

# DESARROLLO PLATAFORMA WEB LITEBI



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA

Alumno: **Juan Carlos Herrera Bedoya**

Director: **Vicente Pelechano Ferragud**

**TESINA MASTER EN INGENIERIA DE SOFTWARE, METODOS FORMALES  
Y SISTEMAS DE INFORMACION**

**Universidad Politécnica de Valencia**

## INDICE

|         |  |    |
|---------|--|----|
| 1       | INTRODUCCION .....                             | 10 |
| 1.1     | Introduccion.....                              | 10 |
| 1.2     | Estado Del Arte .....                          | 10 |
| 1.3     | Descripción De La Actividad De La Empresa..... | 12 |
| 1.4     | Entorno De Trabajo .....                       | 12 |
| 2       | MÓDULOS DE LITEBI .....                        | 14 |
| 2.1     | Litespace.....                                 | 14 |
| 2.2     | Liteintegrator .....                           | 16 |
| 2.3     | Módulos De Análisis.....                       | 18 |
| 2.3.1   | LiteExplorer .....                             | 18 |
| 2.3.2   | LiteMonitor .....                              | 20 |
| 2.4     | Características Del Producto.....              | 21 |
| 2.4.1   | LiteSpace.....                                 | 24 |
| 2.4.1.1 | Definir un cubo .....                          | 24 |
| 2.4.1.2 | Definición de una dimensión compartida .....   | 27 |
| 2.4.2   | LiteIntegrator.....                            | 29 |
| 2.4.2.1 | Paso de Output a Litebi.....                   | 30 |
| 2.4.3   | LiteExplorer .....                             | 31 |
| 2.4.4   | LiteMonitor .....                              | 32 |
| 3       | IMPLEMENTACION TECNOLÓGICA DE LITEBI .....     | 35 |
| 3.1     | Inteligencia de Negocios .....                 | 35 |
| 3.2     | Software as a Service .....                    | 35 |
| 3.3     | ETL.....                                       | 36 |
| 3.4     | Cubos (Olap).....                              | 37 |
| 3.5     | Dimensión .....                                | 37 |
| 3.6     | Jerarquía .....                                | 38 |
| 3.7     | Data Warehouse .....                           | 38 |

|          |   |    |
|----------|---|----|
| 3.8      | Esquema en estrella.....                                | 38 |
| 3.9      | MDX .....   | 40 |
| 3.9.1    | Principales Aspectos de la Sintaxis .....               | 42 |
| 3.9.2    | Miembros calculados, medidas y fórmulas complejas ..... | 44 |
| 3.10     | Arquitectura de la aplicación .....                     | 45 |
| 3.10.1   | GWT (Google Web Toolkit) .....                          | 46 |
| 3.10.2   | Protocolo de transferencia.....                         | 47 |
| 3.10.3   | Hibernate.....  | 48 |
| 3.10.4   | J2EE.....   | 48 |
| 3.10.5   | Servidor de mapa .....                                  | 49 |
| 3.10.5.1 | GeoServer .....   | 50 |
| 3.10.5.2 | Abrir y compartir datos espaciales.....                 | 51 |
| 3.10.5.3 | Uso libre y open software .....                         | 51 |
| 3.10.5.4 | Interacción con Mapping API existentes.....             | 51 |
| 3.10.6   | Mondrian .....  | 52 |
| 3.10.7   | Kettle .....  | 54 |
| 3.10.8   | MySQL .....   | 55 |
| 3.10.9   | Web Service axis2.....                                  | 55 |
| 3.10.9.1 | Arquitectura Axis2 .....                                | 56 |
| 3.11     | Aspectos técnicos de litebi.....                        | 57 |
| 3.11.1   | Implementación de Axis2 en Litebi.....                  | 59 |
| 3.11.1.1 | Modulo de seguridad para Axis2 .....                    | 68 |
| 4        | CONCLUSIONES Y TRABAJOS FUTUROS .....                   | 74 |
| 5        | Bibliografía .....                                      | 78 |

## ANEXOS

|   |    |
|---|----|
| Anexo 1 – Diagrama de clases de Litebi .....      | 80 |
| Anexo 2 – Documento de requisitos de Litebi ..... | 85 |

## INDICE DE FIGURAS

|   |    |
|---|----|
| Figura 1: LiteSpace (Sistema de archivos y carpetas) .....                              | 15 |
| Figura 2: LiteSpace (Editor de cubos) .....   | 16 |
| Figura 3: LiteIntegrator .....  | 18 |
| Figura 4: LiteExplorer .....  | 19 |
| Figura 5: LiteMonitor (Gestor de diseño) .....  | 20 |
| Figura 6: LiteMonitor (Visualización) .....   | 21 |
| Figura 7: Uso módulos de Litebi .....   | 22 |
| Figura 8: Creando un cubo en LiteSpace .....  | 25 |
| Figura 9: Creando una dimensión el LiteSpace .....                                      | 27 |
| Figura 10: Paso de Output to Litebi (Kettle) .....                                      | 30 |
| Figura 11: Visor Olap de Litebi .....   | 31 |
| Figura 12: Creación de cuadros de mando .....   | 33 |
| Figura 13: Diagrama servicios de Litebi .....   | 34 |
| Figura 14: Cugo Olap .....  | 37 |
| Figura 15: Ejemplo de modelo de datos en estrella .....                                 | 39 |
| Figura 16: Arquitectura Litebi .....  | 46 |
| Figura 17: Infraestructura servidores de mapas .....                                    | 49 |
| Figura 18: Comunicación Litebi - GeoServer .....  | 52 |
| Figura 19: Funcionamiento de Mondrian .....   | 53 |
| Figura 20: Ejemplo fichero XML con el esquema de Mondrian en Litebi .....               | 54 |
| Figura 21: Componentes de la arquitectura de Axis2 .....                                | 57 |
| Figura 22: Diagrama de paquetes en Litebi .....   | 59 |
| Figura 23: Modelo comunicación con Axis2 .....  | 60 |
| Figura 24: Relación Espacio y Usuario .....   | 80 |
| Figura 25: Relación Espacio, Usuario, Fichero y Carpeta .....                           | 81 |
| Figura 26: Relación Cubo, Dimensión, Jerarquía, Nivel, Vista, Dashboard y Fichero ..... | 81 |
| Figura 27: Relación Usuario, Rol, ACL, Fichero y Carpeta .....                          | 81 |
| Figura 28: Generalización Dimension .....   | 82 |

|   |     |
|---|-----|
| Figura 29: Relación Cubo, Measure y Dimensión .....                     | 82  |
| Figura 30: Relación Dimensión Compartida, jerarquía y nivel.....        | 83  |
| Figura 31: Relación vista y cubo .....                                  | 83  |
| Figura 32: Relación vista y Dashboard.....                              | 83  |
| Figura 33: Diagrama de clases de Litebi .....                           | 84  |
| Figura 34: Diagrama de caso de uso: Hacer login .....                   | 89  |
| Figura 35: Diagrama de caso de uso: LiteSpace.....                      | 90  |
| Figura 36: Diagrama de caso de uso: Editor de Cubo.....                 | 100 |
| Figura 37: Diagrama casos de uso: Edición Dimensiones Compartidas ..... | 103 |
| Figura 38: Diagrama de caso de uso: LiteMonitor.....                    | 106 |
| Figura 39: Diagrama de caso de uso: LiteExplorer .....                  | 109 |

## INDICE DE TABLAS

|   |     |
|---|-----|
| Tabla 1: Especificación Caso de Uso de Hacer Login.....                         | 89  |
| Tabla 2: Especificación Caso de Uso de Crear Carpeta .....                      | 91  |
| Tabla 3: Especificación Caso de Uso de Copiar Carpeta.....                      | 91  |
| Tabla 4: Especificación Caso de Uso de Cortar Carpeta .....                     | 92  |
| Tabla 5: Especificación Caso de Uso de Eliminar Carpeta .....                   | 92  |
| Tabla 6: Especificación Caso de Uso de Crear Informe .....                      | 93  |
| Tabla 7: Especificación Caso de Uso de Explorar Informe .....                   | 93  |
| Tabla 8: Especificación Caso de Uso de Eliminar Informe.....                    | 93  |
| Tabla 9: Especificación Caso de Uso de Copiar Informe .....                     | 94  |
| Tabla 10: Especificación Caso de Uso de Cortar Informe.....                     | 94  |
| Tabla 11: Especificación Caso de Uso de Crear Dashboard .....                   | 95  |
| Tabla 12: Especificación Caso de Uso de Visualizar Dashboard.....               | 95  |
| Tabla 13: Especificación Caso de Uso de Eliminar Dashboard.....                 | 95  |
| Tabla 14: Especificación Caso de Uso de Copiar Dashboard.....                   | 96  |
| Tabla 15: Especificación Caso de Uso de Cortar Dashboard .....                  | 96  |
| Tabla 16: Especificación Caso de Uso de Crear Cubo .....                        | 97  |
| Tabla 17: Especificación Caso de Uso de Editar Cubo .....                       | 97  |
| Tabla 18: Especificación Caso de Uso de Eliminar Cubo .....                     | 97  |
| Tabla 19: Especificación Caso de Uso de Cortar Cubo .....                       | 98  |
| Tabla 20: Especificación Caso de Uso de Crear Dimensión Compartida.....         | 98  |
| Tabla 21: Especificación Caso de Uso de Eliminar Dimensión Compartida.....      | 99  |
| Tabla 22: Especificación Caso de Uso de Editar Dimensión Compartida .....       | 99  |
| Tabla 23: Especificación Caso de Uso de Cortar Dimensión Compartida.....        | 99  |
| Tabla 24: Especificación Caso de Uso de Crear y Editar Métrica .....            | 100 |
| Tabla 25: Especificación Caso de Uso de Crear y Editar Dimensión Embebida ..... | 101 |
| Tabla 26: Especificación Caso de Uso de Asociar Dimensión Compartida....        | 101 |
| Tabla 27: Especificación Caso de Uso de Desasociar Dimensión Compartida .....   | 102 |

|   |     |
|---|-----|
| Tabla 28: Especificación Caso de Uso de Guardar y Editar Cubo .....                   | 102 |
| Tabla 29: Especificación Caso de Uso de Crear y Editar Niveles .....                  | 103 |
| Tabla 30: Especificación Caso de Uso de Eliminar Nivel .....                          | 104 |
| Tabla 31: Especificación Caso de Uso de Crear y Editar Jerarquías .....               | 104 |
| Tabla 32: Especificación Caso de Uso de Eliminar Jerarquías .....                     | 104 |
| Tabla 33: Especificación Caso de Uso de Asociar nivel a las Jerarquías .....          | 105 |
| Tabla 34: Especificación Caso de Uso de Guardar y Editar Dimensión Compartida.....    | 105 |
| Tabla 35: Especificación Caso de Crear un Informe .....                               | 107 |
| Tabla 36: Especificación Caso de Uso de Importar un Informe.....                      | 107 |
| Tabla 37: Especificación Caso de Uso de Eliminar Vista Creada .....                   | 107 |
| Tabla 38: Especificación Caso de Uso de Eliminar Vista Importada .....                | 108 |
| Tabla 39: Especificación Caso de Uso de Previsualizar Dashboard .....                 | 108 |
| Tabla 40: Especificación Caso de Uso de Guardar Dashboard .....                       | 108 |
| Tabla 41: Especificación Caso de Uso de Explorar datos de forma multidimensional..... | 109 |
| Tabla 42: Especificación Caso de Uso de Exportar Informe .....                        | 110 |
| Tabla 43: Especificación Caso de Uso de Visualizar Datos Gráficamente.....            | 110 |
| Tabla 44: Especificación Caso de Uso de Guardar Vista.....                            | 110 |
| Tabla 45: Especificación Caso de Uso de Editar Vista.....                             | 111 |

## **AGRADECIMIENTOS**

Quiero dar las gracias y admiración a Javier Giménez Aznar y a Jorge López Mateo, fundadores de Lite Internet Solutions S.L. y compañeros de trabajo, por brindarme la posibilidad de desarrollar la tesina en su empresa y ampliar mis conocimientos profesionales. Lugar donde he tenido un recibimiento muy agradable desde el primer día y que siempre me han ayudado en la medida de lo posible.

Quiero agradecer también la indispensable ayuda del tutor de la tesina, Vicente Pelechano, quien me ha guiado durante el proceso de desarrollo de esta tesina y alcanzar los objetivos fijados.

Y sobre todo quiero agradecer el esfuerzo de mi familia, mis padres Oscar Luís y María Mercedes, a mi tía Lourdes, ya que sin su ayuda no habría podido realizar este máster; extendiendo el agradecimiento a mi hermano Oscar Elías y a mi esposa Yuneidis por estar siempre cerca de mí apoyándome, ayudándome en todo durante mi estancia en este país.

Gracias a todos por darme la oportunidad de ser lo que soy.

## **DATOS DE LA PRÁCTICA EN LA EMPRESA**

No. De horas totales:

**960**

Ubicación puesto:

**Desarrollador de software**

No. créditos a reconocer:

**20**

Denominación de la empresa:

**Lite Internet Solutions, S.L.**

Lugar de realización:

**Pascual y Genís 10 - 3ºD, 46002 Valencia, España**

Web:

**[www.litebi.com](http://www.litebi.com)**

## LISTA DE ABREVIATURAS

|              |  |
|--------------|--|
| <b>SaaS</b>  | <b>Software as a Service</b>                 |
| <b>BI</b>    | <b>Business Intelligence</b>                 |
| <b>ETL</b>   | <b>Extract, Transform and Load</b>           |
| <b>OLAP</b>  | <b>On-Line Analytical Processing</b>         |
| <b>SAP</b>   | <b>Sistemas, Aplicaciones y Productos</b>    |
| <b>ERP</b>   | <b>Enterprise Resource Planning</b>          |
| <b>CRM</b>   | <b>Customer Relationship Management</b>      |
| <b>RDBMS</b> | <b>Relational Database Management System</b> |
| <b>MDX</b>   | <b>MultiDimensional eXpressions</b>          |
| <b>SQL</b>   | <b>Structured Query Language</b>             |
| <b>EIS</b>   | <b>Executive Information System</b>          |
| <b>KML</b>   | <b>Keyhole Markup Language</b>               |
| <b>SOA</b>   | <b>Service Oriented Architecture</b>         |
| <b>SOAP</b>  | <b>Simple Object Access Protocol</b>         |
| <b>MPL</b>   | <b>Mozilla Public License</b>                |
| <b>LGPL</b>  | <b>Lesser General Public License</b>         |
| <b>GWT</b>   | <b>Google Web Toolkit</b>                    |
| <b>KML</b>   | <b>Keyhole Markup Language</b>               |
| <b>GML</b>   | <b>Geography Markup Language</b>             |
| <b>SIG</b>   | <b>Sistema de Información Geográfica</b>     |

# 1 INTRODUCCION

## 1.1 Introducción

Durante el tiempo que he estado trabajando, como practicante, en esta empresa, he aprendido diferentes conceptos que se manejan en la inteligencia de negocios, además de múltiples herramientas, lenguajes y modelos para el desarrollo de software, que es el área en que deseo especializarme al finalizar el Máster.

Litebi es una innovadora plataforma Web de Inteligencia de negocios, comercializada bajo el modelo SaaS, cuya tecnología permitirá integrar y analizar información de cualquier origen de forma sencilla y segura.

Se ha desarrollado Litebi para ser la herramienta que resuelva determinado problema empresarial, la toma de decisiones, de la forma más inteligente (potente, fácil, asequible) que se ha sido capaz de concebir. Creo que una herramienta como Litebi es necesaria y que puede ayudar a muchas empresas sin generar gastos innecesarios.

## 1.2 Estado Del Arte

El modelo SaaS es considerado como una de las tendencias actuales más importantes en el sector de las tecnologías de la información.

El porcentaje de empresas y ejecutivos de TI que utilizan al menos una tecnología SaaS aumentó de un 11% a un 26% en 2006.<sup>1</sup>

---

<sup>1</sup> fuente: Saugatuck Technology

La empresa más representativa y exitosa del modelo SaaS es la empresa de CRM Salesforce, cuya facturación el 2006 fue de 300 millones de dólares y que tiene una cuota de mercado dentro del mundo del CRM SaaS del 44% con unas expectativas enormes de crecimiento.

Prácticamente todas las grandes empresas de tecnología están dando pasos hacia el modelo SaaS (SAP, Microsoft, Oracle, etc...).

La inteligencia de negocio o Business Intelligence es actualmente una de las prioridades de los responsables de informática de las empresas.<sup>2</sup>

Las aplicaciones de analíticas para empresas y los servicios relacionados, generaron un volumen de negocio durante el año 2006 de 16.600 Millones de Dólares.

Por todo esto, tanto **SaaS** como el **BI** son dos mercados con gran proyección. **BI + SaaS** es un mercado emergente y muy atractivo por explotar. Este modelo es un complemento al modelo tradicional de implantación de sistemas de BI y abre una gran oportunidad a miles de Pymes que no tendrán que invertir decenas de miles de euros en software, hardware o mantenimiento.

Entre las características de este modelo destaco:

- No requiere inversión en hardware o software.
- Pago por uso.
- Listo en pocos días.
- Informes y cuadros de mando self-service.
- Herramienta SaaS / On Demand para carga de datos.
- Comunicable con cualquier otra tecnología abierta.

---

<sup>2</sup> fuente: *Gartner, IDC*

El enfoque de posicionamiento de Litebi es proporcionar al cliente las ventajas del SaaS (precio más asequible y flexible, simplicidad, inmediatez), pero manteniendo un alto nivel de calidad y de funcionalidad, más propio de las aplicaciones de BI tradicionales. Para ello se desarrollarán tecnologías propias punteras e innovadoras que permitan llevar las posibilidades del Business Intelligence a un nuevo nivel acorde con un mundo en el que Internet es el centro de los procesos.

### **1.3 Descripción De La Actividad De La Empresa**

Es una empresa Start-up, con sede en Valencia (España), que se dedica al desarrollo y comercialización de una aplicación de **Business Intelligence SaaS**.

La empresa comenzó en el año 2008 con un objetivo en mente, **hacer posible el Business Intelligence para todo el mundo**. En este mundo saturado de información su visión es ofrecer una herramienta que haga fáciles las decisiones de negocio más difíciles.

### **1.4 Entorno De Trabajo**

Como ya se ha mencionado, este trabajo se está desarrollando dentro de la empresa Lite Internet Solutions S.L, cuyos fundadores son Javier Giménez Aznar y Jorge López Mateo.

La mayoría de las empresas recopilan una gran cantidad de datos de sus operaciones de negocio. Para el manejo de esa información se tendría que usar una amplia gama de programas de software como Excel, Access y diferentes aplicaciones de bases de datos para los distintos departamentos de su organización. El uso de múltiples programas de software hace que sea difícil recuperar la información de manera oportuna y realizar el análisis de los datos.

Litebi ofrece diferentes servicios a sus usuarios para el manejo inteligente de la información de su negocio y la toma de decisiones. Los servicios de LiteSpace, LiteIntegrator y LiteMonitor han sido desarrollados por mi persona con la colaboración de Jorge López Mateo, director de investigación y desarrollo de Litebi, bajo los requerimientos de Javier Giménez Aznar, director de Litebi.

En este documento se describe la estructura, características, desarrollo y funcionalidad de cada uno de los módulos o servicios desarrollados para dicha aplicación, así como las diferentes herramientas y tecnologías utilizadas para su implementación.

## 2 MÓDULOS DE LITEBI

Litebi [1] es una completa plataforma SaaS de *Inteligencia de negocios* con *cubos OLAP*, Reportes, Cuadros de Mando y una Herramienta *ETL* de integración de datos. Esta plataforma BI permite definir un data warehouse (almacén de datos) completo, o simplemente cargar y analizar datos de un excel en pocos minutos.

La funcionalidad de Litebi esta dividida en **módulos** y varía desde funcionalidades de probada eficacia como sistemas **OLAP** y **Cuadros de Mando**, hasta tecnologías **innovadoras y propias** que hace que Litebi sea una herramienta de Inteligencia de negocios de una **nueva generación** cómo el uso de técnicas de **integración de información**, **inteligencia artificial** o **web semántica** aplicadas a la toma decisiones empresariales.

### 2.1 Litespace

Es el diccionario de metadatos de Litebi, contiene datos y estructuras de metadatos que serán utilizadas en los procesos de análisis de información por el usuario, englobados en dos familias:

- **Estructurados:** Cubos y dimensiones para el análisis OLAP y la minería de datos. Enfocado a análisis cuantitativos
- **No estructurados:** Para análisis semánticos, textuales y de contenido. Enfocado el análisis conceptual.

Una característica fundamental de liteSpace es la posibilidad de que el usuario, de forma **sencilla y visual defina su propio diccionario** de metadatos, creándose automáticamente las estructuras físicas y lógicas necesarias para albergar la información.

Esta innovación supone un gran salto respecto al modo tradicional de desarrollar sistemas de BI (en concreto las tareas relativas al diseño e implementación de Data Warehouse) y **diferencia a LiteBi de todos sus competidores.**

- Cada cliente dispone de su propio liteSpace (espacio) con sus datos analíticos (Cubos) e informes (Vistas de análisis).
- Editor de Cubos y Dimensiones: Permite definir los modelos analíticos en los que se cargaran los datos.
- Seguridad basada en roles.
- Sistema de gestión de contenidos, permite organizar tus elementos de análisis en carpetas. Cada usuario tiene unas carpetas privadas y hay un espacio compartido de carpetas públicas.

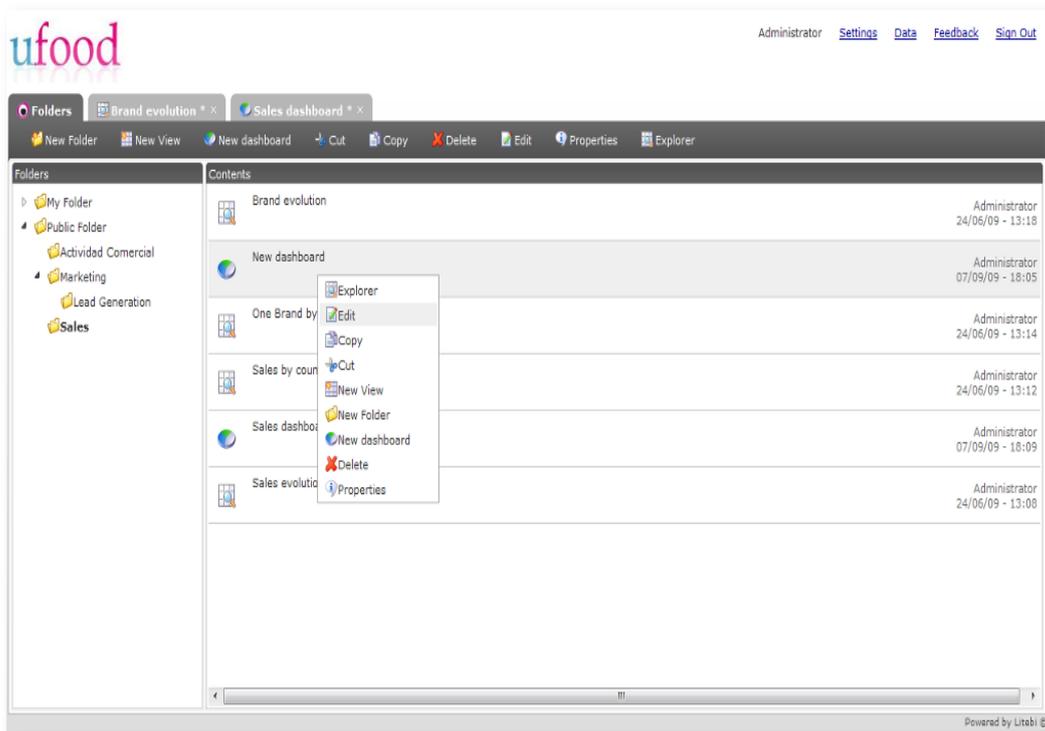
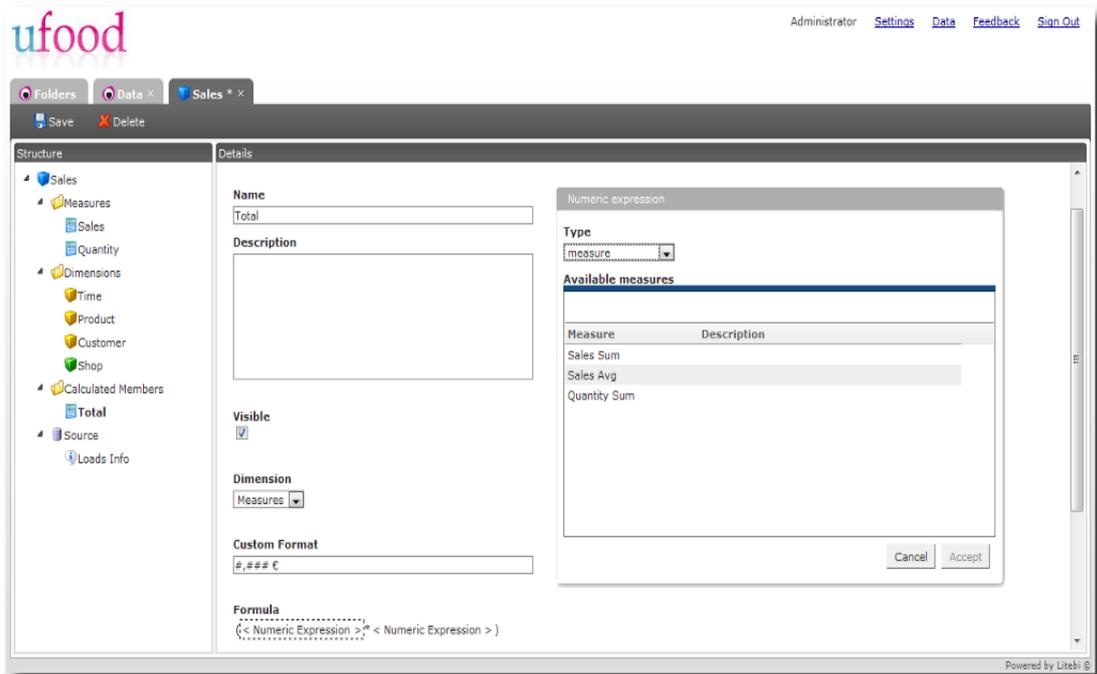


Figura 1: LiteSpace (Sistema de archivos y carpetas)



**Figura 2: LiteSpace (Editor de cubos)**

## 2.2 Liteintegrator

Herramienta de integración de datos basado en la popular y potente herramienta de ETL Open Source Kettle, que pretende llevar la integración de información en organizaciones al entorno actual, en el que cada vez hay más información distribuida en múltiples formatos y fuentes en Internet, que una organización bien informada ha de tener en cuenta junto con los orígenes de datos existentes en sus sistemas.

Permite:

- Definir procesos para integrar, transformar y preparar la información para ser analizada en liteSpace, capaces de ejecutarse de forma periódica en la plataforma de Litebi.
- Integrar datos provenientes de la infraestructura del cliente, situados “detrás del firewall”. Por ejemplo: BBDD corporativas, ficheros excel, texto plano, etc.

- Integrar datos provenientes de orígenes situados en internet: Por ejemplo: Otras aplicaciones SaaS, Plataformas de Cloud Computing, Feeds RSS, servicios web, etc.
- Integrar datos estructurados (Orígenes relacionales, XML, RSS) y no estructurados (Webs, PDFs, Texto, etc.).

