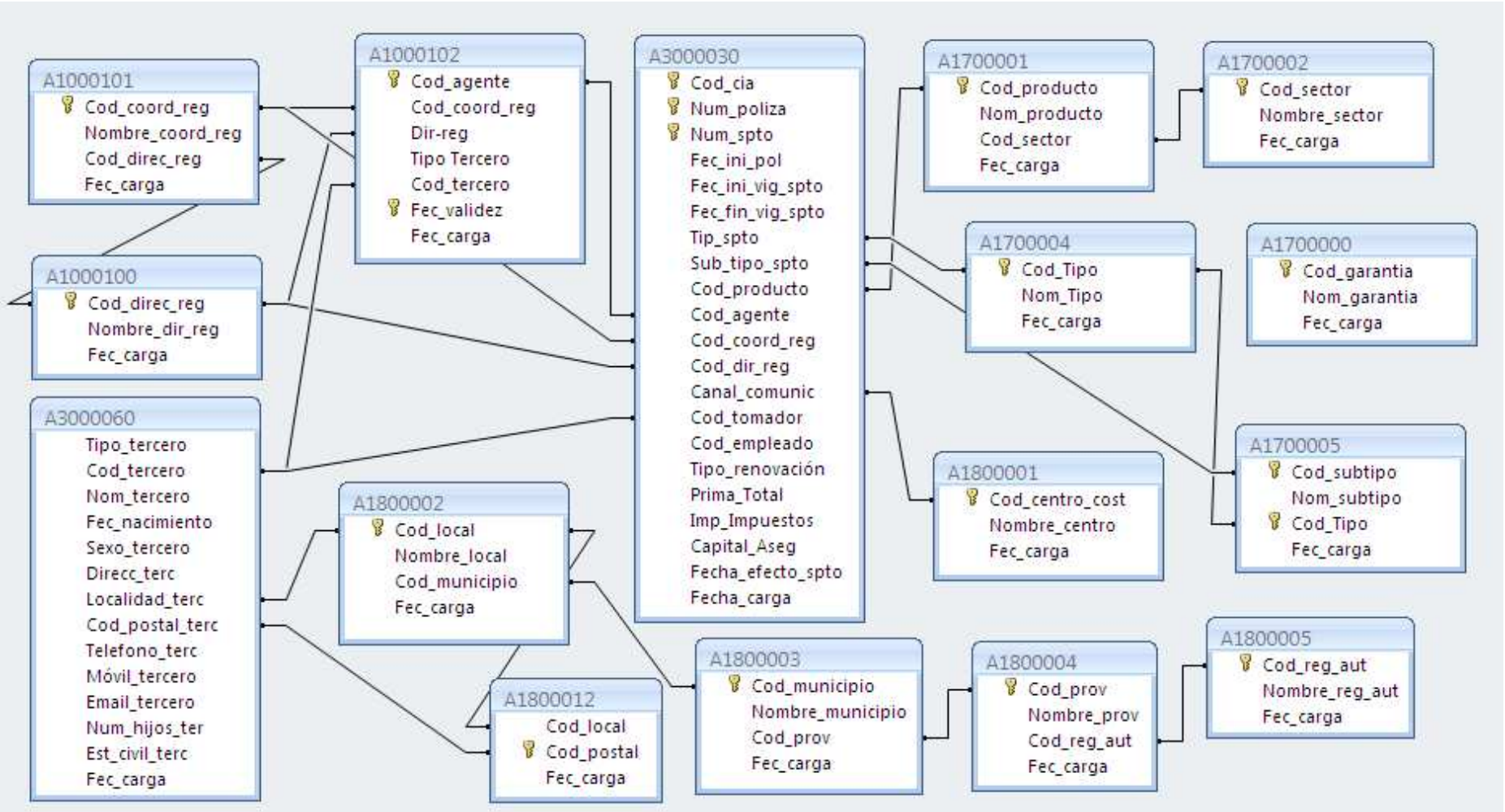



***Figura 19.*** *Modelo Entidad Relación del Sistema Operacional.*

## 4.8 Paso 6: Prototipo de Aplicación

En un proyecto real de BI no es obligatorio realizar este paso, pero es muy recomendable. Hacer prototipos es muy útil para validar y añadir nuevos requisitos, encontrar partes que faltan, capacidad de la tecnología, etc.

Es recomendable hacer prototipos pequeños, que solo representen una parte de todos los requisitos, así es más fácil que la gente del negocio se centre en ellos y puedan analizarlos con mayor facilidad y se involucren más en el proyecto.

En este caso práctico no se han creado prototipos debido a que la extensión del mismo no lo requiere.

## 4.9 Paso 7: Análisis del Repositorio de Metadata

El repositorio de *metadata* es muy importante para los usuarios. Éste les facilitará la comprensión del modelo de datos, de su flujo y acceso. Teniendo acceso a este repositorio los usuarios sabrán por sí mismos dónde deben ir a buscar la información que necesitan.

Se trata de una arquitectura multi-capa con una *Staging Area*. El DW está compuesto por un único *DM*. El acceso a los datos se hace a través del *DM* y de los dos informes predefinidos en la evaluación del caso de negocio. Esta arquitectura se muestra gráficamente en la Figura 20.

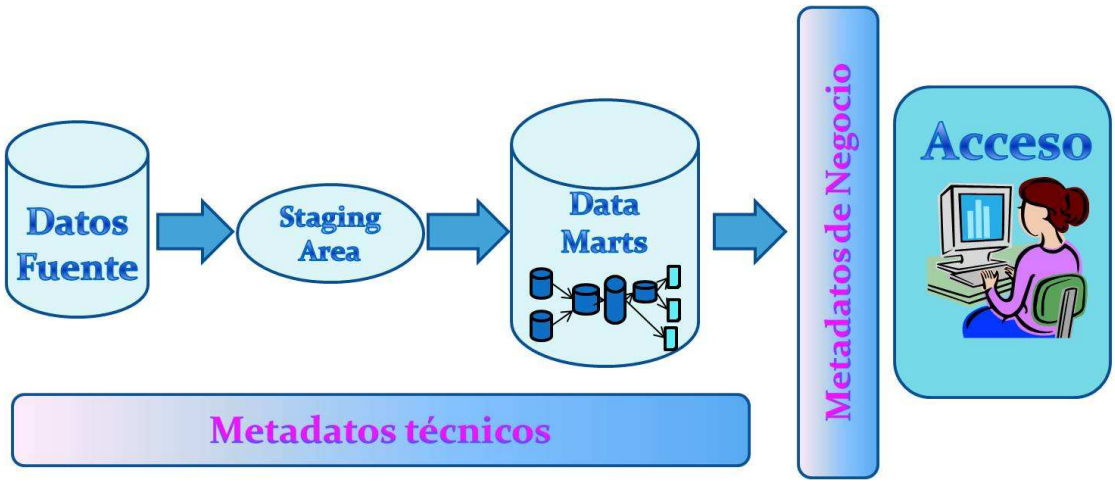



***Figura 20.*** *Intervención de la metadata en la arquitectura BI.*

La capa de metadatos está, en esta arquitectura, dividida en dos grupos:

- *Metadata* Técnica

Tienen todos los aspectos técnicos del proyecto, como diccionarios de datos de los repositorios de datos DM, relaciones fuente-destino y todas las transformaciones de datos, la periodicidad de carga de cada tabla, el histórico existente, etc. Es toda la información técnica acerca de los datos disponibles a los usuarios. En este caso esta información no se va a detallar aquí, ya que las transformaciones de los datos, las relaciones entre origen-destino se crearán automáticamente con la herramienta utilizada para el proceso ETL.

- *Metadata* de Negocio

Datos útiles para la consulta de la información por parte de los usuarios. Algunos ejemplos de *metadata* son los requisitos de negocio, las definiciones inherentes a cada indicador y sus perspectivas de análisis, su fuente y cálculos efectuados (reglas).

La Tabla 7 muestra conceptos de negocio que forman parte de la *metadata*:

| DENOMINACIÓN | DESCRIPCIÓN |
|---|---|
| Acta | Alteración de los datos de una póliza, creación, modificación o eliminación. También nombrado como Suplemento. |
| Agente | Nivel 3 de la estructura comercial y persona que se encarga de llevar a cabo la captación de clientes, así como la formalización de las pólizas con. |
| Centros de Costes | Canal por el cual se ha obtenido una póliza. |
| Coordinación Regional | Nivel 1 de la estructura comercial |

| DENOMINACIÓN | DESCRIPCIÓN |
|---|---|
| Dirección Regional | Nivel 2 de la estructura comercial |
| Prima | Importe que se debe pagar por una póliza, que depende del bien o servicio asegurado, de su capital, así como de las garantías que se pretende contratar |
| Prima comercial | Diferencia entre prima total y prima de impuestos. |
| Prima impuestos | Es la parte de la prima que se debe pagar por el cliente que corresponde a los intereses de dicha póliza. |
| Prima total | Es la cantidad final que el cliente deberá abonar. Siendo ésta la prima comercial más la prima de impuestos. |
| Póliza | Es un contrato entre el tomador y la empresa, en el que se deben especificar las condiciones de cobertura. |
| Tomador | Persona titular de la póliza, quien debe hacerse cargo de la prima. |

***Tabla 7.*** *Descripción de los Conceptos de Negocio.*

Para los distintos tipos de procesos de negocio se deben tener en cuenta los siguientes conceptos:

- Pólizas Emitidas → Prima comercial del Acta.
- Pólizas Vigentes → Prima comercial de pólizas vigentes y Nº de pólizas vigentes.

- Pólizas Anuladas → Prima comercial de pólizas anuladas y Nº de pólizas anuladas.

## 4.10 Paso 8: Diseño del DW

En este apartado se detalla el diseño de la base de datos que se ha creado. Esta es un *DM* del *DW*, donde se podrán crear nuevos DM posteriormente según vayan surgiendo las necesidades del negocio.

El modelo que se ha creado es de tipo dimensional, a diferencia del origen de datos que es de tipo entidad-relación.

El *Staging Area*, por definición, es el área temporal sobre la que se van a ejecutar los procesos de Extracción, Transformación y Carga (ETL) y tiene un carácter volátil. Se usa el *Staging Area* para hacer una primera extracción rápida de los datos fuente (sin ninguna transformación) y almacenarlos temporalmente mientras se analizan, limpian, mejoran y posteriormente se carga al DW.

Se ha optado por no vaciar el *Staging Area* una vez terminados los procesos ETL de carga, manteniendo los datos hasta la siguiente carga. De este modo se puede retornar en cualquier momento a un detalle de datos más granular, aunque este sea un proceso infrecuente y debido generalmente a la necesidad de investigar o depurar problemas.

Las tablas del *Staging Area* conservan las mismas claves primarias que las fuentes originales, pero no tienen relaciones.

Es importante que se conozcan las tablas que forman parte del *Staging Area*, por ello se incluye en este apartado la Tabla 8. En ella se encuentran todas las tablas que forman parte del *Stagging Area* y su descripción.

| ENTIDAD/TABLA | DESCRIPCIÓN |
|---|---|
| **STG_A3000030 (Datos Generales de Pólizas)** | En esta tabla además de ser la tabla operacional donde se registran todos los movimientos de las pólizas están los tipos de renovación, las Primas, el capital Asegurado y las fechas de inicio del suplemento. |
| **STG_A3000060 (Datos Generales de Terceros)** | Tabla de registros de los Datos Personales de cada tercero así como su tipo (asegurado, tomador, empleado, agente). |
| **STG_A1000100 (Datos Direcciones Regionales)** | Es la tabla del nivel 1 de la estructura comercial que por ahora es una tabla con las Comunidades Autónomas. |
| **STG_A1000101 (Datos Coord. Regionales)** | Es la tabla del nivel 2 de la estructura comercial que por ahora es una tabla con las Provincias. |

| ENTIDAD/TABLA | DESCRIPCIÓN |
|---|---|
| **STG_A1000102 (Datos Agentes)** | Tabla de registros de los movimientos de los agentes que llama a la tabla A3000060 para los Datos Personales y dice si el agente permanece en su Coord. Reg. O Dir. Reg. O ha renovado en ella. |
| **STG_A1800001 (Datos Centro de Coste)** | Es la tabla de los Canales de Información. |
| **STG_A1700001 (Datos de Productos)** | Tabla con la descripción de los productos. |
| **STG_A1700002 (Datos de Sectores de Prod)** | Tabla con la descripción de los sectores de productos. |
| **STG_A1700004 (Datos de Tipo de Suplemento)** | Tabla con la descripción de los tipos de suplementos. |
| **STG_A1700005 (Datos de Sub-tipo de Sup.)** | Tabla con la descripción de los sub-tipos de suplementos. |

***Tabla 8.*** *Descripción de las tablas de la BBDD Origen, Staging Area.*

Para obtener el mejor modelo dimensional se ha hecho un estudio de los ejes de análisis deseados. Mediante este estudio es posible empezar a identificar las dimensiones y las tablas de hechos que serán necesarias. Se muestra en la tabla 10, dónde las columnas representan las dimensiones que intervienen en el cálculo de los indicadores deseados. Las filas son los indicadores que se necesitan para el negocio. Una vez creadas todas las filas y columnas se marcan con *X* las relaciones existentes, es decir, si la dimensión *i* (columna) es necesaria para calcular el indicador *j* (fila) se marca con una *X* la casilla *ij* de la tabla. Una vez se han marcado todas las casillas pertinentes se puede saber cuántas tablas de hechos son necesarias. Se agrupan los indicadores que tienen relación exactamente con las mismas dimensiones, y estos formarán parte de la misma tabla de hechos. Dentro de una tabla de hechos todos los registros deben tener valor para todos los campos, así pues no puede haber un campo (indicador) para el cual no existan valores de otros campos relacionados con las dimensiones.

En la Tabla 9 se muestra este análisis, y se puede apreciar que hay cinco indicadores (filas) y nueve dimensiones (columnas). Estos indicadores se agrupan si tienen relación con las mismas dimensiones, y salen tres grupos, que se traducen a tres tablas de hechos. Aquí son necesarias tres tablas de hechos, pólizas emitidas, pólizas vigentes y pólizas anuladas.

| PROCESOS DE NEGOCIO | FECHA | PRODUCTO | CANAL DE DISTRIBUCIÓN | TIPO SUPLEMENTO | CAUSA ANULACIÓN | TOMADOR | ESTRUCTURA COMERCIAL | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | AGENTE | DIRECCION REGIONAL (C. AUTONOMA) | COORDINACION REGIONAL (PROVINCIA) |
| **PÓLIZAS EMITIDAS** | | | | | | | | | |
| Prima comercial del acta | X | X | X | X | | X | X | X | X |
| **PÓLIZAS ANULADAS** | | | | | | | | | |
| Prima comercial pólizas anuladas | X | X | X | | X | X | X | X | X |
| Número de pólizas anuladas | X | X | X | | X | X | X | X | X |
| **PÓLIZAS VIGENTES** | | | | | | | | | |
| Prima comercial pólizas vigentes | X | X | | | | X | X | X | X |
| Número de pólizas vigentes | X | X | | | | X | X | X | X |

***Tabla 9.*** *Ejes de Análisis de indicadores frente a dimensiones, para calcular el número de tablas de hechos..*

En el anterior análisis se ven las dimensiones que intervendrán, y una de las jerarquías que existen. En total hay tres jerarquías en el modelo:

- Producto → Sector: Esta jerarquía está des normalizada.
- Agente → Coordinación → Dirección: Esta jerarquía está normalizada.
- Día → Mes → Anyo: Esta jerarquía está des normalizada.

El *DM* se ha diseñado con un modelo en "*Galaxia*", formado por tres tablas de hechos y varias dimensiones conformes. Estos modelos se encuentran diseñados con todas sus métricas, granularidad asociada y relaciones con las correspondientes dimensiones. El modelo lógico se muestra en las figuras 21, 22 y 23. Este modelo ha sido divido en tres para su mejor visibilidad, pero se debe tener en cuenta que las dimensiones con el mismo nombre se refieren a la misma, sólo se repiten para clarificar. En la Figura 21 se representa el hecho póliza emitida y las relaciones que posee con las dimensiones.

