



PONTIFICIA **UNIVERSIDAD CATÓLICA** DEL PERÚ

Esta obra ha sido publicada bajo la licencia Creative Commons
Reconocimiento-No comercial-Compartir bajo la misma licencia 2.5 Perú.

Para ver una copia de dicha licencia, visite
<http://creativecommons.org/licenses/by-nc-sa/2.5/pe/>



PONTIFICIA UNIVERSIDAD CATOLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



**CONSTRUCCIÓN DE UNA HERRAMIENTA DE DESARROLLO DE
SOLUCIONES PARA INTELIGENCIA DE NEGOCIOS - EXPLOTACIÓN**

Presentada por:

Sergio Daniel Falcón Cisneros

Rosanna Inés Palma Stanciuc

Raúl Alejandro Celi Peña

LIMA - PERU

2007

RESUMEN

Hoy en día, las Tecnologías de Información (TI) están presentes en la mayoría de las organizaciones en todo el mundo. Su uso se hace imprescindible para el manejo de grandes cantidades de datos, pudiéndose manejar millones de transacciones por segundo.

Esta enorme cantidad de datos que pueden procesar las organizaciones tiene un gran potencial para ser convertida en información útil y estratégica, mediante agregaciones y consolidaciones a través de las variables o dimensiones más importantes de análisis, la detección de patrones, y tendencias, etc. Todo esto permitirá a los directores de las organizaciones un panorama más claro y completo para la toma de decisiones.

Es en este punto en el cual se hace útil la Inteligencia de Negocios. En lugar de analizar y procesar manualmente todos estos datos, se utiliza un conjunto de metodologías y herramientas automatizadas que faciliten este trabajo y que permitan dar soporte a la necesidad de información de las organizaciones. Así, para implementar una solución de Inteligencia de Negocios (BI de sus siglas en inglés), se realiza la definición de las dimensiones más importantes de análisis para el *Data Warehouse*, se define el proceso de extracción, transformación y carga de los Datos a este *Data Warehouse*, y finalmente se tiene la explotación de los datos cargados a través de reportes y gráficos.

El presente proyecto de tesis busca implementar el módulo de explotación de una herramienta para desarrollo de aplicaciones de Inteligencia de Negocios integrales. Es decir, una aplicación que cubra las 3 fases del ciclo de desarrollo de un proyecto en BI. La arquitectura planteada permitirá que una organización provea el servicio de Inteligencia de Negocios.

“A mis padres, Sergio y Celmira, quienes me han dado desde siempre su apoyo incondicional y han inculcado en mí el deseo de superación y conocimientos. Y a Susy, mi gran maestra y amiga.”

Sergio Falcón

“Esta tesis quiero dedicársela a a mis padres, Victor y Viorica que dieron mucho para que yo llegue a ser la persona que soy hoy, y a mi madre María que fue mi maestra y compañera”

-Rosanna Palma

“Dedico este trabajo a mi familia que incondicionalmente de alguna manera siempre me ha apoyado y en especial a mis padres a quienes les tendré una eterna deuda por todo el esfuerzo que pusieron sobre mí. También dedico este trabajo a mis compañeros de tesis y a mi mismo por todo el empeño y dedicación que le dimos.”

-Raúl Celi

“A mis padres, Sergio y Celmira, quienes me han dado desde siempre su apoyo incondicional y han inculcado en mí el deseo de superación y conocimientos. Y a Susy, mi gran maestra y amiga.”

Sergio Falcón

"Doy gracias a Dios que me enseñó a perseverar y luchar en todos mis años de carrera, a todos los relacionados a este proyecto de tesis, y especialmente agradezco a Carla por su paciencia cada fin de semana y al ingeniero Abraham.”

-Rosanna Palma

“Agradezco a todos los profesores y asesores de tesis que estuvieron involucrados en el desarrollo de este trabajo por comprometerse enteramente con él y guiarnos hasta su culminación con el mismo interés que demostraron desde un inicio.”

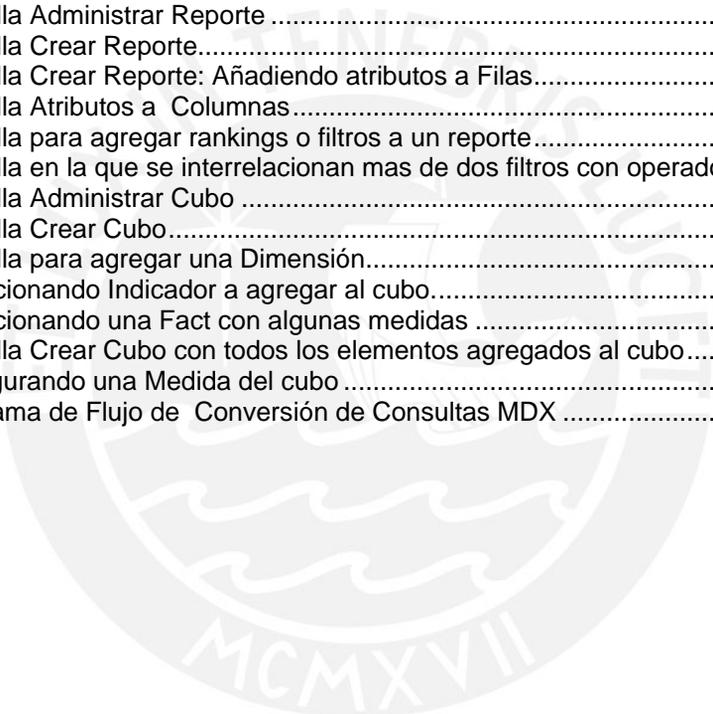
-Raúl Celi

INDICE

RESUMEN.....	II
INDICE.....	V
INDICE DE FIGURAS	VI
INDICE DE TABLAS Y CUADROS	VII
INTRODUCCIÓN.....	1
1. MARCO TEÓRICO	3
1.1. Definición del Problema	3
1.2. Objetivos propuestos	4
1.3. Solución Propuesta: Características de la Herramienta	4
1.4. Productos Existentes	5
2. Análisis de la Herramienta	9
2.1. Diagramas de Casos de Uso	9
2.1.1. Paquete de Cubos	10
2.1.2. Paquete de Reportes.....	14
2.2. Características de los Usuarios	18
2.2.1. Observador.....	18
2.2.2. Constructor	18
2.2.3. Administrador del Proyecto	18
2.3. Características del Entorno.....	19
2.3.1. Herramientas Usadas.....	19
3. Diseño de la Herramienta	25
3.1. Diagrama de clases.....	25
3.2. Diagramas de Secuencia	28
3.3. Diseño de la persistencia	32
3.4. Diagrama de Arquitectura	33
3.5. Diseño de Pantallas Principales.....	37
3.6. Algoritmos Principales.....	46
3.6.1. Conversión de Consultas Multidimensionales	47
4. Construcción de la Plataforma de Integración.....	49
4.1. Consideraciones Preliminares	49
4.2. Metodología de desarrollo.....	50
4.3. Estimación de Esfuerzo para el proyecto	51
4.4. Distribución de las tareas en el equipo	51
4.5. Consideraciones para la Implementación de la Herramienta	52
4.6. Pruebas de Desempeño	52
4.6.1. Casos de Prueba.....	53
4.6.2. Reporte de Pruebas	55
5. Observaciones, Conclusiones y recomendaciones.....	58
5.1. Observaciones	58
5.2. Conclusiones.....	59
5.3. Recomendaciones.....	60
ANEXOS.....	62

INDICE DE FIGURAS

Figura 2.1 Diagrama de casos de uso del paquete de Cubos	11
Figura 2.2 Caso de Uso del paquete de Reportes	15
Figura 2.3 Diagrama de herencia de Usuarios.....	19
Figura 3.1 Diagrama inicialmente planteado.....	26
Figura 3.2 Diagrama de clases del paquete de Reportes.....	27
Figura 3.3 Diagrama de Clases del paquete de Cubos	28
Figura 3.4 Administrar Reporte - Crear Reporte	29
Figura 3.5 Administrar Reporte - Eliminar Reporte	30
Figura 3.6 Administrar Cubo - Crear Cubo.....	31
Figura 3.7 Administrar Cubo - Eliminar Cubo.....	32
Figura 3.8 Diagrama de despliegue	35
Figura 3.9 Diagrama de componentes	37
Figura 3.10 Pantalla Administrar Reporte	39
Figura 3.11 Pantalla Crear Reporte.....	39
Figura 3.12 Pantalla Crear Reporte: Añadiendo atributos a Filas.....	40
Figura 3.13 Pantalla Atributos a Columnas.....	40
Figura 3.14 Pantalla para agregar rankings o filtros a un reporte.....	41
Figura 3.15 Pantalla en la que se interrelacionan mas de dos filtros con operadores lógicos	41
Figura 3.16 Pantalla Administrar Cubo	42
Figura 3.17 Pantalla Crear Cubo.....	43
Figura 3.18 Pantalla para agregar una Dimensión.....	43
Figura 3.19 Seleccionando Indicador a agregar al cubo.....	44
Figura 3.20 Seleccionando una Fact con algunas medidas	44
Figura 3.21 Pantalla Crear Cubo con todos los elementos agregados al cubo.....	45
Figura 3.22 Configurando una Medida del cubo	46
Figura 3.23 Diagrama de Flujo de Conversión de Consultas MDX	48



INDICE DE TABLAS Y CUADROS

Tabla 1.1 Comparación de productos existentes.....	8
Tabla 2.1 Motores de Base de Datos soportados actualmente.	24
Cuadro 3.1 Estructura de Archivo XML <Foodmart.xml>.....	33
Tabla 4.1 Distribución de Responsabilidades de Implementación de Casos de Uso.....	52
Tabla 4.2 Caso de Prueba del Caso de uso Administrar Reporte	54
Tabla 4.3 Caso de Prueba del Caso de uso Administrar Cubos.....	54
Tabla 4.4 Reporte de Pruebas de Caso de Uso	56
Tabla 4.5 Reporte de Pruebas de Caso de Uso de otras funcionalidades	57



INTRODUCCIÓN

Se dice que “la información es poder”, y no poder de autoridad y de opresión, sino poder para la toma de decisiones correctas en el momento oportuno. Día a día más empresas van descubriendo esto y consideran a la información como un activo muy valioso. Esta idea no es errónea del todo, si pensamos en que el manejo correcto de información, puede ofrecer a la empresa, entre otras ventajas, más competitividad. Justamente para cubrir esta necesidad de manejo de información, se van haciendo necesarias nuevas herramientas TI que permitan generar los reportes necesarios para los directores y gerentes de la empresa, y los usuarios de estas herramientas se vuelven cada vez más exigentes, conforme descubren el valor agregado que estas herramientas ofrecen.

Pensando en esta problemática, surgió la idea de este proyecto de tesis, y a lo largo de los capítulos de esta tesis, se describe el estudio y desarrollo realizado para implementar una herramienta de este tipo. Como se quiso que la herramienta a implementar en este proyecto de tesis abarcara todo el proceso de Análisis, Extracción y Explotación de información, usando la metodología de Inteligencia de Negocios, se dividió esta tesis en tres módulos; cada uno con el nombre de la funcionalidad de la herramienta. La documentación presentada a continuación corresponde propiamente, al módulo de Explotación de la herramienta.

En el primer capítulo de este documento, se presenta la definición de la problemática junto con los objetivos de este proyecto de tesis. Además se presenta un breve análisis de los productos actualmente ofrecidos en el mercado, que implementan características similares a los de esta herramienta.

En el segundo capítulo se presenta el análisis de la herramienta, modelando las estructuras correspondientes en lenguaje UML, como lenguaje estándar para la construcción de los artefactos de software. El artefacto de software elaborado para esta etapa fue el diagrama con los casos de uso. También se incluye en este capítulo las características del Entorno consideradas para esta herramienta.

Asimismo, en el tercer capítulo se presenta el diseño de la herramienta. También se modelaron las estructuras en el lenguaje UML. Los artefactos de software elaborados para este capítulo fueron: diagrama de clases, diagrama de secuencia, prototipo y diagrama de arquitectura.

El capítulo tres, también incluye una breve descripción del diseño de la persistencia que utiliza la herramienta OLAP usada por la aplicación desarrollada por este módulo de Explotación, junto con los algoritmos principales usados en la etapa de desarrollo.

En el cuarto capítulo, este documento presenta la implementación de la herramienta, indicando algunas consideraciones previas al desarrollo de esta, la metodología usada en el desarrollo y finalmente una estimación del esfuerzo usado en este proyecto y las pruebas de desempeño que se aplicaron a esta herramienta para asegurar su correcto funcionamiento.

Finalmente en el quinto capítulo se presentan las observaciones, conclusiones y recomendaciones.

1. MARCO TEÓRICO

En el primer capítulo, se presenta una breve definición del problema y objetivos planteados para este proyecto de tesis. Además se hace mención de algunos productos ofrecidos en el mercado actualmente y cuyas funcionalidades son similares en la herramienta de este proyecto.

1.1. Definición del Problema

Cada día las empresas deben enfrentar nuevos desafíos, como el aumentar su nivel de calidad, buscar mayor participación del mercado, lograr mayor satisfacción de sus clientes, mejorar su competitividad, etc. Enfrentar estos retos implica cubrir una mayor necesidad de información. Y aunque la estadística y las tecnologías de información se han desarrollado, generalmente no se cuenta con la información requerida al momento de tomar decisiones, dada la variabilidad y cantidad de esta. Esta dinámica, ligada a la necesidad de rentabilidad de las empresas, las impulsa a requerir información más precisa y de mayor calidad, así como diseñar sistemas que permitan recabarla y manejarla de mejor manera.

En este contexto es donde aparecen conceptos como el de Inteligencia de Negocios que apoyados en técnicas, estrategias, metodologías y herramientas buscan ofrecer información más adecuada para la toma de decisiones. Una adecuada decisión en una organización se traduce en mejoras significativas sea en productividad u otro beneficio. Una solución de Inteligencia de

Negocios puede incluso cambiar el rumbo de una organización hacia escenarios más favorables y beneficiosos.

Existen en el mundo diversas herramientas que apoyan la implementación de Inteligencia de Negocios, pero en el Perú por el alto costo que implican su adquisición, implementación y mantenimiento, son muy pocas las empresas que hacen uso de ellas, y como consecuencia pierden las ventajas competitivas que la Inteligencia de Negocios ofrece.