LiteIntegrator está compuesto de tres componentes fundamentales:

- 1. Capa Servicios Web:** Desarrollados usando la tecnología AXIS2, permiten carga a través de canales securizados de datos en liteSpace desde cualquier lugar a través de Internet. Suponen la base para la construcción de una **API de Servicios Web** que permita embeber con facilidad la funcionalidad de Litebi en productos de terceros, apoyando la línea de negocio de **OEM partners**.
- 2. Agente de Integración:** Accede a cualquier origen de datos y ejecuta los procesos de integración de Información que cargan liteSpace a través de la **capa de servicios web**.
- 3. Servidor de Integración:** Es responsable de coordinar los procesos de integración de información, ya estén estos ejecutándose en un agente o en el propio servidor. Es capaz de ejecutar procesos de integración en la plataforma de Litebi.

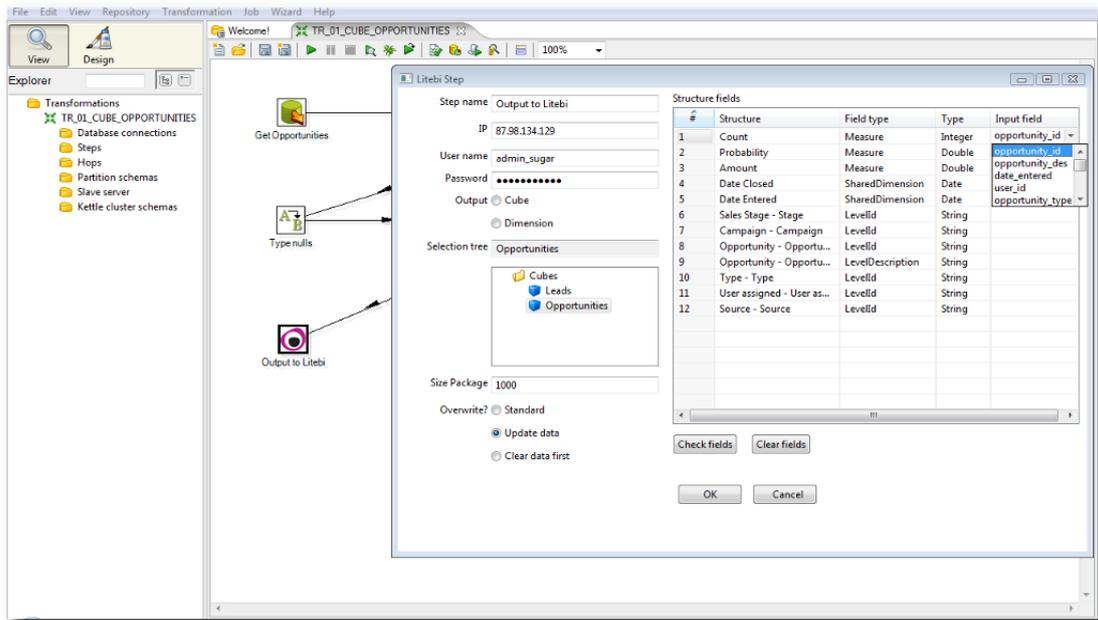


Figura 3: LiteIntegrator

## 2.3 Módulos De Análisis

Una vez los datos están cargados en LiteSpace con LiteIntegrator es posible utilizar diferentes herramientas analíticas.

A continuación se explicarán cada una de estas herramientas.

### 2.3.1 LiteExplorer

Potente herramienta OLAP de análisis y exploración de datos estructurados, de uso sencillo orientada al usuario final, permite:

- Interactuar con la información residente en liteSpace de forma intuitiva, dinámica y potente, ayudando al análisis de grandes cantidad de información para la **toma de decisiones**.
- Realizar operaciones de **exploración multidimensional**, *drill down*, *drill up*, *axis swaping*, filtros, jerarquización, etc...

- Visualizar **gráficamente** los datos explorados, mediante múltiples tipos de gráficas.
- Exportar los informes generados a formatos habituales como Excel y PDF.
- Realizar operaciones de **minería de datos**:
  - **Predicción.** Obtener automáticamente previsiones de resultados futuros en función de los disponibles (ej. *proyección de ventas*)
  - **Extrapolación:** En función de valores numéricos de un área información, predecir posibles valores para otras variables (ej. *Teniendo en cuenta el comportamiento de este producto en esta zona, ¿que he de esperar en esta otra región?*).

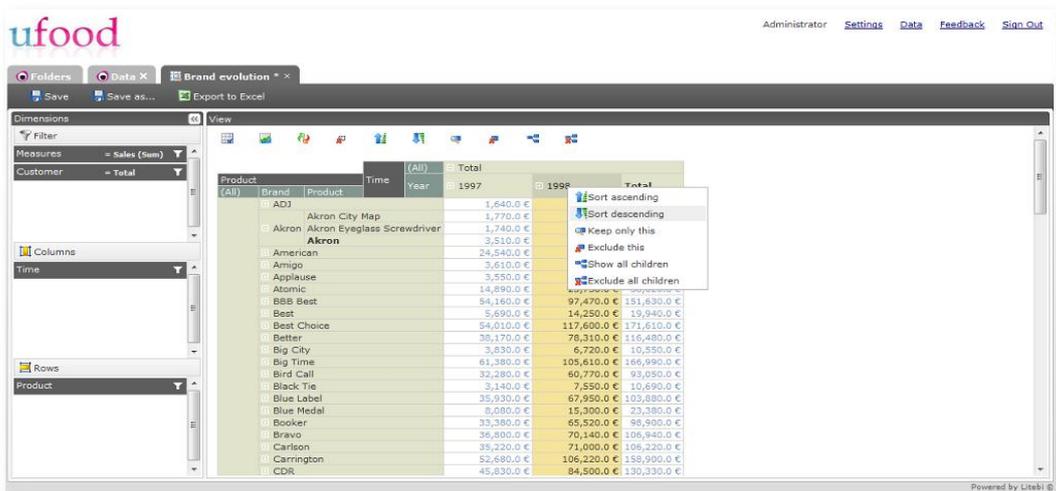


Figura 4: LiteExplorer

### 2.3.2 LiteMonitor

Es la herramienta de cuadros de mando de Litebi (dashboards), permite al usuario diseñar cuadros de mando utilizando **controles visuales** para representar la información existente en *LiteSpace*: Tablas de datos OLAP, gráficas, alarmas, datos textuales, patrones, scorecards, etc., permitiendo la interacción con el mismo mediante **filtros** y operaciones de **drill-through**. En definitiva, LiteMonitor permite al usuario **monitorizar** los aspectos clave del negocio, integrando recursos.

Con LiteMonitor es posible:

- Obtener una vista resumida de la información más relevante de tu empresa.
- Controlar la evolución de tus indicadores clave de rendimiento (KPIs) a menudo.
- Crear cuadros de mandos ricos y potentes a través de una Interfaz Web muy usable.
- Profundizar (Drill-through) desde cualquier dashboard a una vista de detalle en LiteExplorer.

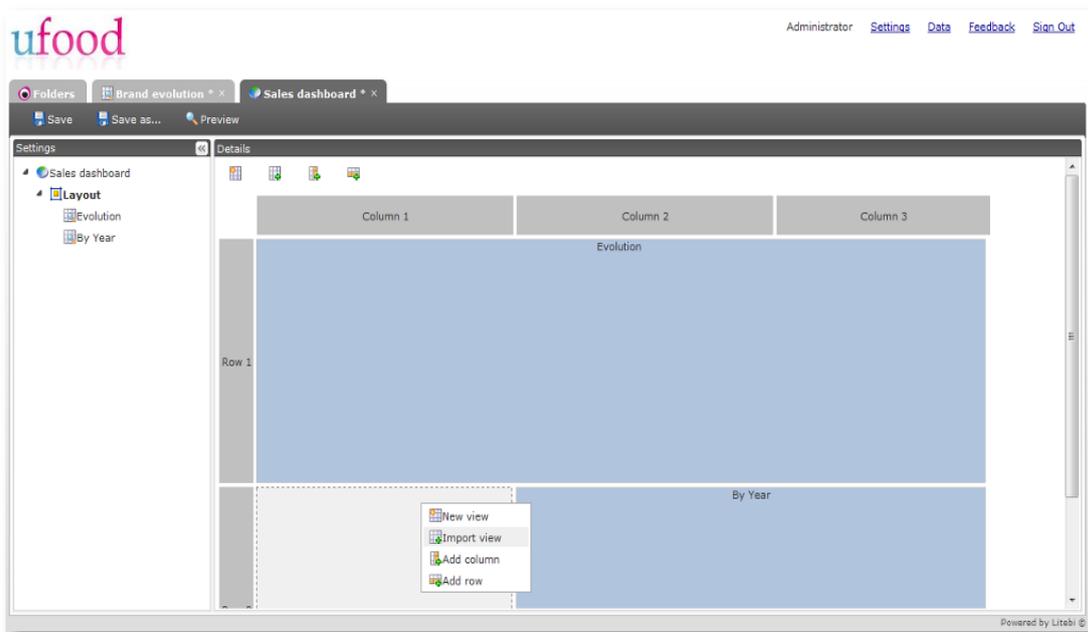


Figura 5: LiteMonitor (Gestor de diseño)

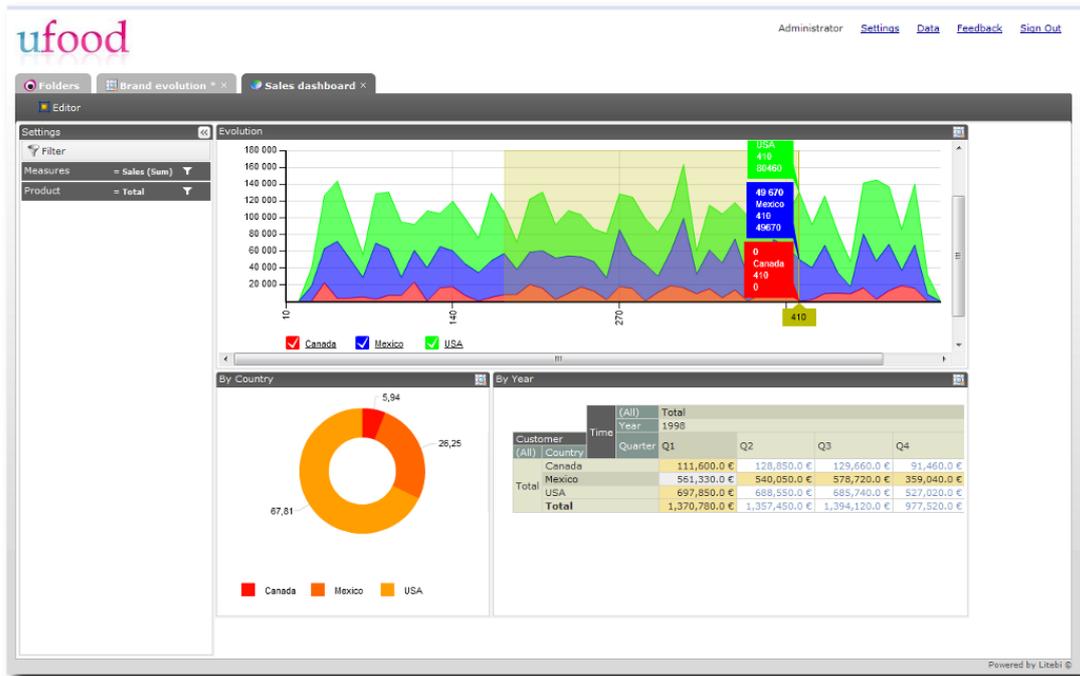
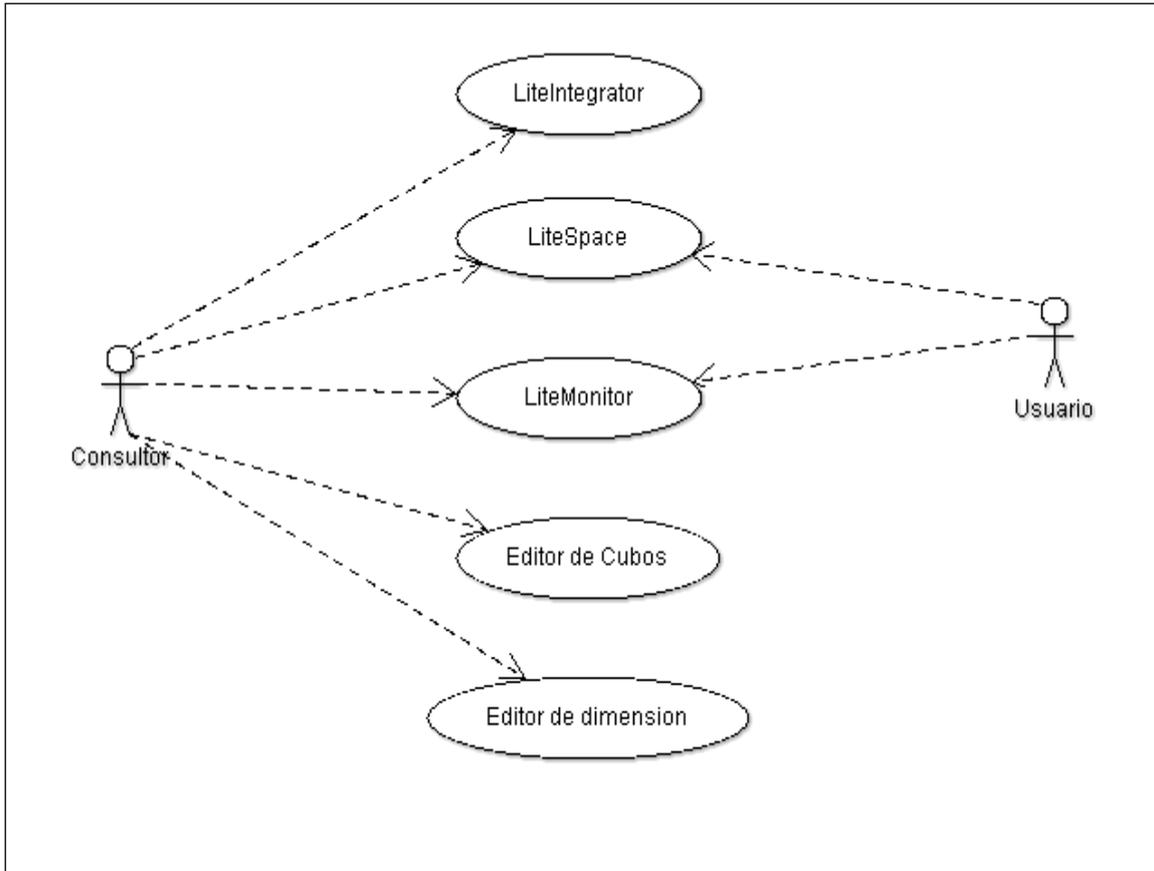


Figura 6: LiteMonitor (Visualización)

## 2.4 Características Del Producto

En un principio los todos los módulos fueron desarrollados para el usuario final, pero en la mayoría de los casos el usuario final no tiene los conocimientos suficientes para generar modelos analíticos eficientes para el manejo de sus datos.

Por lo tanto, los módulos referentes a la manipulación de los modelos analíticos (cubos y/o dimensiones) estarán a cargo de los consultores y/o Partners de Litebi, quienes serán capaces de generar modelos analíticos eficientes para la posterior consulta, manipulación y creación de almacén de datos que podrá estar a cargo del usuario final, cuyos informes consultará para la toma de decisiones inteligentes para su negocio.



**Figura 7: Uso módulos de Litebi**

En la *figura 7* se puede observar los módulos que, en la mayoría de los casos, tienen a cargo el Consultor y el Usuario final. A continuación se explica las características de cada uno de estos módulos.

Básicamente Litebi funciona en tres pasos:

1. **Definir** la información que se desea analizar a través de una interfaz web. Esto genera todas las estructuras de datos y metadatos (datawarehouse y modelos analíticos) necesarios para satisfacer las necesidades de información del cliente. Es a lo que se denomina como **liteSpace**.
2. **Cargar** la información proveniente de los sistemas del cliente (ERP, CRM, Excel, Web Services, etc.) a los modelos definidos

en **liteSpace**. Para hacer esto se apoya en un producto open source de ETL y una capa de servicios web que permite construir de forma sencilla procesos de integración de datos muy potentes.

A este conjunto, que nos permite integrar la información que se desea analizar se llama **liteIntegrator**.

3. **Analizar** una vez los datos residen en **liteSpace**, se cuenta con una herramienta de reporting avanzado (OLAP) con la que el usuario final puede, de forma muy sencilla y potente analizar desde cualquier punto de vista la información y generar sus propios informes que pueden ser compartidos a lo largo de toda la organización. Es el fin de los informes hecho a medida o el caos semi-gestionado a base de excels.

Cabe anotar que el producto ha sido desarrollado sobre el framework GWT (Google Web Toolkit), que permite crear aplicaciones programando desde java y utilizando sus librerías, por ello la posibilidad de asistir a la asignatura Desarrollo de aplicaciones en Java, fue de gran ayuda para una optima utilización de la librerías y poder realizar una programación bien estructurada.

Para almacenar la información necesaria para el funcionamiento de Litebi, se tiene actualmente por un lado en la base de datos lo referente a la configuración de cada espacio y la estructura de los diferentes modelos analíticos, de ahora en adelante llamado SpaceDB; y por otro lado los datos cargados de fuentes externas mapeados para cada modelo analíticos diseñado en Litebi, de ahora en adelante llamado DataMart. *Ver figura 16.*

## 2.4.1 LiteSpace

En el centro de Litebi está liteSpace, que es como un "espacio analítico" o un "Data Warehouse gestionado". La idea es bien sencilla, ser capaz de ofrecer las ventajas de un Data Warehouse clásico (responder a las necesidades analíticas del cliente, integrar información de cualquier origen, dar cobertura a toda la organización, etc.) pero sin la complejidad, el riesgo y el coste que este tipo de proyectos suelen conllevar.

Gracias al modelo SaaS se elimina la necesidad de gestionar el Data Warehouse por el cliente (y el hardware, y el despliegue...), y en parte gracias a que se tiene una tecnología capaz de permitir a cualquiera definir el "que" quiere analizar a través de interfaces web (modelos dimensionales) y dejar que sea Litebi el que haga todo el trabajo duro entre bastidores (construcción de modelos, optimización de los mismo, diseño de metadatos, etc.).

### 2.4.1.1 Definir un cubo

Para definir cubos en Litebi, ha de usarse el editor de cubo en el diccionario de liteSpace. Un cubo coincide a grandes rasgos con el concepto de "tabla de hechos" en un modelo en estrella tradicional (*ver figuras 14 y 15*).

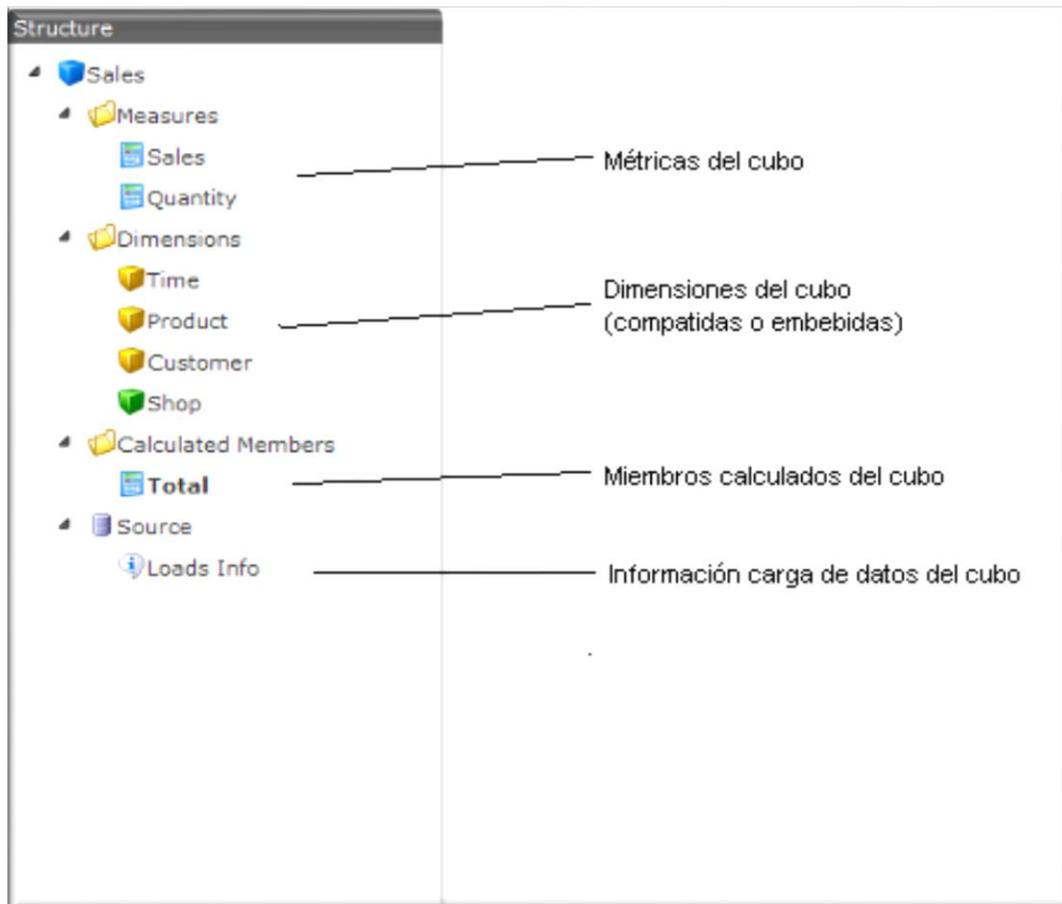


Figura 8: Creando un cubo en LiteSpace

Una vez creado el nuevo cubo, el funcionamiento es muy sencillo:

A la izquierda tenemos un árbol con la estructura del cubo, a la derecha las propiedades del elemento seleccionado.

1. Podemos cambiar el nombre del cubo y su definición.
2. Podemos crear nuevas métricas y definir para cada métrica:
  - **Nombre y descripción**
  - **Formato:** Sigue el estándar de Java (`#,###,###.0`)
  - **Agregador:** Las formas de agregación que deseamos para la métrica al totalizarla (que se sume, que se haga la media de los hijos, que se escoja el máximo o el mínimo de los hijos).
  - **El tipo de datos** (habitualmente Integer o Number para una métrica).

3. Podemos crear dos tipos de dimensiones:

- **Compartidas:** Indicamos que el cubo utilizará una dimensión compartida (definida aparte) y fijamos un alias para el uso. Es posible utilizar una misma dimensión en un cubo más de una vez (*ej: Fecha de venta, Fecha de entrega*).
- **Embebidas o propias:** Son dimensiones que sólo existen para un cubo. Es habitual utilizarlas para definir propiedades (Ej. Estado de una factura) o simplemente para simplificar el modelado. Al crear una dimensión embebida podemos:
  - o Definir su **nombre y descripción**
  - o Especificar si tiene “**Total**” es decir un elemento superior que totalice toda la estructura actuando como cúspide de la jerarquía.
  - o Una **serie de niveles** de detalle que forman una única jerarquía.  
Para cada nivel podemos definir
    - Nombre y descripción
    - Tipo de datos del nivel (de la columna código)
    - Si tiene columna descripción (siempre es de tipo String).  
Además cada nivel puede tener una serie de propiedades asociadas (*Ver definir una dimensiones compartidas*)

*Para más información sobre las columnas código, las columnas descripción de un nivel y propiedades de un nivel ver Definición de una dimensión compartida.*

4. Es posible crear miembros calculados para realizar cálculos avanzados sobre los objetos del cubo, utilizan el lenguaje MDX y es posible definir desde cálculos sencillos (Facturación = Precio unitario \* Cantidad vendida) hasta fórmulas complejas (Incrementos entre periodos de tiempo, porcentajes del total, y un amplio etc.). La edición de miembros calculados requiere de un conocimiento del lenguaje MDX.

### 2.4.1.2 Definición de una dimensión compartida

Desde el diccionario de Litebi es posible definir dimensiones compartidas que serán utilizadas por los cubos. Típicamente en los proyectos hay dimensiones (estructuras jerárquicas) que son utilizadas en varios cubos (sets de datos) cómo por ejemplo la dimensión temporal.

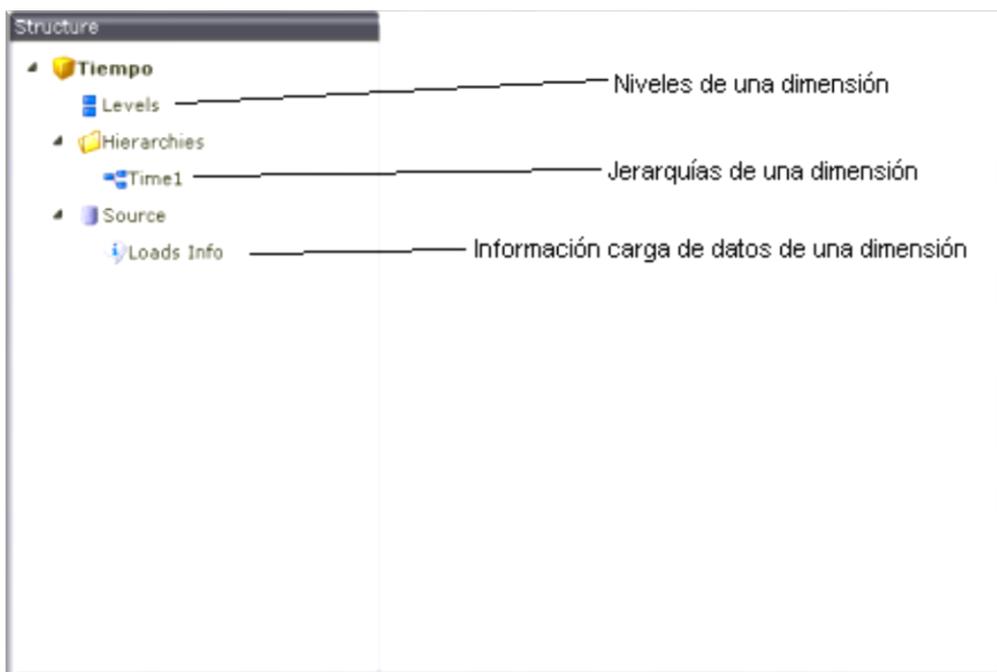


Figura 9: Creando una dimensión el LiteSpace

1. Definir cuál es el **nivel base** de la dimensión.
  - El nivel base es el nivel de máximo detalle de la dimensión, el que posteriormente se utilizará para relacionar la dimensión con el cubo. *Ej. En una dimensión temporal el nivel base podría ser el día, si en el cubo tenemos la información a nivel de día.*
2. Crear los **niveles** que van a ser usados (independientemente de las jerarquías que posteriormente los agrupen). *Ej. Año, Trimestre, Mes, Día de la semana, Semana del año, Quincena, etc...*
  - Al definir estos niveles hay que tener en cuenta que un nivel tiene dos tipos de campos
    - o **Campo Id:**

- Es obligatorio.
- Puede ser de cualquier tipo de datos.
- Define al nivel.

*Ej. En el nivel “Producto” sería el “Código de producto” en el nivel “año” sería el número del año “2007”.*

○ **Campos Descripción:**

- Es opcional.
- Siempre es de tipo String.
- Si existe es utilizado para mostrar el elemento al usuario.
- Si no existe se utiliza el campo Id para mostrar el elemento al usuario.

*Ej. En el nivel “producto” sería el “Nombre del producto”, pero en un nivel “año” no sería necesario.*

**3. Definir las jerarquías:** Una vez creados los niveles que van a utilizarse en la dimensión, se definen las jerarquía que los van a agrupar.

- Una dimensión puede tener tantas jerarquías como se desee, para dar más alternativas de análisis al usuario.
- Toda dimensión ha de tener al menos una jerarquía.
- Todas las jerarquías incluyen el nivel base cómo máximo nivel de detalle.