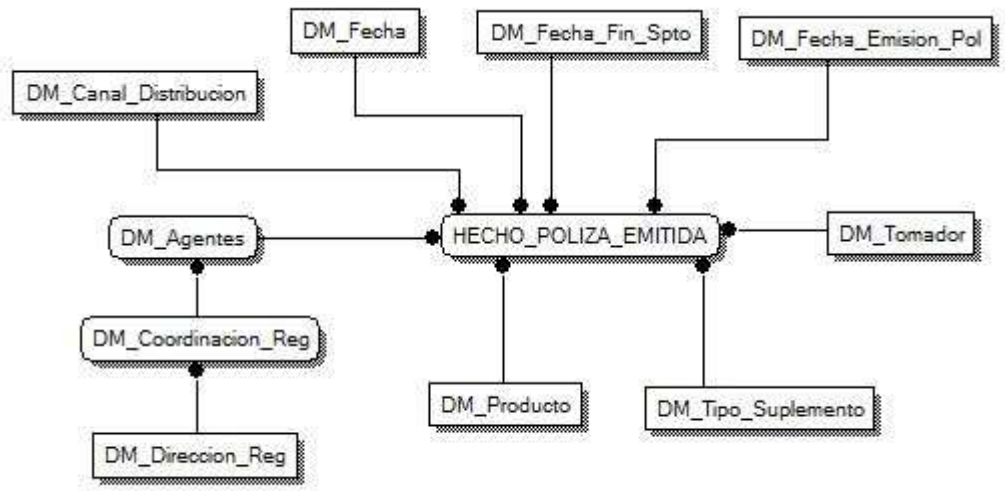



***Figura 21.*** *Diseño lógico del hecho Transaccional de Pólizas Emitidas.*

En la Figura 22 se representa el hecho póliza vigente y las relaciones que posee con las dimensiones. Donde algunas de ellas son las mimas que en la Figura 21.

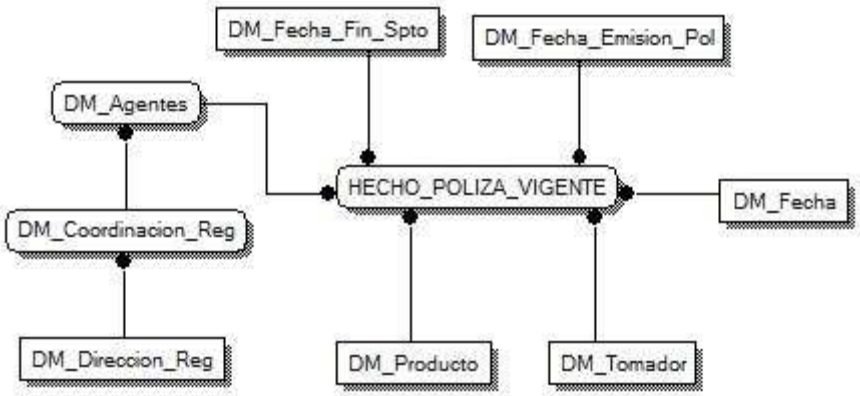



***Figura 22.*** *Diseño lógico del hecho Instantáneo Mensual de Pólizas Vigentes.*

En la Figura 23 se representa el hecho póliza anulada y las relaciones que posee con las dimensiones. Donde algunas de ellas son las mimas que en las Figuras 21 y 22.

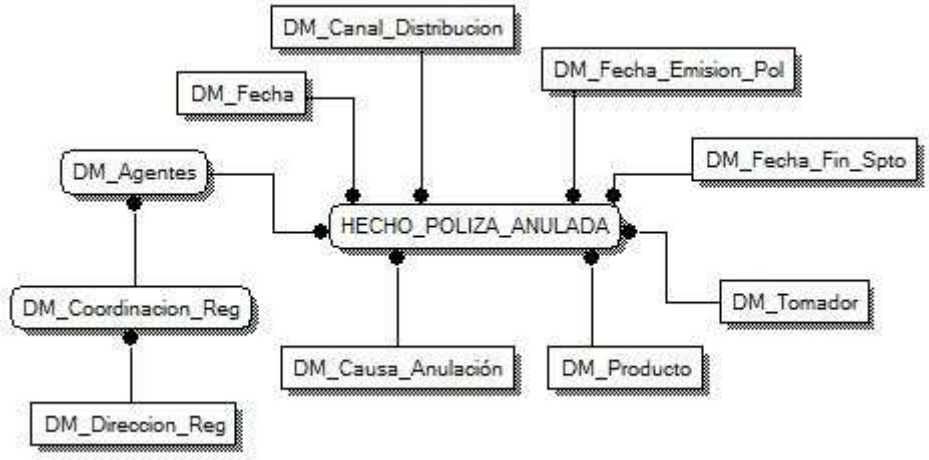



***Figura 23.*** *Diseño lógico del hecho Instantáneo Mensual de Pólizas Anuladas.*

En la Tabla 10 se especifica la descripción de todas las dimensiones que intervienen, así como su tipo de SCD (*Slowly Changing Dimensions*), que son las dimensiones que cambian lentamente. Este tipo puede ser 1, 2 y 3, dependiendo de cómo se acepten los cambios producidos en un registro. En cada caso se actúa de una manera:

- Tipo 1: No se guarda histórico de cambios, el registro modificado se reescribe sobre el mismo, no se crea un nuevo registro. La información antigua se pierde.
- Tipo 2: Se guarda histórico, se inserta un nuevo registro con la información modificada.
- Tipo 3: Se guarda histórico y se registra la información antigua y nueva en el mismo registro. Se añade una columna del campo que se quiere mantener la información antigua y nueva. Se crea un nuevo registro igual que en el tipo 2, y en la columna referida a la información antigua se almacena la información del registro antiguo, que ya existía en la tabla antes de la modificación. Este tipo se usa para obtener esta información rápidamente, de un solo acceso.

| DIMENSIÓN | DESCRIPCIÓN | TIPO SCD |
|---|---|---|
| DM_Fecha_Emision_Pol | Dimensión que almacena la fecha de la emisión de la póliza. | 1 |
| DM_Fecha_Fin_Spto | Dimensión que almacena la fecha de fin de vigencia del suplemento. | 1 |
| DM_Fecha | Dimensión que almacena la fecha de inicio del suplemento. | 1 |
| DM_Producto | Dimensión que almacena la información referida a un producto. | 1 |
| DM_Tomador | Dimensión donde se almacenan los Datos Personales de cada tomador. | 2 |
| DM_Tipo_Suplemento | Dimensión que almacena los diferentes tipos de suplementos que existen. | 1 |
| DM_Canal_Distribucion | Dimensión que almacena la información de los diferentes canales de distribución que existen. | 1 |
| DM_Causa_Anulacion | Dimensión que almacena el tipo de causa de las anulaciones. | 1 |
| DM_Agente | Dimensión que almacena la información de los agentes. | 3 |
| DM_Coordinacion_Reg | Dimensión que almacena la información de las coordinaciones regionales. | 2 |
| DM_Direccion_Reg | Dimensión que almacena la información de las direcciones regionales. | 2 |

***Tabla 10.*** *Tabla de descripción de las Dimensiones del DW y sus Tipos.*

En el caso de los hechos también existen varios tipos:

- Instantáneo Periódico: siempre se cargan todos los registros, con la fecha del momento de la carga. Se duplican registros, incluso los que no han sido modificados. Se usa cuando se van a hacer consultas periódicas del estado de todos los registros. Representa intervalos de tiempos regulares, previsibles.
- Instantáneo Cumulativo: es parecido al anterior, pero en este caso los registros se insertan y luego se modifican, no se duplican. Siempre se tiene el estado del registro actualizado para el momento de la consulta. Sólo es usado para intervalos de tiempo indeterminados, y tienen una vida breve.
- Transaccional: no se duplican ni modifican registros, sólo se insertan. Se inserta cada nuevo registro que refleja una transacción.

En la Tabla 11 se describen las tablas de hechos y los tipos de cada una.

| HECHOS | DESCRIPCIÓN | TIPO |
|---|---|---|
| Hecho_Poliza_Emitida | Tabla de hechos transaccional que almacena todas las actas emitidas sobre una póliza. | Transaccional. |
| Hecho_Poliza_Vigente | Tabla de hechos periódica mensual que almacena la información relativa a las pólizas que están vigentes. | Instantáneo Mensual. |
| Hecho_Poliza_Anulada | Tabla de hechos periódica mensual que almacena la información relativa a las pólizas que están anuladas totalmente. | Instantáneo Mensual. |

***Tabla 11.*** *Tabla de descripción y de los Hechos del DW y sus Tipos.*

Una vez descritas todas las dimensiones y los hechos, así como sus tipos, es necesario detallar todas las dimensiones, especificando la relación fuente-destino para cada campo que forma cada dimensión. Las tablas detalladas se han incluido en el Anexo A.

Una vez descritas las dimensiones se van a detallar el origen de datos y las relaciones de cada tabla de hechos.

1. POLIZAS EMITIDAS

En la Figura 24 se pueden ver las dimensiones con las que tiene relación esta tabla de hechos.

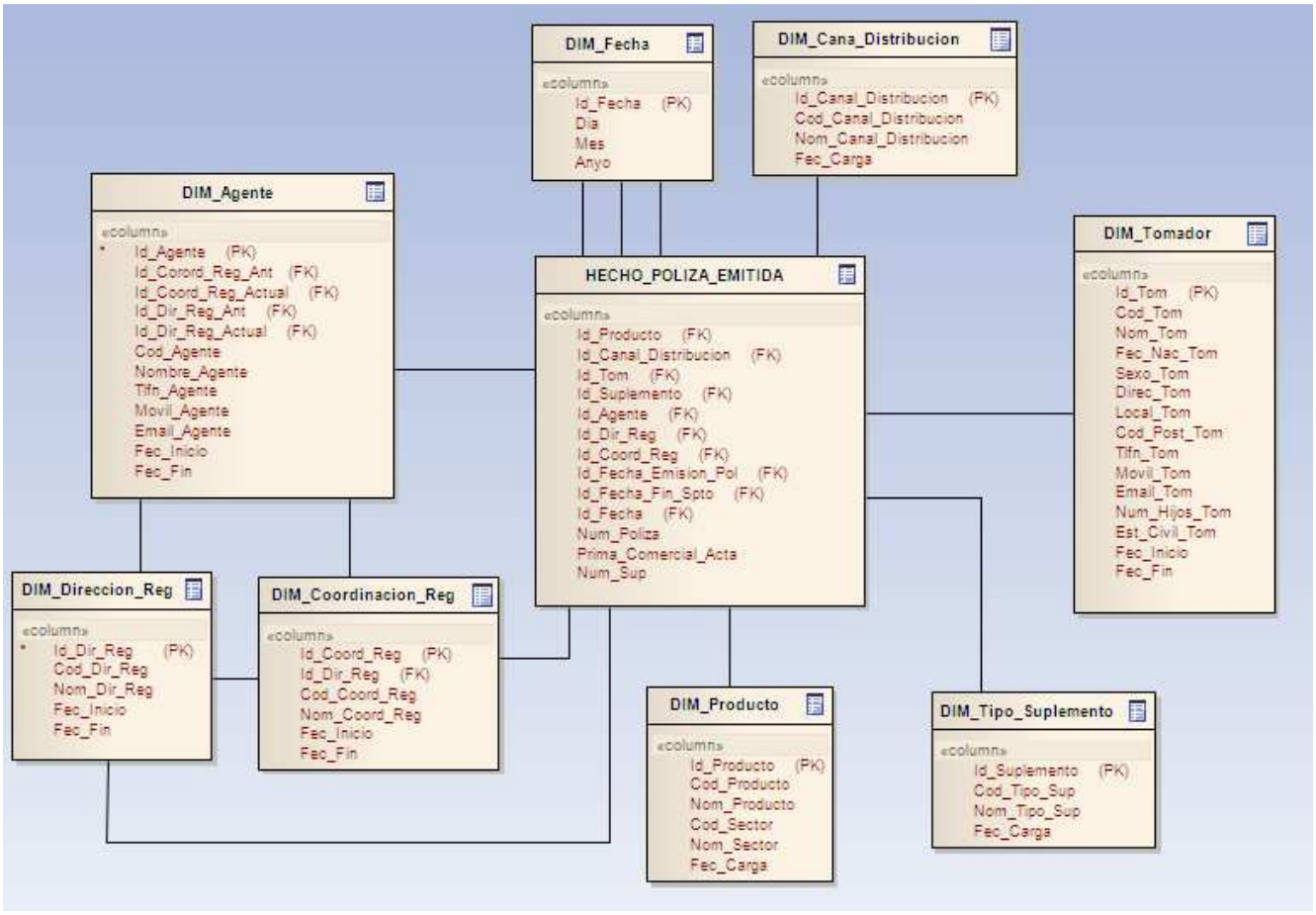



***Figura 24.*** *Diseño físico del hecho transaccional de pólizas emitidas.*

Esta es la base del código SQL para la carga del Hecho:

```
SELECT        *

FROM STG_A3000030

WHERE     STG_A3000030.Fec_ini_vig_spto(“MM/AAAA”)     ==
        DATE(“MM/AAAA”);
```

Esta sentencia SQL hace una consulta a la tabla de pólizas, recuperando solo las que están en vigor en la fecha actual.

Esta carga se hará mensualmente, y cada mes “DATE” tomará el valor de la fecha actual (SYSDATE). Para la carga inicial se ejecutará esta sentencia tantas veces como meses haya comprendidos entre la fecha más antigua y la fecha más reciente de la base de datos, asignando a “DATE” la fecha correspondiente al mes a cargar. Para cada registro leído del SELECT se puede ver una tabla en el anexo que indica la procedencia de cada campo

2. POLIZAS ANULADAS

En la Figura 25 se pueden ver las dimensiones con las que tiene relación esta tabla de hechos.

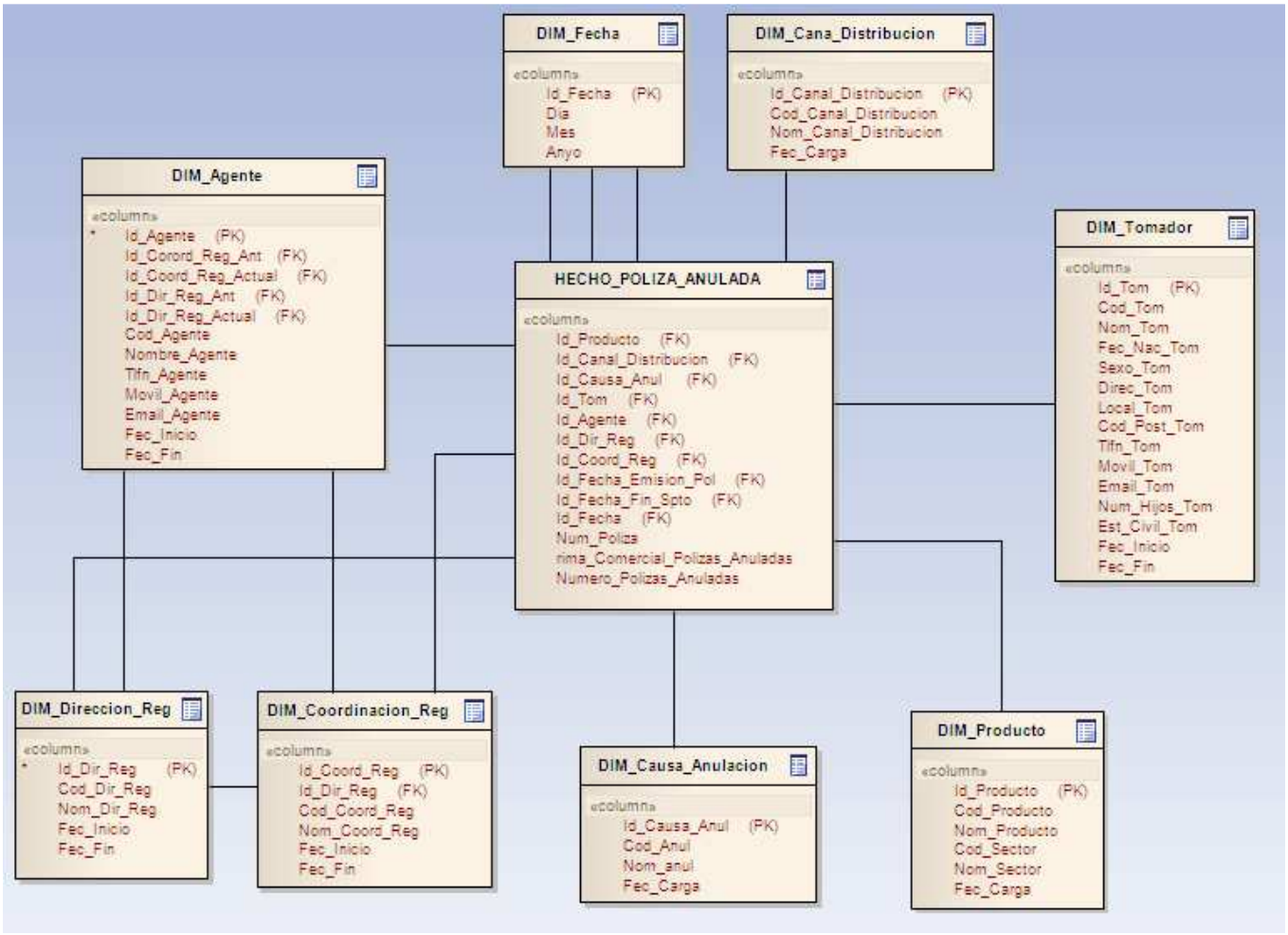



***Figura 25.*** *Diseño físico del hecho instantáneo mensual de pólizas anuladas.*

Esta es la base SQL para la carga del Hecho:

```
SELECT (DISTINCT Num_poliza) *

FROM     HECHO_POLIZA_EMITIDA     H,     DM_FECHA     F,
       DM_TIPO_SUPLEMENTO T

WHERE F.Mes == DATE("MM/AAAA")

AND T.Cod_Tipo_Sup == "AT"

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto

AND T.Id_Suplemento == H.Id_Suplemento;
```

Esta sentencia SQL selecciona las distintas pólizas de la tabla hecho_poliza_emitida que han sido anuladas en el mes en curso.