1.2. Objetivos propuestos

El objetivo principal de esta parte del proyecto es construir una herramienta de Explotación que pueda ser accedida mediante la Web y que permita conectar el flujo de trabajo entre las otras dos partes del proyecto, es decir entre el módulo de Análisis y el módulo de Extracción. El flujo de trabajo se describe a continuación:

Se define el esquema *datamart* en el cual se almacenarán los datos históricos. Luego, se hace uso del módulo de Extracción para construir los flujos de transformación que cargarán los datos en el *datamart*. Finalmente, una vez ejecutada la carga de datos, mediante el módulo de Explotación se podrá diseñar y obtener los reportes que mostrarán la información buscada.

Es importante notar que los datos de origen que serán cargados en el *datamart* pueden provenir de fuentes de datos heterogéneas. Esto hace que sea necesario pensar en una solución ETL que soporte el espectro de fuentes de datos existentes en el mercado en la actualidad, pero que también permita ser ampliada para no perder vigencia en el futuro.

Finalmente, es importante que la herramienta trabaje de forma eficiente para asegurar que los recursos del servidor sean aprovechados correctamente.

1.3. Solución Propuesta: Características de la Herramienta

Para este proyecto de tesis se ha definido un producto con las siguientes características:

La solución se construirá basada en la especificación J2EE e incorporará los siguientes sub-módulos.

- **Administración de Cubos:** En el cual se utilizará uno de los modelo de tipo *Online Analytical Processing* (OLAP) para la explotación de datos; por lo que este módulo

permitirá realizar la administración de los cubos y de todos sus componentes. Además esta herramienta permitirá la configuración de estos componentes, que serán usados como fuentes de los reportes.

- **Confección de Reportes:** El sistema permitirá realizar la explotación de los datos cargados al DataWarehouse; dicha explotación se realizará mediante reportes tabulares o gráficos, los cuales podrán ser exportados en diversos formatos.

1.4. Productos Existentes

En este punto se analiza algunos productos que cumplen características similares al de este proyecto de tesis. Se menciona algunas características generales respecto a la funcionalidad global de la herramienta, pero se pondrá más énfasis, en las características referentes a la explotación de datos.

Cognos 8 Business Intelligence

En enero del 2007 la empresa Cognos anunció la versión preliminar de Cognos 8 Business Intelligence. Esta herramienta ofrece un rango bastante completo de capacidades entre las descritas por la misma página Web oficial del producto **[COGNOS, 2007]**:

1. Generador de reportes, con las siguientes características:
 - ofreciendo una variedad de plantillas de reportes multilingües tanto para el desarrollo como para la publicación de los reportes, adaptables a cualquier conjunto de datos. Esto es, el usuario podrá diseñar el reporte una vez y ejecutarlo en cualquier idioma.
 - La plataforma Cognos provee herramientas de desarrollo y administración de reportes con la conveniencia del uso del Web (usando o no el Portal) como medio de distribución sin comprometer la calidad de presentación. También ofrece otros formatos como MSOffice, PDF, XML, HTML, y CSV.
2. Permite el Desarrollo de Consultas Dinámicas. Una vez definida la estructura de base de datos, los usuarios pueden ejecutar consultas predefinidas o construir nuevas con la ayuda del programa. La fuente de datos para estas consultas a realizar abarca bases de datos múltiples y heterogéneas.

3. Herramienta capacitada para el análisis de datos, que permite la organización lógica y dimensional de la estructura de los datos.
4. Permite el diseño de *balance scorecards* y *dashboards* que ofrecen bastante flexibilidad para la configuración y un software de optimización de *performance*. Estos son accesibles a través de un navegador.
5. Facilidades para la integración de datos, lo que mejora la limpieza y extracción y transformación de los datos a estructuras dimensionales.
6. Administrador de eventos de negocios. Ayudando en la configuración y automatización de procesos del negocio y de decisiones para disminuir el tiempo de acción frente a ellos.

Otras ventajas de la herramienta Cognos:

- Esta herramienta ofrece la posibilidad de exportar y combinar datos de sistemas tipo SAP a datos externos para los reportes.
- Permite acceso y gestión de la información de la información desde MS Office.
- Facilidad de creación de informes: El informe se adapta automáticamente cuando se añaden o se quitan objetos.
- Capacidad de añadir aplicaciones multimedia (video y flash) sitios Web, o cualquier contenido externo a Bussiness Intelligence.

Oracle Bussiness Intelligence

Oracle ofrece también una herramienta que provee la tecnología para la construcción de una solución completa de Inteligencia de Negocios, haciendo uso de la base de los datos Oracle Database 10g. Oracle ofrece herramientas para realizar el proceso de extracción, transformación y carga (ETL), OLAP, *DataMining*, agregados al servidor de datos. Esta herramienta permite el rápido desarrollo y creación de varios datawarehouses y data Marts con un traductor de consultas multidimensionales, generador de reportes, aplicación para hacer análisis, alertas del negocio, administración e integración de datos [ORACLE, 2006].

Según el mismo fabricante, las funcionalidades que este producto ofrece se dividen por aplicaciones, las más resaltantes y de interés para evaluar en este proyecto de tesis son:

Oracle Ware House Builder

- Permite la extracción, transformación y carga de la data (ETL)
- Consolidación de los datos de diferentes y variadas fuentes de datos
- Migración de los datos desde sistemas legados.
- Permite el modelado de estructuras dimensionales y relacionales
- Diseño y administración de los datos de la organización
- Limpieza de los datos, para permitir información de calidad.
- Permite clasificar y auditar la calidad de los datos.

Oracle XML Publisher

- Permite crear formatos de reportes, usando Microsoft Word, Adobe Acrobat y otras herramientas conocidas por los usuarios.
- Acepta o estructura cualquier dato en XML, desde una hasta varias fuentes de datos y crea a partir de ellos un solo documento.
- Permite enviar los datos ya sea a impresión, mail, Fax, o Portal Web. También se puede personalizar algún otro tipo de canal de envío de datos.
- Las plantillas de los reportes pueden ser creados en alguno de los 185 lenguajes que Oracle ofrece. Cada traducción del documento forma un archivo separado, pero la plantilla se modifica una sola vez.
- Esta herramienta es una aplicación open source en Java, lo que facilita la integración con otras aplicaciones Java.

Microsoft SQL Server 2005

Según el mismo fabricante, SQL Server 2005, siguiendo los pasos de su antecesor SQL Server 2000, ofrece una solución integrada de administración y análisis de datos que ayuda a las organizaciones de cualquier magnitud a realizar diversas funciones entre estas: [Microsoft, **2005**]

- Crear, desplegar y administrar aplicaciones empresariales seguras, escalables y confiables.
- Maximizar la productividad de Tecnologías de Información mediante la reducción de la complejidad y el soporte de aplicaciones de bases de datos.
- Compartir datos en múltiples plataformas, aplicaciones y dispositivos para facilitar la conexión de sistemas internos y externos.
- Controlar los costes sin sacrificar el rendimiento, la disponibilidad, la escalabilidad o la seguridad.

Entre las herramientas de administración que SQL Server 2005 ofrece, existen

- **Integration Services (Servicios de Integración)**, que permite la integración de datos desde distintos tipos de fuentes, facilitando de esta manera una visión integral del negocio.
- **Analysis Services (Servicios de Análisis)**, permite el análisis de los datos de manera integral, preparando la estructura necesaria para los reportes, análisis OLAP, datamining y desarrollo de scorecards.
- **Reporting Services (Servicios de Reportes)**, permite la administración y el desarrollo de reportes, ya sea a partir de formatos predefinidos o de manera personalizada, con la posibilidad de ser mostrados tanto en PDF, en Web o con alguna otra herramienta Microsoft. La herramienta desarrollada por nuestro módulo de Explotación incluye algunas de las funcionalidades de estos servicios.

En la tabla 1.1, se observa un cuadro, que compara las funciones principales entre los productos antes expuestos:

Características	Cognos	SQL Server 2005 - Analysis Services	Oracle Business Intelligence
Funciones Integradas de Business Intelligence.	Extracción, transformación y Carga (ETL)	Extracción , transformación y Carga	Extracción, transformación y Carga
Formatos para exportar reportes posibles	Web, XML, PDF, CSV MS Word y Excel	Microsoft Word (.doc) y PDF.	XML, PDF y Word
Herramientas de Consulta Dinámicas	Sí entre ellos Wizards y optimización de Queries.	Si	No menciona
Herramientas extras para reportes	Soporta Reportes Multilingües y añade reportes prediseñados.	Integración con Microsoft Office	Soporta Reportes Multilingües(185 idiomas)
Permite integración de varias fuentes de datos	Si, incluyendo las usadas por sistemas SAP	Si, usando los "Sistemas de Integración"	Si, incluyendo las usadas por fuentes legadas

Tabla 1.1 Comparación de productos existentes

2. Análisis de la Herramienta

En este capítulo se presenta el análisis de la plataforma. Además se realiza una discusión sobre la evolución del producto presentado por el módulo de Explotación desde una experiencia previa en el curso de Desarrollo de Programas 1 (DP1).

2.1. Diagramas de Casos de Uso

En esta sección se exponen los principales casos de uso que se deben desarrollar para satisfacer los requerimientos principales planteados para la herramienta, descritos en la Lista de Exigencias (Anexo B).

Estos requisitos principales, para el caso del módulo de Explotación, exigen que la herramienta desarrollada por este proyecto de tesis, incluya entre sus funcionalidades lo siguiente:

- La generación y configuración de reportes gráficos y tabulares. En estos reportes también se podrían agregar filtros y rankings para seleccionar mejor la información.
- Poder armar un cubo a partir de los elementos creados previamente en el módulo de Análisis.

Los diagramas presentados en la documentación de este proyecto de tesis, fueron desarrollados

usando el lenguaje de modelamiento UML, para estandarizar la forma de presentación de la información.

Cada módulo de este proyecto de tesis, ya sea Análisis, Extracción o Explotación, agrupó los casos de uso en paquetes, de acuerdo a la afinidad respecto del tipo de funcionalidad que implementan en la herramienta. De manera que se logre, una visualización más consistente y una mayor comprensión.

La especificación detallada de los casos de uso se encuentra en el documento de Especificación de Requisitos de Software (ERS), Anexo C. En este documento sólo presentaremos los principales de cada paquete:

2.1.1. Paquete de Cubos

Este paquete es el que contiene todos los casos de uso relacionados a la creación, edición y configuración de los Cubos, como se observa en la figura 2.1. El caso de uso principal de este paquete es “Administrar Cubo”.

En la primera fase del proyecto, en el curso de Desarrollo de Programas 1, se tenía un caso de uso similar, pero en esta fase no existía una integración de los elementos usados entre los tres módulos del proyecto. Es decir, no había una integración entre los objetos del módulo de Explotación y Análisis, de tal manera que se pudieran usar y configurar elementos del cubo de un módulo a otro. Por el contrario, las estructuras de las clases de ambos módulos diferían lo suficiente como para permitir la integración.

En esta segunda fase del proyecto, sin embargo, para este caso de uso el usuario ya tiene la posibilidad de diseñar los Cubos usando elementos previamente creados y configurados en el módulo de Análisis.

Para cada elemento del Cubo el usuario puede además configurar otros parámetros necesarios para las consultas multidimensionales y para la explotación de los datos. Esta nueva opción ofrece al analista una personalización más específica de las características de cada elemento que componen un Cubo.

La descripción de los casos de uso de este paquete y su funcionalidad es como sigue:

Sugerir Cubos: Permite al usuario crear los cubos de manera automática a partir de la metadata física del modelo de datos.

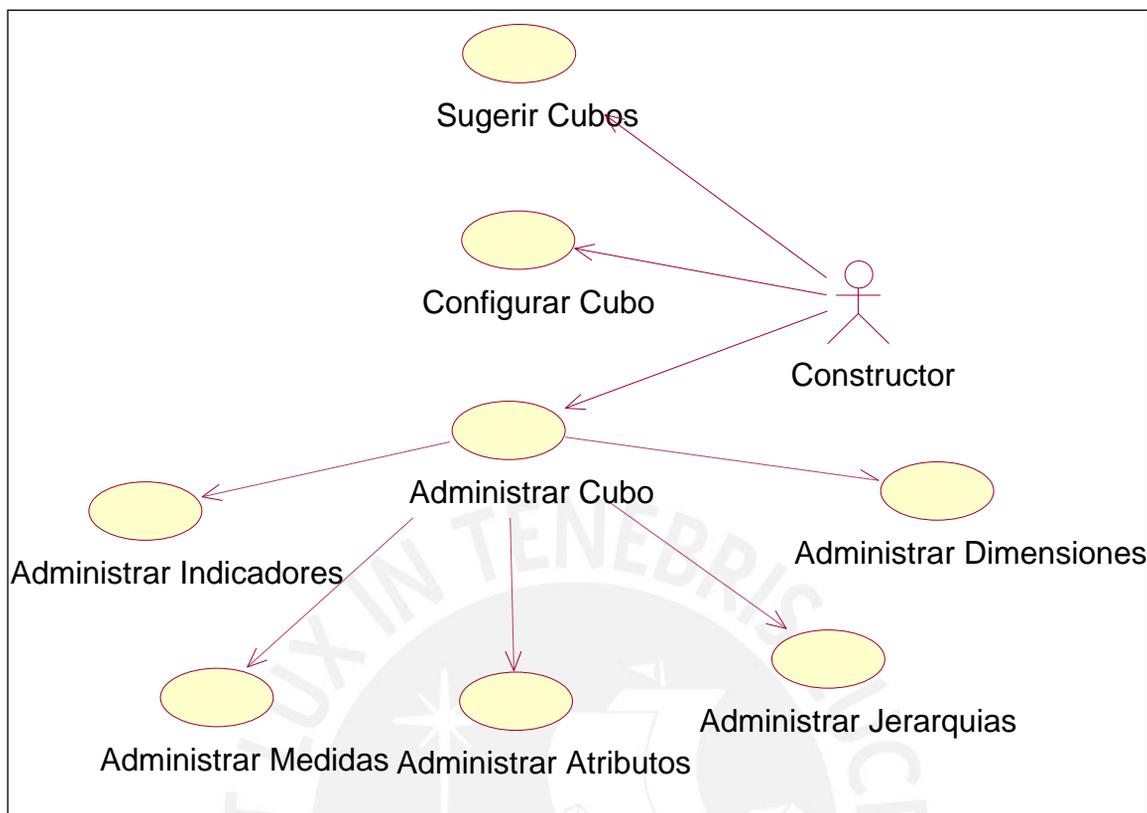


Figura 2.1 Diagrama de casos de uso del paquete de Cubos

Configurar Cubos: Permite al usuario configurar los parámetros de uso del Cubo para el motor OLAP.