*Ej. Jerarquía “Por Trimestre” formada por los niveles “Año > Trimestre > Mes > Nivel base día”*

Primero que todo, partiendo de los requisitos para el desarrollo de los editores, se ha modificado el SpaceDB y se han generado las clases necesarias para empezar a implementar los diferentes servicios para la creación y edición de un cubo y de una dimensión compartida. (Ver figuras 29 y 30). Para esta tarea fueron muy importantes los conocimientos adquiridos en las asignaturas vistas en este Máster, como por ejemplo, **sistema avanzado de base de datos**, para un buen diseño de las tablas y sus relaciones, y **técnicas avanzadas de**

**ingeniería de requisitos**, para el correcto entendimiento y desarrollo de los requisitos dados para la implementación de estos módulos.

### **2.4.2 LiteIntegrator**

Litebi cuenta con una potente herramienta de ETL a la que llamamos liteIntegrator, que consta de una parte que se ejecuta en el lado del cliente, apoyada en la fantástica herramienta open source Kettle, y una capa de servicios web en el lado de Litebi. *Ver figura 16.*

Es una herramienta que sirve para tomar datos de cualquier origen, mapearlos y cargarlos en los diferentes modelos analíticos definidos en Litebi para un espacio privado del usuario, además permite configurarlo de tal manera que los datos sean cargados diariamente, para siempre tener datos actualizados en los modelos analíticos.

Con Litebi el objetivo es construir los procesos que obtengan la información de los orígenes deseados, los organicen de la forma deseada para cargarlos en los cubos y dimensiones de liteSpace.

Los problemas que pueden aparecer a la hora de construir procesos de integración son de diversa índole, así que se centrará aquí en el funcionamiento principal de la herramienta.

Para ver un funcionamiento completo de Kettle lo recomendable es su propia y extensa documentación:

[http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+\(a  
ka+Kettle\)+Documentation](http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+(aka+Kettle)+Documentation)

### 2.4.2.1 Paso de Output a Litebi

Además de todos los pasos de lectura, transformación, lookup, etc., de kettle el paso fundamental para cargar datos a Litebi es el “output a litebi”:

- Permite asociar los campos recibidos en la transformación a la estructura de un cubo o una dimensión en liteSpace de un espacio privado, para ello se necesita el usuario y clave del cliente.
- Una vez los campos están correctamente asociados, Kettle es capaz de subir los datos a los cubos y dimensiones quedando estos listos para su análisis.
- La **IP a utilizar** para cargar datos al entorno de Litebi es: **“app.litebi.com”**
- Se puede configurar el número de datos por paquete de envío, por defecto está de 1000 datos por paquete.

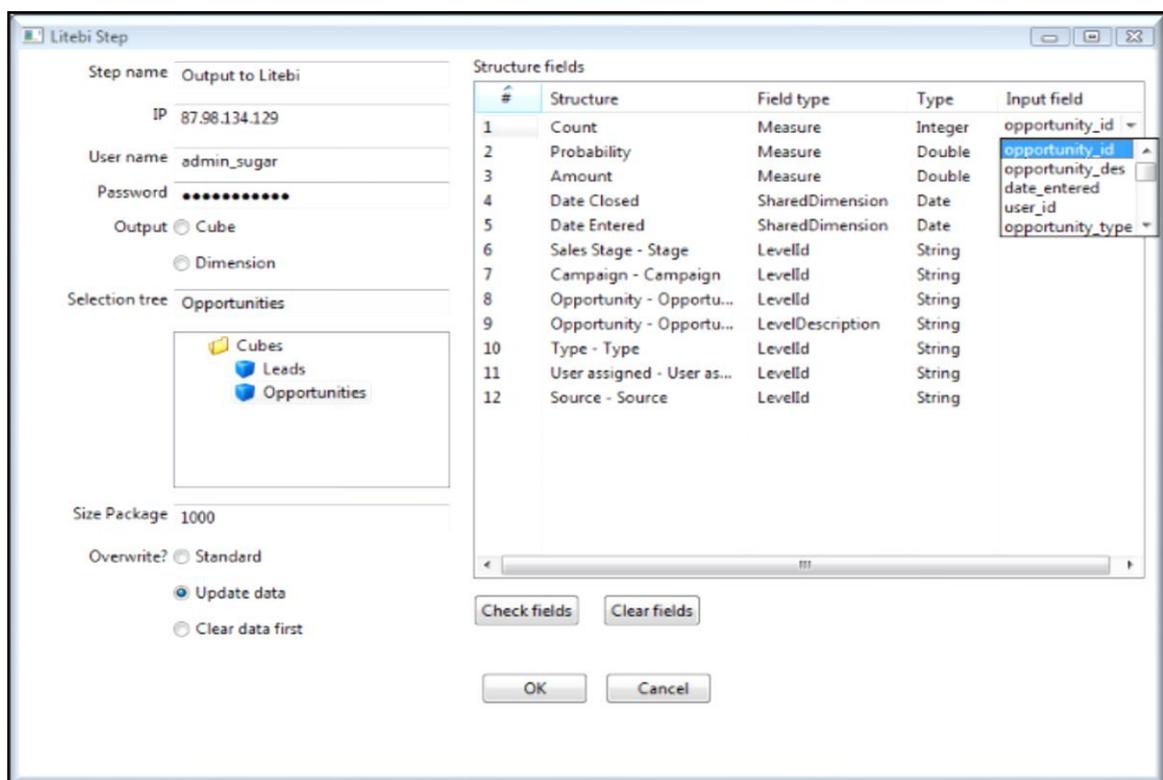


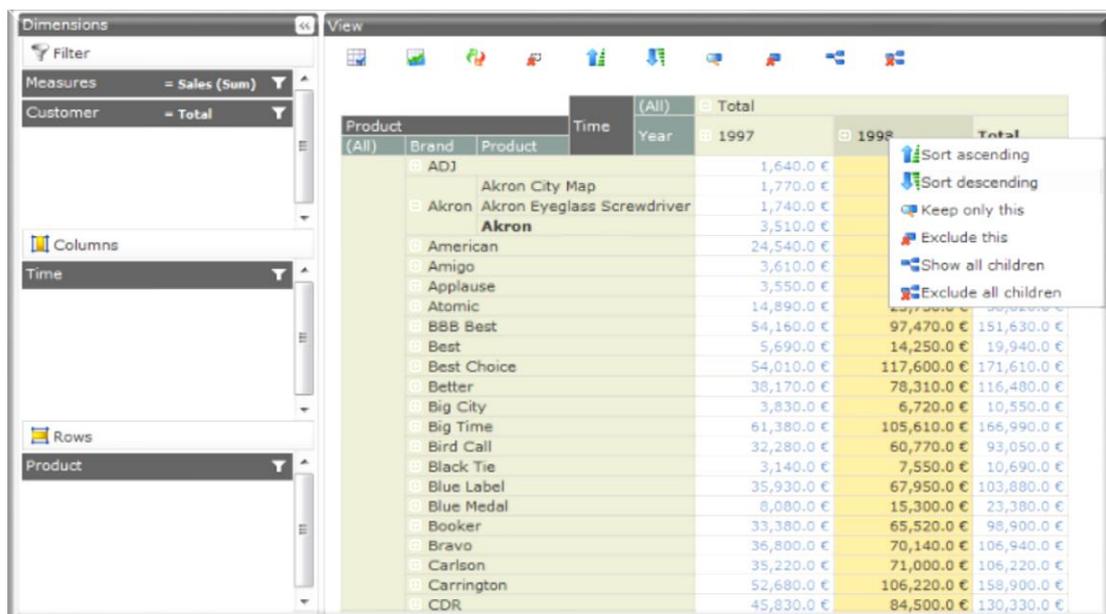
Figura 10: Paso de Output to Litebi (Kettle)

Además de la interfaz gráfica para el mapeo de los datos externos con los modelos analíticos de Litebi, era necesario la implementación de la capa de Servicio Web que tendría la función de conectar a Kettle con el lado del servidor de Litebi, para ello utilizamos AXIS2. Ver figura 23.

Para esta tarea fue importante los conocimientos adquiridos en las asignatura **Modelado, Diseño e Implementación de Servicios Web**, para una buena implementación y configuración de Axis2, ya que se puso en práctica conceptos, arquitecturas y metodologías aprendidas para la correcta implementación del servicio. La implementación tecnológica y configuración de Axis2 será explicado detalladamente en el **capítulo 4**.

### 2.4.3 LiteExplorer

El funcionamiento es muy sencillo, al explorar un cubo o reporte (vista de cubo) se abre liteExplorer quedando en el panel izquierdo disponibles las dimensiones que conforman el cubo y que es posible utilizar para el análisis



The screenshot shows the LiteExplorer OLAP viewer interface. On the left, there are panels for 'Dimensions' (with a 'Filter' icon), 'Columns' (set to 'Time'), and 'Rows' (set to 'Product'). The main area displays a pivot table with the following data:

| Product | Brand | Product | Time  | Year  | 1997       | 1998        | Total       |
|---------|-------|---------|-------|-------|------------|-------------|-------------|
| (All)   | (All) | (All)   | (All) | (All) | Total      | Total       | Total       |
|         |       |         |       | 1997  |            |             |             |
|         |       |         |       | 1998  |            |             |             |
|         |       |         |       | Total |            |             |             |
|         |       |         |       |       | 1,640.0 €  |             |             |
|         |       |         |       |       | 1,770.0 €  |             |             |
|         |       |         |       |       | 1,740.0 €  |             |             |
|         |       |         |       |       | 3,510.0 €  |             |             |
|         |       |         |       |       | 24,540.0 € |             |             |
|         |       |         |       |       | 3,610.0 €  |             |             |
|         |       |         |       |       | 3,550.0 €  |             |             |
|         |       |         |       |       | 14,890.0 € |             |             |
|         |       |         |       |       | 54,160.0 € | 97,470.0 €  | 151,630.0 € |
|         |       |         |       |       | 5,690.0 €  | 14,250.0 €  | 19,940.0 €  |
|         |       |         |       |       | 54,010.0 € | 117,600.0 € | 171,610.0 € |
|         |       |         |       |       | 38,170.0 € | 78,310.0 €  | 116,480.0 € |
|         |       |         |       |       | 3,830.0 €  | 6,720.0 €   | 10,550.0 €  |
|         |       |         |       |       | 61,380.0 € | 105,610.0 € | 166,990.0 € |
|         |       |         |       |       | 32,280.0 € | 60,770.0 €  | 93,050.0 €  |
|         |       |         |       |       | 3,140.0 €  | 7,550.0 €   | 10,690.0 €  |
|         |       |         |       |       | 35,930.0 € | 67,950.0 €  | 103,880.0 € |
|         |       |         |       |       | 8,080.0 €  | 15,300.0 €  | 23,380.0 €  |
|         |       |         |       |       | 33,380.0 € | 65,520.0 €  | 98,900.0 €  |
|         |       |         |       |       | 36,800.0 € | 70,140.0 €  | 106,940.0 € |
|         |       |         |       |       | 35,220.0 € | 71,000.0 €  | 106,220.0 € |
|         |       |         |       |       | 52,680.0 € | 106,220.0 € | 158,900.0 € |
|         |       |         |       |       | 45,830.0 € | 84,500.0 €  | 130,330.0 € |

Figura 11: Visor Olap de Litebi

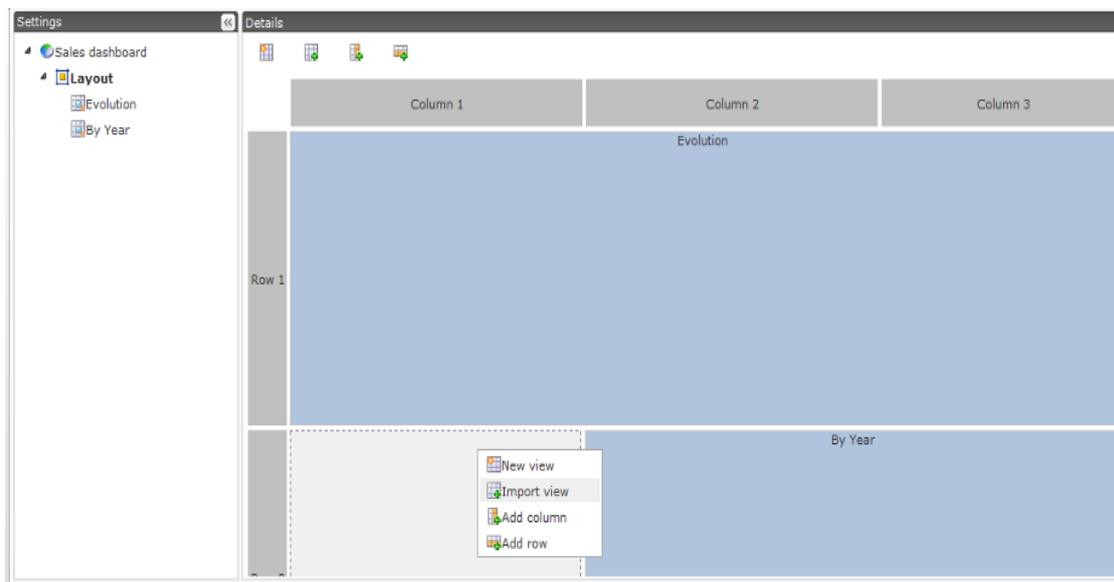
Éste panel tiene tres áreas:

- **Filtros:** Dimensión que no se están visualizando en la vista pero sobre las que es posible seleccionar filtros (acotar la información que se ve). *Ej. No deseo ver los años en filas o columnas, pero deseo ver datos únicamente del año 2008.*
- **Columnas:** Dimensiones que deseo ver en la vista del cubo cómo columnas.
- **Filas:** Dimensiones que deseo ver en la vista del cubo cómo filas.
- **Edición de Filtros:** Sobre cualquier dimensión es posible activar (mediante botón derecho del ratón) los filtros, pudiendo seleccionar los elementos de la dimensión que quiero utilizar en la vista.

Además se da la posibilidad de generar gráficos con respecto a los datos observados en el Visor Olap y además, si existen datos geográficos, es posible obtener un mapa que represente estos datos, para ello se utiliza el servidor de mapas GeoServer **Ver capítulo 4.**

#### **2.4.4 LiteMonitor**

Con LiteMonitor es posible crear cuadros de mando, con los datos cargados en Litebi, en minutos y modificarlos en tiempo record, todo desde una sencilla interfaz web.



**Figura 12: Creación de cuadros de mando**

1. Cada cuadro de mando está asociado a una vista o informe.
2. La vista o informe asociada puede ser importada o creada desde LiteMonitor.
3. En el panel de la izquierda aparecerá en forma de árbol las diferentes vistas asociadas al tablero de los cuadros de mando.
4. Se podrá decidir posición y tamaño de cada cuadro de mando.
5. Se tiene la opción de vista previa.

Para el desarrollo de este servicio fue de gran ayuda tener conocimientos avanzados en HTML, además modificar el SpaceDB y generar y/o modificar las clases necesarias para el correcto funcionamiento del servicio. *Ver figura 32.*

En la *figura 13* se puede observar el diagrama de servicios de Litebi, algunos ya implementados y otros para implementación futura.

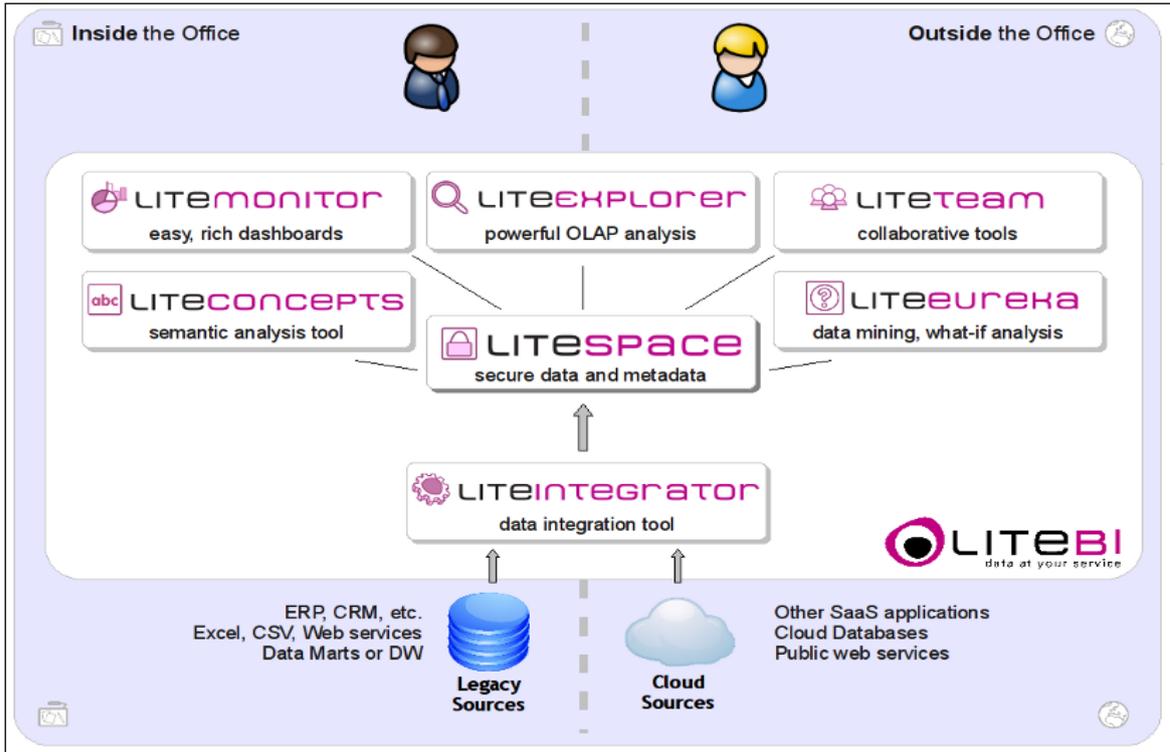


Figura 13: Diagrama servicios de Litebi

### 3 IMPLEMENTACION TECNOLÓGICA DE LITEBI

Para la construcción de Litebi se utilizó la estructura cliente-servidor, que es aquella que se maneja para el uso del modelo SaaS.

#### 3.1 Inteligencia de Negocios

El término Inteligencia de Negocios (BI) representa las herramientas y sistemas que juegan un papel clave en el proceso de planificación estratégica de la corporación. Estos sistemas permiten a una compañía recopilar, almacenar, acceder y analizar datos de la empresa para ayudar a la toma de decisiones. Generalmente estos sistemas ilustran la inteligencia de negocios en las áreas de perfiles de clientes, atención al cliente, estudios de mercado, la segmentación del mercado, la rentabilidad del producto, análisis estadístico, y el inventario y análisis de la distribución por nombrar algunos.

#### 3.2 Software as a Service

*“...También se conoce como 'Software on Demand', permite acceder a todas las funcionalidades de la aplicación a través de la Web.*

*Puede considerarse como una evolución del modelo ASP, con el que a menudo es comparado. En este caso, no se trata de trasladar toda la complejidad de un SAP<sup>3</sup> (Sistemas, Aplicaciones, Productos en procesamiento de datos), por ejemplo, fuera de la empresa, sino de poder customizar y abrir ciertos componentes predefinidos a las características de los usuarios en todo tipo de*

---

<sup>3</sup> Es un sistema ERP (Enterprise Resource Planning), desde el cual se puede administrar toda la empresa. Desde el ingreso de una factura, hasta la salida de mercadería, pasando por la gestión de los recursos humanos, etc.

*aplicaciones, según sus necesidades (presentes y futuras). Esto hace, que económicamente este modelo sea muy beneficioso para los usuarios...” [2]*

*“...El software se hospeda en el servidor del proveedor, controla un proceso del negocio como lo es la gestión de clientes, gestión de proyectos, punto de venta y hasta el control de la totalidad de los procesos de la empresa a través de la integración de un ERP; el software es propiedad del proveedor y el cliente paga una cuota mensual, semestral o anual por tener acceso al software en línea...” [3]*

### **3.3 ETL**

*“...Los procesos de Extracción, Transformación y Carga constan de múltiples pasos, cuyo objetivo es transferir datos desde las aplicaciones de producción a los sistemas de Inteligencia de negocio:*

- *Extracción de los datos desde las aplicaciones y bases de datos de producción (ERP, CRM, RDBMS, archivos, etc.)*
- *Transformación de estos datos para reconciliarlos en todos los sistemas source, realizar cálculos o análisis sintáctico de cadenas, enriquecerlos con información de búsqueda externa y, además, adaptarlos al formato preciso por el sistema objetivo (Third Normal Form, Star Schema, Slowly Changing Dimensions, etc.)*
- *Carga de los datos resultantes en las diversas aplicaciones de BI: almacenes de datos históricos generales (data warehouse) o almacenes de datos empresariales, almacenes de datos históricos individuales (data mart), aplicaciones OLAP (Procesamiento analítico en línea) o “cubos”, etc.*

*La latencia de los procesos ETL varía desde los lotes (a veces, de forma mensual o semanal, pero en la mayoría de los casos diariamente), al tiempo*

casi real con actualizaciones más frecuentes (cada hora, cada pocos minutos, etc.)...” [4]

### 3.4 Cubos (Olap)

Este proceso consiste en obtener datos relevantes entre la gran cantidad de información contenida en el sistema. Se pueden agregar múltiples dimensiones para realizar los cruces que permitirán extraer, en forma rápida y eficiente, la información que se requiere. Los cubos están formados por métricas y dimensiones. Ej. Cubo de ventas, Cubo de contabilidad, etc.

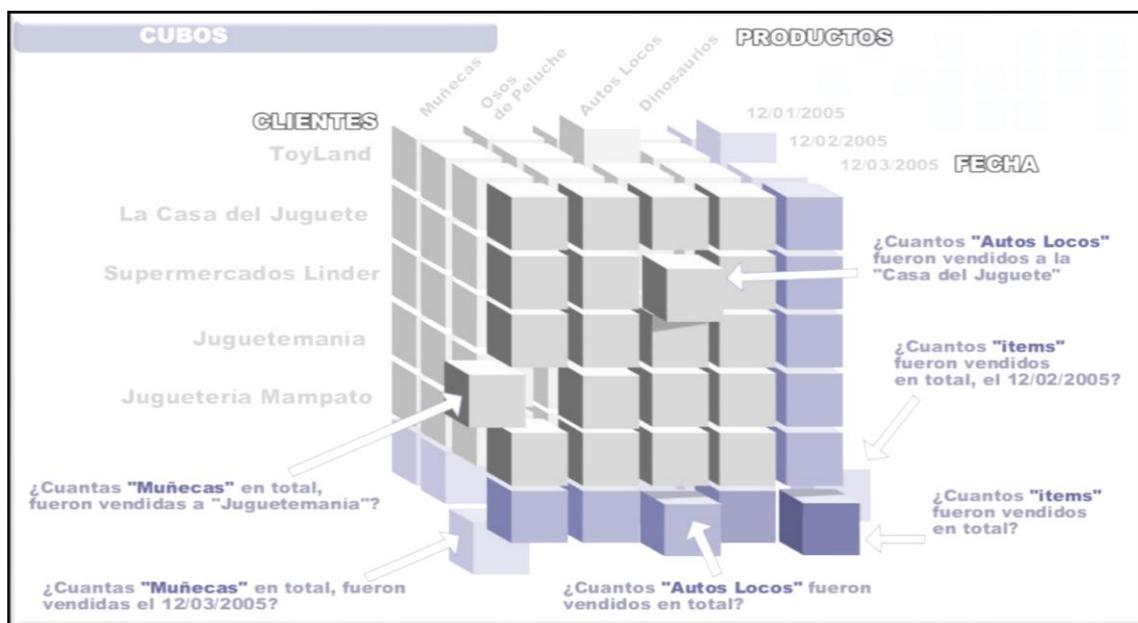


Figura 14: Cubo Olap<sup>4</sup>

### 3.5 Dimensión

Una dimensión representa una perspectiva de los datos. Las dimensiones son usadas para seleccionar y agregar datos a un cierto nivel deseado de detalle.

<sup>4</sup> Figura extraída: DYBOX, Cubos (OLAP). Chile <http://www.dybox.cl/pdf/cubos.PDF>, pág.4, 2007

Las dimensiones se relacionan en jerarquías o niveles. *Ejemplo: la dimensión **Tiempo** que presenta los niveles: día, semana, mes y año.*

### 3.6 Jerarquía

Una dimensión puede tener una o varias jerarquías, una jerarquía es un conjunto de niveles cada uno expresando un nivel de profundidad en la información. Permiten analizar la información de forma agrupada (ventas por año) o al detalle (ventas diarias). *Ejemplo. La dimensión tiempo está formada por la jerarquía “Por Trimestre”, que tiene los niveles “Año > Trimestre > Mes > Día”.*

### 3.7 Data Warehouse

“...El **Data Warehouse** (traducido como Almacén o Bodega de datos normalmente) es un concepto central a partir del cual (junto con los Sistemas de Información ejecutiva y unas cuantas siglas más) se desarrolla hace más de una década el concepto del **Business Intelligence**. En esencia consiste en construir un gran almacén de datos que recoja las informaciones empresariales de fuente de datos heterogéneas, la integre, la limpie, la almacene históricamente, y la deje disponible para el análisis....”. [5]

### 3.8 Esquema en estrella

El esquema en estrella es la representación más importante del modelo dimensional. En el modelo dimensional encontramos hechos y dimensiones. Todo objeto de análisis es un hecho. Este hecho se representa en el modelo dimensional en forma de tabla de hechos o fact table. Los hechos son analizados a su vez, a través de las dimensiones o componentes (tantas como dimensiones participen en la descripción del hecho), que se representan en el modelo dimensional a partir de las tablas de dimensiones.

Si realizamos este esquema mental (tabla de hechos en el centro y tablas de dimensiones alrededor), todo parece dibujar una forma de estrella, origen del nombre.

Los hechos tienen columnas de datos denominadas métricas y las dimensiones tienen columnas que representan los niveles de jerarquías.

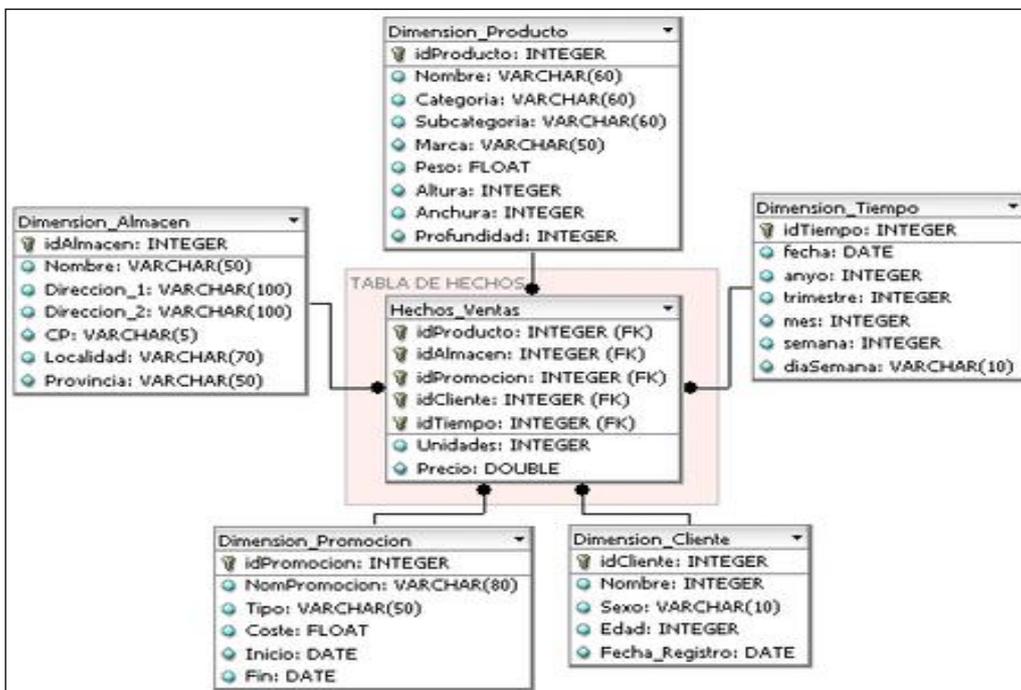


Figura 15: Ejemplo de modelo de datos en estrella

Este esquema es ideal por su simplicidad y velocidad para ser usado en análisis multidimensionales (OLAP, Datamarts, EIS, etc.). Permite acceder tanto a datos agregados como de detalle.