Esta carga es igual que la anterior, el tratamiento de las fechas se hace exactamente igual.

3. POLIZAS VIGENTES

En la Figura 26 se pueden ver las dimensiones con las que tiene relación esta tabla de hechos.

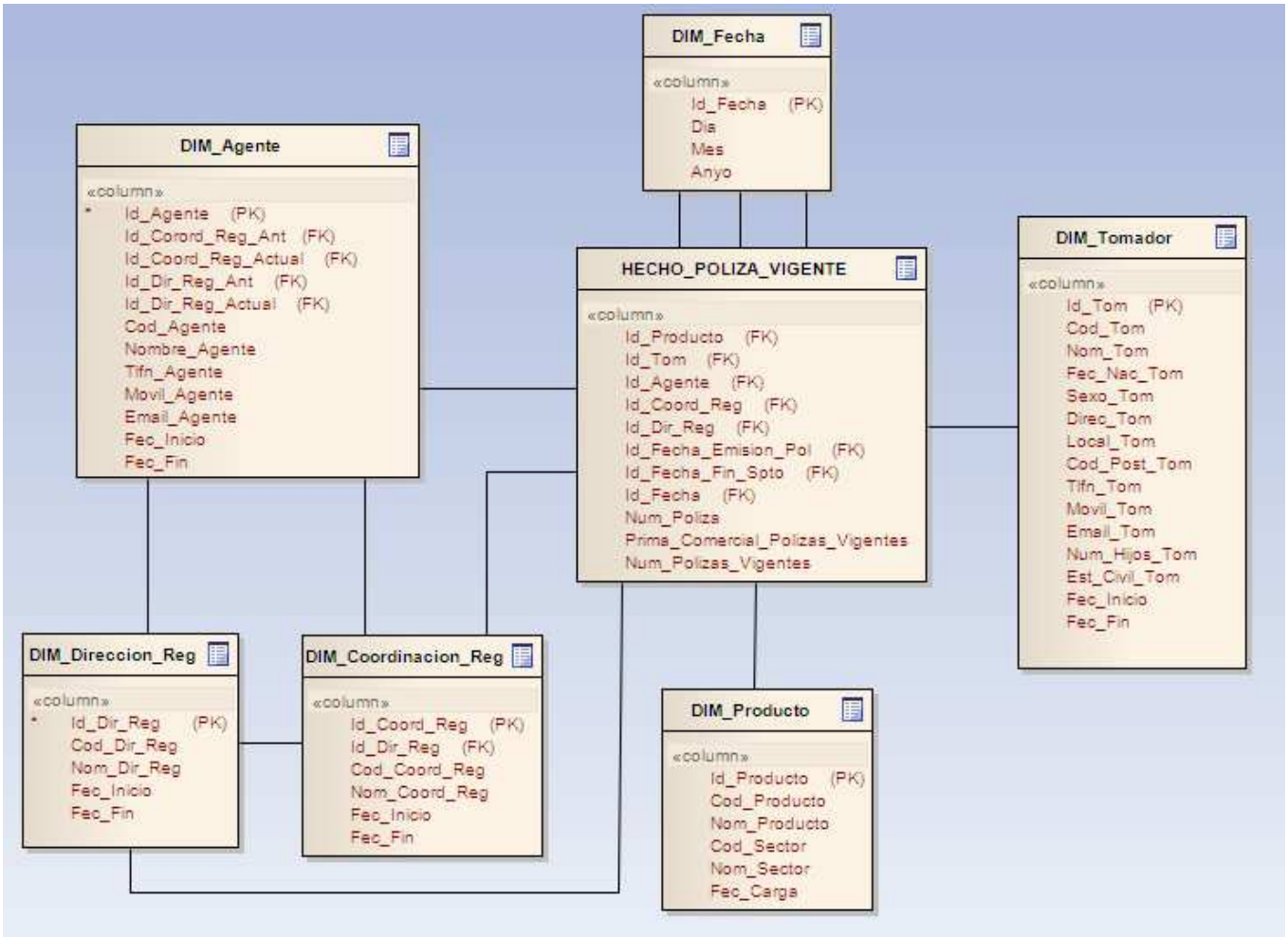



***Figura 26.*** *Diseño físico del hecho instantáneo mensual de pólizas vigentes.*

Esta es la base SQL para la carga del Hecho que se hará el primer mes:

```
SELECT (DISTINCT Num_poliza) *

FROM HECHO_POLIZA_EMITIDA H, DIM_FECHA F,
DIM_TIPO_SUPLEMENTO T

WHERE F.Mes == DATE(“MM/AAAA”)

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto

AND T.Cod_Tipo_Sup != “AT”

AND T.Id_Suplemento == H.Id_Suplemento

AND MAX H.Num_Sup

MINUS

SELECT *

FROM HECHO_POLIZA_ANULADA H, DIM_FECHA F,
DIM_TIPO_SUPLEMENTO T

WHERE F.Mes == DATE (“MM/AAAA”)

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto
```

En esta sentencia se hace una consulta a la tabla hecho_poliza_emitida sobre las distintas pólizas que han sufrido modificaciones en el mes en curso pero no ha sido una anulación. “DATE” tomará el valor del primer mes de que hay datos para cargar.

Esta es la base para la carga del Hecho que se hará cada mes, excepto el primero:

```
(SELECT *

FROM HECHO_POLIZA_VIGENTE H, DIM_FECHA F

WHERE F.Mes == DATE (“MM/AAAA”) -1

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto

MINUS

(SELECT *
```

```
FROM HECHO_POLIZA_ANULADA H, DIM_FECHA F

WHERE F.Mes == DATE(“MM/AAAA”)

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto ) anuladas)

AND

(SELECT (DISTINCT Num_poliza) *

FROM HECHO_POLIZA_EMITIDA H, DIM_FECHA F,
DIM_TIPO_SUPLEMENTO T

WHERE F.Mes == DATE(“MM/AAAA”)

AND F.Id_Fecha == H.Id_Fec_ini_vig_spto

AND T.Cod_Tipo_Sup != “AT”

AND T.Id_Suplemento == H.Id_Suplemento

AND MAX H.Num_Sup

MINUS anuladas)
```

Esta consulta SQL es más compleja que las anteriores. En primer lugar se consultan las pólizas que estaban vigentes el mes anterior y aún siguen estándolo (no se han anulado en el mes en curso), y además se añaden las nuevas del mes en curso que tampoco han sido anuladas.

Esta carga se hará mensualmente, y cada mes “DATE” tomará el valor de la fecha actual (SYSDATE).

Para cada registro leído del SELECT se puede ver una tabla en el Anexo B que indica la procedencia de cada campo.

## 4.11 Paso 9: Diseño del proceso de Extracción, Transformación y Carga (ETL)

Para el diseño ETL, en primer lugar, es necesario definir el o los orígenes de datos. En este caso sólo es uno, el *Stagging Area* que se ha creado a partir del sistema operacional. El sistema operacional consistía en un conjunto de hojas Excel, a partir de las cuales se ha creado una base de datos en *Acces*, el *Stagging Area*. Esta base de datos es la que se usa para el proceso ETL.

La estrategia de implementación es un DM, que consta de tres tablas de hechos y nueve dimensiones. De esta forma siempre se podrá ampliar este DM o crear nuevos, según las necesidades que vayan surgiendo.

Para iniciar el proceso ETL es necesario tener los datos perfectamente preparados, definido su formato, longitud, nombres, etc. La especificación de los nombres de los campos se ha hecho en la fase de diseño de la BBDD, donde también se describe de dónde se obtiene cada campo. El tipo y longitud de cada campo se ha hecho acorde con el origen, si es texto, numérico o fechas. Para ello se han creado unas tablas con toda la información, éstas se pueden ver en el Anexo B.

Una vez definidos los formatos de los datos, se procede a la carga del *Staging Area.* Esto se ha hecho de una forma muy sencilla, ha sido creada una base de datos en *Acces*, a partir de la cual se han importado los datos desde las hojas Excel correspondientes. Durante este proceso se ha encontrado un problema, la duplicidad de registros. En la tabla que contiene la información de las pólizas y sus movimientos existían registros duplicados que contenían exactamente la misma información, todos los campos iguales, para su resolución se han eliminado los duplicados.

El siguiente paso es el diseño de los procesos de extracción, transformación y carga con la herramienta ETL *Power Center*. Para ello se han creado dos ODBC (*Open Database Connectivity*), uno para el origen de datos, de dónde se extraerán los datos, y otro para el Data Warehouse, donde se cargarán los datos. Esta herramienta consta de cuatro módulos:

1. *REPOSITORY MANAGER*: a través de ésta se crean y gestionan los diferentes repositorios y usuarios.

2. *DESIGNER*: con este módulo se desarrollan los *mappings* (procesos estáticos) que describen las diferentes operaciones que se lleva a cabo con los datos, de dónde se obtienen, cómo tratarlos y dónde guardarlos.

3. *WORKFLOW MANAGER*: aquí se gestionan los *mappings* anteriormente desarrollados mediante sesiones. Cada sesión está asociada a un *mapping*, que permite ejecutarlo las veces que sea necesario y con la periodicidad establecida. También se pueden ejecutar varias sesiones a la vez y establecer el orden a seguir.

4. *WORKFLOW MONITOR*: este módulo monitoriza la ejecución de los procesos, pudiendo seguir el estado de cada uno de ellos en cada momento. Además, permite acceder a los diferentes ficheros de registro creados en las diferentes ejecuciones.

Así pues el diseño de los procesos se ha realizado con el módulo Designer. Los procesos se han dividido en dos grandes grupos, el primero que realiza la carga inicial y del histórico, y el segundo para la carga incremental que se realizará mensualmente. Véanse estos dos grupos más detalladamente:

- Carga inicial y de histórico: en primer lugar se han diseñado los procesos que cargan las dimensiones y, una vez hecho esto los procesos para cargar las tablas de hechos. Los *mappings* creados son *M_CanalDistribucionInicial*, *M_CausaAnulacionInicial, M_ProductoInicial, M_TipoSuplementoInicial, M_TomadorInicial, M_CoordinaciónInicial, M_DirecciónInicial, M_Fecha, M_AgenteInicial1 y M_AgenteInicial2.* Cada uno de ellos realiza la carga de la tabla de dimensión correspondiente según su nombre indica. En el caso de la Dimensión Agente ha sido necesario crear dos *mappings*, el primero para cargar todos los registros y el segundo para actualizar las fechas de inicio de vigor, ya que esta se obtiene del registro anterior y no era posible hacerlo todo de una vez. Después se han creado los procesos para cargar las tablas de hechos, éstos son *M_Polizas_Emitidas_Inicial, M_Polizas_Anuladas_Inicial, M_Polizas_Vigentes_Mes1,M_Polizas_Vigentes_Nuevas, M_Polizas_Vigentes_Anteriores* respectivamente. En el caso de las pólizas vigentes es necesario un proceso para cargar los registros del primer mes, y otros dos para los meses siguientes. M_Polizas_Vientes_Nuevas carga las pólizas nuevas o modificadas en el mes x, y M_Polizas_Vigentes_Anteriores carga las pólizas del mes x – 1 que siguen estando vigentes en el mes x (no han sido anuladas) y no han sido cargadas ya ese mes.

- Carga Mensual: en este caso se han diseñado los procesos exactamente igual que para la carga inicial y de histórico, donde se cargan las dimensiones, y una vez hecho esto los procesos para cargar las tablas de hechos. No se van a describir todos los *mappings* utilizados porque es repetitivo. Existe una salvedad en este caso, que solo se cargan los datos referentes al mes en curso, a diferencia del caso anterior donde se cargaban todos los datos existentes hasta la fecha actual.

Para el control de errores se ha creado un registro en cada dimensión con el identificador *-1*, de esta manera cuando se detecte un registro que no existe se le asignará este indicador por defecto. Además todos los campos vacíos en un registro se han rellenado con *-1*, en el caso de ser de tipo numérico, y con '*Error'* en el caso de campos de texto.

Una vez diseñados estos procesos es necesario crear las sesiones, *worklet* y *workflow (*proceso de ejecución de uno o varios *mappings)* necesarios para la ejecución, y establecer el orden con claridad. Al igual que para crear los procesos, se han diseñado dos flujos de ejecución, uno para la carga inicial y de histórico y otro para la Carga mensual. Estos flujos se pueden observar en las Figuras 27 y 28

- Carga inicial y de histórico:

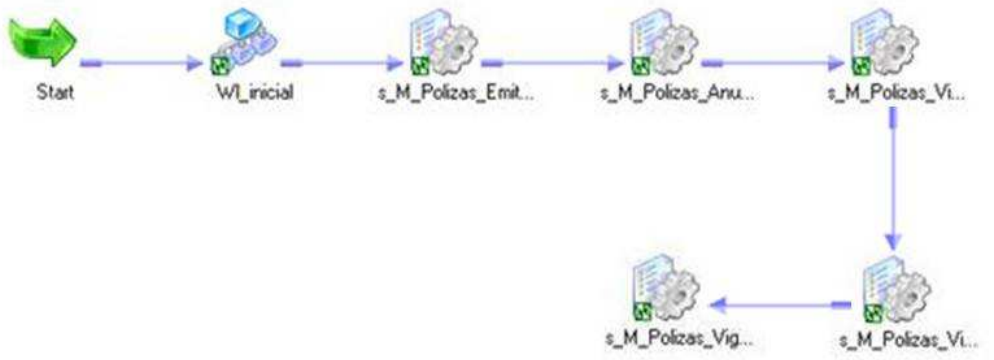



***Figura 27.*** *Workflow Carga Inicial ETL, Power Center.*

El orden de ejecución que se aprecia en la Figura 27 es el siguiente: Wl_inicial, s_M_Polizas_Emitidas_Inicial, s_M_Polizas_Anuladas_Inicial,

s_M_Polizas_Vigentes_Mes1, s_M_Polizas_Vigentes_Nuevas y s_M_Polizas_Vigentes_Anteriores.

En la Figura 28 se puede ver el orden de ejecución del *worklet* WI_inicial. Aquí todas las dimensiones se pueden ejecutar paralelamente excepto DM_Coordinacion_Reg, DM_Direccion_Reg y DM_Agente, que deben seguir este orden respectivamente, con los mapping M_CoordinaciónInicial, M_DirecciónInicial, M_AgenteInicial1 y M_AgenteInicial2 respectivamente.

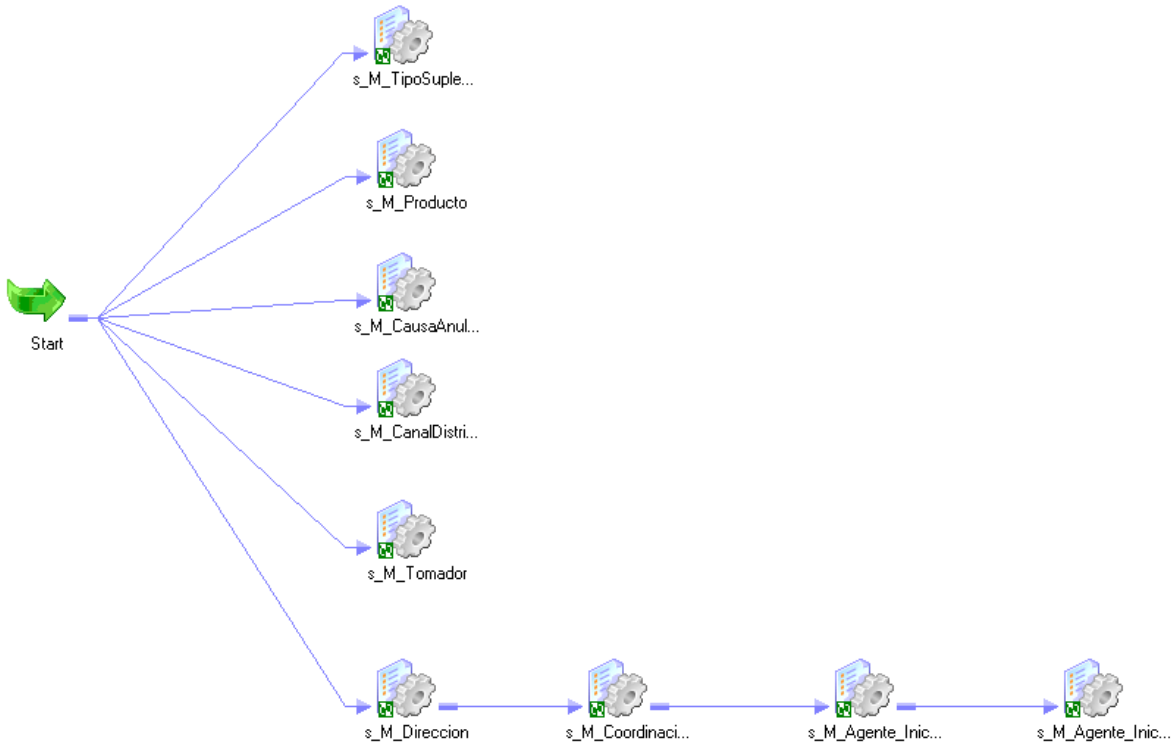



***Figura 28.*** *Worklet Carga Inicial de Dimensiones ETL, Power Center.*

- Carga mensual: En este caso el orden es igual en la carga inicial, pero usando los procesos creados para la carga mensual, así pues no se van a mostrar las imágenes del *workflow*, ya que sería repetitivo.