Administrar Cubo: Permite al usuario crear y buscar los cubos del proyecto.

Administrar Dimensiones: Permite al usuario configurar las dimensiones del proyecto.

Administrar Jerarquías: Permite al usuario crear y administrar las jerarquías del proyecto.

Administrar Indicadores: Permite al usuario crear y administrar los indicadores el proyecto.

Administrar Medidas: Permite al usuario crear y administrar las medidas del proyecto.

Administrar Atributos: Permite al usuario crear y administrar los atributos del proyecto.

La especificación del caso de uso principal “Administrar Cubo, se detalla a continuación:

Caso de Uso Administrar Cubos

Permite al usuario crear, configurar, editar y eliminar Cubos para un tema de análisis.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

El caso de uso se inicia con el usuario.

1. El usuario selecciona la opción “Crear Cubo”.
2. El usuario selecciona un tema al cual pertenecerá el Cubo luego de lo cual, el sistema muestra los botones para la creación de indicadores, dimensiones, dimensiones virtuales, Facts que pertenecerán al cubo.
3. El usuario ingresa el nombre y la descripción del Cubo.
4. Para seleccionar la Fact Table del Cubo, el usuario deberá hacer clic en “Agregar Fact”.
5. El sistema le mostrará en un formulario con la toda la lista de Facts elegibles de acuerdo al tema escogido.
6. El usuario selecciona una Fact y hace clic en aceptar.
7. A partir de este formulario el usuario podrá agregar dimensiones (ya sean virtuales o no), jerarquías e indicadores. Para esto debe hacer clic en el botón del elemento que se quiera agregar.
8. El sistema presentará para cada caso un formulario que muestra las dimensiones compartidas, jerarquías, atributos, medidas, e indicadores existentes en el tema de análisis.
9. El usuario puede crear, modificar, y eliminar dimensiones compartidas, atributos, jerarquías, e indicadores desde este formulario siempre que el usuario cuente con los permisos necesarios.
10. <<include>> Administrar Dimensiones, Administrar Atributos, Administrar Jerarquías, Administrar Medidas, Administrar Indicadores.

11. El usuario selecciona Guardar.
12. El sistema le muestra al usuario un mensaje de confirmación si el cubo se guardó correctamente o no y el caso de uso termina.

1.2. Flujos Alternativos

i. Modificar Cubo

1. El usuario selecciona un Cubo y selecciona la opción Modificar.
2. El sistema presenta un formulario que muestra los datos del Cubo: nombre, descripción, y dimensiones compartidas, jerarquías, atributos, medidas e indicadores que lo componen.
3. El usuario puede cambiar el nombre y la descripción del Cubo.
4. El usuario puede crear, modificar, y eliminar atributos, jerarquías, e indicadores para el Tema de análisis, y dimensiones compartidas siempre que cuente con los permisos necesarios.
5. El usuario puede agregar o quitar dimensiones compartidas, jerarquías, atributos, medidas e indicadores del cubo.
6. El usuario selecciona Guardar.
7. El sistema le muestra al usuario un mensaje de confirmación y el caso de uso termina.

ii. Eliminar Cubo

1. El usuario selecciona un Cubo y selecciona la opción Eliminar.
2. El sistema solicita una confirmación de la eliminación.
3. El usuario acepta la confirmación y el caso de uso termina.

iii. Cancelar

El usuario puede cancelar en cualquier momento las operaciones realizadas, deshaciendo todos los cambios hechos.

2. Requerimiento Especiales

Ninguno

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Tema de análisis creado: El usuario debe haber creado un tema de análisis previamente.

4. **Post condiciones**

Ninguna.

5. **Puntos de extensión**

Ninguno.

2.1.2. Paquete de Reportes

Este paquete encierra todos los casos de uso relacionados a la administración de los reportes y de los elementos que los componen. El diagrama con estos casos de uso son presentados en la Figura 2.2.

La descripción de los casos de uso y su funcionalidad es como sigue:

Administrar Filtros: Permite al usuario, crear, modificar y eliminar los filtros de los datos de los reportes.

Administrar Rankings: Permite al usuario administrar los rankings de los reportes.

Administrar Reportes: Administra los reportes y sus componentes.

Ver Reportes: El usuario puede explotar la información y ver los reportes previamente diseñados.

Exportar Reportes: Este caso de uso es el encargado de la exportación de los reportes en el formato deseado.

El caso de Uso principal de este paquete es “Administrar Reporte”.

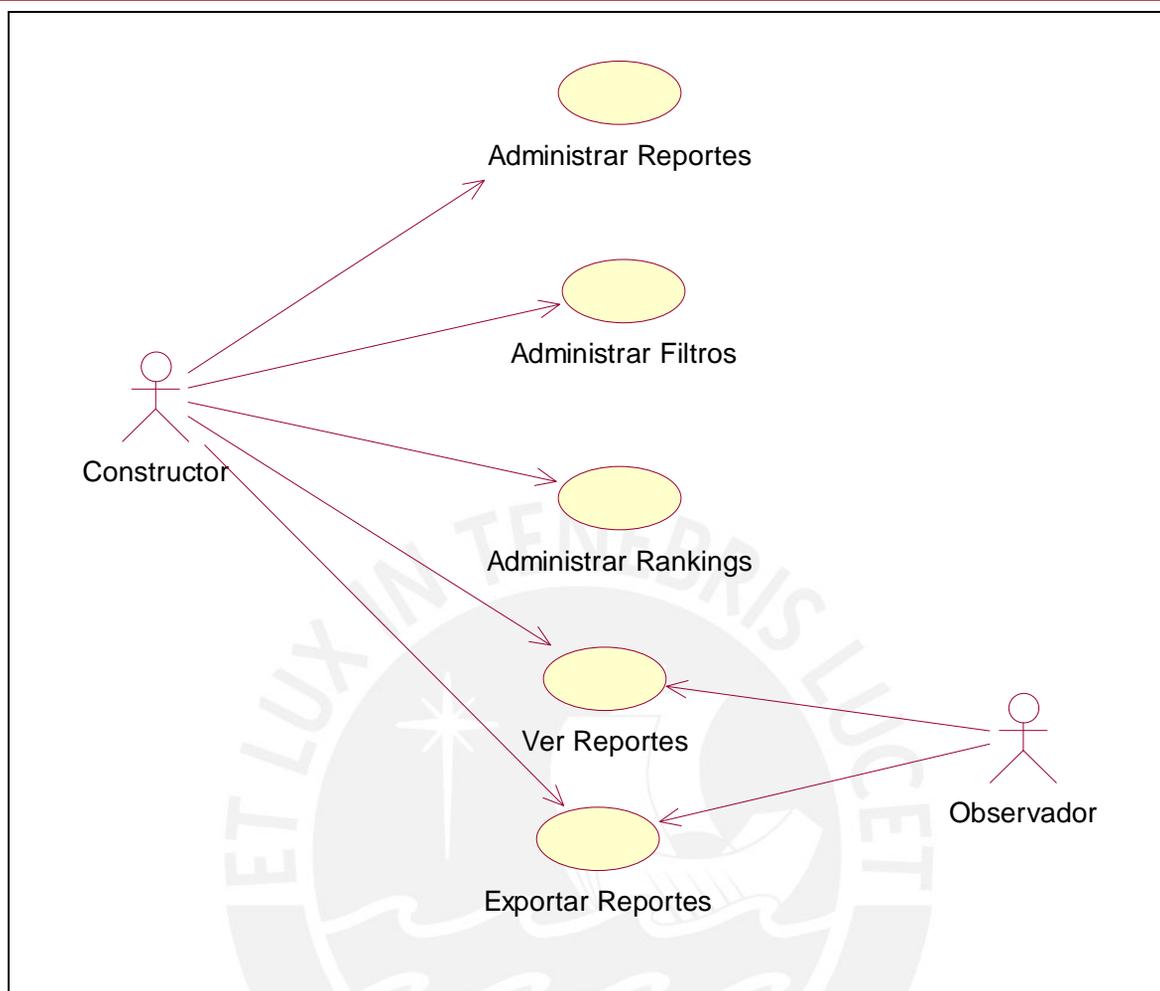


Figura 2.2 Caso de Uso del paquete de Reportes

En la primera fase del proyecto, en el curso de Desarrollo de Programas 1, este caso de uso tenía por nombre “Administrar Informe”, y tenía la misma finalidad que tiene actualmente. Es decir, este caso de uso debe ofrecer una herramienta gráfica que permita ver los reportes generados a partir de de una fuente de datos y una consulta multidimensional dada por el usuario.

En esta segunda fase del proyecto, a este caso de uso se le añadió la opción de exportar los reportes en diferentes formatos, entre ellos: Excel, JPG, PDF, etc.

La especificación de este caso de uso, es como sigue:

Caso de Uso Administrar Reporte.

Permite al usuario configurar los reportes del proyecto.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

El caso de uso se inicia con el usuario.

1. El usuario selecciona un Tema de análisis y selecciona la opción “Crear Reporte”.
2. El sistema muestra el formulario de Especificación de Reportes.
3. El usuario elige un Cubo del Tema de análisis desde el cual crear el reporte.
4. El sistema muestra un formulario que muestra las medidas, jerarquías e indicadores del cubo. Además muestra los filtros y rankings creados en el proyecto.
5. El usuario selecciona las jerarquías, indicadores y medidas que desea agregar al reporte y la posición fila-columna-página en la cual colocará cada elemento.
6. El usuario selecciona los filtros y rankings que desea aplicar al reporte.
7. El usuario selecciona la opción “Guardar Reporte”.
8. El usuario ingresa un nombre, selecciona Acepta, y el caso de uso termina.

1.2. Flujos Alternativos

i. Ver Código Multidimensional (MDX) usado en el Reporte

1. El usuario selecciona el botón “Ver MDX” en la barra de herramientas del reporte.
2. El sistema muestra la consulta MDX usada para el reporte.

ii. Modificar Reporte

1. El usuario selecciona un reporte y selecciona la opción “Modificar Reporte”.
2. El sistema muestra el formulario de Especificación de Reportes, con los campos mencionados en el flujo principal.
3. El usuario puede agregar o eliminar elementos de la grilla del reporte. Además, puede cambiar la posición de los elementos existentes en la grilla.

4. El usuario selecciona la opción “Guardar Cambios”.
5. El sistema guarda los cambios efectuados.

iii. *Eliminar Reporte*

1. El usuario selecciona la opción “Eliminar Reporte” sobre un Reporte.
2. El sistema solicita una confirmación de la eliminación.
3. El usuario acepta la confirmación, y el caso de uso termina.

iv. *Presentación tabular y grafica de promedios, mínimos, máximos, curva de ajuste*

1. El usuario selecciona la opción “Presentación gráfica de cálculos” sobre un Reporte.
2. El usuario selecciona los elementos que desea presentar gráficamente, ya sea promedios, mínimos, máximos, o mediante una curva de ajuste.
3. El usuario ingresa los parámetros necesarios y selecciona aceptar.
4. El sistema muestra un formulario de confirmación.
5. El usuario selecciona la opción “Sí”.
6. El sistema muestra el gráfico correspondiente
7. El caso de uso termina.

v. *Crear cálculos (totales, sumas contadores y promedios mínimo y máximos)*

1. El usuario selecciona la opción “Crear Cálculos.”
2. El usuario selecciona si desea crear totales, sumas, contadores, promedios, mínimos o máximos, sobre un Reporte.
3. El usuario ingresa los valores límites, o según lo que haya elegido.
4. El sistema muestra un formulario de confirmación.
5. El usuario selecciona la opción “Sí”.
6. El sistema crea el cálculo en el reporte.
7. El caso de uso termina.

2. **Requerimientos Especiales**

Ninguno

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Proyecto Abierto: El usuario debe haber abierto un proyecto.

4. Post condiciones

Ninguna.

5. Puntos de extensión

Ninguno.

2.2. Características de los Usuarios

Dentro del módulo de Explotación, los usuarios tendrán la capacidad de ejercer diversos roles que les darán distintos permisos para el manejo de la herramienta de este módulo, entre los cuales tenemos:

2.2.1. Observador

Este usuario tiene sólo la opción de consultar los reportes, creados por otros usuarios. No tendrá la opción de crear nuevos objetos, o en todo caso, de realizar modificación a alguno de ellos.

Se puede decir, que este usuario, es el usuario final de la aplicación desarrollada por este módulo.

2.2.2. Constructor

Se le denomina a la persona cuya tarea es la de diseñar y editar los reportes y cubos del proyecto, con los objetos que ha diseñado el Administrador del Proyecto previamente. Este usuario usará un esquema de objetos desarrollado con la herramienta del módulo de Análisis.

Dado que su tarea es el diseño de reportes, es importante que este usuario tenga pleno conocimiento de la lógica del negocio y los requerimientos del cliente para maximizar la satisfacción de las necesidades de este.

2.2.3. Administrador del Proyecto

Este es el usuario con más privilegios en el sistema. Es el encargado de diseñar las jerarquías, programar los indicadores, cargar las dimensiones de la base de datos, crear campos y especificar los hechos con los que trabaja el sistema. Además este usuario tiene la responsabilidad de configurar los elementos que vaya creando, personalizando así cada elemento dentro del esquema XML del cubo a crear.

Esta personalización es en especial necesaria, para realizar las consultas multidimensionales del cubo por parte del servidor OLAP.