El diseño de esquemas en estrella permite implementar la funcionalidad de una base de datos multidimensional utilizando una clásica base de datos relacional (más extendidas que las multidimensionales).

Otra razón para utilizar los esquemas en estrella es su simplicidad desde el punto de vista del usuario final. Las consultas no son complicadas, ya que las condiciones y las uniones (JOIN) necesarias sólo involucran a la tabla de hechos y a las de dimensiones, no haciendo falta que se encadenen uniones y condiciones a dos o más niveles como ocurriría en un esquema en copo de nieve. En la mayoría de los casos son preferibles los de estrellas por su simplicidad respecto a los de copo de nieve por ser más fáciles de manejar. Finalmente, es la opción con mejor rendimiento y velocidad pues permite indexar las dimensiones de forma individualizada sin que repercuta en el rendimiento de la base de datos en su conjunto.

### **3.9 MDX**

Este lenguaje fue creado en 1997 por Microsoft. No es un lenguaje estándar, sin embargo diferentes fabricantes de herramientas OLAP (entre ellos Mondrian [6]) lo han adoptado como estándar de hecho.

El MDX es en los sistemas OLAP el equivalente al SQL en los sistemas gestores de bases de datos relacionales. Eso significa que es el lenguaje a través del cual podemos explotar la información que reside en los motores OLAP y satisfacer consultas multidimensionales.

El análisis multidimensional consiste en organizar la información y los datos de una organización de forma que los elementos medibles o métricas del negocio (también llamados hechos) puedan ser cualificados utilizando las diferentes "dimensiones" o vistas del negocio.

#### **Ejemplo:**

Tenemos un supermercado con una tarjeta de fidelización que nos permita saber quién compra que y cuando.

Un análisis u organización multidimensional de los datos de ventas de un supermercado nos podría permitir responder a preguntas como:

- ¿Qué artículo es el más vendido este mes?
- ¿Qué tienda ha vendido más artículos del tipo X?
- ¿Qué edad tienen los clientes que más gastan en mi supermercado (de media)?

A continuación se ve el paralelismo entre estas consultas y el modelo multidimensional.

- ¿Que artículo es el más vendido este mes?  
En esta pregunta queremos analizar el hecho artículos vendidos y lo queremos analizar desde 2 dimensiones distintas: tiempo y familia de productos
- ¿Que supermercado ha vendido más artículos del tipo X?  
El hecho a contar aquí es el mismo (cantidad de artículos), pero las dimensiones son aquí las diferentes tiendas y la familia de productos

MDX es un lenguaje de consulta más cercano al lenguaje natural que el sql sin embargo tiene funciones y fórmulas que lo hacen muy potente para el análisis de datos.

La principal diferencia del mundo OLAP respecto al mundo relacional radica en que las estructuras dimensionales están jerarquizadas y se representan en forma de árbol y por lo tanto existen relaciones entre los diferentes miembros de las dimensiones.

Este hecho hace que el lenguaje MDX disponga de funciones y atributos especiales que permiten referenciar los diferentes elementos de las dimensiones con expresiones como MIEMBROS-HIJO, MIEMBROS-PRIMO, MIEMBROS-PADRE, etc., haciendo una analogía con los árboles genealógicos.

Estas estructuras jerarquizadas son especialmente útiles para poder visualizar los datos de forma comparada a nivel temporal.

### 3.9.1 Principales Aspectos de la Sintaxis

El lenguaje MDX permite realizar consultas con una sintaxis parecida al lenguaje natural.

Para ejemplificar las consultas y la sintaxis trabajaremos con un ejemplo de un cubo.

Imaginamos un cubo de ventas con las siguientes dimensiones:

- Dimensión temporal de las ventas con niveles de año y mes.
- Dimensión productos vendidos con niveles de familia de productos y productos.
- Dimensión de medidas con importe de las ventas y unidades vendidas.

#### **Ejemplo:**

Si queremos obtener el importe de las ventas para el año 2008 para una familia de productos lácteos, la consulta sería la siguiente:

```
SELECT  
{ [Medidas].[importe ventas]}  
on columns,  
{ [Tiempo].[2008] }  
on rows  
FROM [cubo ventas]  
WHERE ([Familia].[lacteos])
```

Analizando el ejemplo anterior se puede establecer una analogía con una consulta sql, para así explicar mejor la estructura del lenguaje.

En primer se tiene la estructura idéntica general de la consulta con las cláusulas **SELECT ... FROM ... WHERE** que en MDX se usa idénticamente al sql especificando en el *Select* un conjunto de elementos que queremos visualizar, en el *From* indicamos el cubo del que queremos sacar la información y en el *Where* las condiciones de filtrado.

Una de las particularidades del MDX es que la información la devuelve en forma tabular normalmente y debe ser procesada. En la cláusula *Select* se usa la clausula *on columns*. Esta cláusula sirve para separar los elementos que queremos visualizar en las filas de los que queremos en las columnas. En la consulta de ejemplo se verá el importe en las columnas y el tiempo en las filas. Los { } son para permitir listas de elementos en las selecciones y los [ ] encapsulan elementos de las dimensiones y niveles.

A nivel de funcionalidades y potencia a la hora de realizar consultas el MDX es potente como el SQL aunque su objetivo está orientado a temas de comparaciones, relaciones jerárquicas entre elementos, etc.

Una de las funcionalidades que distinguen al MDX es el acceder a los elementos utilizando estructura de árbol. Así para un determinado nivel de una dimensión tenemos comandos como:

- **[Familia].[lácteos].CurrentMember** : Permite acceder al miembro actual.
- **[Familia].[lácteos].Children** : Permite acceder a los hijos de la familia de los lácteos.
- **[Familia].[lácteos].prevMember**: Permite acceder al miembro anterior de la dimensión.

Por otro lado existen diferentes funciones que permiten realizar cálculos y complementar las consultas como por ejemplo:

- **CrossJoin(conjunto\_a,conjunto\_b)**: Obtiene el producto cartesiano de dos conjuntos de datos.
- **BottomCount(conjunto\_datos,N)**: Obtener un número determinado de elementos de un conjunto, empezando por abajo, opcionalmente ordenado.
- **BottomSum(conjunto\_datos,N,S)**: Obtener de un conjunto ordenado de los N elementos cuyo total es como mínimo el especificado en S.
- **Except(conjunto\_a,conjunto\_b)**: Obtener la diferencia entre dos conjuntos.

MDX también tiene implementada muchas funciones matemáticas, estadísticas y trigonométricas que permiten enriquecer los análisis.

### 3.9.2 Miembros calculados, medidas y fórmulas complejas

Una de las funcionalidades más potentes que ofrece el lenguaje MDX es la posibilidad de realizar cálculos complejos tanto dinámicos (en función de los datos que se están analizando en ese momento) como estáticos. Los cubos multidimensionales trabajan con medidas y con miembros calculados. Las medidas son las métricas de la tabla de hechos a las que se aplica una función de agregación (count, distinct count, sum, max, avg, etc.), en otras palabras son las métricas del negocio en estado puro. Los miembros calculados son tratados como métricas a la hora de explotar los cubos, sin embargo son mucho más potentes que estas.

Un miembro calculado es una métrica que tiene como valor el resultado de la aplicación de una fórmula que puede utilizar todos los elementos disponibles en

un cubo, así como otras funciones de MDX disponibles. Veamos un ejemplo sencillo.

Tenemos dos métricas en nuestro cubo que son: **[Medidas].[importe ventas]** y **[Medidas].[Número de artículos vendidos]**.

Se podría obtener el precio medio de artículo utilizando un miembro calculado con la siguiente formula:

**[Medidas].[precio medio articulo] = [Medidas].[importe ventas] / [Medidas].[Número de artículos vendidos]**

### **3.10 Arquitectura de la aplicación**

A continuación se hará una descripción de la arquitectura y tecnologías usadas para el desarrollo de Litebi.

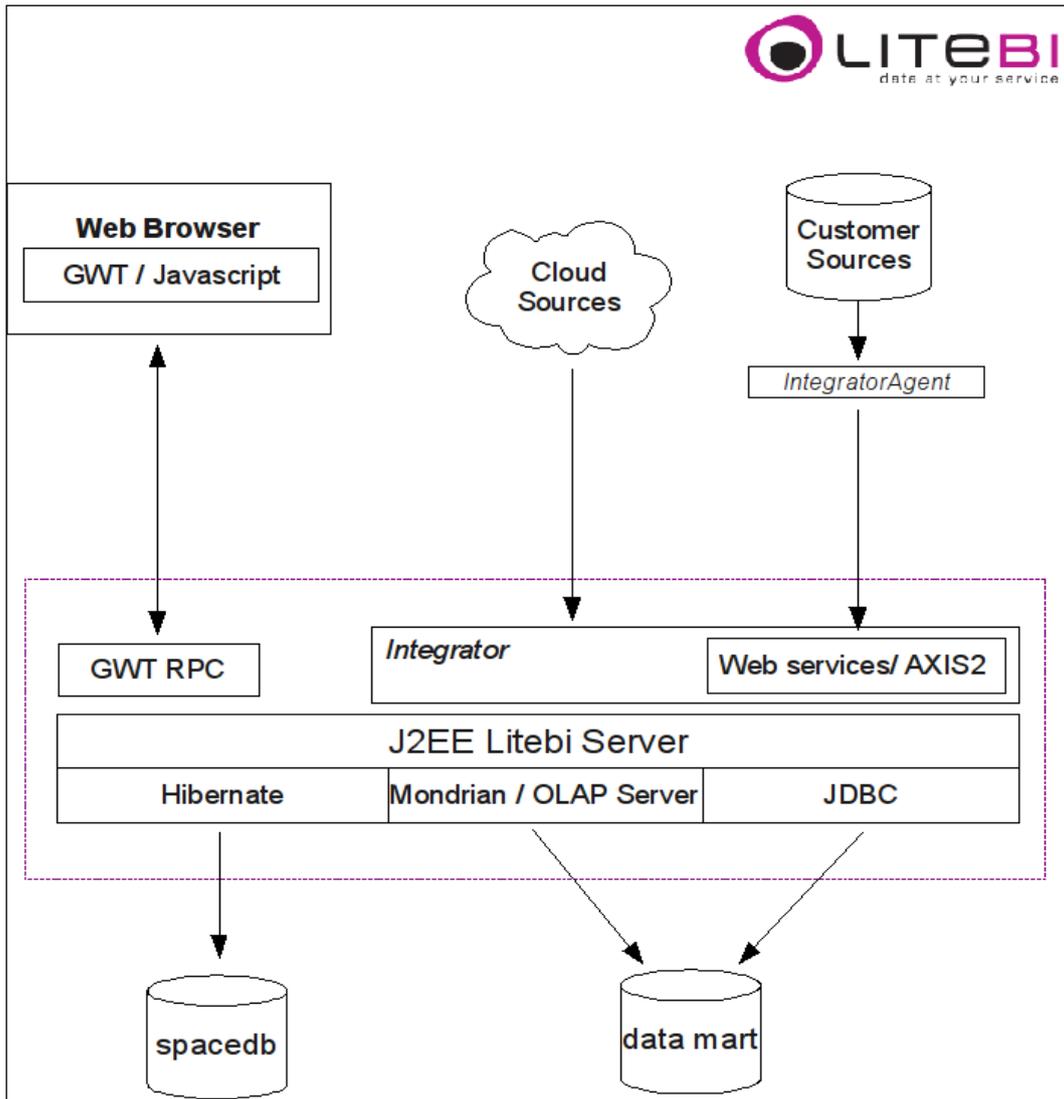


Figura 16: Arquitectura Litebi

### 3.10.1 GWT (Google Web Toolkit)

Para la creación de esta aplicación Web se ha usado el Framework creado por Google, Google Web Toolkit , que básicamente lo que permite hacer es crear el código en Java usando cualquier entorno de desarrollo (IDE) de Java y el compilador lo traducirá a HTML y JavaScript.

GWT [7] permite crear aplicaciones AJAX en el lenguaje de programación Java que son compiladas posteriormente por GWT en código JavaScript ejecutable

optimizado que funciona automáticamente en los principales navegadores. Durante el desarrollo de una aplicación, se puede repetir rápidamente el mismo ciclo "editar - actualizar - ver" típico de JavaScript y aprovechar la ventaja añadida de poder depurar y recorrer una a una todas las líneas de código Java. Cuando se esté listo para la implementación, GWT compilará el código fuente Java en archivos JavaScript optimizados independientes.

Fue seleccionado debido a:

- Uso de **estándares** como javascript en lugar de runtimes propietarios.
- Es **Open Source**
- Ofrece una gran **productividad** al equipo de desarrollo.
- Garantía de **continuidad y calidad** del producto a futuro.

GWT utiliza Java, por lo que se puede utilizar todas las herramientas de desarrollo con Java (Eclipse, IntelliJ, JProfiler, JUnit) al crear las aplicaciones AJAX; por ello para Litebi se utiliza la herramienta de desarrollo integrado **Eclipse**.

### 3.10.2 Protocolo de transferencia

GWT admite un conjunto indefinido de protocolos de transferencia, como JSON (JavaScript Object Notation) y XML, pero el mecanismo de llamada a procedimiento remoto (RPC) de GWT, el utilizado para la transferencia de datos en **Litebi**, permite el establecimiento de comunicaciones Java de una forma especialmente sencilla y eficaz. Al igual que ocurre con el mecanismo de invocación de métodos remotos (RMI) tradicional de Java, tan sólo hay que crear una interfaz que especifique los métodos remotos que se quieran ejecutar. Al realizar una llamada a un método remoto desde el navegador, el mecanismo RPC de GWT serializará automáticamente los argumentos, ejecutará el método adecuado en el servidor y anulará la serialización del valor de retorno del código cliente. El mecanismo RPC de GWT también es bastante

sofisticado: permite gestionar jerarquías de clase polimórfica y ciclos de gráficos de objetos, e incluso generar excepciones durante el proceso.

### 3.10.3 Hibernate

Herramienta de mapeo objeto-relacional para la plataforma Java, que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) u anotaciones que permiten establecer estas relaciones. Un estándar en aplicaciones empresariales. Otorga productividad en el desarrollo e independencia del sistema de gestión de bases de datos.

### 3.10.4 J2EE

Es una plataforma de programación (parte de la plataforma Java), para desarrollar y ejecutar software de aplicaciones en Lenguaje de programación Java con arquitectura de N niveles distribuida, basándose ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.

La tecnología Java es utilizada como espina dorsal de la arquitectura del producto. Algunas de las ventajas que se derivan de su uso son:

- Tecnología estándar en **aplicaciones empresariales**
- Tecnología sólida de **calidad contrastada**.
- Abundancia de librerías **Open Source**
- Integración natural con **GWT**.
- Facilidad para encontrar **profesionales altamente cualificados** con conocimientos de Java / J2EE.

### 3.10.5 Servidor de mapa

Los servidores de mapas permiten la interacción con la información espacial almacenada en servidores de datos espaciales accesibles vía web. El usuario accede a la información de manera que puede visualizarla, consultarla y, en función de las características de los servidores y de los servicios prestados, descargarla o realizar análisis espaciales.

El usuario se conecta a los servicios prestados por estos servidores de mapas a través de cliente tanto ligeros, aplicaciones web que permiten la consulta de estos servidores de mapas desde el navegador, como pesados, aplicaciones SIG de escritorio con módulos que permiten la conexión a servidores de mapas.

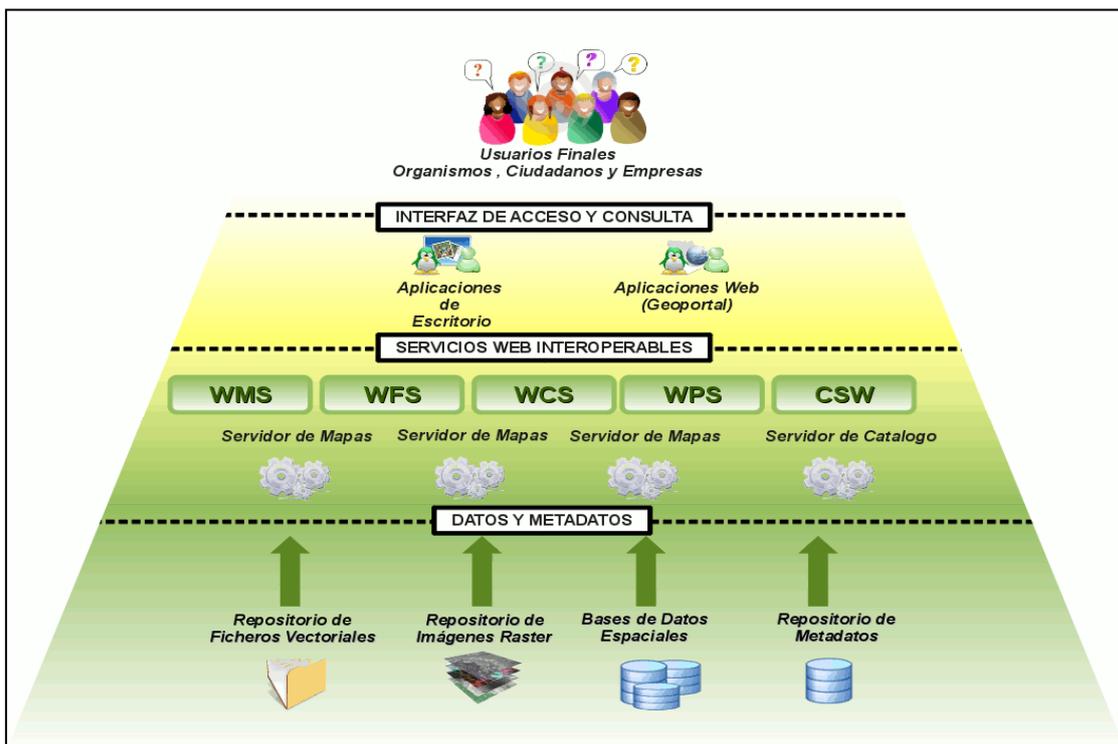


Figura 17: Infraestructura servidores de mapas

Para Litebi se utiliza el servidor GeoServer para el manejo de mapas dinámicos en el Visor Olap de la aplicación.

### **3.10.5.1 GeoServer**

Geoserver [8] es un servidor de código abierto desarrollado en Java, lo que le hace ser multiplataforma, que permite a los usuarios compartir y editar datos geoespaciales. Diseñado para la interoperabilidad, publica datos de cualquier fuente de datos espaciales con estándares abiertos.

Geoserver está desarrollado sobre la base Geotools, una biblioteca de sistemas de información geográfica. Geoserver lee una variedad de formatos de datos, incluyendo PostGIS, Oracle Spatial, ArcSDE, DB2, MySQL, Shapefiles, GeoTIFF, GTOPO30, ECW, MrSID y JPEG2000. A través de protocolos estándares es capaz de generar KML, GML, Shapefile, GeoRSS, PDF, GeoJSON, JPEG, GIF, SVG, PNG y otros. Además, se puede editar datos a través de WFS transaccionales (WFS-T). Geoserver incluye un cliente integrado OpenLayers capaz de visualizar datos para obtener una vista previa.

Geoserver permite la publicación eficiente de datos geoespaciales de Google Earth a través de la utilización de enlaces de red, utilizando KML. Permite utilizar las funciones avanzadas de Google Earth para incluir plantillas de salida, pop-ups, el tiempo, altura de visualizaciones, y "super-overlays".

Geoserver es la implementación de referencia del Open Geospatial Consortium (OGC) para las normas Web Feature Service (WFS) y Web Coverage Service (WCS), además está certificado como servidor de alto rendimiento para Web Map Service (WMS). Geoserver es un componente básico de la Web Geoespacial.

### **3.10.5.2 Abrir y compartir datos espaciales**

GeoServer le permite mostrar la información espacial del mundo. Implementando el estándar de la Web Map Service (WMS), GeoServer puede crear mapas en una variedad de formatos de salida. OpenLayers, una biblioteca de cartografía gratuita, está integrada en GeoServer, lo que hace la generación de mapas rápidos y fáciles. GeoServer se basa en Geotools, un conjunto de herramientas de código abierto de Java SIG.

GeoServer también se ajusta al estándar Web Feature Services (WFS), que permite la participación real y edición de los datos que se utiliza para generar los mapas.

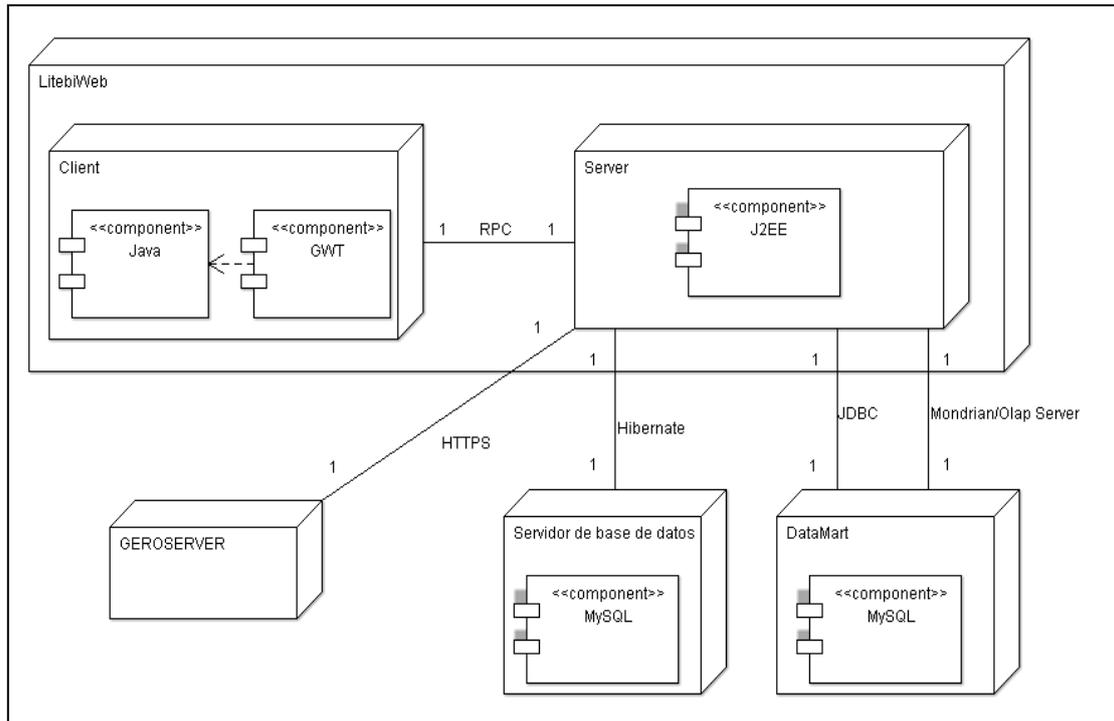
### **3.10.5.3 Uso libre y open software**

GeoServer es software libre. Esto reduce significativamente las barreras económicas en comparación con los productos tradicionales de los SIG.

Además, no sólo está disponible de forma gratuita, también es de código abierto. Las correcciones de errores y mejoras de características en el software de código abierto están más aceleradas en comparación con las soluciones de software tradicional.

### **3.10.5.4 Interacción con Mapping API existentes**

GeoServer puede mostrar los datos en cualquiera de las aplicaciones de mapas populares, tales como Google Maps, Google Earth, Yahoo Maps y Microsoft Virtual Earth. Además, GeoServer puede contactar con las arquitecturas tradicionales de los SIG como ESRI ArcGIS.



**Figura 18: Comunicación Litebi - GeoServer**

### 3.10.6 Mondrian

Mondrian [9], ahora rebautizado como Pentaho Analysis Services, es el motor OLAP integrado en la suite de Business Intelligence Open Source Pentaho.

Mondrian es un proyecto Open Source, licenciado bajo la Mozilla Public License (MPL). Esta licencia es una de las “Business Friendly” lo cual implica que es de las menos restrictivas para su uso desde la mayor parte de los puntos de vista (al igual que la resto de la suite de Pentaho), permitiendo Modificar, Embeber, Modularizar, el software sin restricciones; dejando al parecer de la organización el aporte o no de los cambios realizados al proyecto.

El esquema de Mondrian es el utilizado para la exploración de cubos Olap en LiteExplorer

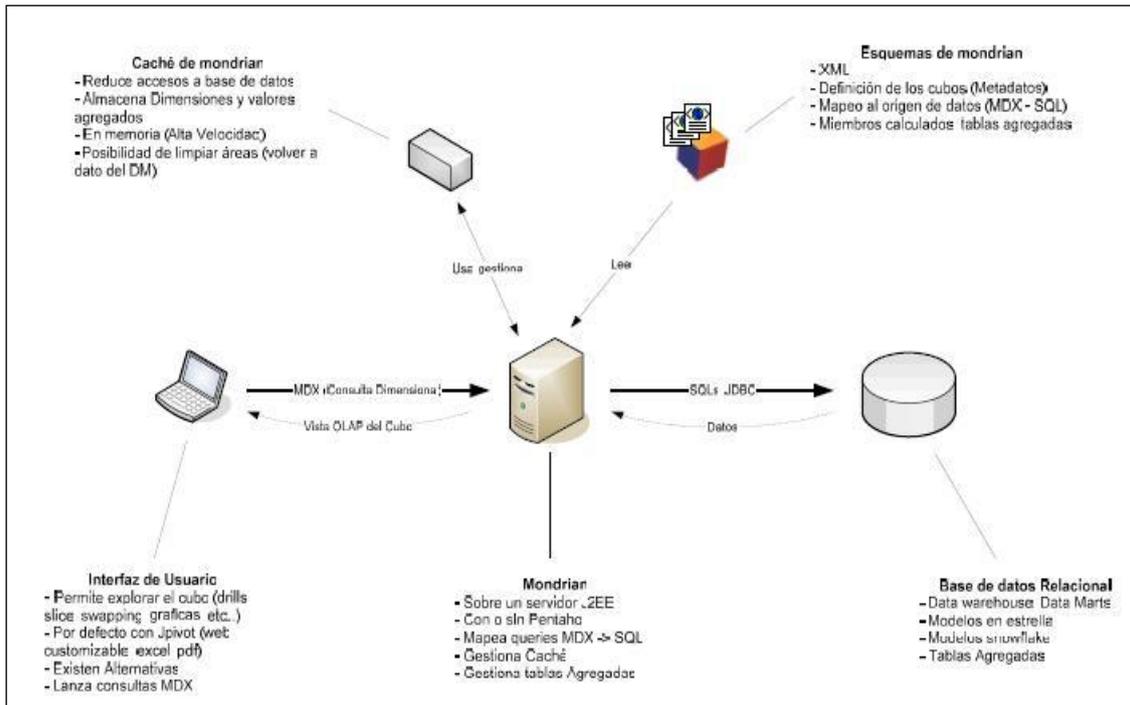


Figura 19: Funcionamiento de Mondrian [10]

Mondrian se encarga de recibir consultas dimensionales (lenguaje MDX) y devolver los datos de un cubo, sólo que este cubo no es algo físico sino un conjunto de metadatos que definen como se han de “mapear” estas consultas que tratan conceptos dimensionales a sentencias SQL ya tratando con conceptos relacionales que obtengan de la base de datos la información necesario para satisfacer la consulta dimensional.