## 4.12 Paso 10: Diseño del Repositorio de Metadata

Este paso no se va a desarrollar, ya que lo hará automáticamente la herramienta utilizada para los procesos ETL. Además, esta herramienta almacenará esta información en un repositorio creado automáticamente.

En proyectos reales y de mayor envergadura se debe tener un diccionario de datos actualizado, así como las reglas que los definen. Esta información se almacena en repositorios, que deben estar bien definidos y situados, para agilizar los accesos desde las herramientas y sistemas que los requieren. Estos aspectos se tienen que detallar en este paso, su distribución y correcta instalación es muy importante.

## 4.13 Paso 11: Desarrollo del Proceso de Extracción, Transformación y Carga (ETL)

En este paso se han realizado las pruebas necesarias para comprobar que los procesos ETL garantizan la calidad del DW creado. Debe existir una concordancia entre los datos origen y destino, y para ello se han realizado los siguientes grupos de pruebas:

- Pruebas Unitarias: estas pruebas se han realizado para cada proceso ETL de forma autónoma, sin intervenir ningún otro proceso, sea dependiente de él o no.

    1. Comprobar que todos los campos del origen son almacenados correctamente en el destino.

    2. Comprobar que el número de registros que se carga del origen es el mismo que en el destino.

    3. Comprobar que realiza la acción deseada al encontrarse un error en un campo numérico o de texto.

- Pruebas de Integración: comprobar que al ejecutar el flujo de procesos completo, en el orden diseñado, se obtienen los mismos resultados que para las pruebas unitarias.

- Pruebas de Aceptación: los procesos son aceptados si todos ellos pasan correctamente toda la batería de pruebas.

No se han realizado pruebas de regresión, rendimiento y seguridad, ya que no se requieren en el ámbito de este proyecto.

En la Tabla 12 se muestran los resultados de las pruebas para todos los procesos que intervienen:

| Proceso ETL | Pruebas Unitarias | | | Prueba Integración | Prueba Aceptación |
|---|---|---|---|---|---|
| | PU1 | PU2 | PU3 | | |
| M_CanalDistribucionInicial | Si | Si | Si | Si | Si |
| M_CausaAnulacionIncial | Si | Si | Si | Si | Si |
| M_ProductoInicial | Si | Si | Si | Si | Si |
| M_TipoSuplementoInicial | Si | Si | Si | Si | Si |
| M_TomadorInicial | Si | Si | Si | Si | Si |
| M_Fecha | Si | Si | Si | Si | Si |
| M_CoordinacionInicial | Si | Si | Si | Si | Si |
| M_DireccionInicial | Si | Si | Si | Si | Si |
| M_AgenteInicial1 | Si | Si | Si | Si | Si |
| M_AgenteInicial2 | Si | Si | Si | Si | Si |
| M_Polizas_Emitidas_Inicial | Si | Si | Si | Si | Si |
| M_Polizas_Anuladas_Inicial | Si | Si | Si | Si | Si |
| M_Polizas_Vigentes_Mes1 | Si | Si | Si | Si | Si |
| M_Polizas_Vigentes_Nuevas | Si | Si | Si | Si | Si |
| M_Polizas_Vigentes_Anteriores | Si | Si | Si | Si | Si |
| M_CanalDistribucion | Si | Si | Si | Si | Si |
| M_CausaAnulacion | Si | Si | Si | Si | Si |
| M_Producto | Si | Si | Si | Si | Si |
| M_TipoSuplemento | Si | Si | Si | Si | Si |
| M_Tomador | Si | Si | Si | Si | Si |
| M_Coordinacion | Si | Si | Si | Si | Si |
| M_Direccion | Si | Si | Si | Si | Si |
| M_Agente | Si | Si | Si | Si | Si |

| Proceso ETL | Pruebas Unitarias | | | Prueba Integración | Prueba Aceptación |
|---|---|---|---|---|---|
| | PU1 | PU2 | PU3 | | |
| M_Polizas_Emitidas | Si | Si | Si | Si | Si |
| M_Polizas_Anuladas | Si | Si | Si | Si | Si |

***Tabla 12.*** *Tabla de Pruebas sobre el desarrollo ETL.*

## 4.14 Paso 12: Desarrollo Aplicación

En este caso práctico no se ha desarrollado una aplicación completa, sino que se han creado informes a partir del D. A continuación se describen los informes creados.

La finalidad de la creación del DW era tener un modelo datos que permitiera generar dos informes anuales sobre la cuantía de las primas de las pólizas. Así pues mediante la herramienta *Business Objects* se han generado los informes requeridos, que se muestran en el Anexo C.

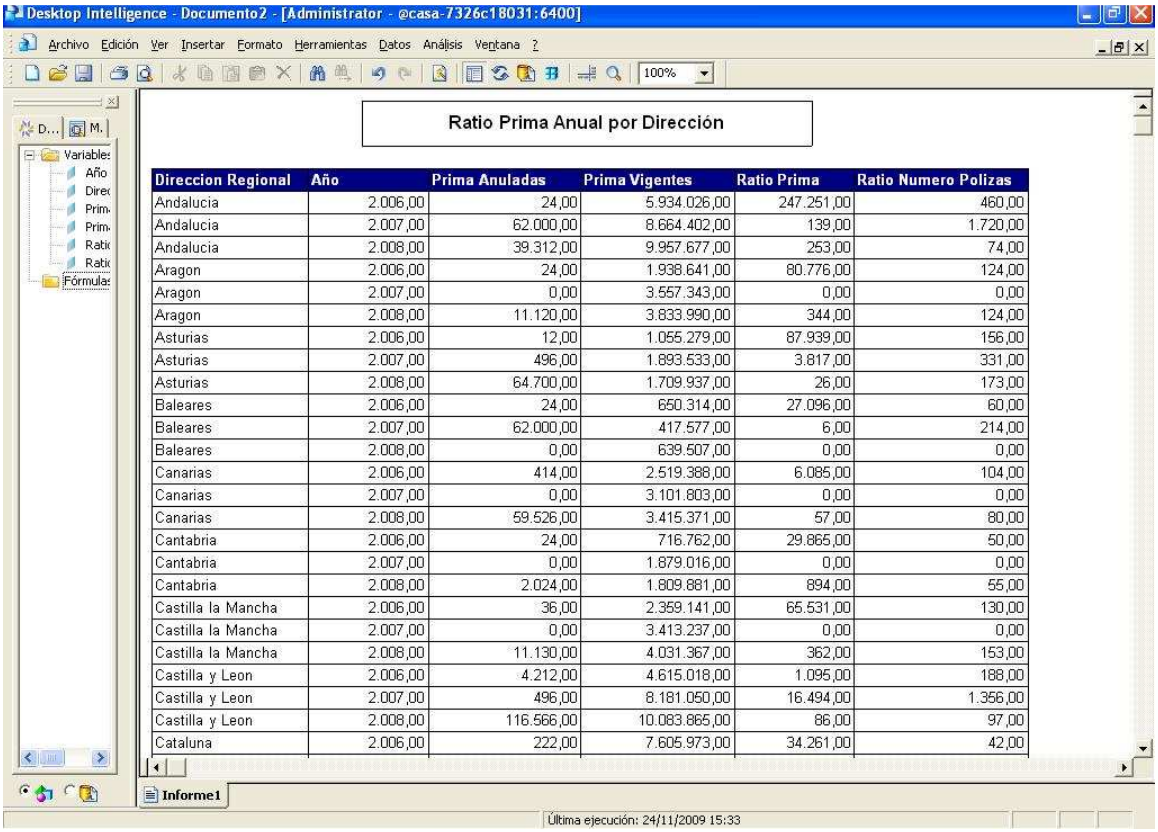
Este informe muestra la cuantía de las primas de cada año en cada dirección regional, así como el ratio de la prima y del número de pólizas, que se calculan de la siguiente manera:

```
Ratio prima = prima pólizas vigentes/prima pólizas anuladas

Ratio pólizas = número pólizas vigentes/número pólizas anuladas
```

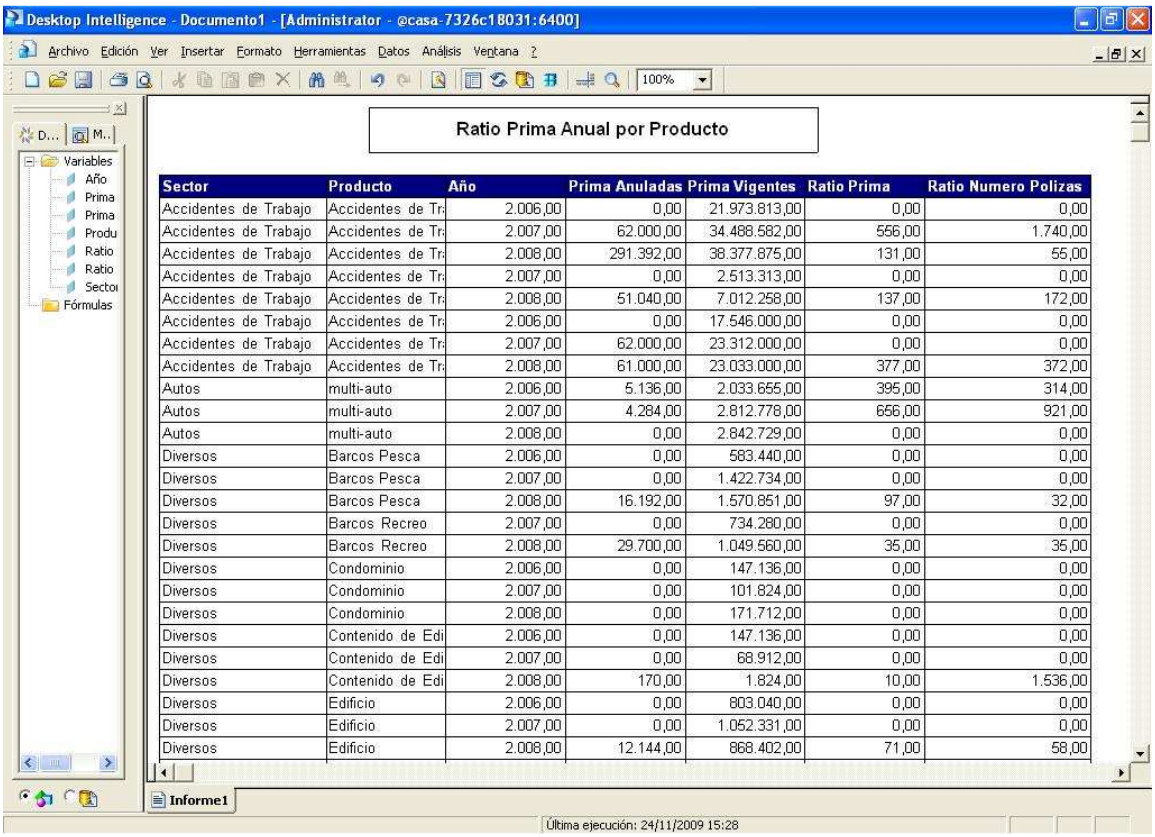
Este informe muestra el ratio por dirección de forma anual, muy útil para tener una visión global al cierre del año. La ventaja de realizar estos informes con herramientas OLAP es que son dinámicos y permiten navegar por las jerarquías. En este caso se tienen dos jerarquías por las que se podría navegar si se quisiera consultar la información relativa a otros niveles de la jerarquía. La primera es por fecha, se podrían visualizar los datos por mes o incluso por día si se requiere, y eso se haría de una forma muy sencilla usando este tipo de herramientas. La segunda jerarquía es la dirección, coordinación regional y el agente.

En este caso particular el interés para el negocio es poder consultar esta información a nivel de dirección y a nivel de producto, que es el siguiente informe que se ha generado. En este también se podría navegar por la jerarquía fecha, y pasar del año al mes o incluso al día. Estos informes se puedes ven en detalle en el Anexo C, y en las Figuras 29 y 30 se muestran los informes con la herramienta *Business Objects*




| Direccion Regional | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Numero Polizas |
|---|---|---|---|---|---|
| Andalucia | 2.006,00 | 24,00 | 5.934.026,00 | 247.251,00 | 460,00 |
| Andalucia | 2.007,00 | 62.000,00 | 8.664.402,00 | 139,00 | 1.720,00 |
| Andalucia | 2.008,00 | 39.312,00 | 9.957.677,00 | 253,00 | 74,00 |
| Aragon | 2.006,00 | 24,00 | 1.938.641,00 | 80.776,00 | 124,00 |
| Aragon | 2.007,00 | 0,00 | 3.557.343,00 | 0,00 | 0,00 |
| Aragon | 2.008,00 | 11.120,00 | 3.833.990,00 | 344,00 | 124,00 |
| Asturias | 2.006,00 | 12,00 | 1.055.279,00 | 87.939,00 | 156,00 |
| Asturias | 2.007,00 | 496,00 | 1.893.533,00 | 3.817,00 | 331,00 |
| Asturias | 2.008,00 | 64.700,00 | 1.709.937,00 | 26,00 | 173,00 |
| Baleares | 2.006,00 | 24,00 | 650.314,00 | 27.096,00 | 60,00 |
| Baleares | 2.007,00 | 62.000,00 | 417.577,00 | 6,00 | 214,00 |
| Baleares | 2.008,00 | 0,00 | 639.507,00 | 0,00 | 0,00 |
| Canarias | 2.006,00 | 414,00 | 2.519.388,00 | 6.085,00 | 104,00 |
| Canarias | 2.007,00 | 0,00 | 3.101.803,00 | 0,00 | 0,00 |
| Canarias | 2.008,00 | 59.526,00 | 3.415.371,00 | 57,00 | 80,00 |
| Cantabria | 2.006,00 | 24,00 | 716.762,00 | 29.865,00 | 50,00 |
| Cantabria | 2.007,00 | 0,00 | 1.879.016,00 | 0,00 | 0,00 |
| Cantabria | 2.008,00 | 2.024,00 | 1.809.881,00 | 894,00 | 55,00 |
| Castilla la Mancha | 2.006,00 | 36,00 | 2.359.141,00 | 65.531,00 | 130,00 |
| Castilla la Mancha | 2.007,00 | 0,00 | 3.413.237,00 | 0,00 | 0,00 |
| Castilla la Mancha | 2.008,00 | 11.130,00 | 4.031.367,00 | 362,00 | 153,00 |
| Castilla y Leon | 2.006,00 | 4.212,00 | 4.615.018,00 | 1.095,00 | 188,00 |
| Castilla y Leon | 2.007,00 | 496,00 | 8.181.050,00 | 16.494,00 | 1.356,00 |
| Castilla y Leon | 2.008,00 | 116.566,00 | 10.083.865,00 | 86,00 | 97,00 |
| Cataluna | 2.006,00 | 222,00 | 7.605.973,00 | 34.261,00 | 42,00 |

***Figura 29.*** *Informe Ratio Prima por Dirección con Business Objects.*




| Sector | Producto | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Numero Polizas |
|---|---|---|---|---|---|---|
| Accidentes de Trabajo | Accidentes de Tr | 2.006,00 | 0,00 | 21.973.813,00 | 0,00 | 0,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.007,00 | 62.000,00 | 34.488.582,00 | 556,00 | 1.740,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.008,00 | 291.392,00 | 38.377.875,00 | 131,00 | 55,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.007,00 | 0,00 | 2.513.313,00 | 0,00 | 0,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.008,00 | 51.040,00 | 7.012.258,00 | 137,00 | 172,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.006,00 | 0,00 | 17.546.000,00 | 0,00 | 0,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.007,00 | 62.000,00 | 23.312.000,00 | 0,00 | 0,00 |
| Accidentes de Trabajo | Accidentes de Tr | 2.008,00 | 61.000,00 | 23.033.000,00 | 377,00 | 372,00 |
| Autos | multi-auto | 2.006,00 | 5.136,00 | 2.033.655,00 | 395,00 | 314,00 |
| Autos | multi-auto | 2.007,00 | 4.284,00 | 2.812.778,00 | 656,00 | 921,00 |
| Autos | multi-auto | 2.008,00 | 0,00 | 2.842.729,00 | 0,00 | 0,00 |
| Diversos | Barcos Pesca | 2.006,00 | 0,00 | 583.440,00 | 0,00 | 0,00 |
| Diversos | Barcos Pesca | 2.007,00 | 0,00 | 1.422.734,00 | 0,00 | 0,00 |
| Diversos | Barcos Pesca | 2.008,00 | 16.192,00 | 1.570.851,00 | 97,00 | 32,00 |
| Diversos | Barcos Recreo | 2.007,00 | 0,00 | 734.280,00 | 0,00 | 0,00 |
| Diversos | Barcos Recreo | 2.008,00 | 29.700,00 | 1.049.560,00 | 35,00 | 35,00 |
| Diversos | Condominio | 2.006,00 | 0,00 | 147.136,00 | 0,00 | 0,00 |
| Diversos | Condominio | 2.007,00 | 0,00 | 101.824,00 | 0,00 | 0,00 |
| Diversos | Condominio | 2.008,00 | 0,00 | 171.712,00 | 0,00 | 0,00 |
| Diversos | Contenido de Edi | 2.006,00 | 0,00 | 147.136,00 | 0,00 | 0,00 |
| Diversos | Contenido de Edi | 2.007,00 | 0,00 | 68.912,00 | 0,00 | 0,00 |
| Diversos | Contenido de Edi | 2.008,00 | 170,00 | 1.824,00 | 10,00 | 1.536,00 |
| Diversos | Edificio | 2.006,00 | 0,00 | 803.040,00 | 0,00 | 0,00 |
| Diversos | Edificio | 2.007,00 | 0,00 | 1.052.331,00 | 0,00 | 0,00 |
| Diversos | Edificio | 2.008,00 | 12.144,00 | 868.402,00 | 71,00 | 58,00 |

***Figura 30.*** *Informe Ratio Prima por Producto con Business Objects.*

## 4.15 Paso 13: Minería de Datos

En este paso se describe la minería de datos que se ha realizado, así como los resultados obtenidos.