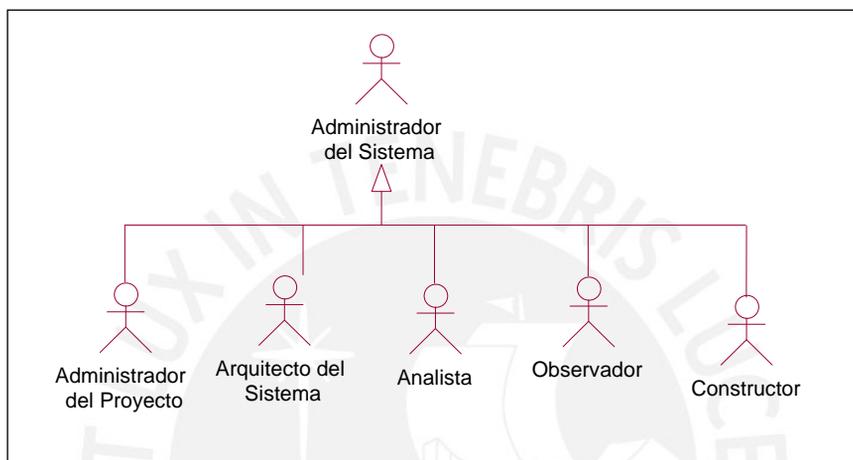


Figura 2.3 Diagrama de herencia de Usuarios

2.3. Características del Entorno

En este punto se describe el entorno de ejecución de la herramienta, es decir, los elementos con los cuales interactúa tanto en el ambiente de desarrollo como en el de producción.

2.3.1. Herramientas Usadas

Aquí se describe las herramientas usadas tanto para el desarrollo de la aplicación, como para las pruebas y la ejecución de esta. Estas fueron definidas por los integrantes del proyecto. Ver anexo A.

2.3.1.1. En la Etapa de Desarrollo

Durante el desarrollo de la aplicación se utilizaron las siguientes herramientas:

J2SDK (Java 2 Software Development kit)

Esta implementación contiene el entorno de desarrollo de código Java de Sun. Brinda las clases básicas necesarias para empezar a programar, y además las funciones necesarias para poder compilar y ejecutar el programa en tiempo de desarrollo.

La versión utilizada en esta implementación fue la 1.5, por las diversas ventajas que ofrecía esta versión con respecto a versiones anteriores, las cuales son:

- Esta versión contiene muchas más clases que las que existían en otras versiones. Estas clases contienen además nuevos métodos, ofreciendo al equipo de desarrollo, muchas más opciones en la programación, como por ejemplo: Facilidades para recorrer arreglos. Esto aceleraría, en el proyecto, el recorrido de los arreglos con los objetos propios de cada módulo.
- Finalmente, la más importante ventaja de esta versión, fue el uso de tipos de objetos para las listas o colecciones. De esta forma, el equipo de desarrollo podía conocer en tiempo de compilación, la clase o el tipo de dato de cada elemento contenido en una lista, dando más orden, y evitando hacer transformaciones entre tipos de dato de los objetos. El factor orden era especialmente importante en este proyecto, ya que al realizar un proyecto con el número de personas que este tiene, se necesitaba tener muy presente el tipo de dato que se manejaba y a que módulo pertenecía.

Eclipse 3.1

Herramienta de desarrollo para código Java. Se usó para escribir, compilar, depurar las clases Java y diseñar las páginas Web. La ventaja que trajo haber usado esta herramienta, fue el uso de los *plug-ins* libres en el mercado para el manejo del *framework* JSF, entre ellos algunos que fueron bastante útiles para la parte gráfica de la herramienta desarrollada, como por ejemplo Amateras Server Faces. Eclipse, además provee facilidades para programar de una manera fácil y rápida.

Al inicio del proyecto se realizó un análisis para comparar las ventajas de otras herramientas de desarrollo Java. Se evaluaron las herramientas NetBeans 4.1 y JDeveloper 10g, y se encontraron diversas limitaciones en comparación con el Eclipse, como la falta de plugins para el desarrollo en el caso de Netbeans. Además estas dos últimas eran herramientas mucho más pesadas de cargar en una computadora promedio, como en las que se desarrolló la aplicación.

Java Server Faces

JSF (Java Server Faces) es un framework de desarrollo basado en el patrón MVC, (Modelo Vista Controlador). [ORACLE, 2005] Este patrón permite separar la lógica de control, la lógica de negocio y la lógica de presentación. Es por esto que se eligió usarlo en el desarrollo de la aplicación, ya que es más sencillo y ordenado que JSP, ofrece un mejor mantenimiento, una mayor normalización, y estandarización en al código de la aplicación del módulo de Explotación. Otra ventaja y razón por la que se eligió este framework, es que existen diversas especificaciones para este que se pueden encontrar en distintos fabricantes. Esto permite elegir la más adecuada al proyecto de acuerdo al número de componentes que proporciona, rendimiento, soporte, etc.

En el caso de este módulo de Explotación, se usó la implementación ofrecida por el proyecto de Apache: MyFaces. Esta es una implementación de JSF de Software Libre que, además de cumplir con el estándar, también proporciona algunos componentes adicionales, junto con una constante validación de las clases y funciones JSP [MyFaces, 2007].

Mondrian

Mondrian es un servidor OLAP escrito en java que permite la traducción de Expresiones Multidimensionales (MDX) a SQL. Fue elegido por este módulo por ser una herramienta Java gratuita bastante eficiente con respecto al manejo de grandes cantidades de información [Pentaho, 2006]. Además ofrece el uso de una gran variedad de conectividad con diversos sistemas de Base de Datos, lo que enriquecía la herramienta propuesta a construirse.

El modelo de servidor OLAP utilizado por Mondrian, es un modelo HOLAP u OLAP híbrido, el cual trabaja sobre bases de datos relacionales y además permite utilizar tablas de agregaciones con datos precalculados de manera similar a como trabaja un modelo MOLAP.

JPivot

Para el desarrollo de de los casos de uso de reportes del módulo de explotación, se utilizó la librería JPivot, que es una librería de código “open-source”, personalizada, de etiquetas JSP, las cuales brindan una tabla OLAP que permite a los usuarios realizar navegaciones típicas, como por ejemplo “slice and dice” y “drill-down” [JPIVOT, 2006]

Esta librería utiliza Mondrian como su servidor OLAP.

Apache Tomcat 5.5.12

En todas las etapas del proyecto, se usó esta herramienta como servidor de aplicaciones Web.

Esta implementación es muy usada en la actualidad en los servidores Linux, y fue presentada al público bajo las licencia de Apache Software Licence. [Apache 2006]

Se eligió este servidor porque presenta una buena compatibilidad con aplicaciones Java, como la nuestra y además presenta más estabilidad y es de fácil configuración.

Dado que Tomcat fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java (JVM).

Internet Explorer (versión 6 y 7)

El Navegador Web que se usó en la etapa de desarrollo y en la etapa de pruebas fue Internet Explorer, dado que cuenta con una consola de java que viene por defecto, lo que facilita la instalación de la herramienta construida en el presente proyecto de tesis. Esta consola de java resultó muy importante sobre todo en la etapa de pruebas para ubicar los puntos de error más rápidamente.

2.3.1.2. En la Etapa de Pruebas

Las herramientas usadas en el entorno de ejecución de la misma son las siguientes:

Mozilla FireFox

Para poder realizar las pruebas correspondientes de la aplicación desarrollada por el módulo de Explotación en el sistema operativo LINUX, se decidió usar este navegador de páginas Web. (Aunque este navegador también se encuentra disponible para Windows actualmente).

Dado que este navegador está ganando cada vez más adeptos en el mercado y haciéndose más popular, y es bastante diferente a Internet Explorer, consideramos importante usarlo en la etapa de pruebas también.

Java 2 Runtime Enviroment

En esta etapa se usó el J2RE (Java 2 Runtime Enviroment), ya que tiene integrados entre los módulos que trae el programa, una maquina Virtual Java (JVM), necesaria para la ejecución de las pruebas, de la aplicación propuesta por el módulo de Explotación.

También se usó J2RE, por la portabilidad del programa, es decir, la herramienta J2RE no tiene dependencia ni del hardware del sistema operativo.

De esta manera se logró probar la aplicación del módulo de Explotación, en un entorno muy parecido al real, usando los navegadores **Internet Explorer** en Windows y **Mozilla FireFox** en LINUX.

2.3.1.3. Entorno de Ejecución

Para lograr la ejecución adecuada de la herramienta, se recomienda tomar todas las medidas de seguridad necesarias en el servidor Web donde se instalará esta, dado que los datos a manejar en la aplicación son bastante importantes para el negocio. Sin embargo, estas medidas de seguridad no deben de ninguna forma, limitar el acceso a los usuarios reales a la aplicación.

Para lograr dicho objetivo, puede ser necesario usar firewalls y una intranet bien manejada en lo que a medidas de seguridad y administración de usuarios se refiera.

Otro punto importante a ser tomado en cuenta es, que el servidor en el cual se piensa instalar y ejecutar la aplicación de Explotación, esté disponible en las horas en las que se tiene planificado el acceso de usuarios. Como la aplicación está desarrollada en un entorno Web, y los usuarios pueden requerir acceder tanto nacional como internacionalmente, (dependiendo del rubro de la empresa) puede que el servidor necesite ser accedido las 24 horas del día, lo que aumentaría los gastos de mantenimiento e incluso las medidas de seguridad en este.

Para ejecutar la aplicación, el servidor Web Apache Tomcat, puede ser instalado un sistema operativo Windows o Linux.

La aplicación de este proyecto de tesis ha sido probada exitosamente en los motores de base de datos mostrados en la tabla 2.1. Se puede aumentar esta lista de manera bastante sencilla, dado que los tres módulos que integran este proyecto han desarrollado sus aplicaciones en herramientas que ofrecen una gran compatibilidad con otros motores de base de datos y el lenguaje de programación usado en el desarrollo, orientado a objetos facilita la tarea.

Motor de Base de datos ¹	Versión
Microsoft SQL Server	7.0
Oracle	8
MySQL	5.0
Microsoft Access	11.66 SP2

Tabla 2.1 Motores de Base de Datos soportados actualmente.



¹ Microsoft SQL Server es un producto y marca registrada de Microsoft Inc., Oracle es un producto y marca registrada de Oracle Corp., MySQL es un producto y marca registrada de MySQL AB, Access es una marca registrada de Microsoft Corporation.

3. Diseño de la Herramienta

En este capítulo se presentan puntos referidos al diseño de la herramienta, tales como diagrama de clases, de secuencia y de arquitectura. Además de otros temas relacionados al diseño.

3.1. Diagrama de clases

Los diagramas de clases aquí presentados fueron desarrollados en el lenguaje UML, al igual que en todo el proyecto, para estandarizar la presentación de estos.

Durante la primera parte del proyecto, basándonos en el esquema de clases definidos en el curso Desarrollo de Programas 1 (DP1), se planteó el siguiente diagrama de clases mostrado en la figura 3.1.

En la segunda parte del proyecto, cuando se reformuló la herramienta desarrollada en el curso de Desarrollo de Programas 1 para convertirlos en una aplicación WEB, se reformó e integró todo el proyecto, se determinó que el modelo debería ser modificado para soportar las funcionalidades de los otros módulos (Análisis y Extracción). Los inconvenientes encontrados, se presentan a continuación:

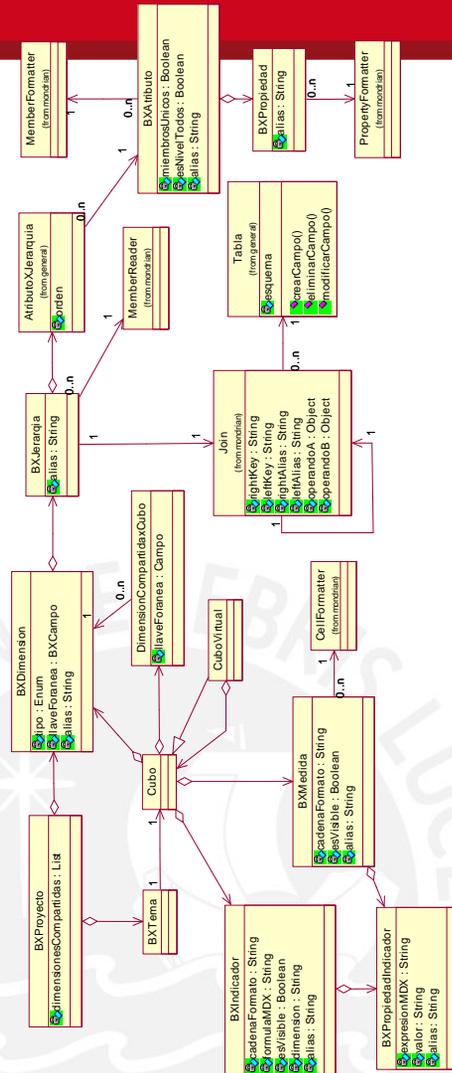


Figura 3.1 Diagrama inicialmente planteado

Según el estándar planteado en el Anexo I, los elementos de las clases del módulo de Explotación llevarían el prefijo BX-. Las clases correspondientes a los elementos del cubo, como: BXDimension, BXIndicador, BXAtributo, BXMedida, etc, si bien heredaban de las clases utilizadas por el módulo de Análisis, eran en si diferentes instancias de un mismo objeto, entre el módulo de Análisis y el de Explotación.

Para ser más claros en este punto se expone el siguiente ejemplo:

Se tiene un objeto perteneciente a la clase "Indicador" creado en el módulo de Análisis y se configura de acuerdo a las necesidades de este módulo. Posteriormente este objeto es tomado por el usuario en el módulo de Explotación. Según el diagrama de clases inicialmente planteado, con este objeto, que tiene las mismas características definidas en el módulo de Análisis, se instanciaba otro objeto llamado BXIndicador (Clase heredada de la clase Indicador de Análisis). Luego, el usuario configuraba en este segundo, características requeridas por el módulo de

explotación, entre ellas las necesarias para generar el archivo XML de definición del cubo para el servidor OLAP Mondrian. Si el módulo de Análisis actualizaba el objeto posteriormente, al tener ahora una instancia para cada módulo, se generarían problemas de sincronización de contenido. De tal manera que se podía tener un mismo objeto con diferentes características, al integrar ambos módulos. Este problema se repetía para todos los objetos compartidos por los módulos de Análisis y de Explotación.

Otro punto corregido se refiere al uso de las interfaces Mondrian, que inicialmente el diagrama de clases proponía (MemberFormatter, MemberReader, etc.). Estas interfaces no fueron finalmente utilizadas en el desarrollo de la herramienta final, porque descubrimos posteriormente que éstas solo son útiles para extender la funcionalidad básica de Mondrian, cosa que no era necesario para este proyecto de tesis. Por lo tanto las retiramos para reducir la complejidad.