Es utilizado como gestor de metadatos dimensionales para LiteSpace.

- **Potente**, rico en características.
- **Utilizado ampliamente** por organizaciones y también, de forma embebida, en productos comerciales, incluso otras alternativas SaaS BI como Lucidera<sup>5</sup>.
- Producto **sólido** y altamente testado (7 años de historia)
- **Independiente** del SGDB

<sup>5</sup> <http://www.lucidera.com/>

- **Adaptable** a las necesidades concretas de Litebi (Open Source)
- Basado en **estándares** del sector (MDX, XMLA, etc.), posibilidades de interoperación con otros sistemas.

```

<Schema name='NoName'>
  <Dimension type='StandardDimension' name='7' caption='Product'>
    <Hierarchy name='11' hasAll='true' allMemberName='Total' primaryKey='BASE_LEVEL_ID' caption='Product'>
      <Table name='DIMENSION_7'>
      </Table>
      <Level name='10' column='LEVEL_10_ID' nameColumn='LEVEL_10_DESC' ordinalColumn='LEVEL_10_DESC' type='String' uniqueMembers='true' levelType='Regular'>
      </Level>
      <Level name='9' column='LEVEL_9_ID' ordinalColumn='LEVEL_9_ID' type='String' uniqueMembers='false' levelType='Regular'>
      </Level>
      <Level name='8' column='BASE_LEVEL_ID' nameColumn='BASE_LEVEL_DESC' ordinalColumn='BASE_LEVEL_DESC' type='String' uniqueMembers='true' levelType='Regular'>
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type='StandardDimension' name='1' caption='Customer'>
    <Hierarchy name='6' hasAll='false' primaryKey='BASE_LEVEL_ID' caption='Customer'>
      <Table name='DIMENSION_1'>
      </Table>
      <Level name='3' column='LEVEL_3_ID' ordinalColumn='LEVEL_3_ID' type='String' uniqueMembers='false' levelType='Regular'>
      </Level>
      <Level name='4' column='LEVEL_4_ID' ordinalColumn='LEVEL_4_ID' type='String' uniqueMembers='false' levelType='Regular'>
      </Level>
      <Level name='5' column='LEVEL_5_ID' ordinalColumn='LEVEL_5_ID' type='String' uniqueMembers='false' levelType='Regular'>
      </Level>
      <Level name='2' column='BASE_LEVEL_ID' nameColumn='BASE_LEVEL_DESC' ordinalColumn='BASE_LEVEL_DESC' type='String' uniqueMembers='true' levelType='Regular'>
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type='StandardDimension' name='12' caption='Tiempo'>
    <Hierarchy name='19' hasAll='true' allMemberName='Total' primaryKey='BASE_LEVEL_ID' caption='Time1'>
      <Table name='DIMENSION_12'>
      </Table>
      <Level name='17' column='LEVEL_17_ID' ordinalColumn='LEVEL_17_ID' type='String' uniqueMembers='false' levelType='Regular'>
      </Level>
      <Level name='13' column='BASE_LEVEL_ID' nameColumn='BASE_LEVEL_DESC' ordinalColumn='BASE_LEVEL_DESC' type='String' uniqueMembers='true' levelType='Regular'>
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type='Cube' name='123' caption='Sales' cache='false' enabled='true'>
    <Table name='CUBE_123'>
    </Table>
    <DimensionUsage source='7' name='7_1' caption='Product' foreignKey='DIMENSION_7_1_ID'>
    </DimensionUsage>
    <DimensionUsage source='1' name='1_2' caption='Customer' foreignKey='DIMENSION_1_2_ID'>
    </DimensionUsage>
    <DimensionUsage source='12' name='12_3' caption='Tiempo' foreignKey='DIMENSION_12_3_ID'>
    </DimensionUsage>
    <Measure name='126_Sum' column='MEASURE_124' aggregator='sum' caption='Ventas (Sum)' visible='true'>
    </Measure>
  </Dimension>
  <Dimension type='Cube' name='136' caption='Nuevo Cubo' cache='false' enabled='true'>
    <Table name='CUBE_136'>
    </Table>
    <DimensionUsage source='7' name='7_4' caption='Product' foreignKey='DIMENSION_7_4_ID'>
    </DimensionUsage>
    <Measure name='137_Sum' column='MEASURE_137' aggregator='sum' caption='Nueva métrica (Sum)' visible='true'>
    </Measure>
  </Dimension>

```

Figura 20: Ejemplo fichero XML con el esquema de Mondrian en Litebi

### 3.10.7 Kettle

Kettle [11] es un proyecto Open Source que incluye un conjunto de herramientas para realizar ETL. Uno de sus objetivos es que el proyecto ETL sea fácil de generar, mantener y desplegar.

Se compone de 4 herramientas:

- **SPOON:** permite diseñar de forma gráfica la transformación ETL
- **PAN:** ejecuta la transformaciones diseñadas con SPOON

- **CHEF:** permite, mediante una interfaz gráfica, diseñar la carga de datos incluyendo un control de estado de los trabajos.
- **KITCHEN:** permite ejecutar los trabajos batch diseñados con Chef.

Es utilizado como base para la versión de LiteIntegrator (cliente/servidor).

- Producto **sólido** con un largo recorrido
- Multiplataforma (java)
- **Ampliamente conocido** en el sector del Business Intelligence.
- **Adaptable** a las necesidades de Litebi (Open Source)

### 3.10.8 MySQL

MySQL es un gestor de base de datos sencillo de usar y increíblemente rápido. También es uno de los motores de base de datos más usados en Internet. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Este reconocido sistema de gestión de base de datos es utilizado en nuestra aplicación; ya que **Litebi** es una aplicación Web, por lo tanto hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

### 3.10.9 Web Service axis2

Para la arquitectura de servicios web (SOA) se utiliza Axis2, que es la versión mejorada del contenedor de Web Services AXIS. El proyecto ha evolucionado independientemente de la primera versión debido a que implementa

especificaciones diferentes. AXIS en su primera versión implementaba JAX-RPC y proveía un contenedor relativamente versátil para su época de lanzamiento y sobre todo de propagación (entre 2004 y 2007).

AXIS originalmente se convirtió en uno de los contenedores más extendidos para implantar soluciones basadas en Web Services o que proveían acceso a sistemas preexistentes mediante Web Services SOAP.

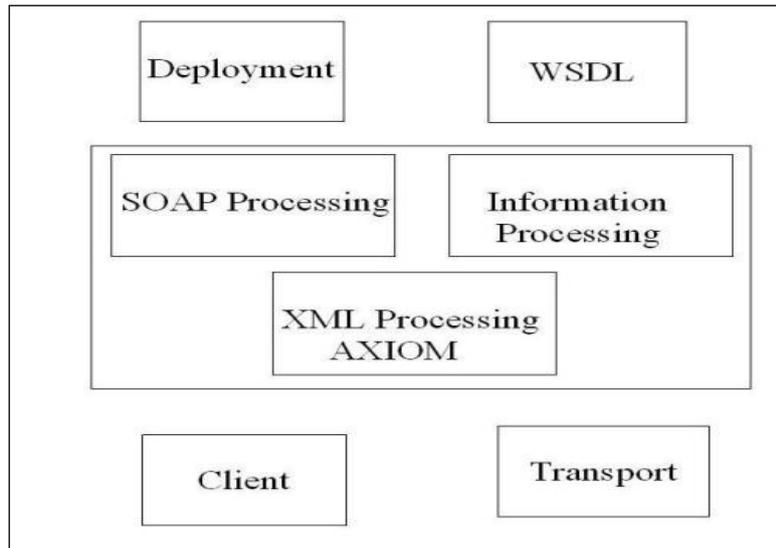
Axis2 soporta SOAP 1.1, SOAP 1.2 y REST. Axis2 es más eficiente, más modular y más orientado a XML que Axis1. Ha sido diseñado para la fácil agregación de plug-in “módulos”, que extienden su funcionalidad para la seguridad y fiabilidad.

Los criterios para su utilización en el proyecto son:

- **Ampliamente usado** en software empresarial.
- Open source
- Basado en estándares (WSDL, SOAP, etc.)

### **3.10.9.1 Arquitectura Axis2**

La arquitectura de Axis2 se ha desarrollado manteniendo la uniformidad de los servicios. Primero separa la lógica y los estados; el código que procesa la lógica es usualmente apátrida. Esto permite que el código sea ejecutado libremente en hilos paralelos. Segundo, que toda la información es mantenida en un modelo de información; esto permite al sistema ser almacenado y reanudado. Y finalmente, la arquitectura es modular; la arquitectura divide a Axis2 en los siguientes siete módulos:



**Figura 21: Componentes de la arquitectura de Axis2 [12]**

- 1) Information Model
- 2) XML processing Model
- 3) SOAP Processing Model
- 4) Deployment
- 5) WSDL and Code Generation
- 6) Client API
- 7) Transports

### **3.11 Aspectos técnicos de litebi**

El patrón utilizado para la implementación de Litebi es el modelo vista controlado, el cual es utilizado para separar los datos, la interfaz de usuario, y la lógica de control en 3 componentes distintos.

Como se ha mencionado anteriormente, Litebi está siendo desarrollada utilizando **Eclipse**, en esta sección se explicará como fue organizada esta aplicación en dicha herramienta.

Se decidió separar Litebi en 6 grandes proyectos o paquetes de desarrollo (ver *figura 22*) que a continuación se enumeran:

- 1) **LitebiTypesOlap:** Contiene todas las clases de datos necesarias para el funcionamiento del Visor Olap (LiteExplorer).
- 2) **LitebiTypesSolution:** Contiene todas las clases de datos necesarias para el funcionamiento de LiteSpace.
- 3) **LitebiWidgets:** Contiene todas las clases referente a la interfaz de usuario, clases que no depende de los demás proyectos.
- 4) **LitebiWeb:** Contiene, básicamente, la lógica de producto y los controladores responsable de recibir los eventos de entrada, invocar cambios en el modelo y en la vista. Este proyecto conoce o depende de los 3 proyectos anteriores.
- 5) **LitebiWebService:** En donde están todas las clases y librerías que necesitamos para generar el servicio Web con Axis2.
- 6) **LitebiKettlePlugin:** Contiene todas las clases necesarias para la creación de un plugin para Kettle que sirve para conectar con Litebi.

Lo que se busca en todo esto, es la reutilización de código y de esa manera hacer más fácil el mantener el código en un futuro, entre otras cosas.

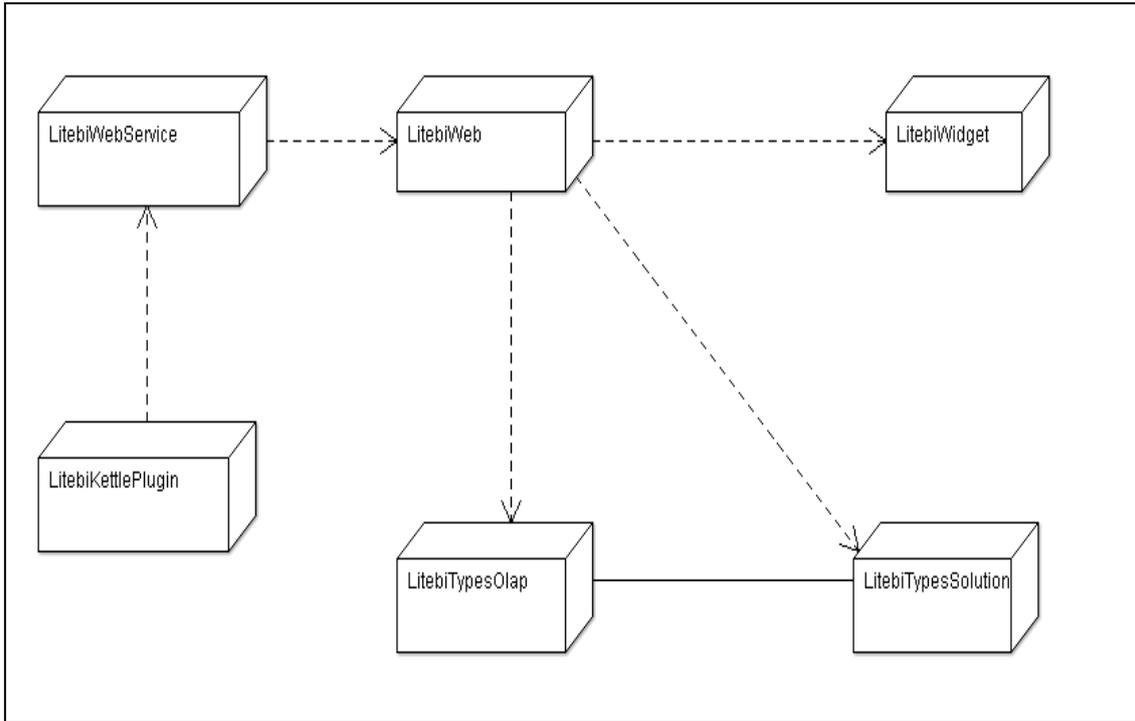
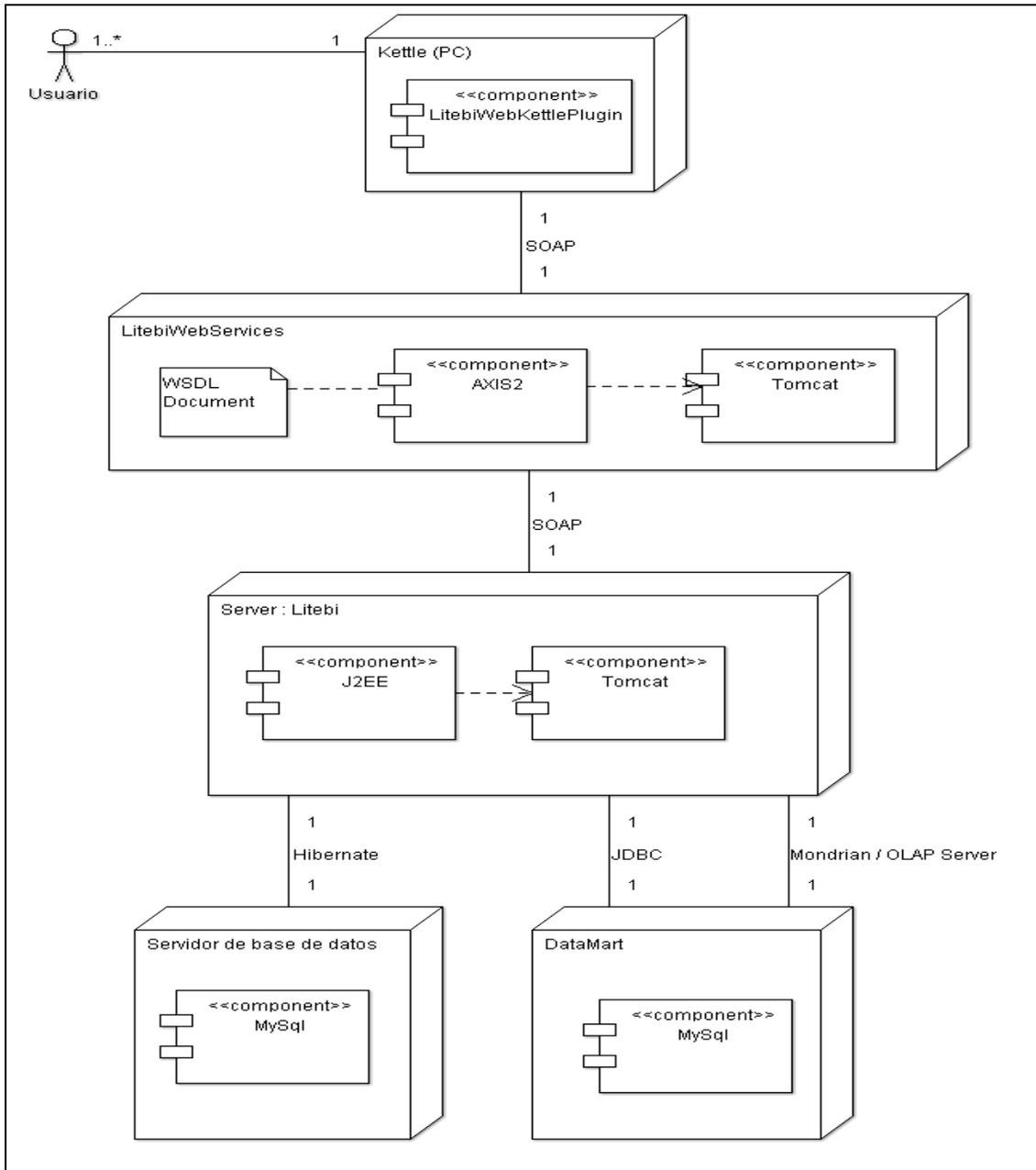


Figura 22: Diagrama de paquetes en Litebi

### 3.11.1 Implementación de Axis2 en Litebi

Para el proceso de carga de datos desde el exterior a **Litebi** utilizamos AXIS2. En un principio tuvimos problemas a la hora de construir el servicio Web por la poca documentación y ejemplos que había en el momento, pero al final hemos logrado implementar los servicios de manera óptima, en la *figura 23* se muestra el modelo de comunicación entre los servicios de Litebi y el plugin de Litebi para Kettle.



**Figura 23: Modelo comunicación con Axis2**

Para empezar el proceso de implementación de Axis2 se descarga la distribución binaria estándar<sup>6</sup>, específicamente en **AXIS2\_HOME/samples/** dentro de los directorios: quickstart, quickstartadb, quickstartaxiom, quickstartjibx y quickstartxmlbeans. La misma incluye los buildfiles (archivos build.xml) para Ant que sirve para hacer una compilación más fácil.

<sup>6</sup> [http://ws.apache.org/axis2/download/1\\_3/download.html#std-bin](http://ws.apache.org/axis2/download/1_3/download.html#std-bin)

A continuación hago una guía rápida para la implementación de un servicio Web con Axis2 utilizando JIBX.

Empiezo explicando el contenido del archivo **build.xml**:

### 1. **build-shared:**

Compila con Jibx, con el comando bind, las clases que le pongamos en binding.xml

- ***binding.xml***: Hace referencia a wsdl-binding.xml y a types-binding.xml.
- ***wsdl-binding.xml***: Excepciones que queremos serializar.
- ***types-binding.xml***: Las clases que queremos serializar.

El bind de build.xml se encarga de generar clases extras para permitir serializar las clases que le configuremos en building.xml, building solo necesita las rutas de todas las clases que estén implicadas, estas clases la necesita el cliente (nuestra aplicación java) y el servidor (tomcat). Las clases generadas son del tipo xxx\_acces.class.

### 2. **build-client:**

- ***generate-client***: Genera las clases del cliente.
- ***compile-client***: Compila las clases generadas en el cliente.

### 3. **build-server:**

- ***generate\_server***: Genera las clases del servidor
- ***compile-server***: Compila las clases generadas del servidor

La diferencia entre generate-server con generate-client, es que generate-server tiene un argumento **-ss** que le dice que genere clases para la parte servidor.

Las clases generadas sirven para enviar y recibir los objetos que se serializan en Jibx. La parte cliente consta básicamente de `WebServiceClient.java` (nuestra aplicación) que utiliza clases generadas por axis2 como `JibxUnwrapLibraryStub`, etc. y nuestras clases, es decir, `Measure`, `Dimension`, etc...

- **packaged:** Genera el archivo `.aar` (que en realidad es una jar con la extensión aar), que es lo que necesita axis2 para definir un servicio web, dentro del `.aar`, aparte de clases, hay un archivo `services.xml`, en la carpeta `META-INF`, con la definición del servicio Web.

#### 4. deploy:

Copia el archivo `.aar` junto con todos los archivos `.xml`, `.class`, `.jar`, etc., que se necesitan para que el servicio web funcione, en las carpetas respectivas de `Axis2/WEB-INFO`, que se encuentra dentro de la carpeta de `tomcat/webapps`.

Con todo lo anterior tenemos lo necesario para ejecutar sin problemas el Servicio Web.

El archivo **binding.xml** hace referencia a `wsl_binding.xml` y a `types-binding.xml`.

En **ws\_binding.xml** agregamos las excepciones que queremos serializar.

En **types\_binding.xml** agregamos las clases que queremos serializar, para nuestra aplicación queremos serializar las clases: `Measure`, `SelfDimension`, `SharedDimension`, `Cube`, `Hierarchy`, `Level`, `File` y `Folder`.

Si queremos agregar una clase con solo atributos, se hace de la siguiente forma:

Agregando la clase File, por ejemplo:

```
<mapping abstract = "true"  
class="com.litebi.web.client.shared.solution.File"  
type-name = "tns:File">  
  <value name="id" style="attribute" field="id"/>  
  <value name="name" style="attribute" field="name"/>  
  <value name="description" style="attribute" field="description"/>  
  <value name="owner" style="attribute" field="owner"/>  
  <value name="date" style="attribute" field="date"/>  
</mapping>
```

Para referenciar cada clase agregada, se utiliza el atributo type-name de la etiqueta mapping, como se puede ver en el ejemplo anterior.

Si queremos agregar una clase que además tenga colecciones de objetos, se puede hacer de la siguiente manera:

Agregando la clase Cube, por ejemplo:

```
<mapping abstract = "true"  
class="com.litebi.web.client.shared.dictionary.Cube"  
type-name="tns:cube">  
  <value name="id" style="attribute" field="id"/>  
  <value name="name" style="attribute" field="name"/>  
  <value name="description" style="attribute" field="description"/>  
  <collection field="measures"  
    item-type="com.litebi.web.client.shared.dictionary.Measure">  
    <structure name="measure" map-as="tns:Measure"/>  
  </collection>  
  <collection field="selfDimensions"
```

```

        tem-type="com.litebi.web.client.shared.dictionary.SelfDimension">
        <structure name="selfDimension" map-as="tns:SelfDimension"/>
</collection>
<collection field="sharedDimensions"
    item-
type="com.litebi.web.client.shared.dictionary.SharedDimension">
    <structure name="sharedDimension"
        map-as="tns:SharedDimension"/>
</collection
</mapping>

```

Nótese que para la clase Cube, existen 3 colecciones de objetos (Measure, SelfDimension, SharedDimension), los cuales previamente ya han sido agregados a types-binding.xml. Al definir las colecciones como lo muestra ejemplo, se permite que todas las clases sean abstractas.

Si queremos agregar una clase que tenga como atributo algún objeto creado por nosotros, lo hacemos de la siguiente manera:

Agregamos la clase SelfDimension, por ejemplo:

```

<mapping abstract = "true"
class="com.litebi.web.client.shared.dictionary.SelfDimension"
type-name = "tns:SelfDimension">
<value name="id" style="attribute" field="id"/>
<value name="name" style="attribute" field="name"/>
<value name="description" style="attribute" field="description"/>
<structure name="hierarchy" field="hierarchy" map-as="tns:Hierarchy"
usage="optional"/>
</mapping>

```

En este ejemplo, además de los atributos simples de la clase, hemos agregado un objeto (Hierarchy), que también se ha agregado a types-binding.xml previamente.

Por ejemplo: si queremos crear un método para recibir un objeto Cube, escribimos lo siguiente:

Primero especificamos el nombre del método y cual es el nombre y el tipo del parámetro de entrada, es posible que no haya parámetros de entrada, en ese caso dejamos la etiqueta *sequence* vacía.

```
<element name="getCube">
  <complexType>
    <sequence>
      <element name="id" type="int"/>
    </sequence>
  </complexType>
</element>
```

Luego se especifica que tipo de respuesta se recibirá, en este caso se especifica recibir un objeto Cube, lo cual especificamos en el atributo *type*.

```
<element name="getCubeResponse">
  <complexType>
    <sequence>
      <element name="getCubeReturn" minOccurs="0"
        type="tns:cube"/>
    </sequence>
  </complexType>
</element>
```

Mas adelante agregamos las siguientes instrucciones, para que Axis2 sepa cuáles son los métodos que hemos implementado y que hacer con ellos.

```
<wsdl:message name="getCubeRequest">
<wsdl:part element="wns:getCube" name="parameters"/>
</wsdl:message>
<wsdl:message name="getCubeResponse">
<wsdl:part element="wns:getCubeResponse" name="parameters"/>
</wsdl:message>
<wsdl:operation name="getCube">
<wsdl:input message="wns:getCubeRequest" name="getCubeRequest"/>
<wsdl:output message="wns:getCubeResponse"
name="getCubeResponse"/>
</wsdl:operation>
<wsdl:operation name="getCube">
<wsdlsoap:operation soapAction="urn:getCube"/>
<wsdl:input name="getCubeRequest">
<wsdlsoap:body use="literal"/>
</wsdl:input>
<wsdl:output name="getCubeResponse">
<wsdlsoap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
```

En **JibxUnwrapLibraryImpl.java** se sobrescriben los métodos del servicio Web, definidos en library.wsdl.

En **WebServiceClient.java** Se llaman los métodos especificados en JibxUnwrapLibraryImpl.java, hay dos maneras de llamar a los métodos, de forma síncrona y asíncrona. Para nuestro desarrollo se ha utilizado la forma asíncrona.

Para ilustrar un poco, hagamos un ejemplo de una llamada asíncrona para el método getCubes().

- **Primer paso:**

Creamos la interfaz estática CubesCallback que extiende de JibxUnwrapLibraryCallbackHandler, con dos métodos, que se encarga de recibir la respuesta si todo sale bien y otro método que recibe la excepción si hay algún tipo de error:

```
public static class CubesCallback extends  
JibxUnwrapLibraryCallbackHandler  
{  
    private boolean m_done;  
    private Exception m_exception;  
    private Folder cubesFolder;  
    public synchronized void receiveResultgetCubes(Folder resp) {  
        cubesFolder = resp;  
        m_done = true;  
    }  
    public synchronized void receiveErrorgetCubes(Exception e) {  
        m_exception = e;  
        m_done = true;  
    }  
}
```

- **Segundo paso:**

Llamar al método, para esto escribimos lo siguiente:

```
CubesCallback cbs = new CubesCallback();  
stub.startgetCubes(cbs);  
long start = System.currentTimeMillis();  
synchronized (cbs) {  
    while (!cbs.m_one) {
```

```

        try {
            cbs.wait(100L);
        } catch (Exception e) {}
    }
}
System.out.println("Asynchronous operation took " +
(System.currentTimeMillis()-start3) + " millis");
if (cbs.cubesFolder != null) {
    Folder cubesFolder = cbs.cubesFolder;
} else {
    cbs.m_exception.printStackTrace(System.out);
}

```

### 3.11.1.1 Modulo de seguridad para Axis2

Incluimos Rampart [13] que es un módulo basado en Apache WSS4J [14], que a su vez implementa OASIS Web Services Security, estándar para la seguridad de servicios Web mediante SOAP.

Se utilizó para la validación de usuario entre el plugin de Litebi para Kettle y el servicio Web, a continuación se muestra la clase en el cliente, donde se especifica el sistema de seguridad:

```

package com.sosnoski.ws.library.jibx;
import org.apache.ws.security.WSPasswordCallback;
import javax.security.auth.callback.Callback;
import javax.security.auth.callback.CallbackHandler;
import javax.security.auth.callback.UnsupportedCallbackException;
import java.io.IOException;

public class ClientPWCBHandler implements CallbackHandler {

```

```

public void handle(Callback[] callbacks) throws IOException,
UnsupportedCallbackException {
    System.out.println("client handle callback!");
    for (int i = 0; i < callbacks.length; i++) {
        WSPasswordCallback pwcb = (WSPasswordCallback) callbacks[i];
        String id = pwcb.getIdentifer();
        System.out.println("client id :" + id);
        if("bob".equals(id)) {
            pwcb.setPassword("bobPW");
        }
    }
}
}
}
}
}

```

Axis2 llamará al método **handle** cuando necesite obtener la clave del usuario. Se puede implementar de cualquier manera, el resultado debe ser que sobre el objeto **WSPasswordCallback** se le asigne la clave que le corresponda al usuario (con la línea `pwcb.setPassword("bobPW ");`).