El desarrollo del DW tenía como objetivo poder generar los informes deseados así como hacer un pequeño análisis mediante minería de datos sobre el incremento de la prima de las pólizas, con el fin de poder predecir este incremento en el futuro. Para realizar esta predicción se ha seleccionado la clasificación como método de minería de datos. En primer lugar ha sido necesario obtener los datos necesarios en el formato adecuado. El conjunto de datos está formado por registros (individuos) que indican el incremento de la prima de una póliza por periodo, adjuntando a esta información características sobre los individuos, que se pueden ver en la Tabla 13.

| Nº Atributo | Atributo | Tipo |
|---|---|---|
| 1 | Producto | Numérico |
| 2 | Tomador | Numérico |
| 3 | Agente | Numérico |
| 4 | Dirección Regional | Numérico |
| 5 | Coordinación Regional | Numérico |
| 6 | Fecha de Emisión | Numérico |
| 7 | Fecha de nacimiento del Tomador | Numérico |
| 8 | Sexo del Tomador | Numérico |
| 9 | Código postal del Tomador | Numérico |
| 10 | Número hijos del Tomador | Numérico |
| 11 | Estado civil del Tomador | Numérico |
| 12 | Número de periodo | Numérico |
| 13 | Prima Comercial | Numérico |
| 14 | Número de Póliza | Numérico |
| 15 | Incremento de la Póliza | Clase |

***Tabla 13.*** *Lista de atributos para el análisis de Data Mining.*

Estas características suman un total de 14 atributos y la clase a predecir, incremento. Para poder tener este atributo clasificado se ha discretizado en tres clases (1, 2, 3). La clase 1 indica que el incremento es negativo, la 2 que casi no hay incremento y la 3 que hay un incremento positivo. Además, se han obtenido los datos para dos tipos de periodos, por trimestre y por semestre, así se podrá saber por qué tipo de periodo se hace mejor predicción. Para cada tipo de periodo se han creado dos

archivos de datos, uno para el entrenamiento (training) y otro para la validación (test), en ambos casos el conjunto de validación está formado por los datos de último periodo. En la Tabla 14 se puede ver la distribución de los datos:

| CLASE | Clasificación Incremento | | | |
|---|---|---|---|---|
| | Trimestrales | | Semestrales | |
| 1 | 2.018 | 8,66% | 1.569 | 15,71% |
| 2 | 17.103 | 73,42% | 4.881 | 48,88% |
| 3 | 4.173 | 17,91% | 3.535 | 35,40% |
| Total | 23.294 | 100,00% | 9.985 | 100,00% |

***Tabla 14.*** *Distribución de los conjuntos de datos para el análisis de Data Mining, Weka.*

Como se puede observar los datos están más balanceados en el caso del periodo semestral, aún así se realizó el estudio para los dos casos. Para la discretización de los datos y su clasificación se ha utilizado la parte de pre-procesado de datos de *Weka*.

En primer lugar, se ha realizado una batería de pruebas con los algoritmos de clasificación para los dos conjuntos de entrenamiento. Los algoritmos que se han usado son árboles, reglas y funciones. Se han utilizado varios de este tipo de algoritmos que están descritos en el Anexo D en forma de tabla.

El método aplicado ha sido el de validación cruzada, que consiste en dividir el conjunto de datos en subconjuntos, en este caso 5, y con esos subconjuntos se ejecuta el algoritmo tantas veces como número de subconjuntos se tienen. En cada ejecución se usa uno de esos subconjuntos como datos de validación, y se selecciona la media de todos los resultados obtenidos. Estos resultados se han representado en la Tabla 15:

| Instancias Clasificadas Correctamente (%) | | | |
|---|---|---|---|
| ALGORITMOS | | Incremento | |
| | | Trimestral (73,42%) | Semestral (48,88%) |
| ÁRBOLES | BFTree | **97,47%** | 97,27% |
| | DecisionStamp | 60,74% | 77,46% |
| | FT | 93,51% | 94,24% |
| | J48 | **97,75%** | **97,90%** |
| | J48graf | **97,65%** | **97,79%** |
| | LADTree | 82,43% | 82,12% |
| | LMT | 96,63% | **97,54%** |
| | NBTree | 96,92% | 95,67% |
| | RandomForest | 97,08% | 96,25% |

| Instancias Clasificadas Correctamente (%) | | | |
|---|---|---|---|
| ALGORITMOS | | Incremento | |
| | | Trimestral (73,42%) | Semestral (48,88%) |
| | RandomTree | 92,93% | 92,47% |
| | REPTree | 97,08% | 97,10% |
| | SimpleCart | **97,47%** | **97,40%** |
| | UserClassifier | 48,88% | 73,42% |
| REGLAS | ConjuntiveRule | 60,74% | 77,44% |
| | DecisionTable | 95,85% | 91,42% |
| | DTNB | 95,57% | 91,75% |
| | JRip | 97,05% | 96,58% |
| | OneR | 65,29% | 77,46% |
| | PART | **97,59%** | **97,53%** |
| | Ridor | 96,58% | 96,05% |
| | ZeroR | 48,88% | 73,42% |
| FUNCIONES | Logistic | 70,20% | 80,61% |
| | MultilayerPerceptron | 88,25% | 91,14% |
| | RBFNetwork | 61,09% | 73,65% |
| | SimpleLogistic | 69,94% | 80,60% |
| | SMO | 72,15% | 81,34% |

***Tabla 15.*** *Resultados de clasificación con algoritmos de Data Mining, Weka.*

La Tabla 15 muestra el porcentaje de acierto de cada algoritmo para cada conjunto de datos. Debajo de trimestral y semestral pone un porcentaje, este indica el porcentaje de registros que pertenecen a la clase mayoritaria, mostrado en la anterior tabla. Como estas clases no están balanceadas, este porcentaje es muy alto en el caso de los datos trimestrales. Para que un resultado sea aceptable debe superar este porcentaje, ya que si no sería mejor clasificar siempre como la clase mayoritaria, sin necesidad de aplicar ningún algoritmo. Así pues dentro de los resultados más aceptables se han seleccionado los mejores para cada tipo de periodo, los que se han marcado en negrita en la Tabla 15.

Una vez realizada esta primera etapa, se han elegido algoritmos de selección de atributos para ver qué resultados se pueden conseguir con menor número de atributos, y así poder saber qué influencia tienen. Los algoritmos de selección de atributos utilizados son:

- *CfsSubsetEval:* Evalúa un subconjunto de atributos, considerando la capacidad de predicción individual de cada función, junto con el grado de redundancia entre ellos.

- *ChiSquaredAttributeEval:* Evalúa un atributo mediante el cálculo del valor estadístico de la chi-cuadrado con respecto a la clase.
- *ConsistencySubsetEval:* Evalúa un subconjunto de atributos por el nivel de coherencia con los valores de la clase.
- *PrincipalComponents:* Realiza un análisis de componentes principales y la transformación de los datos. La reducción de dimensiones se lleva a cabo con la elección de vectores propios, suficientes para tener un porcentaje de la varianza en los datos originales por defecto (95%).

Con ellos se han realizado las pruebas que se muestran en la Tabla 16.

El siguiente paso es aplicar los algoritmos, anteriormente elegidos, conjuntamente con éstos de selección de atributos. Las Tablas 17 y 18 contienen los resultados de cada algoritmo junto con selección de atributos. La Tabla 17 muestra los resultados para el conjunto de datos trimestrales y la Tabla 18 para semestrales.

| SELECCIÓN ATRIBUTOS | | Trimestral | Semestral |
|---|---|---|---|
| CfsSubsetEval | Num. Atrb. | 2 | 2 |
| | Atrb. Selecc. | 12, 14 | 12, 14 |
| ChiSquaredAttributeEval | Num. Atrb. | 14 | 14 |
| | Atrb. Selecc. | 1,2,3,4,5,6,7,8,9,10,11,12,13,14 | 1,2,3,4,5,6,7,8,9,10,11,12,13,14 |
| ConsistencySubsetEval | Num. Atrb. | 11 | 10 |
| | Atrb. Selecc. | 2,3,4,5,6,7,9,10,12, 13,14 | 2,4,5,6,7,9,10,12, 13,14 |
| PrincipalComponents | Num. Atrb. | 10 | 10 |
| | Atrb. Selecc. | 1,2,3,4,5,6,7,8,9,10 | 1,2,3,4,5,6,7,8,9,10 |

***Tabla 16.*** *Resultados de selección de atributos con Weka.*

| ALGORITMO | SELECCIÓN | Trimestral |
|---|---|---|
| BFTree | Sin selección | 97,47% |
| | CfsSubsetEval | 85,83% |

| ALGORITMO | SELECCIÓN | Trimestral |
|---|---|---|
| | ConsistencySubsetEval | 97,27% |
| | PrincipalComponents | 78,48% |
| J48 | Sin selección | 97,75% |
| | CfsSubsetEval | 85,82% |
| | ConsistencySubsetEval | **97,89%** |
| | PrincipalComponents | 74,91% |
| J48graf | Sin selección | 97,65% |
| | CfsSubsetEval | 85,83% |
| | ConsistencySubsetEval | **97,79%** |
| | PrincipalComponents | 74,91% |
| SimpleCart | Sin selección | 97,47% |
| | CfsSubsetEval | 85,83% |
| | ConsistencySubsetEval | 97,40% |
| | PrincipalComponents | 74,88% |
| PART | Sin selección | **97,59%** |
| | CfsSubsetEval | 85,87% |
| | ConsistencySubsetEval | 97,53% |
| | PrincipalComponents | 74,63% |

***Tabla 17.*** *Resultados algoritmos-selección de atributos con trimestrales.*

| ALGORITMO | SELECCIÓN | Semestral |
|---|---|---|
| LMT | Sin selección | **97,90%** |
| | CfsSubsetEval | 84,61% |
| | ConsistencySubsetEval | 97,23% |
| | PrincipalComponents | 64,27% |
| J48 | Sin selección | **97,79%** |
| | CfsSubsetEval | 85,84% |
| | ConsistencySubsetEval | 97,74% |
| | PrincipalComponents | 64,09% |
| J48graf | Sin selección | 97,54% |
| | CfsSubsetEval | 85,83% |
| | ConsistencySubsetEval | **97,64%** |
| | PrincipalComponents | 64,08% |
| SimpleCart | Sin selección | 97,40% |
| | CfsSubsetEval | 85,86% |
| | ConsistencySubsetEval | 97,45% |
| | PrincipalComponents | 63,78% |
| PART | Sin selección | 97,53% |
| | CfsSubsetEval | 85,83% |

| ALGORITMO | SELECCIÓN | Semestral |
|---|---|---|
| | ConsistencySubsetEval | 97,54% |
| | PrincipalComponents | 62,05% |

***Tabla 18.*** *Resultados algoritmos-selección de atributos con semestrales.*

En ambos casos se han marcado en negrita las tres pruebas que mejores resultados han dado.

Hasta ahora todas las pruebas han sido realizadas con los conjuntos de datos de entrenamiento. Para validar estos resultados es necesario hacer los experimentos con los datos de *Test*. Estos datos son los correspondientes al último periodo, tanto para los trimestrales como para los semestrales. Con estos datos de validación sólo se van a ejecutar los algoritmos que se han considerado mejores: J48, J48graf, PART y LMT. Vamos a comentar cada uno de ellos antes de mostrar los resultados.

- J48: este algoritmo está implementado por Weka, basándose en el algoritmo c4.5. Se encuentra dentro de los algoritmos basados en árboles de decisión, y es una mejora del ID3. El j48 tiene como principal característica la poda de ramas del árbol de clasificación, que se lleva a cabo después de haber sido construido el árbol, una vez inducido. El criterio que sigue este algoritmo es podar aquellas ramas que tienen menos capacidad de predicción. Además aplica el mismo concepto que ID3 para seleccionar el atributo, elige la que proporciona mayor cantidad de información entre el atributo clase y el elegido. [8]

- J48graf: este algoritmo de clasificación es igual que el j48 pero incluye una característica nueva, consiste en realizar injertos en el árbol. Añade ramas, que han podido ser o no podadas de otra, a un nuevo nodo.

- PART: este es una mezcla de los algoritmos C4.5 y RIPPER, tratando de eliminar la lentitud de la post-clasificación. Está basado en el método divide y vencerás. Construye una regla y elimina las instancias que cubre, y así recursivamente hasta que no queda ninguna instancia por cubrir. De esta forma al final sólo quedan nodos hoja en el árbol de decisión, y estas hojas contienen las mejores reglas. La idea clave consiste en construir un árbol de decisión parcial en lugar de uno completamente explorado.

- LMT: clasificador para la construcción de *Logistic model trees*, que son árboles de clasificación con funciones de regresión logística en las hojas. El algoritmo puede hacer frente a las variables objetivo binarios y multi-clase, los atributos numéricos y nominales y los valores faltantes.

Los resultados obtenidos se muestran en las Tablas 19 y 20, para los conjuntos de datos trimestrales y semestrales respectivamente. Se han marcado en negrita los mejores resultados.

| ALGORITMO | SELECCIÓN | | Trimestral |
|---|---|---|---|
| J48 | ConsistencySubsetEval | TRAINING | 97,89% |
| | | TEST | **99,92%** |
| J48graf | ConsistencySubsetEval | TRAINING | 97,79% |
| | | TEST | **99,92%** |
| PART | Sin selección | TRAINING | 97,59% |
| | | TEST | **99,92%** |

***Tabla 19.*** *Resultados de validación con datos trimestrales en Data Mining, Weka.*

| ALGORITMO | SELECCIÓN | | Semestral |
|---|---|---|---|
| LMT | Sin selección | TRAINING | 97,90% |
| | | TEST | **98,18%** |
| J48 | Sin selección | TRAINING | 97,79% |
| | | TEST | 97,55% |
| J48graf | ConsistencySubsetEval | TRAINING | 97,64% |
| | | TEST | 97,17% |

***Tabla 20.*** *Resultados de validación con datos semestrales en Data Mining, Weka.*

En ambos casos los resultados de la validación son buenos, se mejora el porcentaje de acierto con respecto a los datos de entrenamiento, aunque estos resultados ya eran muy buenos. Para ambos conjuntos de datos los resultados son más que aceptables.

En el caso de los datos semestrales el mejor algoritmo es LMT sin selección de atributos, y en los trimestrales son igual de buenos los tres seleccionados.

En el Anexo D se puede observar el árbol generado por el algoritmo J48, con el que se ha obtenido el mejor resultado para el conjunto de datos semestrales.

## 4.16 Paso 14: Desarrollo del Repositorio de Metadata

En este paso se debe construir el repositorio *metadata* que fuera descrito y definido en el paso 7. Además se tendrían que hacer las pruebas pertinentes para garantizar su fiabilidad.

En este caso práctico no ha sido necesario definir ni construir un repositorio *metadata*.