Habiendo detectado los problemas anteriormente expuestos, se propuso inicialmente en crear funcionalidades en la herramienta misma, que permitieran una sincronización entre los módulos de Análisis y Explotación, realizada de manera manual por el usuario. Pero no se encontró en esta decisión tampoco una respuesta adecuada al problema.

Por tal razón se desarrolló un segundo modelo de clases para ambos, presentado en la figura 3.2 y 3.3. En la figura 3.2 se ven las clases del paquete de Reportes y en la 3.3 las del paquete de Cubos.

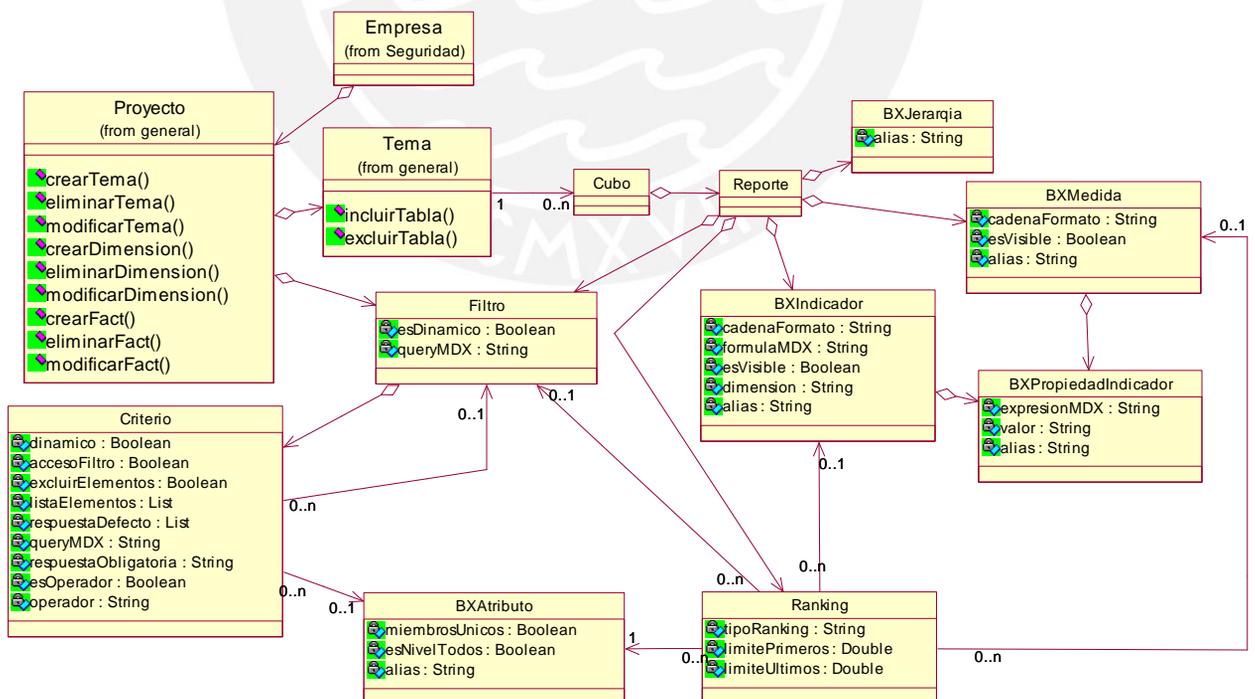


Figura 3.2 Diagrama de clases del paquete de Reportes

El flujo Crear Reporte del caso de uso Administrar Reporte se presentan en las figuras 3.4, y el flujo Eliminar Reporte en la figura 3.5.

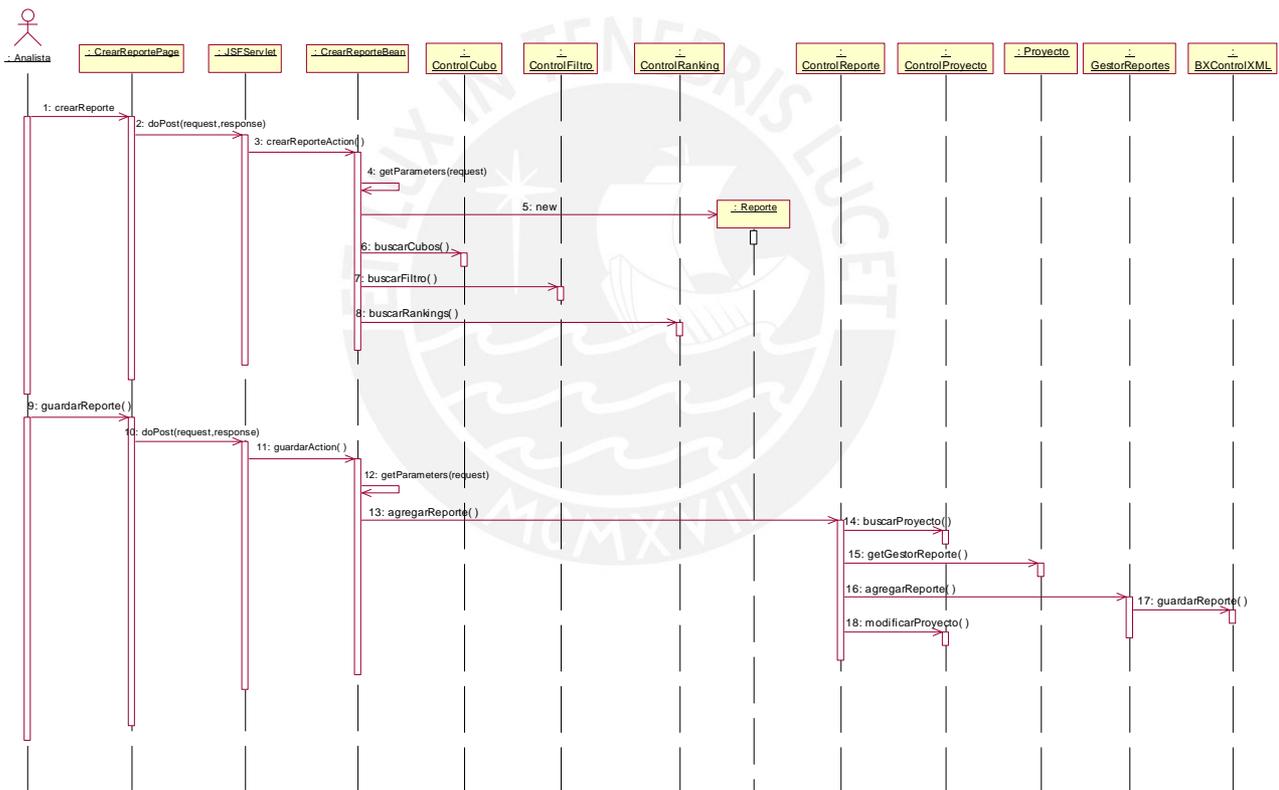


Figura 3.4 Administrar Reporte - Crear Reporte

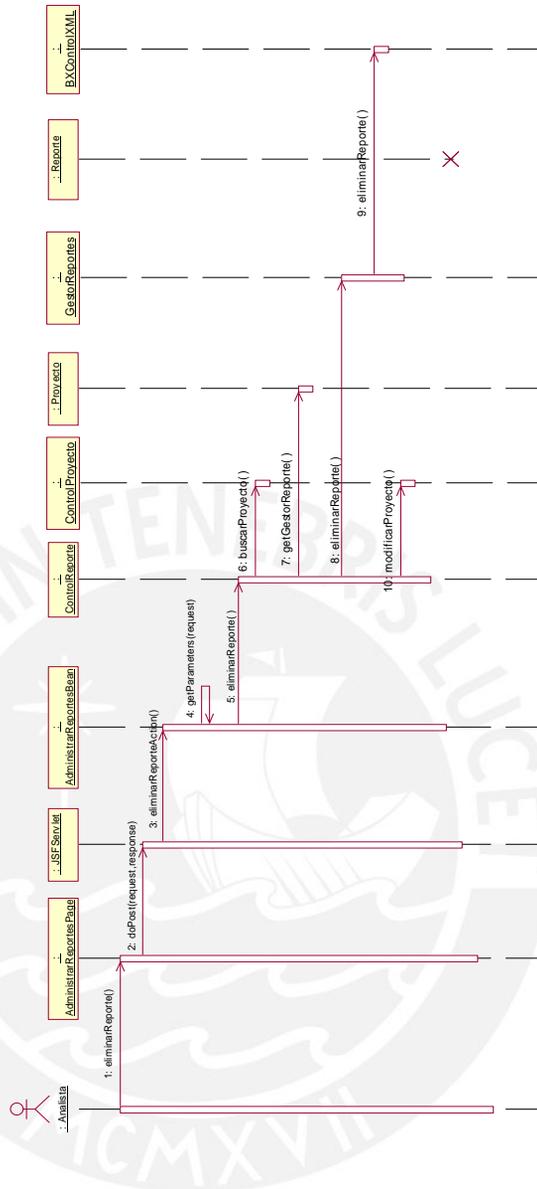


Figura 3.5 Administrar Reporte - Eliminar Reporte

Tanto en los diagramas presentados en la figura 3.4 como en la 3.5, podemos notar el uso de la clase BXControlXML, que es la encargada de la creación, modificación y eliminación de los archivos XML correspondientes a cada Reporte que se cree.

Los diagramas de secuencia del caso de uso Administrar Cubo en cambio se presentan en las figuras 3.6 y 3.7. En la figura 3.6 se muestra el flujo de creación y el flujo de eliminación en la figura 3.7.

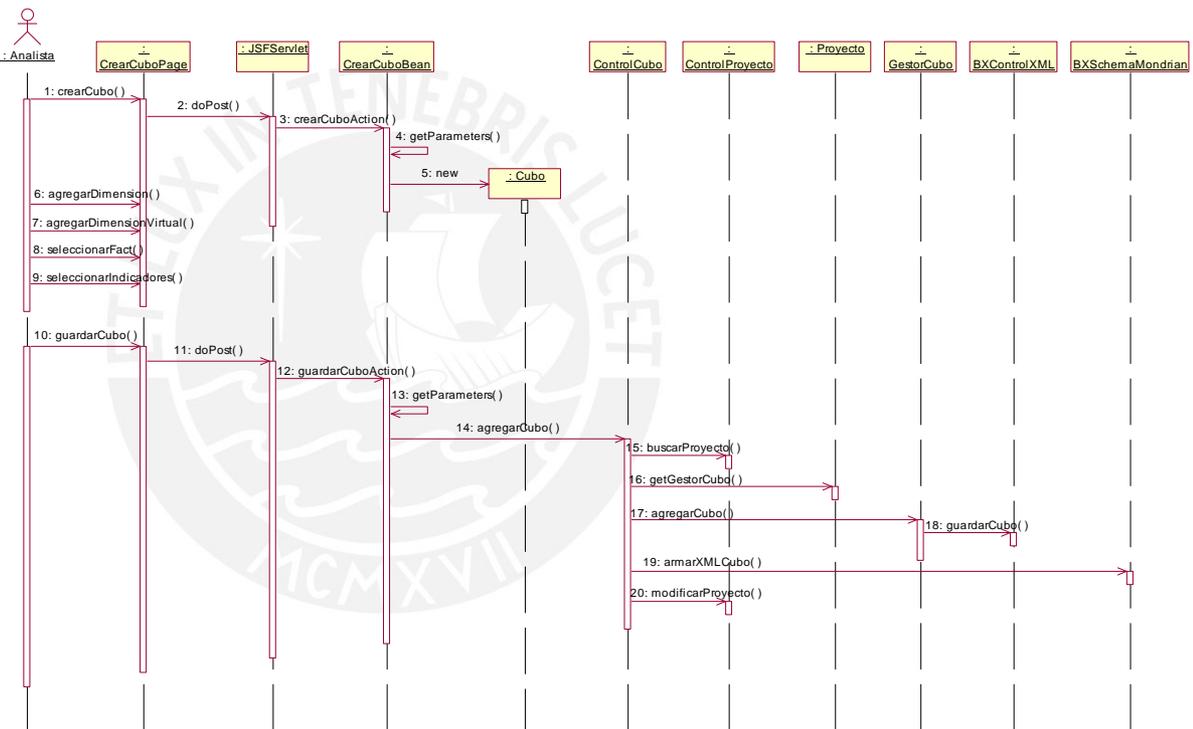


Figura 3.6 Administrar Cubo - Crear Cubo

En el diagrama de secuencias del caso de uso Crear Cubo, se puede observar el uso de una nueva clase no mostrada en los demás casos de uso llamada BXSchemaMondrian. Esta clase es la encargada de la creación del esquema XML Mondrian, a partir de un cubo y sus componentes. Este archivo XML a crearse en especial tiene un formato apropiado para las consultas multidimensionales a ser procesadas por Mondrian, herramienta usada por este proyecto de tesis.

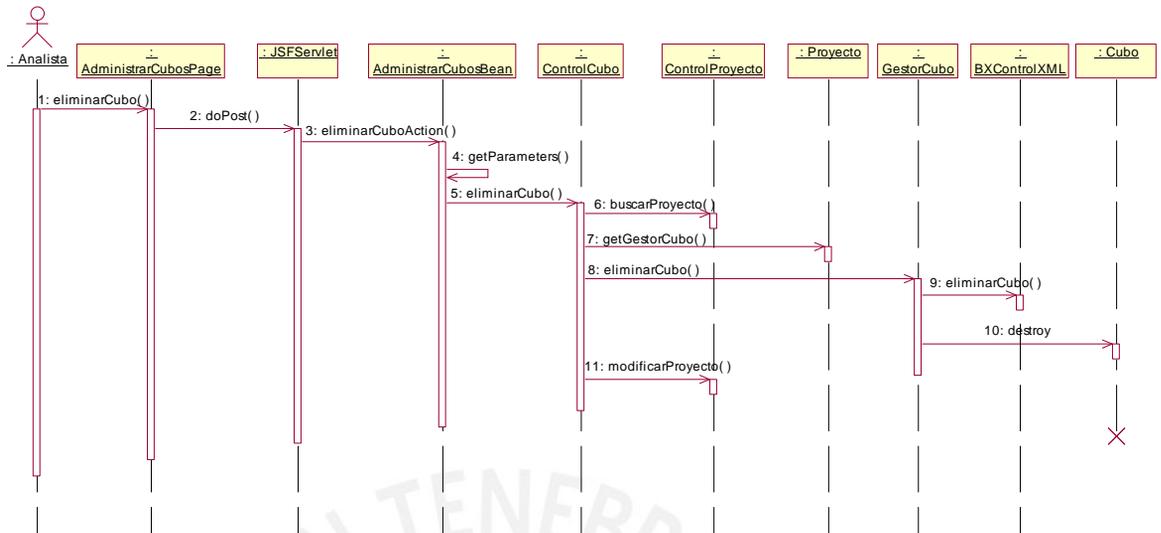


Figura 3.7 Administrar Cubo - Eliminar Cubo

Igual que en el caso del caso de uso Administrar Reporte, se decidió usar la clase BXControlXML como traductora y constructora de los archivos XML a partir de las clases de cada objeto a usarse dentro del cubo.