### 3.11.1.1.1 Seguridad y generación de certificados

Es muy importante la seguridad en la carga de datos entre el plugin de Litebi para Kettle y Litebi, para ello se utiliza el protocolo de capa de conexión segura- (SSL).

Tanto si se emplea un certificado autofirmado como si se crea una autoridad de certificación, los cambios en el fichero **server.xml** de Tomcat son los mismos.

Hay que buscar el siguiente elemento Connector que aparece comentado:

```

<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"

```

```
port="8443" minProcessors="5" maxProcessors="75"
enableLookups="true"
acceptCount="10" debug="0" scheme="https" secure="true"
useURIVValidationHack="false">
<Factory
  className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"
  clientAuth="false" protocol="TLS" />
</Connector>
```

SSL proporciona autenticación y privacidad de la información entre extremos sobre Internet mediante el uso de criptografía.

Para activar el uso de SSL, hay que dejar sin comentarios el elemento Connector. Los aspectos más destacables de la configuración que se activa son:

- Se habilita el acceso al servidor de forma segura a través del puerto 8443. Por tanto, las URL que se tienen que emplear son de la forma: o `https://servidor:8443/aplicacionweb/recurso`.
- No se solicita certificado al cliente durante el establecimiento de la conexión SSL. Por tanto, no es necesario cambiar nada en el navegador Web que usemos.

Por medio de una tarea **Ant** generamos el certificado un par de claves clave-pública/clave-privada, hay que señalar que la clave privada sirve para cifrar los mensajes de Litebi, mientras que la clave pública solo sirve para descifrar; luego repartimos la claves, la clave privada solo la tiene el servicio Web y la clave pública la tendrá el plugin de Litebi para Kettle. Como Kettle es una aplicación de escritorio, no es seguro que tenga la clave privada.

SSL implica una serie de fases básicas:

- Negociar entre las partes el algoritmo que se usará en la comunicación

- Intercambio de claves públicas y autenticación basada en certificados digitales
- Cifrado del tráfico basado en cifrado simétrico

En el primer contacto entre el Cliente (plugin de Litebi para Kettle) y servidor (LitebiWebService) tienen un intercambio secreto de claves por medio un sistema de cifrado asimétrico (protocolo Diffie-Hellman [15]), la demás comunicación se hace de forma simétrica, por cuestiones de velocidad.

A continuación se muestra la tarea **Ant** para generar el certificado.

- 1) Esta tarea genera el par de claves, lo mete en el keystore (**tomcat\_keystore**) con el password del keystore(**tomcat\_keystore\_pass**), el password para poder utilizar el par de claves es **keypass**.

```
<exec dir="." executable="keytool" failonerror="true">
  <arg value="-v"/>
  <arg value="-genkey"/>
  <arg value="-validity"/>
  <arg value="3650"/>
  <arg value="-keyAlg"/>
  <arg value="RSA"/>
  <arg value="-alias"/>
  <arg value="{tomcat_alias}"/>
  <arg value="-keystore"/>
  <arg value="{tomcat_keystore}" />
  <arg value="-keypass"/>
  <arg value="{keypass}"/>
  <arg value="-dname"/>
  <arg value="{dname}" />
  <arg value="-storepass"/>
  <arg value="{tomcat_keystore_pass}"/>
</exec>
```

`</exec>`

- 2) Con esta tarea se separa la clave publica en un archivo separado .cert

```
<exec dir="." executable="keytool" failonerror="true">
  <arg value="-v"/>
  <arg value="-exportcert"/>
  <arg value="-alias"/>
  <arg value="{tomcat_alias}"/>
  <arg value="-keystore"/>
  <arg value="{tomcat_keystore}" />
  <arg value="-keypass"/>
  <arg value="{keypass}"/>
  <arg value="-storepass"/>
  <arg value="{tomcat_keystore_pass}"/>
  <arg value="-file"/>
  <arg value="{plugin_cert_file}"/>
</exec>
```

- 3) con esta tarea se crea un nuevo keystore (para uso con el plugin) en el que importamos la clave pública que hemos separado antes. El password para el keystore es **plugin\_keystore\_pass** y el password para acceder a la clave pública es **keypass**; estos passwords no son criticos, porque en el keystore que generamos solo existe la clave pública de litebi.

```
<exec dir="." executable="keytool" failonerror="true">
  <arg value="-v"/>
  <arg value="-importcert"/>
  <arg value="-noprompt"/>
  <arg value="-alias"/>
  <arg value="{plugin_alias}"/>
  <arg value="-keystore"/>
```

```
<arg value="${plugin_keystore}" />  
<arg value="-keypass"/>  
<arg value="${keypass}"/>  
<arg value="-storepass"/>  
<arg value="${plugin_keystore_pass}"/>  
<arg value="-file"/>  
<arg value="${plugin_cert_file}"/>  
</exec>
```

## 4 CONCLUSIONES Y TRABAJOS FUTUROS

Litebi ofrece una aproximación diferente al Business Intelligence. Se pretende eliminar el coste y la complejidad asociada a este tipo de soluciones mediante el modelo de Software como Servicio (SaaS), es decir accesible a través de Internet y con pago por uso. En resumen se elimina la necesidad de implantar y mantener la solución de BI en el cliente, con el ahorro que supone en inversión inicial en hardware y software, y en mantenimiento. Además Litebi va más allá y simplifica enormemente el proceso de creación de las soluciones analíticas.

Litebi es el primer producto con vocación integral y autónoma de Business Intelligence desarrollado en España del que se tenga noticia. El producto no es Open Source, y ha sido construido desde cero con el modelo SaaS en mente. Ahora bien, cómo todo software desarrollado en la actualidad, nos hemos apoyado en algunos productos de Software Libre (bajo licencias adecuadas, MPL, LGPL, etc.) en especial para la herramienta de ETL, aunque hay más componentes, por ejemplo AXIS2 para la capa de servicios Web o algunas otras librerías Java. Sin profundizar más, el producto está desarrollado sobre el framework GWT, que también es Open Source y una excelente plataforma que nos ha facilitado muchísimo el desarrollo.

En este informe se ha mostrado las diferentes tecnologías y arquitectura utilizada para el desarrollo de los diferentes módulos de Litebi. Hasta el momento el trabajo realizado es realmente sorprendente y bastante rápido; pero aún faltan muchas funcionalidades por desarrollar y que permitan que el producto sea más novedoso y competitivo. Entre estos módulos futuros encontramos.

## 1) liteEureka:

A partir de la información contenida en liteSpace, liteEureka aplica **técnicas de minería de datos e inteligencia artificial** para localizar **patrones de información** que puedan ser útiles al usuario.

El objetivo es el **descubrimiento automático** de información relevante entre los datos estructurados residentes en liteSpace utilizando las estructuras dimensionales.

liteEureka funciona como un framework para crear **sistemas expertos** que se adapten a las necesidades del usuario (bien entrenándolo, bien definiendo reglas). Será posible crear sistemas expertos verticalizados adaptados a las necesidades de distintos ámbitos (*ej. sistema experto de control de calidad en una línea de fabricación*).

El concepto clave en liteEureka es el de “**descubrimiento**” el sistema funciona como un sugeridor/descubridor capaz de adaptarse a las necesidades del cliente.

Se plantea el **desarrollo propio desde cero de liteEureka** apoyándose en librerías y técnicas de IA y Minería de Datos que permita al sistema trasladar al usuario descubrimientos como:

- **Clasificación:** Dadas unas variables determinadas (para un cliente: sexo, edad y región), genera y propone una serie de “arquetipos” para comprender la información (ej. Un 12% son mujeres entre 25-35 años de Madrid, Un 9% hombres de 40-50 años de Barcelona, etc.)
- **Información potencialmente relevante.** En base a las reglas definidas en el sistema experto mostrar información similar a la información que anteriormente se consideró valiosa.

- **Datos anómalos** (limpieza de datos, detección de casos exóticos ). *Ej. El cliente A efectuó una compra 20 veces superior a su media en determinado día, etc.*

## 2) liteTeam:

Como parte del enfoque de Software como Servicio, Litebi incorpora funcionalidades que optimizan el uso de la aplicación por parte de equipos de trabajo.

liteTeam aporta:

- **Mail Integrado:** El sistema es capaz de enviar correos electrónicos con **informes adjuntos**. Se puede programar el envío de estos correos o bien enviarlos en el momento del análisis.
- **Alarmas:** Es posible definir Alarmas de forma que, ante determinados eventos (*ej. Un valor supera el umbral indicado*) desencadenen un suceso (*ej. envío de un correo o aviso a un usuario*). Permite controlar la información de forma pasiva.
- **Colaboración:** Es posible agregar comentarios a los elementos de análisis, o llamar la atención de otro usuario sobre datos determinados. (*ej. proceso de revisión de objetivos de ventas en el que el responsable va marcando elementos de datos y explicando los motivos detrás del valor*).

## 3) liteConcepts

Herramienta para el análisis de los datos no estructurados (textuales) almacenados en liteSpace o existentes en Internet.

LiteConcepts supone la creación un lenguaje de representación de ontologías relacionadas con el modelo dimensional almacenado en liteSpace. Planteamos

un desarrollo propio apoyándose en librerías y técnicas de minería web y tecnologías semánticas (ver anexo liteConcepts) que permitirá:

- **Relacionar información no estructurada con información estructurada.** Por ejemplo mientras se realiza un análisis OLAP ser capaz de ofrecer un contexto relacionado de Noticias, documentos o webs relacionadas.
- **Vigilar** un determinado **sector** de Internet. El **sector** se define como un conjunto de **conceptos** construidos usando técnicas de **minería web** a partir de un input suministrado por el usuario (keywords, webs, etc.). (*Ej. Vigilancia competitiva del sector del calzado, posibilitando recibir noticias relacionadas, descubrimiento de información relevante, monitorización de webs, etc.*)
- Usar para el proceso de toma de decisiones información proveniente de diferentes **orígenes ya existentes en internet** (Google Trends, Noticias, Google Sets, Digg, Freebase, Technorati, etc.) (*Ej. Uso de noticias de internet, datos internos de ventas, y tendencias detectadas en Google Trends para presentar un Cuadro de Mando sobre tendencias de marketing en un determinado sector*)

Para finalizar, pienso que la gran mayoría de la información, privada y pública, residirá en Internet. Las personas y las empresas establecerán relaciones sobre plataformas en Internet, la red será ubicua y plana, sin dejar por ello de ser segura o de tener áreas privadas. Con este proyecto se quiere aportar un granito de arena a esa tendencia que creo natural y sana.

## 5 BIBLIOGRAFÍA

[1]. **Lite Internet Solutions S.L.** Litebi. *Litebi*. [En línea] 2008. [Citado el: 8 de abril de 2009.] [www.litebi.com](http://www.litebi.com).

[2]. **Todo BI <business intelligence>**. 9 cosas que debes saber sobre saas. [En línea] 27 de Diciembre de 2007. [Citado el: 10 de abril de 2009.] <http://todobi.blogspot.com/2007/12/9-cosas-que-debes-saber-sobre-saas.html>.

[3]. **Piattini, M y Garzas, J.** *Fabricas del Software: Experiencias, Tecnologías y Organización*. s.l. : Alfaomega Grupo Editor, 2007. ISBN: 8478978097 ISBN-13: 9788478978090.

[4]. **Talend open data solutions**. Etl for analytics. *Talend open data solutions*. [En línea] [Citado el: 10 de abril de 2009.] <http://es.talend.com/solutions-data-integration/etl-for-analytics.php>.

[5]. **Giménez Aznar, Javier**. liteSpace: ¿Análisis de datos sin un Data Warehouse? *Todo On Demand*. [En línea] 25 de mayo de 2009. [Citado el: 11 de noviembre de 2009.] <http://noticias.litebi.com/2009/05/litespace-analisis-de-datos-sin-un-data.html>.

[6]. **Pentaho**. Pentaho open source business intelligence. [En línea] 2000. [Citado el: 11 de noviembre de 2009.] <http://mondrian.pentaho.org/>.

[7]. **Google**. Google Web Toolkit. [En línea] 2007. [Citado el: 11 de noviembre de 2009.] <http://code.google.com/intl/es-ES/webtoolkit/>.

[8]. **Garnett, Jody y Pumphrey, Mike.** Geoserver. [En línea] 26 de marzo de 2005. [Citado el: 2009 de noviembre de 14.] <http://geoserver.org/display/GEOS/Welcome>.

[9]. **Pentaho.** Mondrian Schema. *Pentaho Analysis Services*. [En línea] 2000. [Citado el: 11 de noviembre de 2009.] <http://mondrian.pentaho.org/documentation/schema.php>.

[10]. **Giménez Aznar, Javier.** Análisis del estado de Mondrian (OLAP Open Source). *Todo BI*. [En línea] 21 de mayo de 2007. [Citado el: 20 de abril de 2009.] [http://www.telefonica.net/web2/todobi/Mayo07/Analisis\\_de\\_Mondrian.pdf](http://www.telefonica.net/web2/todobi/Mayo07/Analisis_de_Mondrian.pdf).

[11]. **Pentaho.** Kettle pentaho data integration. [Online] 2000. [Cited: noviembre 11, 2009.] <http://kettle.pentaho.org/>.

[12]. **Jayasingue, Deepal.** Understanding Axis2 Deployment Architecture. *Developer.com*. [En línea] 20 de octubre de 2005. [Citado el: 20 de abril de 2009.] <http://www.developer.com/open/article.php/3557741/Understanding-Axis2-Deployment-Architecture.htm>.

[13]. **The apache software foundation.** Securing SOAP Messages with Rampart. [En línea] 23 de octubre de 2009. [Citado el: 20 de abril de 2009.] [http://ws.apache.org/axis2/modules/rampart/1\\_3/security-module.html](http://ws.apache.org/axis2/modules/rampart/1_3/security-module.html).

[14]. —. Apache WSS4J. [En línea] 24 de julio de 2009. [Citado el: 11 de noviembre de 2009.] <http://ws.apache.org/wss4j/>.

[15]. **Universidad de Buenos Aires.** Algoritmo Diffie-Hellman. [En línea] 10 de octubre de 2007. [Citado el: 11 de noviembre de 2009.] <http://materias.fi.uba.ar/6669/docs/Diffie-Hellman.pdf>.

## Anexo 1 – Diagrama de clases de Litebi

El objetivo de este documento es mostrar las clases utilizadas para el diseño de los datos en Litebi y su relación.

A continuación algunos puntos a explicar sobre el diagrama de la *figura 33*:

- Litebi da la posibilidad de crear espacios para cada cliente (empresa o autónomo).
- Un espacio puede tener uno o muchos usuarios

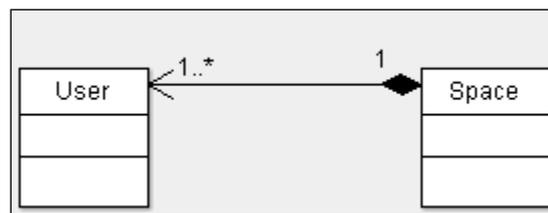
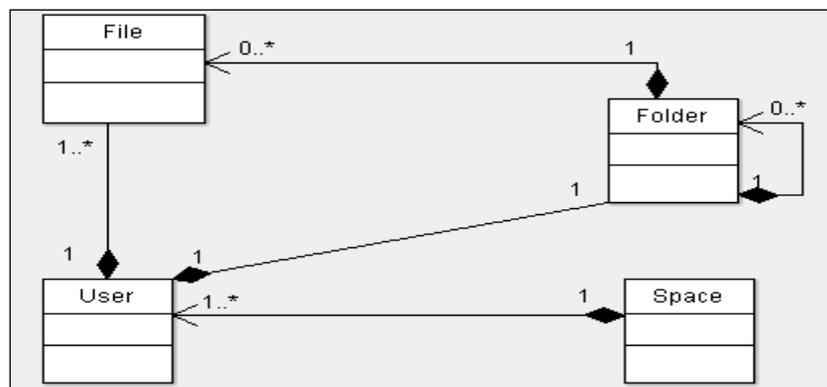


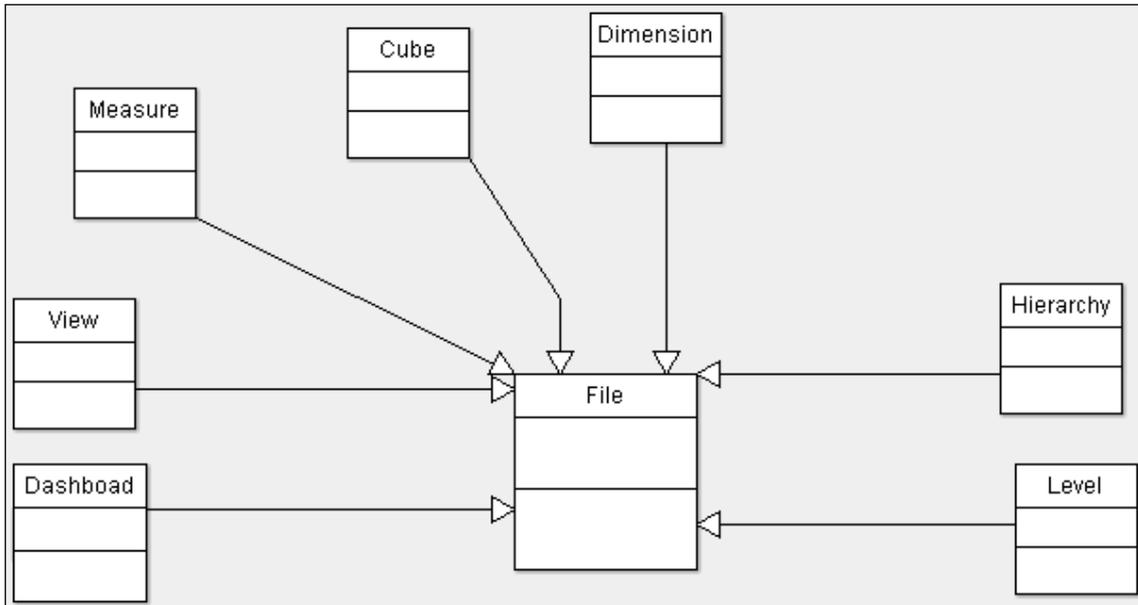
Figura 24: Relación Espacio y Usuario

- Cada espacio tiene un conjunto de carpetas y ficheros, hay 3 tipos de carpetas: las que contienen los modelos analíticos cubos o dimensiones y las que contienen informes y dashboards; y esta última pueden ser carpetas públicas (espacio compartido) o privadas (espacio único para el usuario actual).



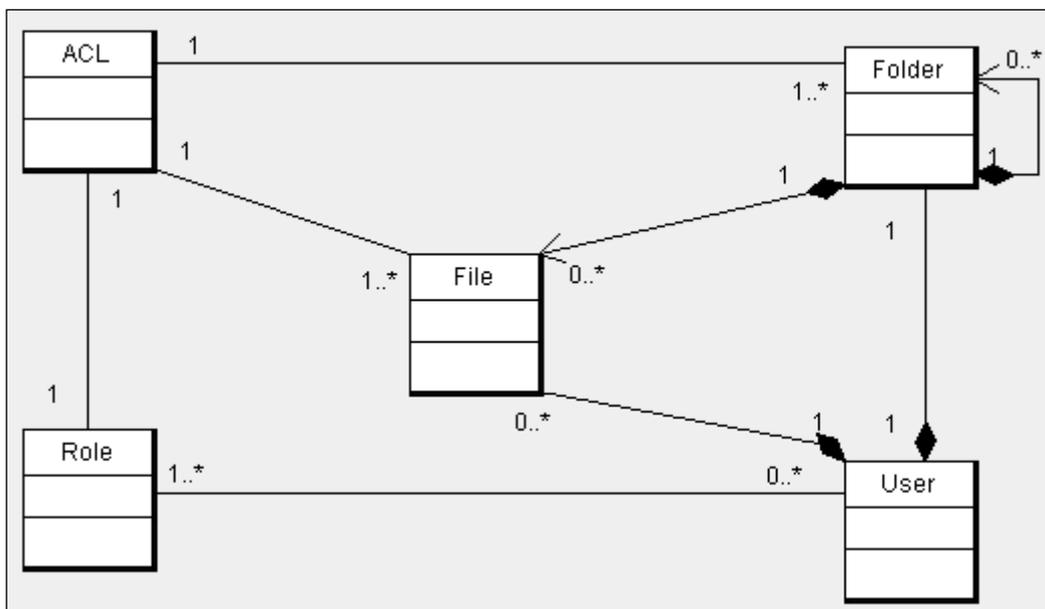
**Figura 25: Relación Espacio, Usuario, Fichero y Carpeta**

- Las carpetas, cubos, dimensiones, jerarquías, niveles, informes (vistas de análisis) y dashboard extienden de File.



**Figura 26: Relación Cubo, Dimensión, Jerarquía, Nivel, Vista, Dashboard y File**

- La seguridad de carpetas, archivos y usuarios está basado en roles y acs (escritura, lectura y borrado).



**Figura 27: Relación Usuario, Rol, ACL, Fichero y Carpeta**

- Una dimensión puede ser compartida o propia.

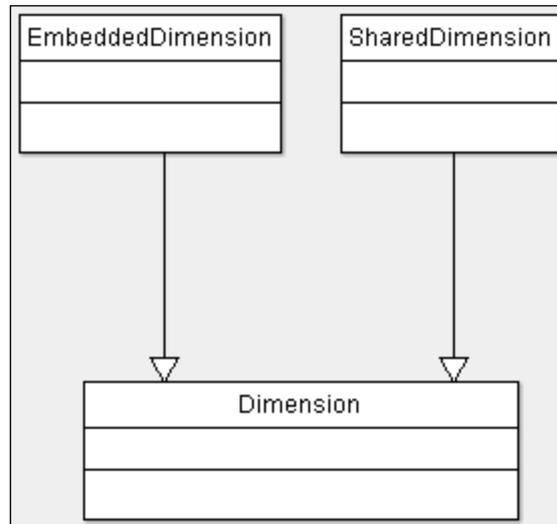


Figura 28: Generalización Dimension

- Un cubo debe tener al menos una Medida, miembro calculado y dimensión.

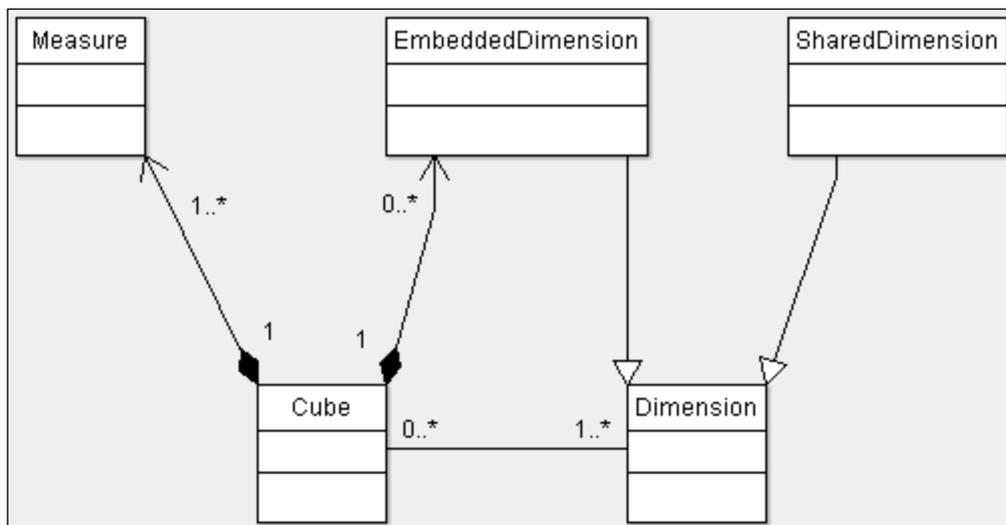
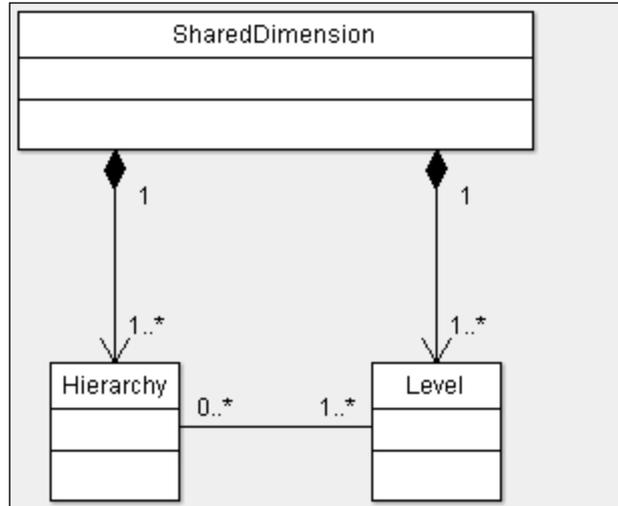


Figura 29: Relación Cubo, Measure y Dimensión

- Una dimensión compartida debe tener al menos un nivel y jerarquía. Una jerarquía debe tener al menos un nivel asociado.



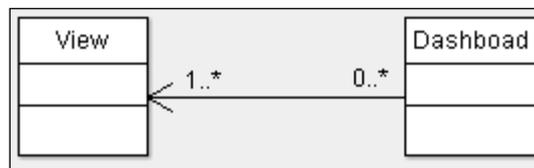
**Figura 30: Relación Dimensión Compartida, jerarquía y nivel**

- Una vista debe estar asociada a un cubo.