## 4.17 Paso 15: Implementación

Cuando se llega a este paso la aplicación ya está construida y probada para ser implementada en el entorno de producción. Para ello se debe crear un plan de implementación, preparar el entorno, instalar los componentes que sean necesarios y poner el esquema en producción. Entonces se podrán cargar las BBDD y preparar el soporte en curso.

En este caso práctico no se ha realizado implementación, ya que no se va a implantar en ningún entorno de producción, sólo es un caso de estudio.

## 4.18 Paso 16: Evaluación

En este paso se debe evaluar la solución implementada y ver cuáles serían las siguientes mejoras o ampliaciones que se pueden realizar. En este caso se da por finalizado, ya que era un caso de estudio para poner en práctica la metodología de sistemas BI.

Los resultados obtenidos han sido buenos, los informes realizados proporcionan toda la información que se requería para este caso práctico.

Respecto al análisis predictivo que se requería, los resultados han sido buenos. Los porcentajes de acierto son superiores al 95%. Esto significa que se hace una clasificación de las pólizas según el incremento de su prima trimestral y semestral casi exacta. Esta información podría ser muy útil para una compañía de seguros, pero se debe tener en cuenta que este es un caso basado en datos ficticios.

Para este caso práctico la evaluación es positiva, pero se da por finalizado ya que no se van a ampliar ni modificar los requisitos. En el caso de que sí se hiciera no habría ningún problema en ampliarlo, las herramientas están preparadas por hacer

modificaciones y ampliaciones de los modelos creados. Sería necesario volver a realizar todos los pasos según su orden, e ir añadiendo o modificando todo lo que sea necesario para cumplir todos los requisitos, antiguos y nuevos.

## 4.19 Herramientas Utilizadas

Se han utilizados varias herramientas para llevar a cabo el caso práctico. Estas herramientas se muestran en la Tabla 21.

| Herramienta | Utilización |
|---|---|
| Power Center | Herramienta utilizada para los procesos ETL. |
| Business Objects | Herramienta utilizada para generar los informes. |
| Weka | Herramienta utilizada para hacer el análisis de datos de *Data Mining*. |

***Tabla 21.*** *Resultados de validación con datos trimestrales en Data Mining, Weka.*

## 4.20 Resultados

Los resultados obtenidos han sido buenos. Primero se ha hecho un estudio bastante amplio sobre BI, en el cual se describen las nociones básicas para empezar a trabajar con ello. También se ha seguido una metodología de una forma muy extensa, que permite tener éxito en este tipo de proyectos. Con el desarrollo de esta parte más teórica se pretendía proporcionar los conocimientos básicos para poder detectar este tipo de sistemas.

Con el conocimiento transmitido se pueden detectar problemas reales en organizaciones, y proporcionar posibles soluciones al problema. Para conseguir esto también es necesario que se esté muy involucrado e informado sobre la organización, ya que es imprescindible que se conozca cómo funciona, hacia dónde se dirigen y cómo se pretende hacer, sólo de esta forma se será capaz descubrir la necesidad y los requisitos a cumplir. Con este proyecto se tiene la información suficiente para crear la solución más correcta.

En este proyecto ha sido muy importante realiza el caso práctico siguiendo rigurosamente todos los pasos propuestos en la metodología, de esta forma se han clarificado muchos conceptos y posibles problemas en el desarrollo. Por otro lado, el caso práctico puede servir de guía para futuros casos de aplicación de BI.

Dentro del caso práctico se ha hecho un análisis de predicción con minería de datos para el cual no se requería un resultado con porcentajes de acierto elevados. En los requisitos se especificó que se quería hacer un estudio sobre análisis de los datos para ver si a partir de ellos se podría obtener una predicción fiable sobre el incremento de la prima por trimestre y semestre. Afortunadamente los resultados no sólo han sido los mínimos exigidos, hacer el estudio, sino que además son más que aceptables, tienen un porcentaje de acierto muy elevado (más del 95%). Esto significa que, si los datos fueran reales, la organización podría invertir más en lo clientes que menos van a incrementar sus primas, ofrecerles más atención, más ofertas, etc.

En conjunto se han conseguido todos los objetivos propuestos y además con muy buenos resultados.

# Capítulo 5

# CONCLUSIONES Y FUTURAS LÍNEAS

La principal conclusión es que se debe hacer una distinción entre los sistemas de información clásicos y los sistemas de apoyo a la decisión. Saber que la principal diferencia entre ellos es la evolución de su ciclo de vida. En un SI clásico se siguen pasos secuenciales que tienen que tener un fin para poder empezar el siguiente, y en un sistema de apoyo a la decisión los pasos son iterativos, se puede volver sobre pasos anteriores tantas veces como sea necesario, sin que esto signifique un error en el proceso, y no siempre es obligatorio haber finalizado un paso para empezar el siguiente, depende de qué paso sea.

Ya en los sistemas clásicos de información era muy importante seguir cuidadosamente una metodología para conseguir crear con éxito un sistema, cumplir objetivos, tiempos y costes principalmente. En el caso de los sistemas de apoyo a la decisión, sigue siendo igual de importante o más si cabe. Las organizaciones no poseen mucho conocimiento sobre BI, así pues tampoco todo lo que implica, y por lo tanto puede que no le den importancia a crear un diccionario de datos, *metadata*, etc, pero son pasos que se encuentran dentro de la metodología y deben ser realizados con el mismo interés que el resto, sólo así se conseguirá el éxito del proyecto.

Es importante que no se confunda el éxito del proyecto con cumplir los objetivos actuales de la organización, esto puede dar lugar a frustración, pero no es real. Se puede dar el caso de que los requisitos hayan cambiado cuando se ha finalizado el proyecto, pero este cumple con los requisitos que se definieron inicialmente. Esto no significa que el proyecto no haya tenido éxito, sino que se necesita una evaluación y realizar los pasos pertinentes para ampliar o modificar el estado actual y que se cumplan los nuevos requisitos.

Este tipo de sistemas aportan mucho conocimiento, y por lo tanto valor, a las organizaciones. Sin embargo, la mayoría de éstas no le dan esta importancia y prescinden de ellos pensando que es un gasto, cuando en realidad es una gran inversión que podría darles muchos beneficios, entre ellos económicos.

Como futuras líneas de este proyecto se propone crear un nuevo DM con nuevas tablas de hechos con indicadores sobre distribución territorial, los agentes y los tomadores. Con estas nuevas tablas de hechos se podrían hacer más estudios sobre estos campos con minería de datos, y poder generar más informes dinámicos. Se podría hacer un estudio con minería de datos sobre los agentes, clasificándolos según sus ventas, así se podrá saber cuándo éstos tienen ventas muy inferiores a la media y quizás averiguar por qué. Esto se podría deber a un cambio de zona del agente, y si se detecta a tiempo se puede rectificar el traslado o tomar otras medidas.

# BIBLIOGRAFÍA

**[1].** "*Data Warehouse*", Deakin University.

**[2].** *"Business Inteligence Roadmap",* Larissa T. Moss, Shaku Atre.

**[3].** *"Data Warehousing",* Caludio Cesares.

**[4].** *"Formación en Tecnologías para la Inteligencia de Negocio"*, Curso Plan Avanza.

**[5].** *"Business Intelligence for Dummies",* Swain Scheps.

**[6].** *"Maximizing competitive advantage with highend Business Intelligence technology",* Olapatwork, (2000).

**[7].** *"BI market fraught with instability" ,* Hilson, G. (2001*).*

**[8].** *Mehmed K. Data Mining Concepts, models, methods and algorithms. IEEE Press. 2001.*

**[9].** Business Objects: http://www.sap.com/solutions/sapbusinessobjects/index.epx

**[10].** Acutate: http://www.actuate.com/home/

**[11].** Cognos: http://www.cognos.com.bo/

**[12].** Information Builders: http://www.informationbuilders.es/

**[13].** Microsoft: http://www.microsoft.com/spain/dynamics/default.mspx

**[14].** SAP: http://www.sap.com/spain/index.epx

**[15].** Hyperion Solutions: http://ir.hyperion.com/directors.cfm

**[16].** SAS: http://www.sas.com/

**[17].** Oracle: http://www.oracle.com/index.html

**[18].** Microstrategy: http://www.microstrategy.es/

**[19].** Power Center: http://www.informatica.com

**[20].** Pentaho: http://www.pentaho.com/

**[21].** Weka: http://www.cs.waikato.ac.nz/ml/weka/

**[22].** *"Inteligencia de negocio",* Universidad de Panamá y Carlos III. Agapito Ledezma.

# ANEXO A

En las siguientes tablas se muestra el origen de los datos para cada campo de cada dimensión y de cada tabla de hechos.

| DM_FECHA | |
|---|---|
| **Destino** | **Transformación** |
| Id_Fecha | Llenar de forma secuencial. |
| Dia | 'DD/MM/AAAA' |
| Mes | 'MM/AAAA' |
| Anyo | 'AAAA' |

***Tabla 22.*** *Dimensión DM_FECHA.*

| DM _PRODUCTO | |
|---|---|
| **Destino** | **Transformación** |
| Id_Producto | Llenar de forma secuencial. |
| Cod_Producto | SELECT Cod_producto<br>FROM STG_A1700001 |
| Nom_Producto | SELECT Nom_producto<br>FROM STG_A1700001 |
| Cod_Sector | SELECT Cod_sector<br>FROM STG_A1700001 |
| Nom_Sector | SELECT Nombre_sector<br>FROM STG_A1700001, STG_A1700002<br>WHERE STG_A1700001.Cod_sector == STG_A1700002.Cod_sector |
| Fec_Carga | SYSDATE |

***Tabla 23.*** *Dimensión DM_PRODUCTO.*

| DM_CANAL_DISTRIBUCION | |
|---|---|
| **Destino** | **Transformación** |
| Id_Canal_Distrib | Llenar de forma secuencial |
| Cod_Canal_Distribucion | SELECT Cod_centro_cost<br>FROM STG_A1800001 |
| Nom_Canal_Distribucion | SELECT Nombre_centro<br>FROM STG_A1800001 |
| Fec_Carga | SYSDATE |

***Tabla 24.*** *Dimensión DM_CANAL_DISTRIBUCION.*

| DM_TIPO SUPLEMENTO | |
|---|---|
| **Destino** | **Transformación** |
| Id_Suplemento | Llenar de forma secuencial |
| Cod_Tipo_Sub | SELECT Cod_Tipo<br>FROM STG_A1700004 |
| Nom_Tipo_Sub | SELECT Nom_Tipo<br>FROM STG_ A1700004 |
| Fec_Carga | SYSDATE |

***Tabla 25.*** *Dimensión DM_TIPO_SUPLEMENTO.*

| DM_CAUSA_ANULACION | |
|---|---|
| **Destino** | **Transformación** |
| Id_Causa_Anul | Llenar de forma secuencial |
| Cod_Anul | SELECT Cod_subtipo<br>FROM STG_A1700005<br>WHERE STG_A1700005.Cod_Tipo == “AT” |
| Nom_Anul | SELECT Nom_subtipo<br>FROM STG_ A1700005<br>WHERE STG_A1700005.Cod_Tipo == “AT” |
| Fec_Carga | SYSDATE |

***Tabla 26.*** *Dimensión DM_CAUSA_ANULACION.*

| DM_TOMADOR | |
|---|---|
| **Destino** | **Transformación** |
| Id_Tom | Llenar de forma secuencial |
| Cod_Tom | SELECT Cod_tercero<br>FROM STG_A3000060 |
| Nom_Tom | SELECT Nom_tercero<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Fec_Nac_Tom | SELECT Fec_nacmiento<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Sexo_Tom | SELECT Sexo_tercero<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Direc_Tom | SELECT Direcc_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |

| DM_TOMADOR | |
|---|---|
| **Destino** | **Transformación** |
| Local_Tom | SELECT Localidad_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Cod_Post_Tom | SELECT Cod_postal_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Tlfn_Tom | SELECT Telefono_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Movil_Tom | SELECT Movil_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Email_Tom | SELECT Email_tercero<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Num_Hijos_Tom | SELECT Num_hijos_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Est_Civil_Tom | SELECT Est_civil_terc<br>FROM STG_ A3000060<br>WHERE STG_ A3000060.Tipo_tercero == 2 |
| Fec_Inicio | SYSDATE |
| Fec_Fin | Llenar con 2100 |

***Tabla 27.*** *Dimensión DM_ TOMADOR.*

| DM_AGENTE | |
|---|---|
| **Destino** | **Transformación** |
| Id_Agente | Llenar de forma secuencial |
| Id_Coord_Reg_Ant | SELECT Id_Coord_Reg<br>FROM DM_COORDINACION_REGIONAL, STG_ A1000102<br>WHERE STG_ A1000102.Cod_coord_reg ==<br>DM_COORDINACION_REGIONAL. Cod_coord_reg<br>AND STG_ A1000102.Fec_carga -1 BETWEEN<br>DM_COORDINACION_REGIONAL.Fec_Inicio AND<br>DM_COORDINACION_REGIONAL.Fec_Fin |
| Id_Coord_Reg_Actual | SELECT Id_Coord_Reg<br>FROM DM_COORDINACION_REGIONAL, STG_ A1000102<br>WHERE STG_ A1000102.Cod_coord_reg ==<br>DM_COORDINACION_REGIONAL. Cod_coord_reg<br>AND DM_COORDINACION_REGIONAL.Fec_Fin == 31/12/2100 |

| DM_AGENTE | |
|---|---|
| **Destino** | **Transformación** |
| Id_Dir_Reg_Ant | SELECT Id_Dir_Reg<br>FROM DM_DIRECCION_REGIONAL, STG_ A1000102<br>WHERE STG_ A1000102.Dir-reg == DM_DIRECCION_REGIONAL.<br>Cod_Dir_Reg<br>AND STG_ A1000102.Fec_carga - 1 BETWEEN<br>DM_DIRECCION_REGIONAL.Fec_Inicio AND<br>DM_DIRECCION_REGIONAL.Fec_Fin |
| Id_Dir_Reg_Actual | SELECT Id_Dir_Reg<br>FROM DM_DIRECCION_REGIONAL, STG_ A1000102<br>WHERE STG_ A1000102.Dir-reg == DM_DIRECCION_REGIONAL.<br>Cod_Dir_Reg<br>AND DM_ DIRECCION _REGIONAL.Fec_Fin == 31/12/2100 |
| Cod_Agente | SELECT Cod_agente<br>FROM STG_ A1000102 |
| Nombre_Agente | SELECT Nom_tercero<br>FROM STG_ A3000060, STG_A1000102<br>WHERE STG_ A3000060.Cod_tercero == STG_A1000102.<br>Cod_tercero<br>AND STG_ A3000060.Tipo_tercero == 4 |
| Tlfn_Agente | SELECT Telefono_terc<br>FROM STG_ A3000060, STG_A1000102<br>WHERE STG_ A3000060.Cod_tercero == STG_A1000102.<br>Cod_tercero<br>AND STG_ A3000060.Tipo_tercero == 4 |
| Movil_Agente | SELECT Movil_tercero<br>FROM STG_ A3000060, STG_A1000102<br>WHERE STG_ A3000060.Cod_tercero == STG_A1000102.<br>Cod_tercero<br>AND STG_ A3000060.Tipo_tercero == 4 |
| Email_Agente | SELECT Email_tercero<br>FROM STG_ A3000060, STG_A1000102<br>WHERE STG_ A3000060.Cod_tercero == STG_A1000102.<br>Cod_tercero<br>AND STG_ A3000060.Tipo_tercero == 4 |
| Fec_Inicio | SELECT Fec_validez<br>FROM STG_ A1000102<br>WHERE STG_ A1000102.Cod_agente == Cod_agente |
| Fec_Fin | Llenar con fecha 2100<br>AND<br>UPDATE DM_AGENTE.Fec_Fin = Fec_Inicio<br>WHERE STG_ A1000102. Cod_agente == DM_AGENTE.Cod_Agente<br>AND STG_ A1000102.Fec_carga BETWEEN DM_AGENTE.Fec_Inicio<br>AND DM_AGENTE.Fec_Fin |

***Tabla 28.*** *Dimensión DM_AGENTE.*

| DM_COORDINACION_REGIONAL | |
|---|---|
| **Destino** | **Transformación** |
| Id_Coord_Reg | Llenar de forma secuencial |
| Id_Dir_Reg | SELECT Id_Direc_Reg<br>FROM DM_DIRECCION_REGIONAL, STG_ A1000101<br>WHERE STG_ A1000101.Cod_direc_reg ==<br>DM_DIRECCION_REGIONAL.Cod_Direc_Reg |
| Cod_Coord_Reg | SELECT Cod_coord_reg<br>FROM STG_ A1000101 |
| Nom_Coord_Reg | SELECT Nombre_coord_reg<br>FROM STG_ A1000101 |
| Fec_Inicio | SYSDATE |
| Fec_Fin | Llenar con fecha 2100<br>AND<br>UPDATE DM_ COORDINACION _REGIONAL.Fec_Fin = Fec_Inicio<br>WHERE STG_A1000101.Cod_coord_reg ==<br>DM_COORDINACION_REGIONAL. Cod_coord_reg<br>AND STG_ A1000101.Fec_carga BETWEEN<br>DM_COORDINACION_REGIONAL.Fec_Inicio AND<br>DM_COORDINACION_REGIONAL.Fec_Fin |