Finalmente cabe mencionar que se decidió modelar los diagramas de secuencia de manera detallada, incluyendo las clases necesarias para el la interfaz JSF y los objetos que se van creando durante los procesos; esto para facilitar la estructura del código en la etapa de desarrollo del proyecto.

3.3. Diseño de la persistencia

En el caso de la persistencia de los objetos llamados Cubos, el servidor Mondrian, usado en la aplicación del módulo de Explotación, requería el uso de archivos con formato XML, con sus propias estructuras y estándares que la misma herramienta Mondrian denomina “esquema Mondrian”.

Para cada cubo se generan dos archivos XML. Un archivo propio que es la utiliza la herramienta de este módulo de Explotación y el archivo usado por mondrian cuya estructura es similar al Cuadro 3.1. El archivo está compuesto por los elementos que conforman en si la estructura de un cubo de DataWareHouse.

```

<Schema>
  <Cube name="Sales">
    <Table name="sales_fact_1997"/>
    <Dimension name="Gender" foreignKey="customer_id">
    <Hierarchy hasAll="true" allMemberName="All Genders" primaryKey="customer_id">
    <Table name="customer"/>
    <Level name="Gender" column="gender" uniqueMembers="true"/>
    </Hierarchy>
    </Dimension>
    <Dimension name="Time" foreignKey="time_id">
    <Hierarchy hasAll="false" primaryKey="time_id">
    <Table name="time_by_day"/>
    <Level name="Year" column="the_year" type="Numeric" uniqueMembers="true"/>
    <Level name="Quarter" column="quarter" uniqueMembers="false"/>
    <Level name="Month" column="month_of_year" type="Numeric" uniqueMembers="false"/>
    </Hierarchy>
    </Dimension>
    <Measure name="Unit Sales" column="unit_sales" aggregator="sum" formatString="#,###"/>
    <Measure name="Store Sales" column="store_sales" aggregator="sum" formatString="#,###.##"/>
    <CalculatedMember name="Profit" dimension="Measures" formula="[Measures].
    [Store Sales]-[Measures].[Store Cost]">
    <CalculatedMemberProperty name="FORMAT_STRING" value="$#,##0.00"/>
    </CalculatedMember>
  </Cube>
</Schema>

```

Cuadro 3.1 Estructura de Archivo XML <Foodmart.xml>

El esquema Mondrian contiene un modelo físico compuesto por la estructura del cubo en sí: jerarquías, dimensiones y sus atributos. Además tiene un modelo lógico que, consiste en las construcciones usadas para escribir las consultas en lenguaje multidimensional (MDX). El modelo físico describe la fuente de datos del modelo lógico [PENTAHO, 2006].

3.4. Diagrama de Arquitectura

Uno de los desarrollos más importantes dentro de la construcción del software ha sido el desarrollo de su arquitectura, que permite representar la estructura de un sistema a un nivel mayor que el dado por la programación o incluso el diseño. [Boasson, 95]

Para definir la arquitectura del proyecto, lo que se planteó primero fue cual era el objetivo de esta herramienta y el como debía ser accedida. Se plantearon entonces las siguientes características en la arquitectura de la aplicación desarrollada por el módulo de Explotación para satisfacer este primer punto.

Plataforma Web

Dado que se quería acceder a la aplicación creada por este proyecto de tesis, desde cualquier sitio y de manera remota, se concluyó que era aconsejable desarrollarla en plataforma Web.

Uno de los objetivos principales de la aplicación desarrollada por el presente proyecto de tesis es el poder explotar datos haciendo uso de reportes personalizables; esto implicaba usar una plataforma Web y buscar herramientas de desarrollo que permitieran los elementos gráficos con la calidad y versatilidad necesaria. Justamente se eligieron por eso las herramientas descritas en el punto 2.3 del presente documento (Entorno del Sistema).

Entre las ventajas de usar una plataforma Web para este proyecto de tesis se incluyen:

- Los clientes pueden crear, configurar los reportes y los elementos que los conforman desde cualquier lugar y en cualquier computadora que tenga acceso a Internet y compatibilidad con JAVA.
- Además de estos dos requisitos de software, no se necesita configurar ningún software extra en la máquina del cliente, lo que a su vez disminuye los costos de hardware de la empresa.
- Para realizar cualquier actualización y mantenimiento de software, el cliente solo necesitaría conectarse a la Web, evitando de esta manera los costos de reinstalación en los equipos clientes.
- Es por eso que al cliente de esta plataforma se le llama también: cliente ligero, porque el cliente solo se encarga de la presentación visual de la herramienta así como algunas validaciones para no recargar el servidor.

El servidor central es el que realiza toda la lógica del negocio de la herramienta integral de este proyecto de tesis.

Repositorio de Datos

Los objetos generados por la aplicación, tales como Indicadores, Medidas, Dimensiones, etc, no se guardarán en un servidor de base de datos sino en un repositorio de archivos XML. En el caso particular de los cubos se guardarán dos archivos:

- Uno con la información de los elementos que forman el cubo.
- Otro con la información necesaria para las consultas MDX.

Se decidió conveniente usar dos archivos, ya que el primero podía ser usado internamente por el módulo de Explotación; y el segundo podía ser usado por el servidor OLAP para las consultas

multidimensionales. La razón por la que se decidió usar dos archivos, fue que ambos formatos eran diferentes.

Comunicación del Nodo Cliente con el Servidor

La comunicación con el cliente se realiza a través de mensajes sobre protocolo de hipertexto (http) los cuales son transmitidos a través de Internet. El Cliente podrá acceder con este protocolo haciendo uso de cualquier navegador para Internet existente en el mercado. (Particularmente Se han realizado pruebas exitosas usando el Internet Explorer en Windows y Mozilla FireFox tanto en Windows como en Linux. Se decidió usar este tipo de comunicación en el proyecto ya que era el protocolo usado por la mayoría de navegadores actualmente.

El diagrama de despliegue para la herramienta de los tres módulos se puede describir a continuación en la figura 3.8

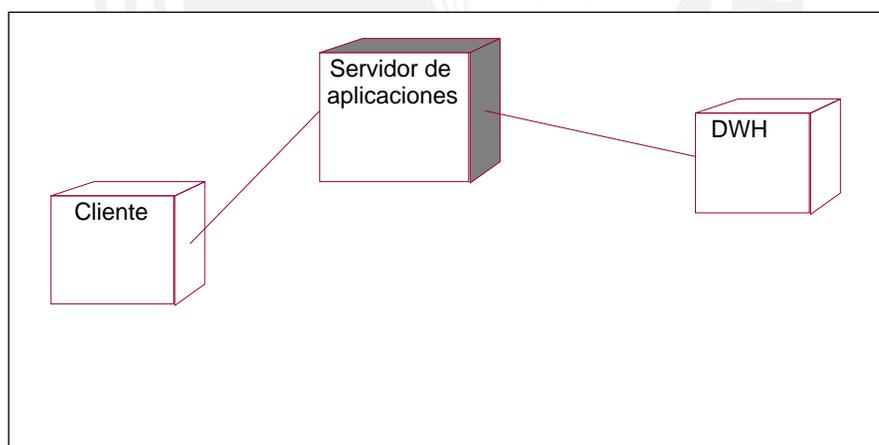


Figura 3.8 Diagrama de despliegue

Para entender mejor cada nodo de este diagrama de despliegue indicamos para cada uno lo siguiente:

Cliente: Representa el equipo remoto desde el cual el usuario accede a la herramienta haciendo uso de un navegador, desde el cual realizará la creación y configuración tanto de los cubos como de los reportes.

Servidor de Aplicaciones: Este nodo representa el elemento clave donde se realizan todos los procesos centrales para la configuración de los elementos del cubo, además de creación y administración de los reportes, junto con las otras funcionalidades que la herramienta permite

DWH (DataWareHouse) Base de datos relacional en la cual se almacenan los datos y que sirve como fuentes para los reportes.

La organización y la relación de los componentes que forman la herramienta Axebit en el diagrama de componentes en la figura 3.9 se describe a continuación:

Los principales componentes son los que corresponden a cada módulo de la herramienta, es decir los componentes de Análisis, Extracción y Explotación. Adicionalmente se cuenta con el componente Control, el cual se encarga de las funcionalidades de seguridad y manejo de sesiones, necesario para poder controlar el acceso de los usuarios a las funcionalidades de los 3 módulos. Los módulos de Análisis y Extracción utilizan un applet como pantalla principal para interacción con el usuario, y cuentan con un área de dibujo para lo cual comparten muchas clases comunes, las cuales corresponden al componente de Gráficos. El componente Común es el que contiene todas las clases comunes a todos los módulos. En este componente se encuentran las clases que representan empresas, proyectos temas de análisis, etc.

Finalmente, los componentes Mondrian y JPivot son utilizados por el módulo de Explotación para interactuar con la base de datos y presentar gráficos y reportes tabulares. En el Anexo F, Documento de Arquitectura, se encuentra una descripción más detallada de la arquitectura, de los componentes y la organización interna de cada uno de ellos.

La ventaja de presentar el diagrama de despliegue de esta manera, es que nos permite separar en controles diferentes a cada uno de los módulos que integran este proyecto de tesis y los controles que cada uno de los módulos usa. Además se indican las relaciones que entre todos estos existe. Para el caso del módulo de Explotación, el diagrama de despliegue muestra claramente como este módulo usa de manera independiente el código de las herramientas Mondrian y JPivot.

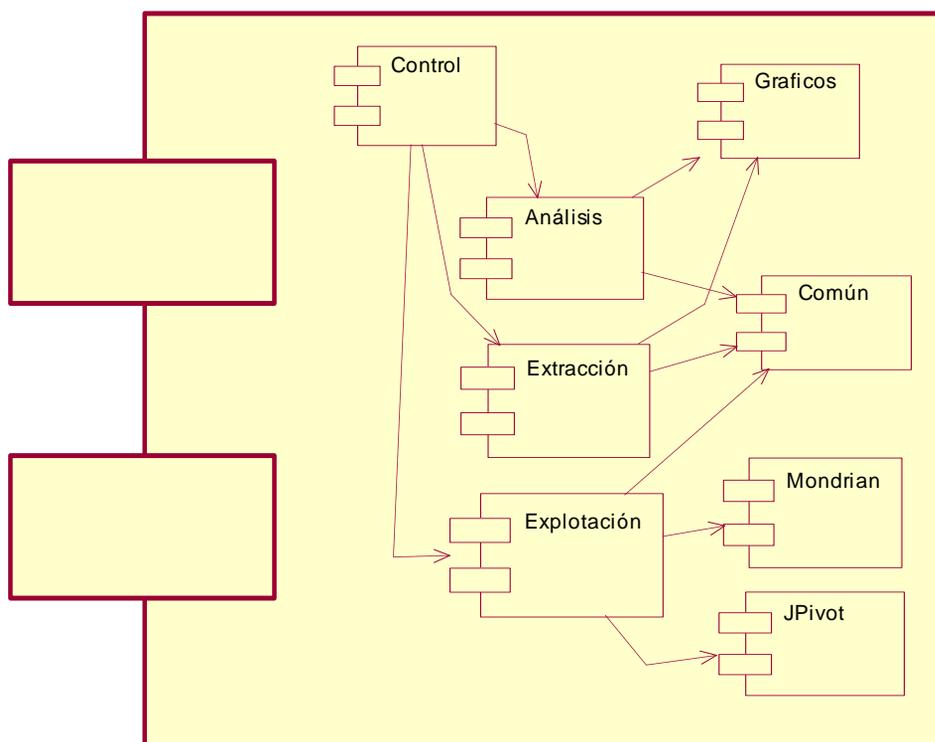


Figura 3.9 Diagrama de componentes

3.5. Diseño de Pantallas Principales

La funcionalidad del módulo de Explotación se concentra principalmente en la administración de reportes y de los cubos. Ambos se crean y configuran a partir de elementos diseñados en los otros módulos.

Para cumplir ambas funcionalidades, en especial la primera, de manera satisfactoria, se necesitaban incluir interfaces bastante complejas para ser desarrolladas en Web, por lo que el diseño de las pantallas de la herramienta, se veía inicialmente limitado. Posteriormente se vieron superadas estas limitaciones con la ayuda de las herramientas descritas en el punto 2.3.

Las pantallas representativas de este módulo son las correspondientes a su vez también con los casos de uso principales indicados inicialmente en el punto 2.1 implementadas en el *framework* JSF. Las pantallas de reportes sin embargo usaron además la herramienta gráfica JPivot (mencionada en el punto 2.1 de este documento)

En todo este proyecto de tesis, para el caso de las pantallas de la herramienta Web a desarrollarse, se tomaron en cuenta las siguientes características:

- Para unificar el estilo de las pantallas entre los tres módulos y facilitar el mantenimiento se usaron las mismas hojas de estilo o *cascading style sheets* (CSS). Un cambio en la hoja de estilos se verá reflejado en todas las páginas inmediatamente.
- Los colores elegidos en las pantallas son colores pasteles para evitar el agotamiento visual luego de usar la herramienta por tiempo prolongado.
- Se hizo uso de archivos *properties* o *bundle* para obtener el texto que se muestra en las páginas. De esta forma se puede establecer un archivo *properties* por idioma y permitir que se seleccione dinámicamente el archivo de idioma apropiado según la configuración del explorador Web que accede a la herramienta. Esto permite que la herramienta soporte múltiples idiomas con un mínimo esfuerzo, pues no hay que modificar las páginas *JSP*, sino solamente definir nuevos archivos *properties* para los idiomas adicionales.