**Figura 31: Relación vista y cubo**

- Un Dashboard debe tener al menos una vista asociada



**Figura 32: Relación vista y Dashboard**

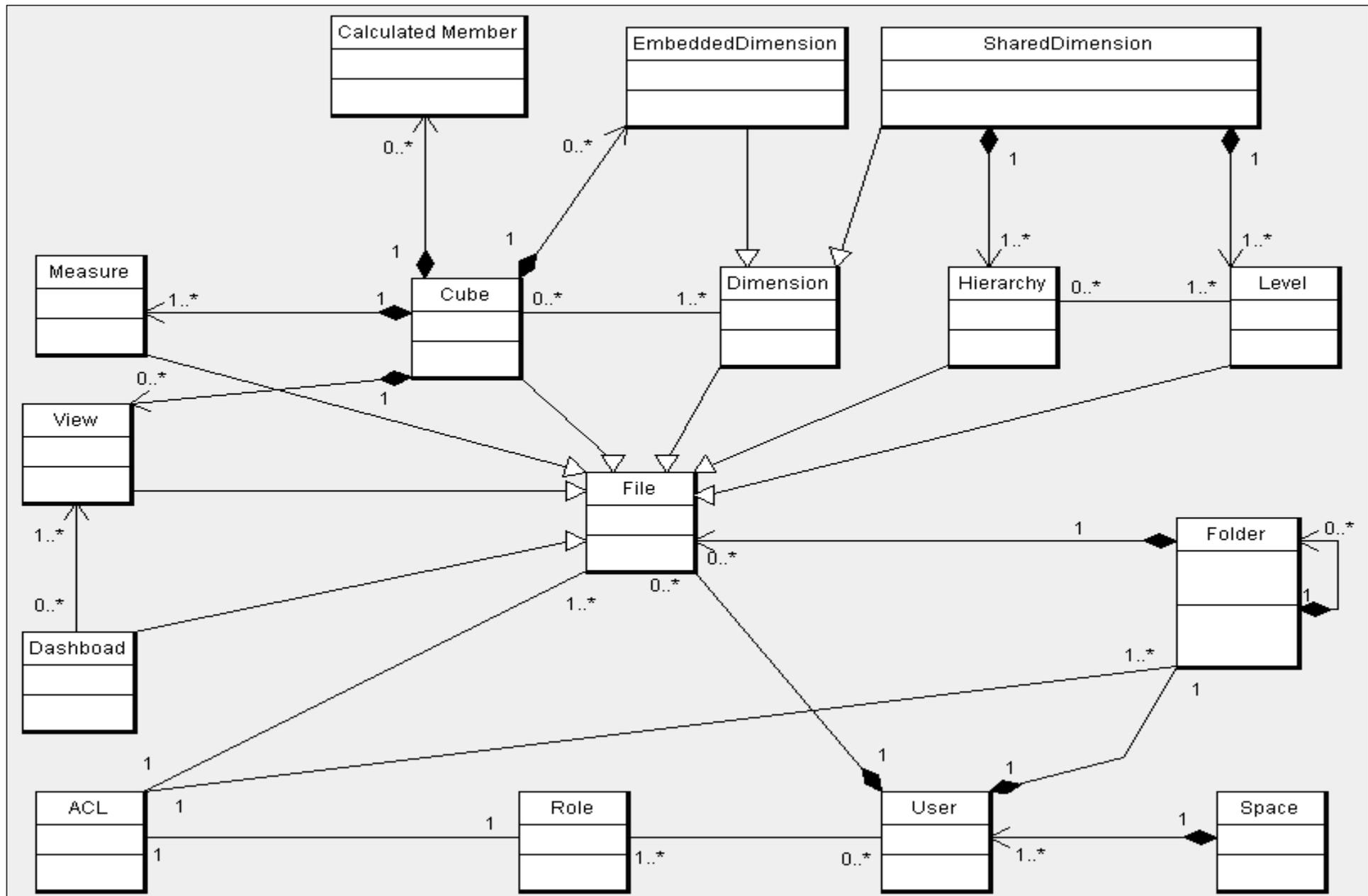


Figura 33: Diagrama de clases de Litebi

**Anexo 2 – Documento de requisitos de Litebi**

El siguiente documento es una especificación de requisitos de Software de Litebi.

# **Proyecto: Litebi**

## **Documento de Requisitos del Sistema**

Versión 1.0

Fecha: 01-11-2009

## Historial de Revisiones

| Fecha          | Descripción                   | Autor          |
|----------------|-------------------------------|----------------|
| 01 – 11 – 2009 | Documento inicial de creación | Juan Carlos H. |
|                |                               |                |
|                |                               |                |

## **DESCRIPCION DEL SISTEMA**

Litebi es una innovadora plataforma web de Inteligencia de negocios cuya tecnología permitirá integrar y analizar información de cualquier origen de forma sencilla y segura.

Una vez los datos han sido cargados en el diccionario de datos de Litebi, es posible realizar potentes análisis (desde cuadros de mando y olap hasta técnicas de inteligencia artificial o data mining), desde Internet, de forma segura, y sin necesidad de hardware o proyectos de implantación.

## **CARACTERISTICAS DEL SISTEMA**

- Subir e integrar datos en nuestro diccionario de datos a través de un canal altamente seguro.
- Disponer de un Espacio de Análisis para la empresa propio, customizable y seguro, con Reporting, Análisis Dinámico (Cubos), Cuadros de Mando, Minería de Datos, IA, etc...
- Acceder a Litebi desde cualquier plataforma con acceso a Internet
- Mejora el proceso de toma de decisiones, haciendo a la organización más eficaz y competitiva a partir de muy pocos días de uso.
- Reduce el coste total de disponer de un sistema de Business Intelligence.

## **CATALOGOS DE REQUISITOS DEL SISTEMA**

La especificación de requisitos del sistema ha sido realizada a través de la identificación de las principales características (*features*) del sistema y de la identificación de las características necesarias que deben existir en el entorno para que el sistema funcione correctamente.

## REQUISITOS FUNCIONALES

*Principales características (Features) del sistema*

1. Login
2. LiteSpace
  - Gestión de contenido
  - Editor de cubo
  - Editor de dimensión
3. LiteExplorer
4. LiteMonitor

## DEFINICIÓN DE LOS ACTORES

|              |  |
|--------------|--|
| Actor:       | <i>Cliente</i>   |
| Descripción: | <i>Denominación de la persona antes de hacer login en el sistema</i> |
| Comentarios: |  |

|              |   |
|--------------|---|
| Actor:       | <i>Usuario</i>  |
| Descripción: | <i>Denominación del cliente después de hacer login en el sistema</i>  |
| Comentarios: | <i>El usuario es una persona con los suficientes conocimientos para la creación de modelos analíticos en Litebi, normalmente es el consultor, pero puede ser cualquier persona que conozca el proceso para el manejo de los modelos analíticos.</i> |

## DIAGRAMAS DE CASOS DE USO

Los casos de uso del sistema se organizan de acuerdo a las características principales (features) del sistema, para cada uno de ellos se muestra el diagrama de caso de uso y su especificación.

### LOGIN :

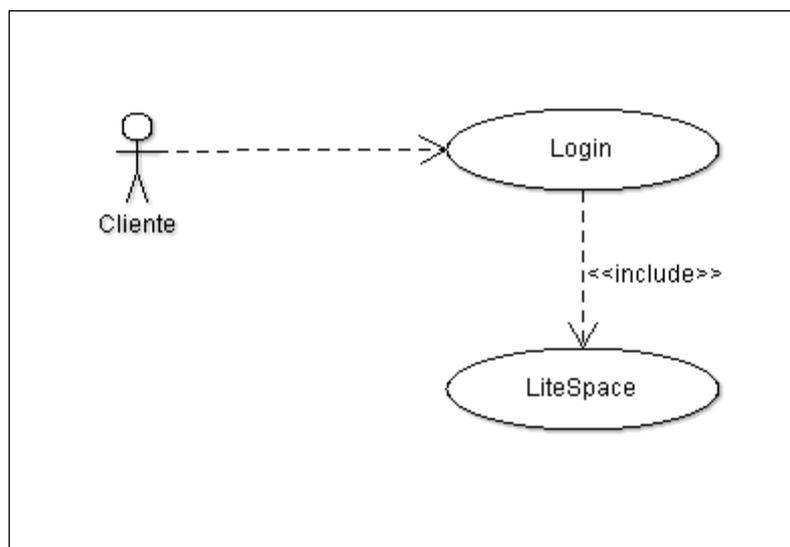


Figura 34: Diagrama de caso de uso: Hacer login

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Login</b>   |
| <b>Descripción:</b>         | El sistema verifica si el cliente se encuentra registrado en el sistema y si el espacio al cual pertenece se encuentra habilitado  |
| <b>Actores:</b>             | Cliente  |
| <b>Precondición:</b>        |  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"><li>1. El actor inserta su usuario y clave.</li><li>2. El sistema verifica si el usuario y clave concuerdan con los registrados en el sistema.</li><li>3. El sistema muestra el espacio de carpetas y ficheros del cliente (LiteSpace)</li></ol> |

Tabla 1: Especificación Caso de Uso de Hacer Login

## LITESPACE:

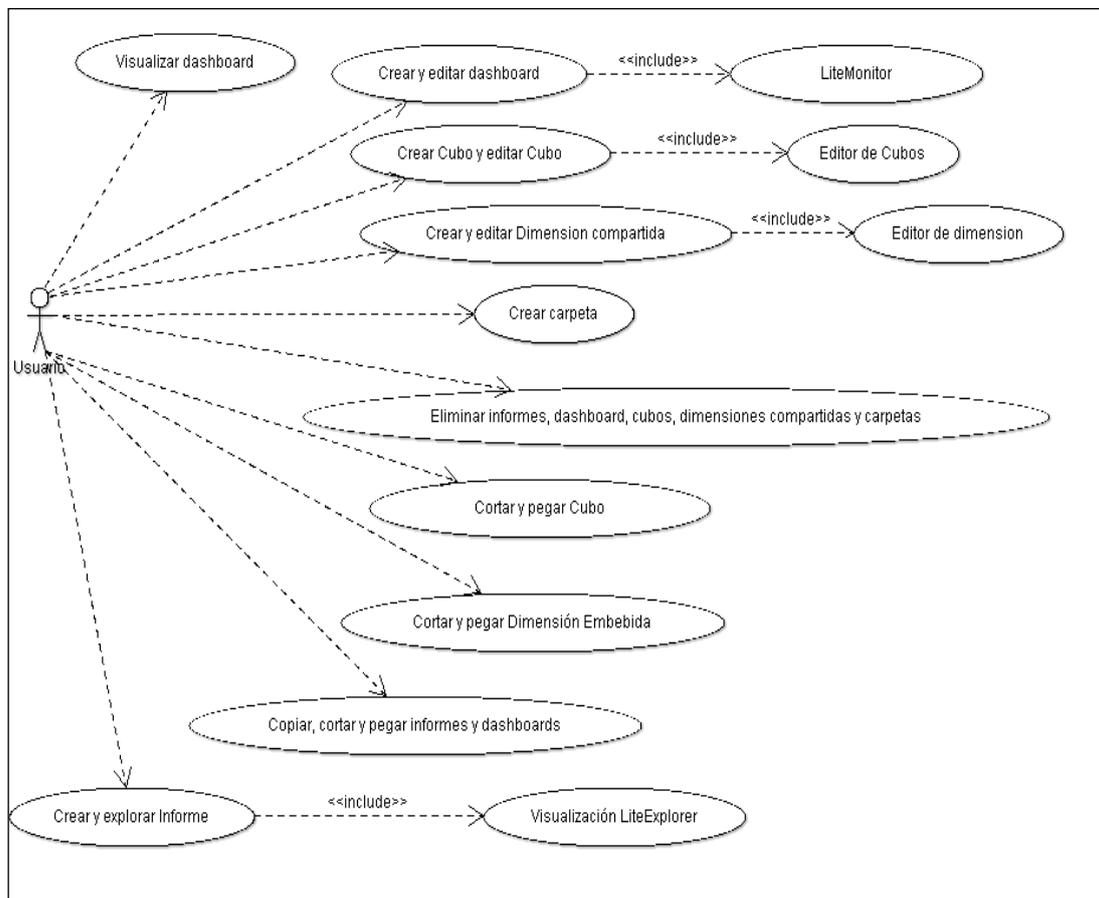


Figura 35: Diagrama de caso de uso: LiteSpace

## Gestión de contenido

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Crear Carpeta</b>  |
| <b>Descripción:</b>         | El sistema permite crear carpetas, donde se podrán almacenar diferentes tipos de ficheros, como informes, dashboard, cubos y dimensiones.   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>La carpeta debe tener una carpeta padre</li> <li>Para esta acción el usuario debe tener permiso necesarios sobre la carpeta padre.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona la carpeta donde crear la nueva carpeta</li> <li>El actor selecciona la opción: Nueva Carpeta.</li> <li>El sistema despliega una ventana con el campo</li> </ol> |

|  |  |
|--|--|
|  | <p>de nombre y descripción para la nueva carpeta.</p> <ol style="list-style-type: none"> <li>4. El actor selecciona la opción aceptar.</li> <li>5. El sistema creará la carpeta dentro de la carpeta seleccionada</li> </ol> |
|--|--|

**Tabla 2: Especificación Caso de Uso de Crear Carpeta**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Copiar Carpeta</b>   |
| <b>Descripción:</b>         | El sistema permite copiar y pegar carpetas entre carpetas   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• La carpeta destino debe ser diferente a la carpeta padre</li> <li>• Para esta acción el usuario debe tener permiso necesarios sobre las carpetas origen y destino.</li> <li>• Solo se puede copiar carpetas entre las carpetas publicas y privadas o copiar carpetas que pertenezcan solo a la de Cubos o solo a la de Dimensiones</li> </ul>                            |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la carpeta donde crear la nueva carpeta</li> <li>2. El actor selecciona la opción: Nueva Carpeta.</li> <li>3. El sistema despliega una ventana con el campo de nombre y descripción para la nueva carpeta.</li> <li>4. El actor selecciona la opción aceptar.</li> <li>5. El sistema creará la carpeta dentro de la carpeta seleccionada</li> </ol> |

**Tabla 3: Especificación Caso de Uso de Copiar Carpeta**

|                      |  |
|----------------------|--|
| <b>Caso de uso:</b>  | <b>Cortar Carpeta</b>  |
| <b>Descripción:</b>  | El sistema permite cortar y pegar carpetas entre carpetas  |
| <b>Actores:</b>      | Usuario  |
| <b>Precondición:</b> | <ul style="list-style-type: none"> <li>• La carpeta destino debe ser diferente a la carpeta padre</li> <li>• Para esta acción el usuario debe tener permiso necesarios sobre las carpetas origen y destino.</li> <li>• Solo se puede copiar carpetas entre las carpetas publicas y privadas o copiar carpetas que pertenezcan solo a la de Cubos o solo a la de</li> </ul> |

|                             |   |
|-----------------------------|---|
|                             | Dimensiones <ul style="list-style-type: none"> <li>No se pueden cortar las carpetas padres</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona la carpeta a cortar</li> <li>El actor selecciona la opción: Cortar.</li> <li>El actor selecciona la carpeta destino.</li> <li>El actor selecciona la opción: Pegar.</li> <li>El sistema vincula la carpeta cortada a la carpeta destino</li> </ol> |

Tabla 4: Especificación Caso de Uso de Cortar Carpeta

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Eliminar Carpeta</b>  |
| <b>Descripción:</b>         | El sistema permite eliminar carpetas   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>No se puede eliminar una carpeta padre.</li> <li>Solo se pueden eliminar carpetas vacías.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona la carpeta a eliminar</li> <li>El actor selecciona la opción: Eliminar.</li> <li>El sistema elimina el registro de la carpeta.</li> </ol> |

Tabla 5: Especificación Caso de Uso de Eliminar Carpeta

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Crear Informe</b>  |
| <b>Descripción:</b>         | El sistema crea informes a partir de un cubo, el cual es un modelo analítico con datos cargados.  |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>El cubo debe existir en el sistema y tener datos cargados</li> <li>Solo se puede crear informes en las carpetas privadas, públicas e hijas.</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona la carpeta donde crear el informe</li> <li>El actor selecciona la opción: Nueva Vista.</li> <li>El sistema despliega una lista de cubos y un campo para el nombre del informe.</li> <li>El actor selecciona el cubo al cual desea crear el informe y escribe el nombre del informe (obligatorio).</li> <li>El actor selecciona la opción aceptar.</li> <li>El sistema verifica si el usuario tiene permisos necesarios para crear datos en la carpeta</li> </ol> |

|  |  |
|--|--|
|  | <p>seleccionada.</p> <p>7. El sistema creará el registro del informe y se ejecutará el LiteExplorer.</p> |
|--|--|

**Tabla 6: Especificación Caso de Uso de Crear Informe**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Explorar Informe</b>  |
| <b>Descripción:</b>         | Explorar los informes registrados en el sistema  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Los informes deben estar registradas en el sistema</li> <li>• El cubo y las dimensiones asociadas al informe deben estar registradas en el sistema y tener datos cargados.</li> <li>• El actor debe tener los permisos necesarios para explorar el informe</li> </ul> |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el informe a explorar</li> <li>2. El actor selecciona la opción explorar</li> <li>3. El sistema ejecutará el LiteExplorer</li> </ol>   |

**Tabla 7: Especificación Caso de Uso de Explorar Informe**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Eliminar Informe</b>  |
| <b>Descripción:</b>         | Eliminar un informe registrado en el sistema   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El informe debe estar registrado previamente en el sistema.</li> <li>• El informe debe pertenecer al actor que desea eliminarlo.</li> </ul>                     |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el informe a eliminar</li> <li>2. El actor selecciona la opción eliminar</li> <li>3. El sistema elimina el registro de la vista</li> </ol> |

**Tabla 8: Especificación Caso de Uso de Eliminar Informe**

|                     |   |
|---------------------|---|
| <b>Caso de uso:</b> | <b>Copiar informe</b>   |
| <b>Descripción:</b> | El sistema permite copiar y pegar informes de una carpeta a otra. |
| <b>Actores:</b>     | Usuario   |

|                             |  |
|-----------------------------|--|
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se puede copiar informes entre carpetas pertenecientes a las públicas y privadas.</li> <li>• El actor debe tener los permisos necesarios sobre el informe y la carpeta para realizar esta acción.</li> <li>• La carpeta destino debe ser diferente a la carpeta origen</li> </ul>                            |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el informe a copiar</li> <li>2. El actor selecciona la opción: Copiar.</li> <li>3. El actor selecciona la carpeta destino a pegar el informe.</li> <li>4. El actor selecciona la opción pegar</li> <li>5. El sistema creará un registro del informe pegado en la carpeta destino.</li> </ol> |

Tabla 9: Especificación Caso de Uso de Copiar Informe

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Cortar informe</b>   |
| <b>Descripción:</b>         | El sistema permite cortar y pegar informes de una carpeta a otra.   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se puede cortar informes entre carpetas pertenecientes a las públicas y privadas.</li> <li>• El actor debe tener los permisos necesarios sobre el informe y la carpeta para realizar esta acción.</li> <li>• La carpeta destino debe ser diferente a la carpeta origen</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el informe a copiar</li> <li>2. El actor selecciona la opción: Copiar.</li> <li>3. El actor selecciona la carpeta destino a pegar el informe.</li> <li>4. El actor selecciona la opción pegar</li> <li>5. El sistema vincula el registro del informe a la carpeta destino.</li> <li>6. El sistema eliminará el registro de la carpeta origen</li> </ol> |

Tabla 10: Especificación Caso de Uso de Cortar Informe

|                      |  |
|----------------------|--|
| <b>Caso de uso:</b>  | <b>Crear Dashboard</b>   |
| <b>Descripción:</b>  | El sistema crea ficheros tipo dashboard  |
| <b>Actores:</b>      | Usuario  |
| <b>Precondición:</b> | <ul style="list-style-type: none"> <li>• El actor debe tener los permisos necesarios en la carpeta para crear ficheros.</li> <li>• Solo se puede crear informes en las carpetas</li> </ul> |

|                             |  |
|-----------------------------|--|
|                             | privadas, públicas e hijas.  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la carpeta donde crear el dashboard</li> <li>2. El actor selecciona la opción: Nuevo Dashboard.</li> <li>3. El sistema despliega una ventana para escribir el nombre y descripción del dashboard.</li> <li>4. El actor selecciona el cubo al cual desea crear el informe, escribe el nombre (obligatorio) y la descripción (opcional).</li> <li>5. El actor selecciona la opción aceptar.</li> <li>6. El sistema creará el registro del informe y se ejecutará el LiteExplorer.</li> </ol> |

Tabla 11: Especificación Caso de Uso de Crear Dashboard

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Visualizar Dashboard</b>  |
| <b>Descripción:</b>         | Visualizar dashboard diseñado por los actores  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El dashboard debe estar registrado previamente en el sistema</li> <li>• El actor debe tener los permisos necesarios para visualizar el dashboard</li> </ul>                                     |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el dashboard a visualizar</li> <li>2. El actor selecciona la opción visualizar</li> <li>3. El sistema abrirá una página donde se podrá visualizar el dashboard.</li> </ol> |

Tabla 12: Especificación Caso de Uso de Visualizar Dashboard

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Eliminar Dashboard</b>  |
| <b>Descripción:</b>         | Eliminar un dashboard registrado en el sistema   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El dashboard debe estar registrado previamente en el sistema.</li> <li>• El actor debe ser el propietario del dashboard.</li> </ul>                             |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el informe a eliminar</li> <li>2. El actor selecciona la opción eliminar</li> <li>3. El sistema elimina el registro de la vista</li> </ol> |

Tabla 13: Especificación Caso de Uso de Eliminar Dashboard

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Copiar Dashboard</b>  |
| <b>Descripción:</b>         | El sistema permite copiar y pegar dashboard de una carpeta a otra.   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se puede copiar dashboards entre carpetas pertenecientes a las públicas y privadas.</li> <li>• El actor debe tener los permisos necesarios sobre el dashboard y la carpeta para realizar esta acción.</li> <li>• La carpeta destino debe ser diferente a la carpeta origen</li> </ul>                            |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el dashboard a copiar</li> <li>2. El actor selecciona la opción: Copiar.</li> <li>3. El actor selecciona la carpeta destino a pegar el dashboard.</li> <li>4. El actor selecciona la opción pegar</li> <li>5. El sistema creará un registro del informe pegado en la carpeta destino.</li> </ol> |

Tabla 14: Especificación Caso de Uso de Copiar Dashboard

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Cortar Dashboard</b>   |
| <b>Descripción:</b>         | El sistema permite cortar y pegar dashboard de una carpeta a otra.  |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se puede cortar dashboard entre carpetas pertenecientes a las públicas y privadas.</li> <li>• El actor debe tener los permisos necesarios sobre el dashboard y la carpeta para realizar esta acción.</li> <li>• La carpeta destino debe ser diferente a la carpeta origen</li> </ul>                        |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el dashboard a copiar</li> <li>2. El actor selecciona la opción: Copiar.</li> <li>3. El actor selecciona la carpeta destino a pegar el dashboard.</li> <li>4. El actor selecciona la opción pegar</li> <li>5. El sistema vincula el registro del dashboard a la carpeta destino.</li> </ol> |

Tabla 15: Especificación Caso de Uso de Cortar Dashboard

## Gestión de Cubos y Dimensiones

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Crear Cubo</b>  |
| <b>Descripción:</b>         | El sistema crea modelos analíticos para cargar datos, en este caso un cubo   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se pueden crear cubos en la carpeta Cubos.</li> <li>• El número de cubos creados debe ser menor o igual al número máximo de cubos permitidos para el usuario.</li> </ul> |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la carpeta donde crear el cubo</li> <li>2. El actor selecciona la opción: Nuevo Cubo.</li> <li>3. El sistema ejecutará el Editor de Cubos.</li> </ol>    |

Tabla 16: Especificación Caso de Uso de Crear Cubo

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Editar Cubo</b>  |
| <b>Descripción:</b>         | El sistema permite editar los cubos   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se pueden editar cubos registrados en el sistema.</li> <li>• El usuario debe tener los permisos necesarios sobre el cubo para editarlo</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el cubo a editar</li> <li>2. El actor selecciona la opción: Editar.</li> <li>3. El sistema ejecutará el Editor de Cubos.</li> </ol> |

Tabla 17: Especificación Caso de Uso de Editar Cubo

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Eliminar Cubo</b>  |
| <b>Descripción:</b>         | Eliminar un cubo registrado en el sistema   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El cubo debe estar registrado en el sistema.</li> <li>• El actor debe tener los permisos necesarios sobre el cubo para eliminarlo.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el cubo a eliminar.</li> <li>2. El actor selecciona la opción eliminar</li> <li>3. El sistema elimina el registro del cubo y los datos cargados en él.</li> </ol> |

Tabla 18: Especificación Caso de Uso de Eliminar Cubo

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Cortar Cubo</b>   |
| <b>Descripción:</b>         | El sistema permite cortar cubos de una carpeta a otra  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>Las carpetas origen y destino deben ser parte de la carpeta Cubos.</li> <li>La carpeta origen no puede ser la misma que la carpeta destino</li> <li>El actor debe tener los permisos necesarios sobre el cubo para cortarla y pegarla en otra carpeta.</li> </ul>     |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona el cubo a cortar.</li> <li>El actor selecciona la opción: Cortar.</li> <li>El actor selecciona la carpeta destino</li> <li>El actor selecciona la opción: Pegar</li> <li>El sistema vincula el registro del cubo a la carpeta destino.</li> </ol> |

Tabla 19: Especificación Caso de Uso de Cortar Cubo

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Crear Dimensión Compartida</b>  |
| <b>Descripción:</b>         | El sistema crea modelos analíticos para cargar datos, en este caso una dimensión compartida  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>Solo se pueden crear dimensiones compartidas en la carpeta Dimensiones.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona la carpeta donde crear la dimensión compartida</li> <li>El actor selecciona la opción: Nueva Dimensión Compartida.</li> <li>El sistema ejecutará el Editor de Dimensiones.</li> </ol> |

Tabla 20: Especificación Caso de Uso de Crear Dimensión Compartida

|                      |   |
|----------------------|---|
| <b>Caso de uso:</b>  | <b>Eliminar Dimensión Compartida</b>  |
| <b>Descripción:</b>  | Eliminar una dimensión compartida registrado en el sistema  |
| <b>Actores:</b>      | Usuario   |
| <b>Precondición:</b> | <ul style="list-style-type: none"> <li>La dimensión compartida debe estar registrada en el sistema.</li> <li>La dimensión compartida no puede estar en uso por un cubo.</li> <li>El actor debe tener los permisos necesarios sobre</li> </ul> |

|                             |   |
|-----------------------------|---|
|                             | la dimensión compartida para eliminarlo.  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la dimensión compartida a eliminar.</li> <li>2. El actor selecciona la opción eliminar</li> <li>3. El sistema elimina el registro de la dimensión compartida y los datos cargados en ella.</li> </ol> |

**Tabla 21: Especificación Caso de Uso de Eliminar Dimensión Compartida**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Editar Dimensión Compartida</b>  |
| <b>Descripción:</b>         | El sistema permite editar las dimensiones compartidas   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Solo se pueden editar dimensiones compartidas registradas en el sistema.</li> <li>• El actor debe tener los permisos necesarios sobre la dimensión compartida para editarla</li> </ul> |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la dimensión compartida a editar</li> <li>2. El actor selecciona la opción: Editar.</li> <li>3. El sistema ejecutará el Editor de Dimensiones.</li> </ol>         |

**Tabla 22: Especificación Caso de Uso de Editar Dimensión Compartida**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Cortar Dimensión Compartida</b>  |
| <b>Descripción:</b>         | El sistema permite cortar dimensiones compartidas de una carpeta a otra   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Las carpetas origen y destino deben ser parte de la carpeta Dimensiones.</li> <li>• La carpeta origen no puede ser la misma que la carpeta destino.</li> <li>• El actor debe tener los permisos necesarios sobre la dimensión compartida para cortarla y pegarla en otra carpeta.</li> </ul>                         |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la dimensión compartida a cortar.</li> <li>2. El actor selecciona la opción: Cortar.</li> <li>3. El actor selecciona la carpeta destino</li> <li>4. El actor selecciona la opción: Pegar</li> <li>5. El sistema vincula el registro de la dimensión compartida a la carpeta destino.</li> </ol> |