***Tabla 29.*** *Dimensión DM_COORDINACION_REGIONAL.*

| DM_DIRECCION_REGIONAL | |
|---|---|
| **Destino** | **Transformación** |
| Id_Dir_Reg | Llenar de forma secuencial |
| Cod_Dir_Reg | SELECT Cod_direc_reg<br>FROM STG_ A1000100 |
| Nom_Dir_Reg | SELECT Nombre_dir_reg<br>FROM STG_ A1000100 |
| Fec_Inicio | SYSDATE |
| Fec_Fin | Llenar con fecha 2100<br>AND<br>UPDATE DM_DIRECCION_REGIONAL.Fec_Fin = Fec_Inicio<br>WHERE STG_A1000100.Cod_direc_reg ==<br>DM_DIRECCION_REGIONAL. Cod_Dir_Reg<br>AND STG_ A1000100.Fec_carga BETWEEN<br>DM_DIRECCION_REGIONAL.Fec_Inicio AND<br>DM_DIRECCION_REGIONAL.Fec_Fin |

***Tabla 30.*** *Dimensión DM_DIRECCION_REGIONAL.*

| HECHOS_POLIZAS_EMITIDAS | |
|---|---|
| **Destino** | **Transformación** |
| Id_Producto | SELECT Id_producto<br>FROM DM_PRODUCTO<br>WHERE DM_PRODUCTO.Cod_producto =<br>STG_A3000030.Cod_producto; |
| Id_Canal_Distribucion | SELECT Id_canal_distrib<br>FROM DM_CANAL_DISTRIBUCION<br>WHERE DM_CANAL_DISTRIBUCION.Cod_centro_coste =<br>STG_A3000030.Cod_centro_coste; |
| Id_Tom | SELECT Id_tomador<br>FROM DM_TOMADOR<br>WHERE DM_TOMADOR.Cod_tomador =<br>STG_A3000030.Cod_tomador; |
| Id_Suplemento | SELECT Id_suplemento<br>FROM DM_TIPO_SUPLEMENTO<br>WHERE DM_TIPO_SUPLEMENTO.Cod_tipo = STG_A3000030.<br>Cod_tipo; |
| Id_Agente | SELECT Id_agente<br>FROM DM_AGENTE<br>WHERE DM_AGENTE.Cod_agente =<br>STG_A3000030.Cod_agente; |
| Id_Dir_Reg | SELECT Id_direc_reg<br>FROM DM_DIRECCION_REGIONAL<br>WHERE DM_DIRECCION_REGIONAL.Cod_direc_reg =<br>STG_A3000030.Cod_dir_reg; |
| Id_Coord_Reg | SELECT Id_coord_reg<br>FROM DM_COORDINACION_REGIONAL<br>WHERE DM_COORDINACION_REGIONAL.Cod_coord_reg =<br>STG_A3000030.Cod_coord_reg; |
| Id_Fecha_Emision_Pol | SELECT Id_fecha<br>FROM DM_FECHA<br>WHERE DM_FECHA.dia = STG_A3000030.Fec_ini_pol |
| Id_Fecha_Fin_Spto | SELECT Id_fecha<br>FROM DM_FECHA<br>WHERE DM_FECHA.dia = STG_A3000030.Fec_fin_vig_spto |
| Id_Fecha | SELECT Id_fecha<br>FROM DM_FECHA<br>WHERE DM_FECHA.dia = STG_A3000030.Fec_ini_vig_spto |
| Num_Poliza | Num_poliza |
| Prima_Comercial_Acta | STG_A3000030.Prima_Total - STG_A3000030. Imp_Impuestos |
| Num_Sup | Num_spto |

***Tabla 31.*** *Tabla HECHO_POLIZA_EMITIDA.*

| HECHO_POLIZA_ANULADA | |
|---|---|
| **Destino** | **Transformación** |
| Id_Producto | Id_Producto |
| Id_Canal_Distribucion | Id_Canal_Distribucion |
| Id_Causa_Anul | Id_Causa_Anul |
| Id_Tom | Id_Tom |
| Id_Agente | Id_Agente |
| Id_Dir_Reg | Id_Dir_Reg |
| Id_Coord_Reg | Id_Coord_Reg |
| Id_Fecha_Emision_Pol | Id_Fecha_Emision_Pol |
| Id_Fecha_Fin_Spto | Id_Fecha_Fin_Spto |
| Id_Fecha | Id_Fecha |
| Num_Poliza | Num_Poliza |
| Prima_Comercial_Polizas_Anuladas | Prima_Comercial_Polizas_Anuladas |
| Num_Polizas_Anuladas | 1 |

***Tabla 32.*** *Tabla HECHO_POLIZA_ANULADA.*

| HECHOS_POLIZAS_VIGENTES | |
|---|---|
| **Destino** | **Transformación** |
| Id_Producto | Id_Producto |
| Id_Tom | Id_Tom |
| Id_Agente | Id_Agente |
| Id_Coord_Reg | Id_Coord_Reg |
| Id_Dir_Reg | Id_Dir_reg |
| Id_Fecha_Emision_Pol | Id_Fecha_Emision_Pol |
| Id_Fecha_Fin_Spto | Id_Fecha_Fin_Spto |
| Id_Fecha | Id_Fecha |
| Num_Poliza | Num_poliza |
| Prima_Comercial_Polizas_Vigentes | Prima_Comercial_Polizas_Vigentes |
| Num_Polizas_Vigentes | 1 |

***Tabla 33.*** *Tabla HECHO_POLIZA_VIGENTE.*

# ANEXO B

En las siguientes tablas se muestra el tipo y la descripción de todos los campos de todas las dimensiones y tablas de hechos.

| Nombre Columna | PK | Null? | Tipo Datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Producto | Si | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Producto | | No | DOUBLE(3) | El pk del operacional | |
| Nom_Producto | | No | TEXT(18) | Descripción del Producto | |
| Cod_Sector | | No | DOUBLE (3) | El pk del Código del Sector del Producto en el operacional | |
| Nom_Sector | | No | TEXT(18) | Descripción del sector al que pertenece el producto | |
| Fec_Carga | | No | DATE/TIME | Fecha de carga del dato en el DW | |

***Tabla 34.*** *Diccionario de datos. DM_PRODUCTO.*

| Nombre Columna | PK | Null? | Tipo Datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Canal_Distrib | Si | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Canal_Distribucion | | No | DOUBLE(3) | El pk del operacional | |
| Nom_Canal_Distribucion | | No | TEXT(18) | Descriptivo del tipo de centro de distrib. | |
| Fec_Carga | | No | DATE/TIME | Fecha de carga del dato en el DW | |

***Tabla 35.*** *Diccionario de datos. DM_CANAL_DISTRIBUCIÓN.*

| Nombre Columna | PK | Null? | Tipo Datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Suplemento | Si | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Tipo_Sup | | No | TEXT(18) | El pk del operacional | |
| Nom_Tipo_Sup | | No | TEXT(18) | Descriptivo del tipo de Suplemento | |
| Fec_Carga | | No | DATE/TIME | Fecha de carga del dato en el DW | |

***Tabla 36.*** *Diccionario de datos. DM_TIPO_SUPLEMENTO*

| Nombre Columna | Pk | Null? | Tipo datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Tom | Si | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Tom | | No | TEXT(18) | El pk del operacional | |
| Nom_Tom | | No | TEXT(40) | Descriptivo del tipo de Suplemento | |
| Fec_Nac_Tom | | No | DATE/TIME | Fecha nacimiento del tomador | |
| Sexo_Tom | | No | INTEGER | 1=Femenino, 2=Masculino, 3=Empresa | |
| Direc_Tom | | No | TEXT(80) | Dirección del tomador | |
| Local_Tom | | No | TEXT(18) | Localidad de residencia | |
| Cod_Post_Tom | | No | INTEGER | Código Postal de la Localidad | |
| Tlfn_Tom | | Si | INTEGER | Teléfono fijo del tomador | |
| Movil_Tom | | Si | INTEGER | Móvil del tomador | |
| Email_Tom | | Si | TEXT(18) | e-mail del tomador | |
| Num_Hijos_Tom | | Si | INTEGER | Nº de hijos | |
| Est_Civil_Tom | | Si | INTEGER | 1-Soltero, 2-Casado, 3-Divorciado, 4-Viudo | |
| Fec_Inicio | | No | Date/TIME | Fecha de inicio de validez de los datos del tomador en el Data Mart. | |
| Fec_Fin | | No | Date/TIME | Fecha de fin de validez de los datos del tomador en el Data Mart. | |

***Tabla 37.*** *Diccionario de datos: DM_TOMADOR.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Causa_Anul | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Anul | | No | TEXT(18) | El pk del operacional (viene de Cod_subtipo de la tabla A1700005) | |
| Nom_Anul | | No | TEXT(18) | Descriptivo de la Anulación (viene de Nom_subtipo de la tabla A1700005) | |
| Fec_Carga | | No | DATE/TIME | Fecha de carga del dato en el DW | |

***Tabla 38.*** *Diccionario de datos – DIM_CAUSA_ANULACION.*

| Nombre Columna | Pk | Null? | Tipo datos | Descripción Columna | FK |
|---|---|---|---|---|---|
| Id_Agente | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Agente | | No | DOUBLE(3) | El pk del operacional | |
| Id_Coord_Reg_Actual | | No | INTEGER | La coordinación regional a la que pertenece el agente | DM_Coordinación_Reg |
| Id_Dir_Reg_Actual | | No | INTEGER | La dirección regional a la que pertenece el agente | DM_Direccion_Reg |
| Id_Coord_Reg_Ant | | No | INTEGER | La coord. Reg. anterior del agente | DM_Coordinación_Reg |
| Id_Dir_Reg_Ant | | No | INTEGER | La Dir. Reg. anterior del agente | DM_Direccion_Reg |
| Fec_Inicio | | No | DATE/TIME | Fecha de inicio de validez del agente en una determinada coord. y dir. | |
| Fec_Fin | | No | DATE/TIME | Fecha de fin de validez del agente en una determinada coord. Y dir. | |
| Tlfn_Agente | | Sí | INTEGER | Teléfono fijo del agente | |
| Movil_Agente | | Sí | INTEGER | Móvil del agente | |
| Email_Agente | | Sí | TEXT(18) | e-mail del agente | |

***Tabla 39.*** *Diccionario de datos: DM_ AGENTE.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Id_Dir_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Dir_Reg | | No | DOUBLE(3) | El pk del operacional | |
| Nom_Dir_Reg | | No | TEXT(18) | Descriptivo de la Dirección Regional | |
| Fec_Inicio | | No | Date/TIME | Fecha de inicio de validez de los datos de la dirección en el Data Mart. | |
| Fec_Fin | | No | Date/TIME | Fecha de fin de validez de los datos de la dirección en el Data Mart. | |

***Tabla 40.*** *Diccionario de datos: DM_DIRECCION_REGIONAL.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Id_Coord_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | |
| Cod_Coord_Reg | | No | DOUBLE(3) | El pk del operacional | |
| Nom_Coord_Reg | | No | TEXT(18) | Descriptivo del a Coordinación Regional | |
| Id_Dir_Reg | | No | INTEGER | La dirección Regional a la que pertenece la Coordinación | DM_Direccion_Reg |
| Fec_Inicio | | No | Date/TIME | Fecha de inicio de validez de los datos de la coordinación en el Data Mart. | |
| Fec_Fin | | No | Date/TIME | Fecha de fin de validez de los datos de la coordinación en el Data Mart. | |

***Tabla 41.*** *Diccionario de datos: DM_COORDINACION_REGIONAL.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Id_Producto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Producto |
| Id_Canal_Distrubucion | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Canal_Distribucion |
| Id_Causa_Anul | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Causa_Anulacion |
| Id_Tom | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Tomador |
| Id_Agente | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Agente |
| Id_Dir_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Direccion_Reg |
| Id_Coord_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Coordinacion_Reg |
| Id_Fecha_Emision_Pol | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha_Fin_Spto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Num_Poliza | | No | TEXT(18) | Número identificativo de la póliza | |
| Prima_Comercial_Polizas_anuladas | | No | DOUBLE(10) | Cuantía de la póliza | |
| Numero_Polizas_Anuladas | | No | LONG INTEGER | 1 | |

***Tabla 42.*** *Diccionario de datos: HECHO_POLIZA_ANULADA.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Id_Producto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Producto |
| Id_Canal_Distrubucion | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Canal_Distribucion |
| Id_Tom | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Tomador |
| Id_Suplemento | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Tipo_Suplemento |
| Id_Agente | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Agente |
| Id_Dir_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Direccion_Reg |
| Id_Coord_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Coordinacion_Reg |
| Id_Fecha_Emision_Pol | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha_Fin_Spto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Num_Sup | | No | INTEGER(3) | Número identificativo del número de suplemento de la póliza | |
| Prima_Comercial_del_Acta | | No | DOUBLE(10) | Cuantía de la póliza | |
| Num_Poliza | | No | TEXT(18) | Número identificativo de la póliza | |

***Tabla 43.*** *Diccionario de datos: HECHO_POLIZA_EMITIDA.*

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Id_Producto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Producto |
| Id_Tom | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Tomador |
| Id_Agente | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Agente |
| Id_Dir_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Direccion_Reg |
| Id_Coord_Reg | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Coordinacion_Reg |
| Id_Fecha_Emision_Pol | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha_Fin_Spto | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |
| Id_Fecha | Sí | No | LONG INTEGER | Código Secuencial y Consecutivo | DM_Fecha |

| Nombre columna | Pk | Null? | Tipo datos | Descripción columna | FK |
|---|---|---|---|---|---|
| Num_Poliza | | No | TEXT(18) | Número identificativo de la póliza | |
| Prima_Comercial_Polizas_Vigentes | | No | DOUBLE(10) | Cuantía de la póliza | |
| Numero_Polizas_Vigentes | | No | LONG INTEGER | 1 | |

***Tabla 44.*** *Diccionario de datos: HECHO_POLIZA_VIGENTE.*

# ANEXO C

En las Tablas 44 y 45 se muestran todos los datos que forma los informes generados.

| Dirección Regional | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Número Pólizas |
|---|---|---|---|---|---|
| Andalucia | 2006 | 24 | 5934026 | 247251 | 460 |
| Aragon | 2006 | 24 | 1938641 | 80776 | 124 |
| Asturias | 2006 | 12 | 1055279 | 87939 | 156 |
| Baleares | 2006 | 24 | 650314 | 27096 | 60 |
| Pais Basco | 2006 | 12 | 1336886 | 111407 | 436 |
| Canarias | 2006 | 414 | 2519388 | 6085 | 104 |
| Cantabria | 2006 | 24 | 716762 | 29865 | 50 |
| Cataluna | 2006 | 222 | 7605973 | 34261 | 42 |
| Castilla la Mancha | 2006 | 36 | 2359141 | 65531 | 130 |
| Castilla y Leon | 2006 | 4212 | 4615018 | 1095 | 188 |
| Extremadura | 2006 | 36 | 180236 | 5006 | 61 |
| Galicia | 2006 | 24 | 675951 | 28164 | 111 |
| La Rioja | 2006 | 72 | 7032367 | 97671 | 199 |
| Madrid | 2006 | 24 | 3097310 | 129054 | 82 |
| Murcia | 2006 | 36 | 982168 | 27282 | 47 |
| Navarra | 2006 | 36 | 2880623 | 80017 | 51 |
| Valencia | 2006 | 96 | 344013 | 3583 | 30 |
| Andalucia | 2007 | 62000 | 8664402 | 139 | 1720 |
| Aragon | 2007 | 0 | 3557343 | 0 | 0 |
| Asturias | 2007 | 496 | 1893533 | 3817 | 331 |
| Baleares | 2007 | 62000 | 417577 | 6 | 214 |
| Pais Basco | 2007 | 248 | 2321181 | 9359 | 1591 |
| Canarias | 2007 | 0 | 3101803 | 0 | 0 |
| Cantabria | 2007 | 0 | 1879016 | 0 | 0 |
| Cataluna | 2007 | 5524 | 12272259 | 2221 | 379 |
| Castilla la Mancha | 2007 | 0 | 3413237 | 0 | 0 |
| Castilla y Leon | 2007 | 496 | 8181050 | 16494 | 1356 |
| Extremadura | 2007 | 0 | 718604 | 0 | 0 |
| Galicia | 2007 | 496 | 1693016 | 3413 | 505 |
| La Rioja | 2007 | 1736 | 10843666 | 6246 | 614 |
| Madrid | 2007 | 0 | 3416008 | 0 | 0 |
| Murcia | 2007 | 0 | 2051302 | 0 | 0 |
| Navarra | 2007 | 0 | 3200824 | 0 | 0 |
| Valencia | 2007 | 0 | 511626 | 0 | 0 |
| Andalucia | 2008 | 39312 | 9957677 | 253 | 74 |
| Aragon | 2008 | 11120 | 3833990 | 344 | 124 |
| Asturias | 2008 | 64700 | 1709937 | 26 | 173 |

| Dirección Regional | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Número Pólizas |
|---|---|---|---|---|---|
| Baleares | 2008 | 0 | 639507 | 0 | 0 |
| Pais Basco | 2008 | 14078 | 2571092 | 182 | 136 |
| Canarias | 2008 | 59526 | 3415371 | 57 | 80 |
| Cantabria | 2008 | 2024 | 1809881 | 894 | 55 |
| Cataluna | 2008 | 59934 | 13350765 | 222 | 108 |
| Castilla la Mancha | 2008 | 11130 | 4031367 | 362 | 153 |
| Castilla y Leon | 2008 | 116566 | 10083865 | 86 | 97 |
| Extremadura | 2008 | 18212 | 1415242 | 77 | 229 |
| Galicia | 2008 | 0 | 2361175 | 0 | 0 |
| La Rioja | 2008 | 6370 | 13137977 | 2062 | 185 |
| Madrid | 2008 | 10624 | 2379104 | 223 | 63 |
| Murcia | 2008 | 28836 | 2840226 | 98 | 44 |
| Navarra | 2008 | 9106 | 2446148 | 268 | 287 |
| Valencia | 2008 | 22260 | 317507 | 14 | 51 |