Una descripción más detallada de las características de las pantallas de la herramienta se puede ver en el Documento de Diseño de Pantallas, anexo J.

Para el caso de uso de Administrar Reporte se pensó hacer las pantallas lo más intuitivas posibles, ya que con ellas el usuario podía realizar la explotación de la data.

Para este caso en particular, el usuario elegiría los elementos a usar en las columnas y en las filas. Además en las otras viñetas, el usuario podrá elegir otras configuraciones del reporte, así como también los filtros y rankings a aplicar en este. Con todos estos elementos se formará la consulta multidimensional, que será procesada por JPivot para mostrar los reportes en el caso de uso “Ver Reporte”.

Pantallas usadas para crear, modificar y eliminar el reporte:

La pantalla mostrada en la figura 3.10 ofrece como ventaja que a partir de esta se puede buscar un reporte previamente creado. Y una vez encontrado este, puede ser incluso eliminado, seleccionando el reporte y haciendo clic en Eliminar. Además se puede acceder a un reporte anteriormente creado para modificarlo, haciendo clic en el nombre del reporte.

Para crear un reporte nuevo, el usuario solo tendrá que hacer clic en el botón Crear, después de lo cual aparecerá la siguiente la pantalla mostrada en la figura 3.11.

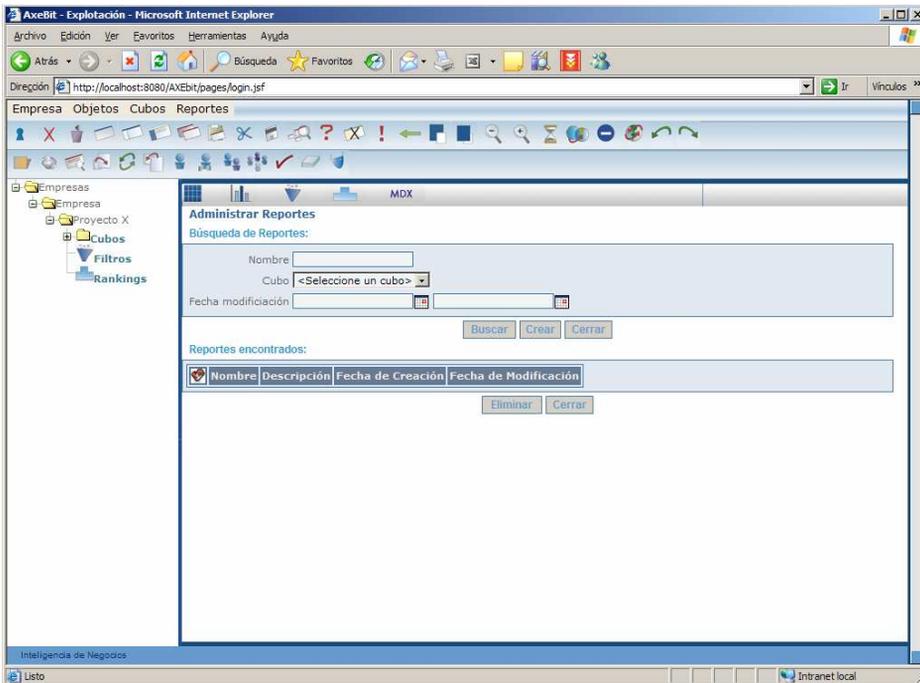


Figura 3.10 Pantalla Administrar Reporte

En la pantalla mostrada en la figura 3.11 se irá seleccionando una a una las jerarquías o dimensiones a agregar en las fila, columnas del reporte.

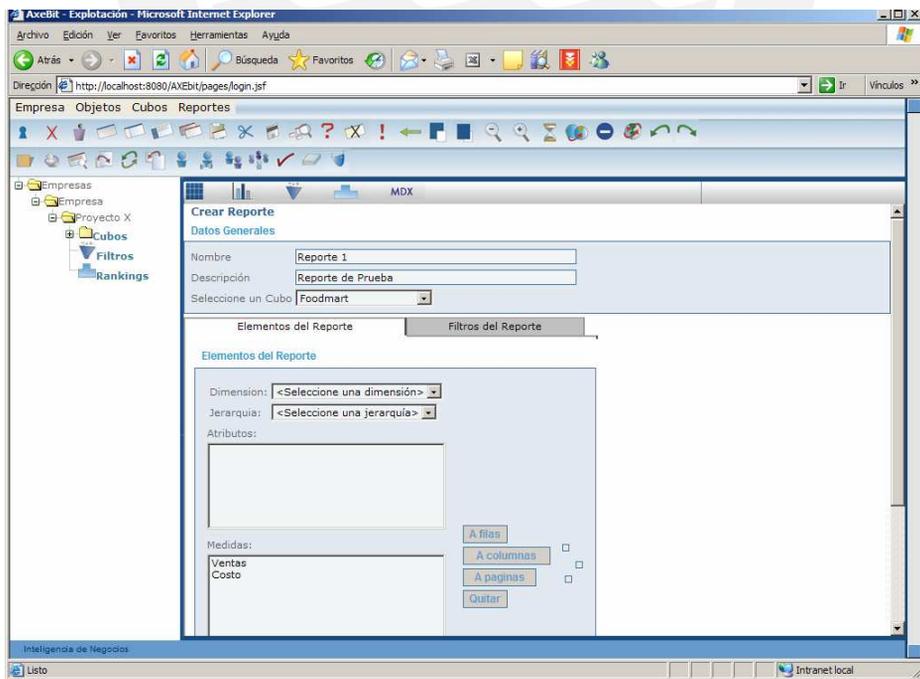


Figura 3.11 Pantalla Crear Reporte

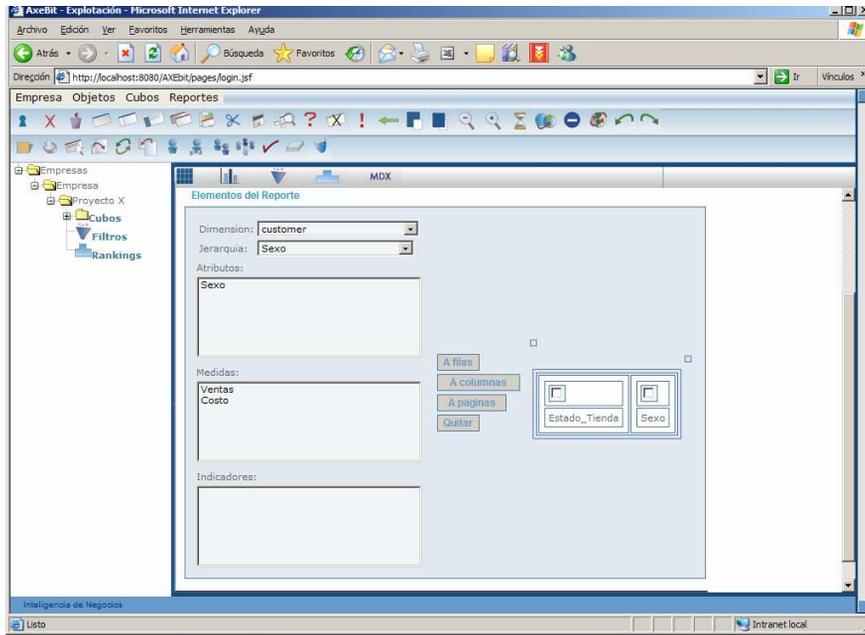


Figura 3.12 Pantalla Crear Reporte: Añadiendo atributos a Filas

En la figura 3.12 se muestra un ejemplo de cómo se añade uno o más atributos a las filas del reporte. De la misma manera en la figura 3.13 el ejemplo se aplica para agregar atributos a las columnas del reporte.

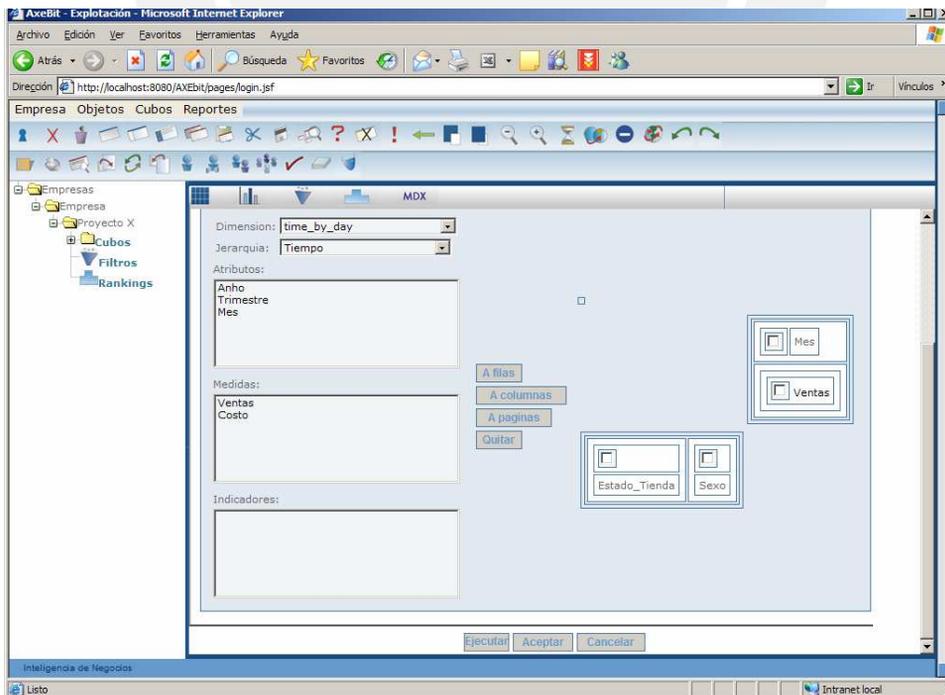


Figura 3.13 Pantalla Atributos a Columnas

El caso de uso Administrar Reporte también permite el agregar rankings o filtros al reporte, el mismo al que se le definió inicialmente sólo los atributos en las filas y columnas, como se muestra en la figura 3.14.

En la figura 3.15, además, se muestra como se pueden definir operadores lógicos para relacionar más de un filtro.

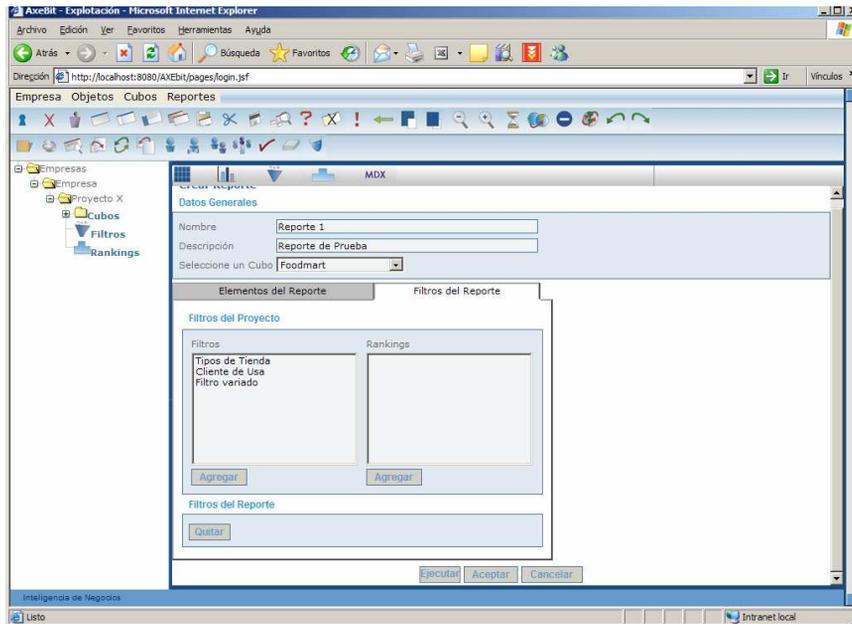


Figura 3.14 Pantalla para agregar rankings o filtros a un reporte

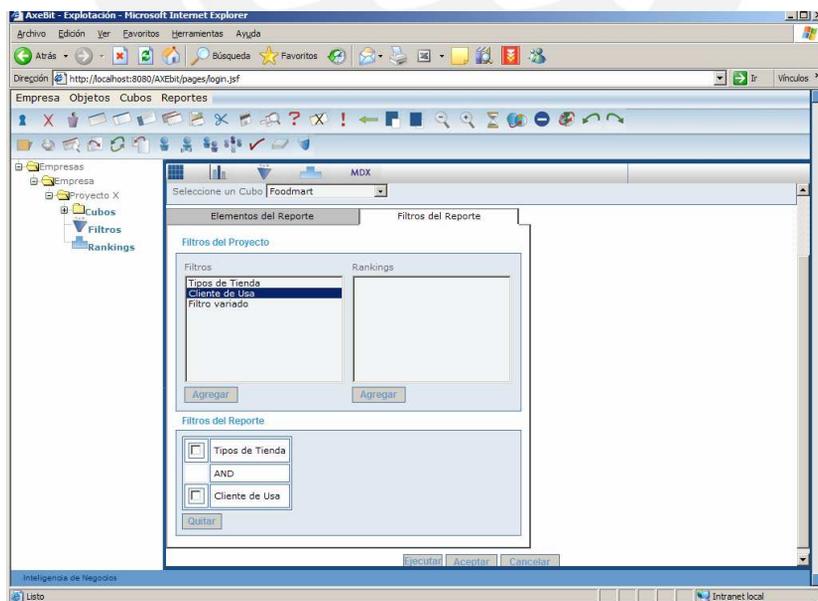


Figura 3.15 Pantalla en la que se interrelacionan mas de dos filtros con operadores lógicos

En la pantalla desarrollada para el caso de uso Ver Reporte, el usuario tendrá la posibilidad de elegir el tipo de reporte a mostrar (Reporte con gráfico de barras, tabulares, etc) y exportar los reportes en el formato deseado, que puede ser tanto Excel como gráfico.

Pantallas usadas para crear un cubo:

Como en el caso de uso Administrar Reporte, en la pantalla mostrada en la figura 3.16, el usuario tiene la posibilidad de buscar entre los reportes creados en el sistema previamente y borrar o modificarlo.