**Tabla 23: Especificación Caso de Uso de Cortar Dimensión Compartida**

## GESTOR EDITOR DE CUBO:

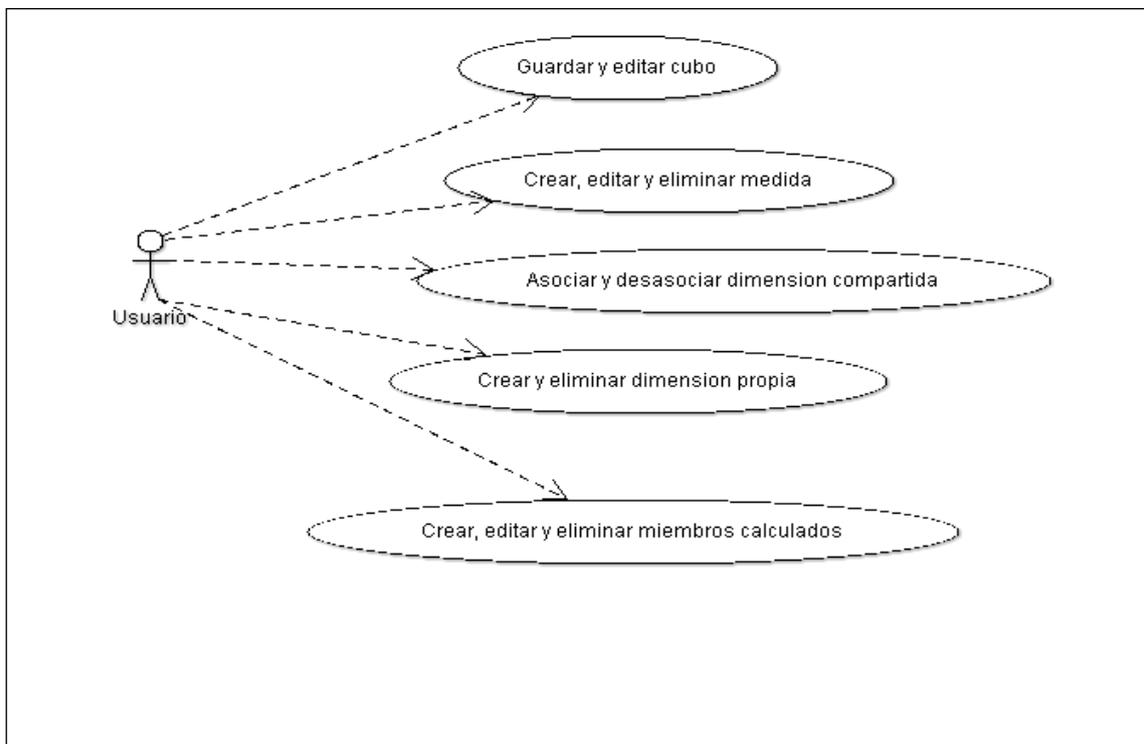


Figura 36: Diagrama de caso de uso: Editor de Cubo

## Edición de Cubos

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Crear y Editar Métrica</b>   |
| <b>Descripción:</b>         | El sistema permite crear y editar una o mas métricas en un cubo   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>El cubo no debe tener datos cargados</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el item Métricas.</li> <li>2. El actor selecciona la opción: Crear Métrica.</li> <li>3. El sistema crea una métrica nueva.</li> <li>4. El actor llena los campos de la métrica.</li> <li>5. El sistema editará laos campos de la métrica mientras que el actor llena los campos de la métrica.</li> </ol> |

Tabla 24: Especificación Caso de Uso de Crear y Editar Métrica

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Crear y Editar Dimensión Embebida</b>  |
| <b>Descripción:</b>         | El sistema permite crear y editar una o mas dimensiones embebidas en un cubo  |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El cubo no debe tener datos cargados</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el item Dimensiones.</li> <li>2. El actor selecciona la opción: Nueva Dimensión Embebida.</li> <li>3. El sistema crea una dimensión embebida.</li> <li>4. El actor llena los campos de la dimensión embebida.</li> <li>5. El sistema editará los campos de la dimensión embebida mientras que el actor llena los campos de la dimensión.</li> </ol> |

**Tabla 25: Especificación Caso de Uso de Crear y Editar Dimensión Embebida**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Asociar Dimensión Compartida</b>   |
| <b>Descripción:</b>         | <p>El sistema permite asociar una o mas dimensiones compartidas en un cubo.</p> <p>Se puede asociar la misma dimensión compartida varias veces en el mismo cubo.</p>  |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El cubo no debe tener datos cargados</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el item Dimensiones.</li> <li>2. El actor selecciona la opción: Nueva Dimensión Compartida.</li> <li>3. Aparecerá una lista con las dimensiones compartidas registradas en el sistema.</li> <li>4. El actor escoge la dimensión compartida que se quiere asociar.</li> <li>5. El actor selecciona la opción: Aceptar.</li> <li>6. El sistema asocia la dimensión compartida al cubo.</li> </ol> |

**Tabla 26: Especificación Caso de Uso de Asociar Dimensión Compartida**

|                     |   |
|---------------------|---|
| <b>Caso de uso:</b> | <b>Desasociar Dimensión Compartida</b>                                  |
| <b>Descripción:</b> | El sistema permite desasociar una o mas dimensiones compartidas al cubo |
| <b>Actores:</b>     | Usuario   |

|                             |   |
|-----------------------------|---|
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• La dimensión compartida debe estar registrada en el sistema y asociada al cubo</li> <li>• El cubo no debe tener datos cargados</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la dimensión compartida a desasociar.</li> <li>2. El actor selecciona la opción: Eliminar.</li> <li>3. El sistema desvincula la dimensión compartida del cubo.</li> </ol> |

**Tabla 27: Especificación Caso de Uso de Desasociar Dimensión Compartida**

### Guardar registros de un Cubo

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Guardar y Editar Cubo</b>   |
| <b>Descripción:</b>         | El sistema permite guardar los registros de un cubo  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El cubo no debe tener datos cargados</li> <li>• El usuario debe tener los permisos necesarios sobre el cubo para guardarlo</li> <li>• Un cubo debe tener al menos una dimensión y métrica asociada</li> </ul> |
| <b>Escenario Principal:</b> | <p>El actor selecciona la opción: Guardar.</p> <p>El sistema eliminará los datos cargados, si los tiene, y guardará los registros del cubo.</p>  |

**Tabla 28: Especificación Caso de Uso de Guardar y Editar Cubo**

## GESTOR EDITOR DIMENSIONES COMPARTIDAS

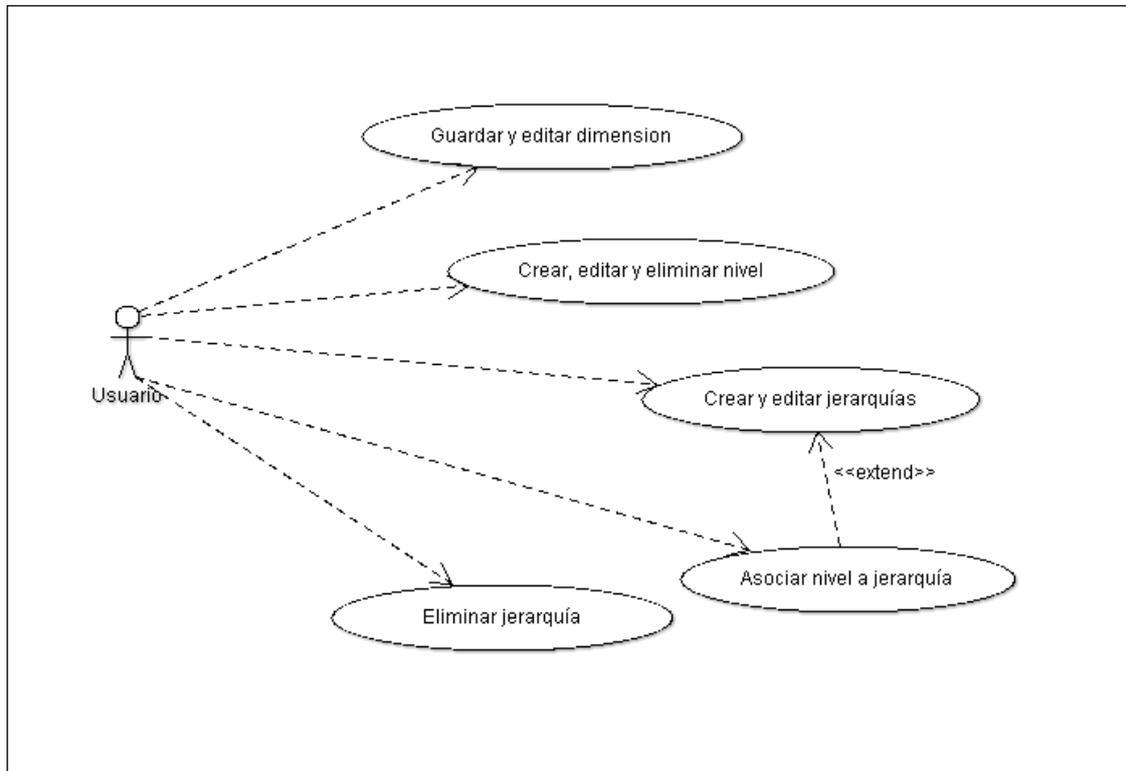


Figura 37: Diagrama casos de uso: Edición Dimensiones Compartidas

### Edición de dimensiones compartidas

|                                |   |
|--------------------------------|---|
| <b>Caso de uso:</b>            | <b>Crear y Editar Niveles</b>   |
| <b>Descripción:</b>            | El sistema permite crear y editar uno o mas niveles en la dimensión   |
| <b>Actores:</b>                | Usuario   |
| <b>Precondición:</b>           | <ul style="list-style-type: none"> <li>La dimensión no debe tener datos cargados, ni estar asociado a uno o más cubos</li> </ul>  |
| <b>1. Escenario Principal:</b> | <ol style="list-style-type: none"> <li>El actor selecciona el item Niveles.</li> <li>El actor selecciona la opción: Crear.</li> <li>El sistema crea un nivel nuevo.</li> <li>El actor llena los campos del nivel.</li> <li>El sistema editará los campos del nivel mientras que el actor llena los campos del nivel.</li> </ol> |

Tabla 29: Especificación Caso de Uso de Crear y Editar Niveles

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Eliminar Nivel</b>   |
| <b>Descripción:</b>         | El sistema permite eliminar uno o mas niveles   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• La dimensión no debe tener datos cargados, ni estar asociado a uno o más cubos.</li> <li>• Solo se pueden eliminar niveles creados por el actor</li> </ul>               |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el item Niveles.</li> <li>2. El actor selecciona la opción: Eliminar.</li> <li>3. El sistema elimina el registro del nivel seleccionado.</li> </ol> |

**Tabla 30: Especificación Caso de Uso de Eliminar Nivel**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Crear y Editar Jerarquías</b>   |
| <b>Descripción:</b>         | El sistema permite crear y editar una o mas jerarquías en una dimensión compartida   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        |  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el item Jerarquías.</li> <li>2. El actor selecciona la opción: Nueva.</li> <li>3. El sistema crea una jerarquía.</li> <li>4. El actor asocia uno o más niveles de la dimensión compartida a la jerarquía y llena los campos de sus propiedades.</li> <li>5. El sistema editará la jerarquía a métrica que el actor asocia niveles y llena los campos de la jerarquía.</li> </ol> |

**Tabla 31. Especificación Caso de Uso de Crear y Editar Jerarquías**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Eliminar Jerarquías</b>   |
| <b>Descripción:</b>         | El sistema permite eliminar las jerarquías de la dimensión compartida  |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        |  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la jerarquía a eliminar.</li> <li>2. El actor selecciona la opción: Eliminar.</li> <li>3. El sistema elimina los registros de la jerarquía.</li> </ol> |

**Tabla 32: Especificación Caso de Uso de Eliminar Jerarquías**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Asociar nivel a las jerarquías</b>  |
| <b>Descripción:</b>         | El sistema permite asociar uno o más niveles a una jerarquía   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• Los niveles a asociar deben estar registrados en el sistema.</li> <li>• Las jerarquías deben estar registradas en el sistema.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona una jerarquía.</li> <li>2. El actor selecciona la opción: Agregar Nivel.</li> <li>3. El sistema muestra una lista de niveles</li> <li>4. El actor escoge el nivel a agregar</li> <li>5. El actor escoge la opción: Agregar</li> <li>6. El sistema asocia el nivel escogido con la jerarquía actual.</li> </ol> |

**Tabla 33: Especificación Caso de Uso de Asociar nivel a las Jerarquías**

### **Guardar registros de una dimensión compartida**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Guardar y Editar Dimensión Compartida</b>   |
| <b>Descripción:</b>         | El sistema permite guardar los registros de una dimensión compartida   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• La dimensión compartida no debe tener datos cargados</li> <li>• El usuario debe tener los permisos necesarios sobre la dimensión compartida para guardarlo</li> <li>• Una dimensión compartida debe tener un nivel base y al menos una jerarquía asociada.</li> </ul> |
| <b>Escenario Principal:</b> | <p>El actor selecciona la opción: Guardar.</p> <p>El sistema eliminará los datos cargados, si los tiene, y guardará los registros del cubo.</p>  |

**Tabla 34: Especificación Caso de Uso de Guardar y Editar Dimensión Compartida**

## GESTOR DE DASHBOARDS :

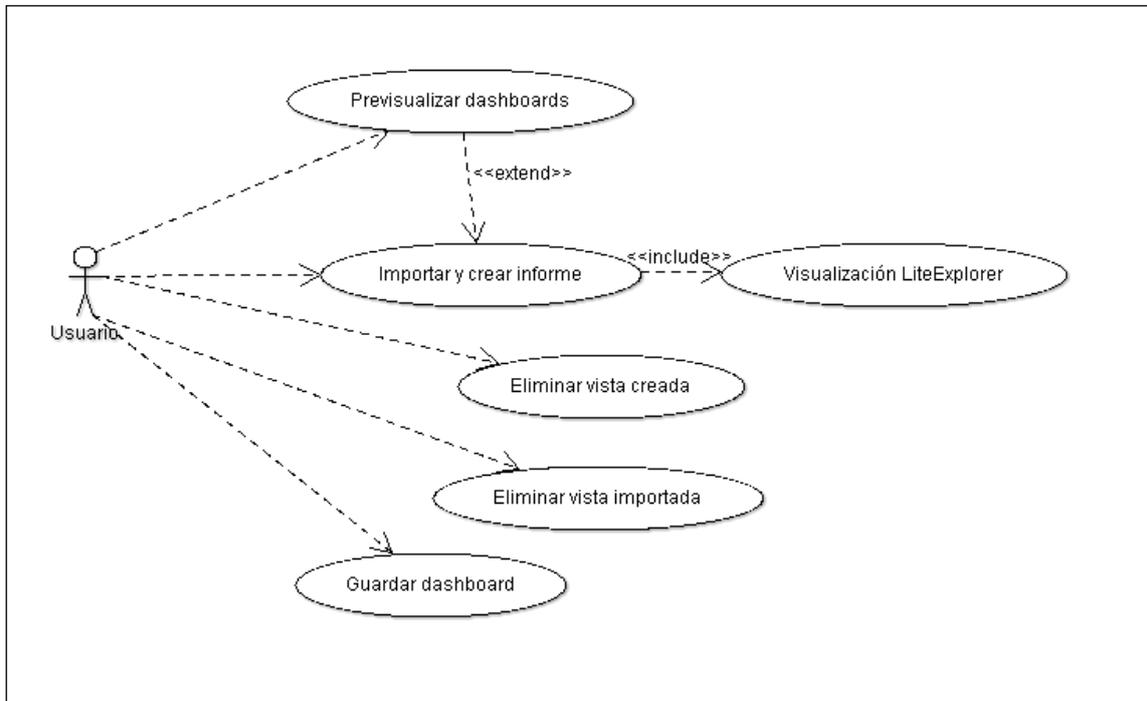


Figura 38: Diagrama de caso de uso: LiteMonitor

### Gestión de informes o vistas

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Crear un informe</b>   |
| <b>Descripción:</b>         | El sistema permite crear uno o mas informes para asociar a un dashboard   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>El cubo asociado al informe debe tener datos cargados.</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el lugar que se le asignará a la vista en el dashboard.</li> <li>2. El actor selecciona la opción: Nueva Vista.</li> <li>3. El sistema mostrará una lista de cubos.</li> <li>4. El actor selecciona el cubo deseado</li> <li>5. El actor llena los campos para el nuevo informe</li> <li>6. El actor selecciona la opción: Aceptar</li> <li>7. El sistema crea un registro de la nueva vista</li> </ol> |

|  |   |
|--|---|
|  | asociada al dashboard<br>8. El sistema ejecuta LiteExplorer |
|--|---|

Tabla 35: Especificación Caso de Crear un Informe

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Importar un informe</b>  |
| <b>Descripción:</b>         | El sistema permite importar uno o mas informes registrados en el sistema para asociar a un dashboard  |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>El informe debe estar registrado en el sistema.</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona el lugar que se le asignará a la vista en el dashboard.</li> <li>2. El actor selecciona la opción: Importar Vista.</li> <li>3. El sistema mostrará una lista de vistas registradas en el sistema.</li> <li>4. El actor selecciona la vista deseada</li> <li>5. El actor llena los campos para el importe de la vista</li> <li>6. El actor selecciona la opción: Aceptar</li> <li>7. El sistema asocia la vista asociada al dashboard</li> <li>8. El sistema ejecuta LiteExplorer</li> </ol> |

Tabla 36: Especificación Caso de Uso de Importar un Informe

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Eliminar vista creada</b>  |
| <b>Descripción:</b>         | El sistema permite desvincular y eliminar el registro de uno o mas informes creados de un dashboard   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>El informe debe estar registrado en el sistema.</li> </ul>   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la vista a eliminar</li> <li>2. El actor selecciona la opción: Eliminar.</li> <li>3. El sistema desvincula la vista del dashboard y elimina el registro de la misma.</li> </ol> |

Tabla 37: Especificación Caso de Uso de Eliminar Vista Creada

|                      |   |
|----------------------|---|
| <b>Caso de uso:</b>  | <b>Eliminar vista importada</b>   |
| <b>Descripción:</b>  | El sistema permite desvincular uno o mas informes creados de un dashboard                         |
| <b>Actores:</b>      | Usuario   |
| <b>Precondición:</b> | <ul style="list-style-type: none"> <li>El informe debe estar registrado en el sistema.</li> </ul> |

|                             |   |
|-----------------------------|---|
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la vista a eliminar</li> <li>2. El actor selecciona la opción: Eliminar.</li> <li>3. El sistema desvincula la vista del dashboard.</li> </ol> |
|-----------------------------|---|

**Tabla 38: Especificación Caso de Uso de Eliminar Vista Importada**

### Gestión de dashboard

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Previsualizar dashboard</b>  |
| <b>Descripción:</b>         | El sistema permite previsualizar un dashboard   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El dashboard debe estar registrado en el sistema.</li> <li>• El dashboard debe tener al menos una vista o informe asociado</li> </ul>                      |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Previsualizar</li> <li>2. El sistema abrirá una página donde se podrá ver como se verá el diseño del dashboard.</li> </ol> |

**Tabla 39: Especificación Caso de Uso de Previsualizar Dashboard**

|                             |  |
|-----------------------------|--|
| <b>Caso de uso:</b>         | <b>Guardar dashboard</b>   |
| <b>Descripción:</b>         | El sistema permite guardar los registros de un dashboard   |
| <b>Actores:</b>             | Usuario  |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• El dashboard debe tener al menos una vista o informe asociado.</li> </ul>                                     |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Guardar</li> <li>2. El sistema guardará el registro del dashboard.</li> </ol> |

**Tabla 40: Especificación Caso de Uso de Guardar Dashboard**

## LITEEXPLORER :

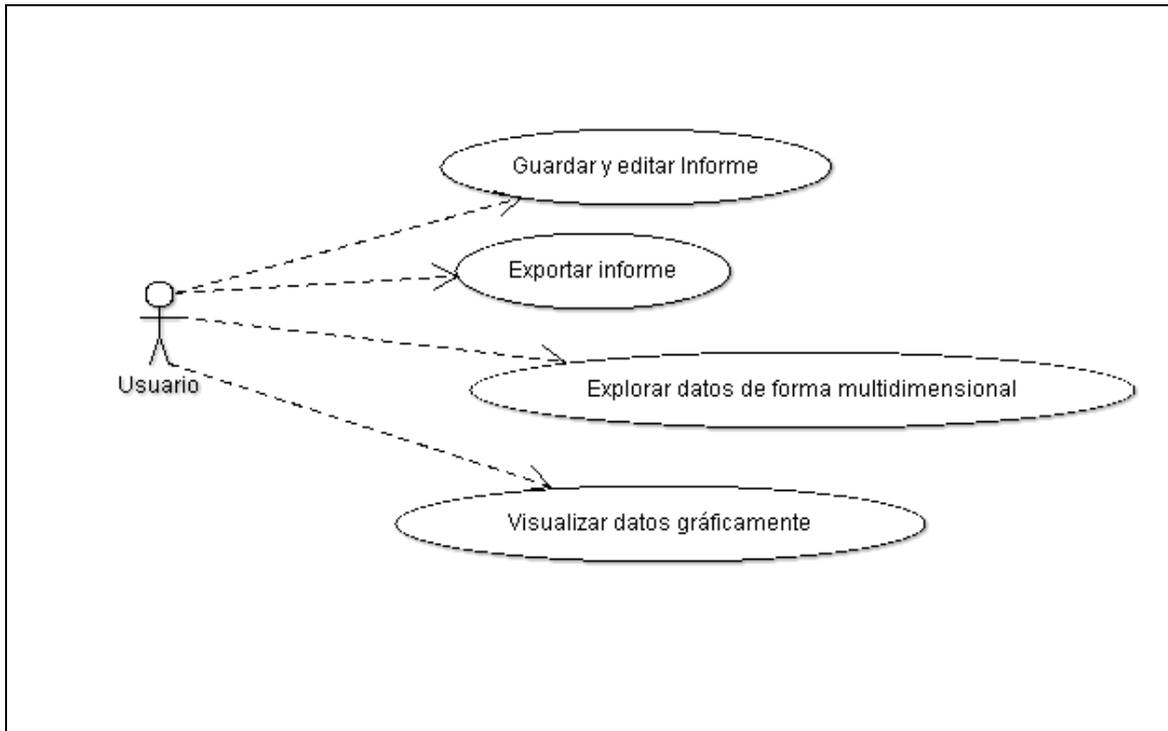


Figura 39: Diagrama de caso de uso: LiteExplorer

## Gestión de Datos

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Explorar datos de forma multidimensional</b>   |
| <b>Descripción:</b>         | El sistema permite explorar los datos cargados a los modelos analíticos de litebi (cubos y/o dimensiones) |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        |   |
| <b>Escenario Principal:</b> | 1. El actor selecciona las diferentes opciones navegacionales sobre los datos cargados.                   |

Tabla 41: Especificación Caso de Uso de Explorar datos de forma multidimensional

|                     |  |
|---------------------|--|
| <b>Caso de uso:</b> | <b>Exportar informe</b>  |
| <b>Descripción:</b> | El sistema permite exportar a un fichero tipo Excel un informe |
| <b>Actores:</b>     | Usuario  |

|                             |  |
|-----------------------------|--|
| <b>Precondición:</b>        |  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Exportar a Excel</li> <li>2. El sistema genera un fichero tipo Excel</li> <li>3. El actor selecciona la ubicación donde se guardará el fichero</li> <li>4. El fichero se guarda en la ubicación escogida</li> </ol> |

**Tabla 42: Especificación Caso de Uso de Exportar Informe**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Visualizar datos gráficamente</b>  |
| <b>Descripción:</b>         | El sistema permite visualizar los datos explorados de forma gráfica   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        |   |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Tipo de gráfica</li> <li>2. El sistema muestra la lista de las gráficas disponibles</li> <li>3. El actor selecciona la gráfica deseada y activa la opción mostrar gráfico.</li> <li>4. El sistema genera y muestra la gráfica seleccionada.</li> </ol> |

**Tabla 43: Especificación Caso de Uso de Visualizar Datos Gráficamente**

|                             |   |
|-----------------------------|---|
| <b>Caso de uso:</b>         | <b>Guardar vista</b>  |
| <b>Descripción:</b>         | El sistema permita guardar la vista explorada   |
| <b>Actores:</b>             | Usuario   |
| <b>Precondición:</b>        | <ul style="list-style-type: none"> <li>• La vista está registrada en el sistema</li> </ul>  |
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Guardar vista de cubo</li> <li>2. El sistema muestra una ventana donde se podrá escoger la carpeta donde se guardará el informe y ponerle un nombre y descripción al informe.</li> <li>3. El actor selecciona: Aceptar.</li> <li>4. El sistema guarda el informe explorado.</li> </ol> |

**Tabla 44: Especificación Caso de Uso de Guardar Vista**

|                      |  |
|----------------------|--|
| <b>Caso de uso:</b>  | <b>Editar vista</b>  |
| <b>Descripción:</b>  | El sistema permita editar y guardar la vista explorada                                     |
| <b>Actores:</b>      | Usuario  |
| <b>Precondición:</b> | <ul style="list-style-type: none"> <li>• La vista está registrada en el sistema</li> </ul> |

|                             |   |
|-----------------------------|---|
| <b>Escenario Principal:</b> | <ol style="list-style-type: none"> <li>1. El actor selecciona la opción: Guardar Vista</li> <li>2. El actor selecciona: Aceptar.</li> <li>3. El sistema guarda el informe explorado.</li> </ol> |
|-----------------------------|---|

Tabla 45: Especificación Caso de Uso de Editar Vista

## REQUISITOS NO FUNCIONALES

Para hacer uso de las opciones para construir soluciones analíticas avanzadas con Litebi, el usuario debe tener una sesión iniciada en el sistema con los respectivos privilegios asignados al rol que le corresponde a su usuario.

## GLOSARIO DE TÉRMINOS

- **Editor de Cubos y Dimensiones:** Permite definir los modelos analíticos en los que se cargaran los datos.
- **Cubo:** Elemento fundamental de liteSpace. Conjunto de datos organizado orientado al análisis, estructurado de forma dimensional. Los cubos están formados por **métricas** (measures) y **dimensiones**. *Ej. Cubo de ventas, Cubo de contabilidad, etc...*
- **Métrica:** Cantidades numéricas que deseamos controlar en un cubo. Es lo que aparece en las celdas al analizar un cubo. *Ej. Cantidad vendida, Cantidad facturada, Precio unitario, etc...*
- **Dimensión:** Punto de vista de la información que deseamos en el análisis. *Ej. Dimensión geográfica (¿donde sucedió la venta?), Dimensión Cliente (¿a quién le vendí?), Dimensión Producto (¿Qué vendí?).* Las dimensión pueden ser **compartidas** (entre varios cubos) o **embebidas** (pertenecientes a un sólo cubo).
- **Jerarquías:** Una dimensión puede tener una o varias jerarquías, una jerarquía es un conjunto de **niveles** cada uno expresando un nivel de profundidad en la información. Permiten analizar la información de forma agrupada (ventas por año) o al detalle (ventas diarias). *Ej. La dimensión tiempo está formada por la jerarquía "Por Trimestre", que tiene los niveles "Año > Trimestre > Mes > Día".*

- **Embebidas:** Son dimensiones que sólo existen para un cubo. Es habitual utilizarlas para definir propiedades (Ej. Estado de una factura) o simplemente para simplificar el modelado.
- **liteExplorer** es la herramienta de reporting avanzado OLAP que permite realizar consultas multidimensionales sobre los cubos construidos y almacenar informes.
- **liteMonitor** es la herramienta de cuadros de mando de Litebi, permite componer representaciones visuales integrales con diferentes puntos de vista. Es una herramienta de control.
- **Compartidas:** Indicamos que el cubo utilizará una dimensión compartida (definida aparte) y fijamos un alias para el uso. Es posible utilizar una misma dimensión en un cubo más de una vez (*ej: Fecha de venta, Fecha de entrega*).
- **Miembros calculados:** Permiten realizar cálculos avanzados sobre los objetos del cubo, utilizan el lenguaje MDX y es posible definir desde cálculos sencillos ( $\text{Facturación} = \text{Precio unitario} * \text{Cantidad vendida}$ ) hasta fórmulas complejas (Incrementos entre periodos de tiempo, porcentajes del total, y un amplio etc...). La edición de miembros calculados requiere de un conocimiento del lenguaje MDX y queda fuera del ámbito del presente manual.
- **Nivel base:** es el nivel de máximo detalle de la dimensión, el que posteriormente se utilizará para relacionar la dimensión con el cubo. *Ej. En una dimensión temporal el nivel base podría ser el día, si en el cubo tenemos la información a nivel de día.*
- **Datos:** Información de orígenes externos cargada a los modelos analíticos de Litebi (cubos y/o dimensiones).