***Tabla 45.*** *Informe por Dirección.*

| Sector | Producto | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Número Pólizas |
|---|---|---|---|---|---|---|
| Accidentes de Trabajo | Accidentes de Trabajo con Itinerario | 2006 | 0 | 21973813 | 0 | 0 |
| Accidentes de Trabajo | Accidentes de Trabajo sin Itinerario | 2006 | 0 | 17546000 | 0 | 0 |
| Diversos | Barcos Pesca | 2006 | 0 | 583440 | 0 | 0 |
| Diversos | Edificio | 2006 | 0 | 803040 | 0 | 0 |
| Diversos | Contenido de Edificio | 2006 | 0 | 147136 | 0 | 0 |
| Diversos | Condominio | 2006 | 0 | 147136 | 0 | 0 |
| Diversos | Maquinas Industriales | 2006 | 0 | 647598 | 0 | 0 |
| Autos | multi-auto | 2006 | 5136 | 2033655 | 395 | 314 |
| Diversos | Transportes Terrestres | 2006 | 768 | 42278 | 55 | 18 |
| Accidentes de Trabajo | Accidentes de Trabajo con Itinerario | 2007 | 62000 | 34488582 | 556 | 1740 |
| Accidentes de Trabajo | Accidentes de Trabajo sin Itinerario | 2007 | 62000 | 23312000 | 0 | 0 |
| Accidentes de Trabajo | Accidentes de Trabajo Domesticas | 2007 | 0 | 2513313 | 0 | 0 |
| Diversos | Barcos Recreo | 2007 | 0 | 734280 | 0 | 0 |
| Diversos | Barcos Pesca | 2007 | 0 | 1422734 | 0 | 0 |
| Diversos | Edificio | 2007 | 0 | 1052331 | 0 | 0 |
| Diversos | Contenido de Edificio | 2007 | 0 | 68912 | 0 | 0 |
| Diversos | Condominio | 2007 | 0 | 101824 | 0 | 0 |
| Diversos | Maquinas Industriales | 2007 | 4712 | 1221798 | 259 | 81 |
| Autos | multi-auto | 2007 | 4284 | 2812778 | 656 | 921 |
| Diversos | Transportes Terrestres | 2007 | 0 | 407895 | 0 | 0 |
| Accidentes de Trabajo | Accidentes de Trabajo co Itinerario | 2008 | 291392 | 38377875 | 131 | 55 |

| Sector | Producto | Año | Prima Anuladas | Prima Vigentes | Ratio Prima | Ratio Número Pólizas |
|---|---|---|---|---|---|---|
| Accidentes de Trabajo | Accidentes de Trabajo sin Itinerario | 2008 | 61000 | 23033000 | 377 | 372 |
| Accidentes de Trabajo | Accidentes de Trabajo Domesticas | 2008 | 51040 | 7012258 | 137 | 172 |
| Diversos | Barcos Recreo | 2008 | 29700 | 1049560 | 35 | 35 |
| Diversos | Barcos Pesca | 2008 | 16192 | 1570851 | 97 | 32 |
| Diversos | Edificio | 2008 | 12144 | 868402 | 71 | 58 |
| Diversos | Contenido de Edificio | 2008 | 170 | 1824 | 10 | 1536 |
| Diversos | Condominio | 2008 | 0 | 171712 | 0 | 0 |
| Diversos | Maquinas Industriales | 2008 | 11160 | 892428 | 79 | 34 |
| Autos | multi-auto | 2008 | 0 | 2842729 | 0 | 0 |
| Diversos | Transportes Terrestres | 2008 | 0 | 483840 | 0 | 0 |

***Tabla 46.*** *Informe por producto.*

# ANEXO D

En la Tabla 46 se describen los algoritmos utilizados en *Weka.*

| ALGOTIRMOS | | DESCRIPCIÓN |
|---|---|---|
| ÁRBOLES | BFTree | Árbol binario que puede utilizar atrubutos numéricos y nominales. |
| | DecisionStamp | Árbol de decisión que se usa para regresión y clasificación. Los campos vacíos se tratan como un valor aparte. |
| | FT | Árbol funcional de clasificación que puede tener funciones de regresión en los nodos interiores y/u hojas. |
| | J48 | Árbol de decisión basado en ID3 que realiza la poda de las ramas que tienen menos capacidad de predicción. |
| | J48graf | Árbol basado en el J48 que incluye injertos. |
| | LADTree | Árbol de decisión multiclase que utiliza la estrategia LogitBoost para decidir. |
| | LMT | Árbol de clasificación con funciones de regresión logística en las hojas. |
| | NBTree | Árbol de decisión que utiliza Bayes para clasificar en las hojas. |
| | RandomForest | Crea un número de árboles al azar. |
| | RandomTree | Selecciona *K* atributos al azar en cada hoja, pero no hace poda. |
| | REPTree | Construye un árbol de regresión basándose en la varianza. |
| | SimpleCart | Utiliza la mínima complejidad para la poda. |
| | UserClassifier | Clasifica de forma iterativa con medios visuales, muestra gráficos de dispersión de los datos. |
| REGLAS | ConjuntiveRule | Implementa una sola regla conjuntiva que puede predecir para clases numéricas y nominales. |

| ALGOTIRMOS | | DESCRIPCIÓN |
|---|---|---|
| | DecisionTable | Construye y utiliza una tabla de decisión simple para clasificar a la mayoría. |
| | DTNB | En cada punto de búsqueda divide en tabla de decisión y Bayes, los evalúa y elimina un atributo. |
| | JRip | Utiliza una regla proposicional que se va repitiendo incrementalmente. |
| | OneR | Utiliza el menor error para el atributo de predicción. |
| | PART | Construye una regla y elimina las instancias que cubre, y así recursivamente hasta que no queda ninguna instancia por cubrir. |
| | Ridor | Se crea una regla por defecto y sus excepciones, se selecciona la mejor excepción y a partir de ahí se vuelve a iterar. |
| | ZeroR | Utiliza un clasificador basado en la media. |
| FUNCIONES | Logistic | Utiliza un modelo de regresión logística multinominal. |
| | MultilayerPerceptron | Clasificador que utiliza *backpropagation* para clasificar. |
| | RBFNetwork | Implementa una red radial de Gauss normalizada. |
| | SimpleLogistic | Clasificador lineal para la construcción de modelos de regresión logística. |
| | SMO | Sustituye valores ausentes, transforma atributos nominales en binarios y normaliza. |

***Tabla 47.*** *Descripción de los algoritmos utilizados en Data Mining, Weka.*

El árbol J48 generado con los datos semestrales y sin selección de atributos es el siguiente:

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:      Datos Semestrales con Incremento Discreto-Test(6)-
weka.filters.unsupervised.attribute.Discretize-F-B3-M-1.0-R14-
weka.filters.unsupervised.attribute.Remove-R9
Instances:    2405
Attributes:   15
              Id_Producto
              Id_Tom
              Id_Agente
              Id_Dir_Reg
              Id_Coord_Reg
              Id_Fec_Emision
              Fec_Nac_Tom
              Sexo_Tom
              Cod_Postal_Tom
              Num_hijos_Tom
              Estado_Civil_Tom
              Num_Periodo
              Prima
              Num_Poliza
              Incremento
Test mode:    10-fold cross-validation
```

```
=== Classifier model (full training set) ===


J48 pruned tree

------------------


Id_Fec_Emision <= 1277
|   Prima <= -20
|   |   Id_Coord_Reg <= 128
|   |   |   Id_Coord_Reg <= 9
|   |   |   |   Id_Coord_Reg <= 6: 2 (7.0)
|   |   |   |   Id_Coord_Reg > 6: 1 (3.0)
|   |   |   Id_Coord_Reg > 9: 2 (81.0)
|   |   Id_Coord_Reg > 128
|   |   |   Id_Coord_Reg <= 151: 1 (45.0)
|   |   |   Id_Coord_Reg > 151
|   |   |   |   Id_Coord_Reg <= 159: 2 (9.0)
|   |   |   |   Id_Coord_Reg > 159: 1 (45.0)
|   Prima > -20
|   |   Prima <= 6350
|   |   |   Id_Coord_Reg <= 641
|   |   |   |   Prima <= 227: 2 (711.0)
|   |   |   |   Prima > 227
|   |   |   |   |   Id_Coord_Reg <= 158
|   |   |   |   |   |   Prima <= 900
|   |   |   |   |   |   |   Id_Coord_Reg <= 67: 2 (77.0)
```

```
    |   |   |   |   |   |   |   Id_Coord_Reg > 67

    |   |   |   |   |   |   |   |   Estado_Civil_Tom <= -1

    |   |   |   |   |   |   |   |   |   Id_Producto <= 10

    |   |   |   |   |   |   |   |   |   |   Prima <= 490

    |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg <= 142: 1
(3.0)

    |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg > 142

    |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg <=
157: 2 (3.0)

    |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg >
157: 1 (3.0)

    |   |   |   |   |   |   |   |   |   |   Prima > 490

    |   |   |   |   |   |   |   |   |   |   |   Id_Producto <= 9: 2
(6.0)

    |   |   |   |   |   |   |   |   |   |   |   Id_Producto > 9

    |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg <=
74: 1 (3.0)

    |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg > 74

    |   |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg
<= 95: 2 (8.0)

    |   |   |   |   |   |   |   |   |   |   |   |   |   Id_Coord_Reg >
95

    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Id_Coord_Reg <= 106: 1 (3.0)

    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Id_Coord_Reg > 106

    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Id_Coord_Reg <= 118: 2 (6.0)

    |   |   |   |   |   |   |   |   |   |   |   |   |   |   |
Id_Coord_Reg > 118: 1 (5.0/1.0)
```

```
|   |   |   |   |   |   |   |   Id_Producto > 10: 2 (3.0)
|   |   |   |   |   |   |   |   Estado_Civil_Tom > -1
|   |   |   |   |   |   |   |   |   Id_Fec_Emision <= 1223: 2
(31.0/1.0)
|   |   |   |   |   |   |   |   |   Id_Fec_Emision > 1223
|   |   |   |   |   |   |   |   |   |   Id_Coord_Reg <= 142: 1
(3.0)
|   |   |   |   |   |   |   |   |   |   Id_Coord_Reg > 142: 2
(10.0)
|   |   |   |   |   |   Prima > 900: 2 (105.0)
|   |   |   |   |   Id_Coord_Reg > 158
|   |   |   |   |   |   Sexo_Tom <= 2
|   |   |   |   |   |   |   Prima <= 1350: 2 (117.0)
|   |   |   |   |   |   |   Prima > 1350
|   |   |   |   |   |   |   |   Id_Fec_Emision <= 1169: 2 (13.0)
|   |   |   |   |   |   |   |   Id_Fec_Emision > 1169: 3 (5.0)
|   |   |   |   |   |   Sexo_Tom > 2: 2 (308.0)
|   |   |   Id_Coord_Reg > 641
|   |   |   |   Id_Coord_Reg <= 797
|   |   |   |   |   Sexo_Tom <= 2
|   |   |   |   |   |   Id_Coord_Reg <= 739
|   |   |   |   |   |   |   Id_Producto <= 4: 1 (3.0)
|   |   |   |   |   |   |   Id_Producto > 4: 2 (7.0)
|   |   |   |   |   |   Id_Coord_Reg > 739: 1 (12.0)
|   |   |   |   |   Sexo_Tom > 2: 2 (15.0)
|   |   |   |   Id_Coord_Reg > 797: 2 (42.0)
|   |   Prima > 6350
```

```
|   |   |   Id_Coord_Reg <= 795
|   |   |   |   Id_Coord_Reg <= 438: 2 (117.0)
|   |   |   |   Id_Coord_Reg > 438
|   |   |   |   |   Id_Producto <= 2: 3 (9.0)
|   |   |   |   |   Id_Producto > 2: 2 (38.0/4.0)
|   |   |   Id_Coord_Reg > 795: 3 (25.0)
Id_Fec_Emision > 1277
|   Prima <= 490
|   |   Prima <= 16: 1 (43.0)
|   |   Prima > 16
|   |   |   Prima <= 340
|   |   |   |   Prima <= 227: 3 (112.0)
|   |   |   |   Prima > 227: 1 (76.0)
|   |   |   Prima > 340: 3 (74.0)
|   Prima > 490
|   |   Prima <= 798: 2 (41.0)
|   |   Prima > 798
|   |   |   Prima <= 6380
|   |   |   |   Num_Poliza <= 4100601242
|   |   |   |   |   Prima <= 4680
|   |   |   |   |   |   Prima <= 3003
|   |   |   |   |   |   |   Id_Fec_Emision <= 1312: 3 (10.0)
|   |   |   |   |   |   |   Id_Fec_Emision > 1312
|   |   |   |   |   |   |   |   Id_Producto <= 3
|   |   |   |   |   |   |   |   |   Id_Fec_Emision <= 1325: 1
(3.0)
```

```
|   |   |   |   |   |   |   |   |   Id_Fec_Emision > 1325: 3 (4.0)

|   |   |   |   |   |   |   |   Id_Producto > 3

|   |   |   |   |   |   |   |   |   Cod_Postal_Tom <= 29007: 2
(7.0)

|   |   |   |   |   |   |   |   |   Cod_Postal_Tom > 29007: 1
(20.0/1.0)

|   |   |   |   |   |   Prima > 3003

|   |   |   |   |   |   |   Cod_Postal_Tom <= 40005: 1 (35.0)

|   |   |   |   |   |   |   Cod_Postal_Tom > 40005

|   |   |   |   |   |   |   |   Estado_Civil_Tom <= -1: 1
(3.0/1.0)

|   |   |   |   |   |   |   |   Estado_Civil_Tom > -1: 3 (5.0/1.0)

|   |   |   |   |   Prima > 4680

|   |   |   |   |   |   Id_Coord_Reg <= 612

|   |   |   |   |   |   |   Id_Producto <= 2: 1 (3.0)

|   |   |   |   |   |   |   Id_Producto > 2: 3 (5.0)

|   |   |   |   |   |   Id_Coord_Reg > 612: 3 (19.0/2.0)

|   |   |   |   Num_Poliza > 4100601242: 3 (35.0)

|   |   |   Prima > 6380: 2 (29.0)


Number of Leaves  :      53


Size of the tree :       105



Time taken to build model: 0.26 seconds
```

```
=== Stratified cross-validation ===

=== Summary ===


Correctly Classified Instances        2347               97.5884 %
Incorrectly Classified Instances        58                2.4116 %
Kappa statistic                          0.9406
Mean absolute error                      0.0226
Root mean squared error                  0.1237
Relative absolute error                  8.1953 %
Root relative squared error             33.3578 %
Total Number of Instances             2405


=== Detailed Accuracy By Class ===


               TP-Rate FP-Rate Precision  Recall F-Measure  ROC-Area  Class
                 0.913  0.008    0.946     0.913   0.929     0.972      1
                 0.994  0.059    0.98      0.994   0.987     0.981      2
                 0.931  0.003    0.979     0.931   0.955     0.985      3
Weighted Avg.    0.976  0.045    0.976     0.976   0.976     0.981


=== Confusion Matrix ===


   a    b    c   <-- classified as
 282   23    4 |    a = 1
   8 1781    2 |    b = 2
   8   13  284 |    c = 3
```