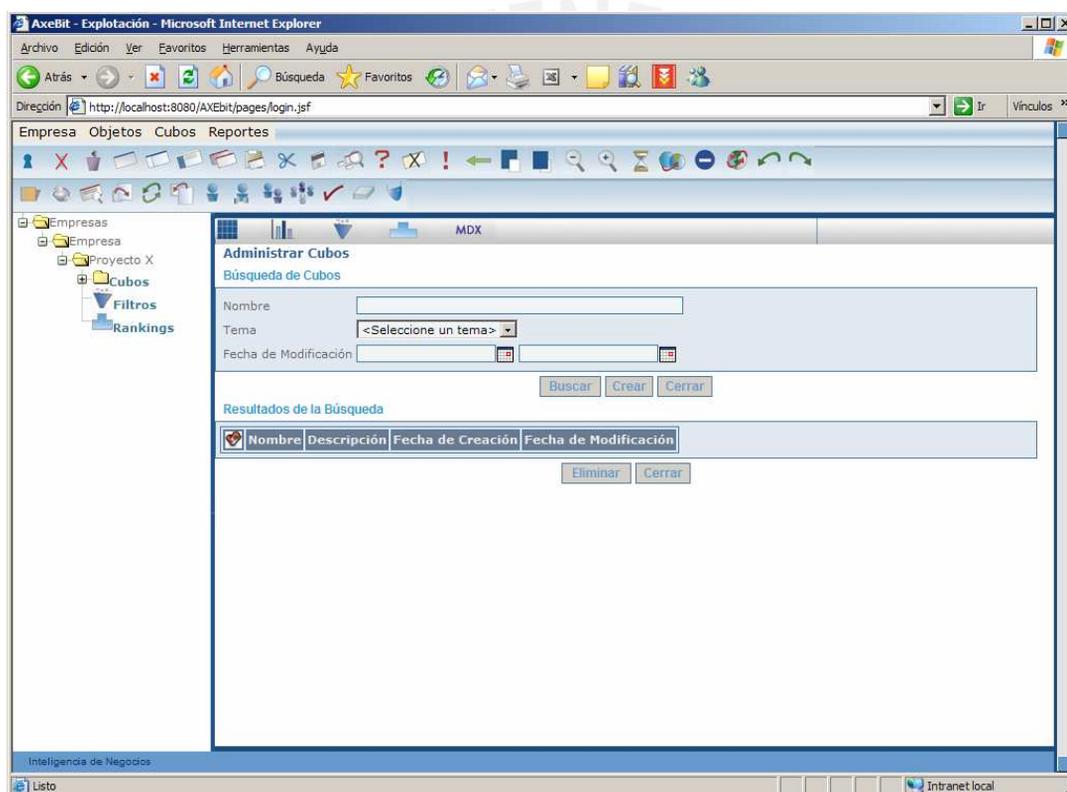


Figura 3.16 Pantalla Administrar Cubo

Para crear un cubo, igual que en el caso de los reportes, el usuario debe hacer clic en el botón “Crear” de la pantalla anterior, luego de lo cual el sistema mostrará una pantalla similar a la mostrada en la figura 3.17.

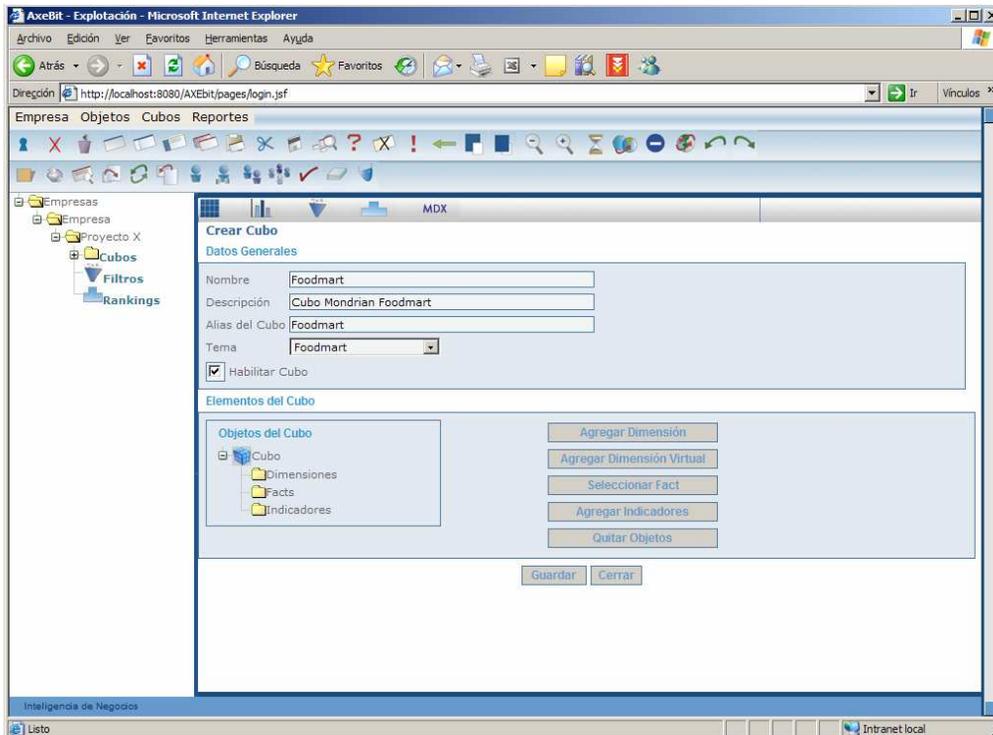


Figura 3.17 Pantalla Crear Cubo

Esta pantalla es bastante intuitiva y tiene como ventaja que muestra un árbol de dependencia de los elementos que se van integrando al cubo.

Una vez elegido el tema al cual va a pertenecer el cubo, se elige añadir el elemento que se desee, ya sea indicadores, la fact del cubo con sus medidas, dimensiones, etc.

Un ejemplo que muestra una pantalla en donde se añade una dimensión en el cubo se puede observar en la figura 3.18

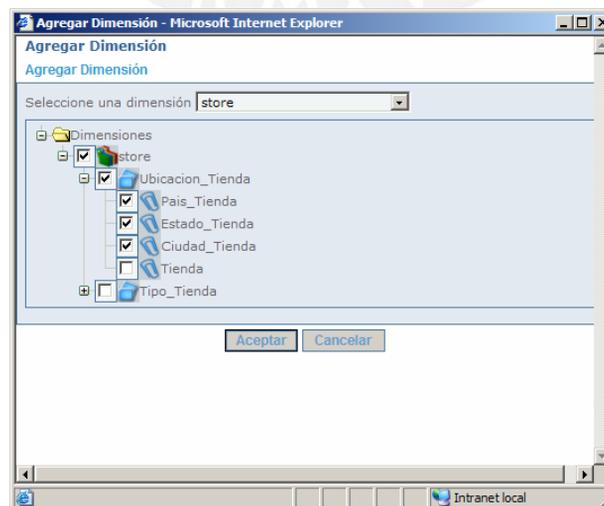


Figura 3.18 Pantalla para agregar una Dimensión

Esto se realiza para todos los elementos a agregar en un cubo tales como indicadores (pantalla mostrada en la figura 3.19) y una fact con sus medidas (Figura 3.20)

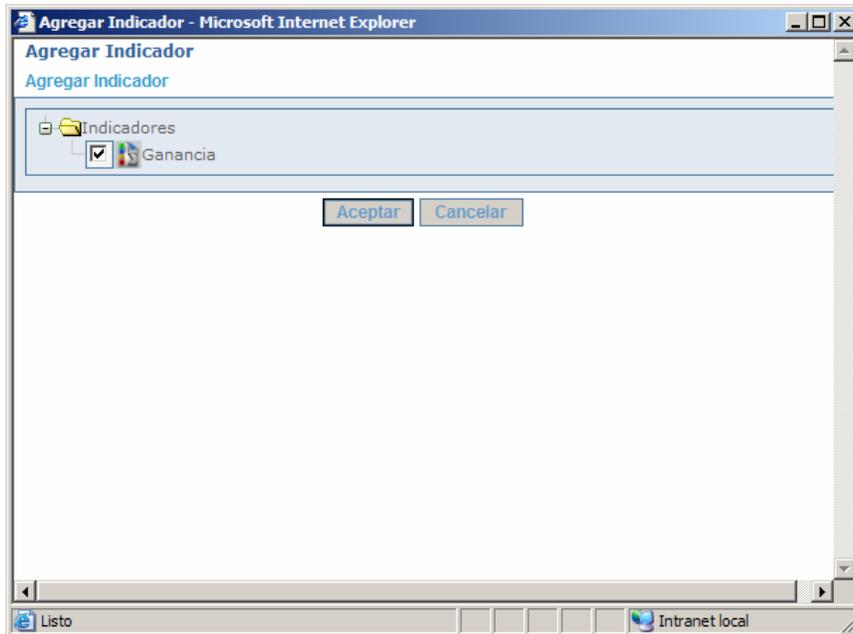


Figura 3.19 Seleccionando Indicador a agregar al cubo.

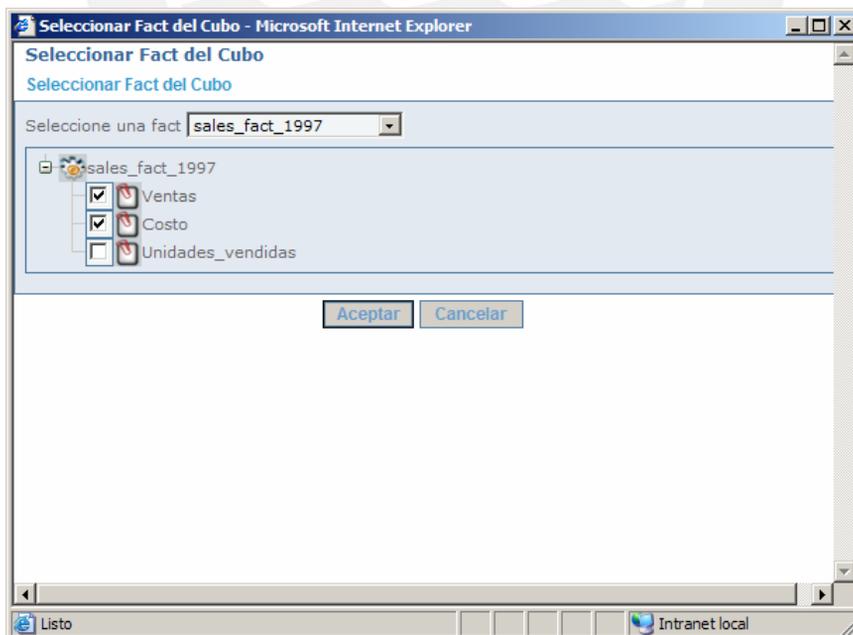


Figura 3.20 Seleccionando una Fact con algunas medidas

Luego de agregados todos los elementos que se deseen, se podrá ver el árbol de dependencias de una manera similar a lo mostrado en la figura 3.21.

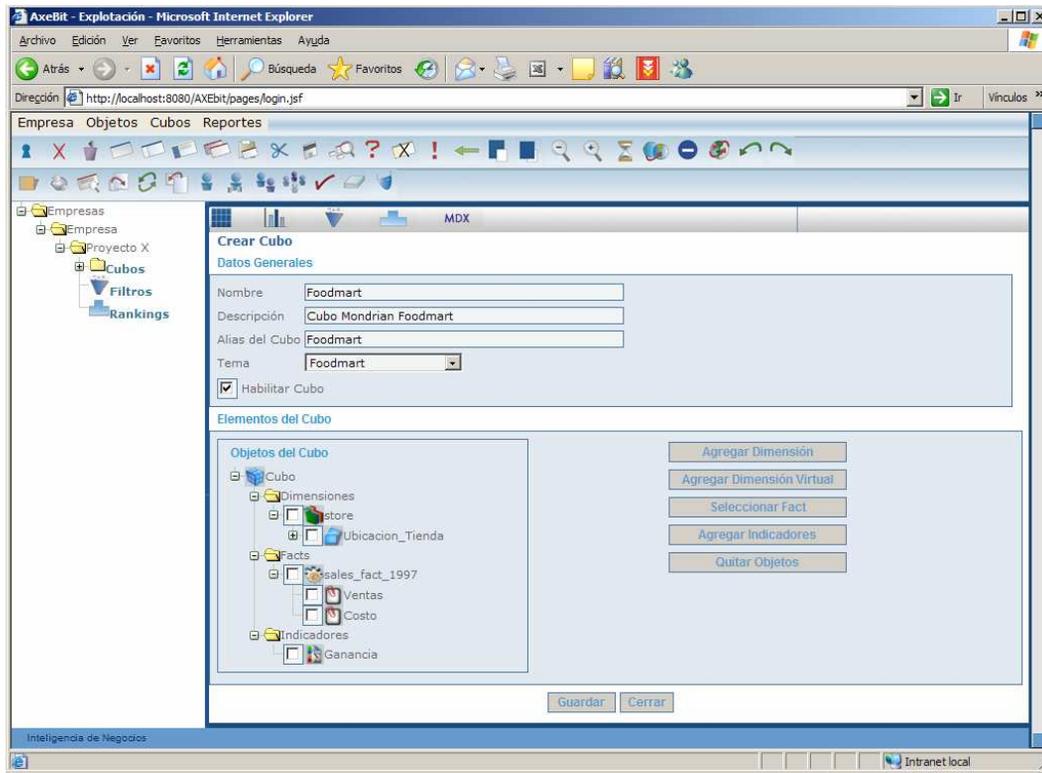


Figura 3.21 Pantalla Crear Cubo con todos los elementos agregados al cubo

Una vez elegidos todos los elementos del cubo, o en el proceso de esta selección, si se desea, se puede dar características especiales a cada elemento del cubo, haciendo clic al elemento del cubo y eligiendo “Configurar <elemento a configurar>”.

Se abrirá una pantalla similar a la figura 3.22, en donde por ejemplo se configura una medida. Cuando se configura un elemento del cubo se podrán elegir las características concretas de cada elemento. Estas pueden ser el alias del elemento, el tipo, la visibilidad, etc. Estas características varían según elemento a configurar.

Ejemplo de pantalla que configura una medida del cubo:

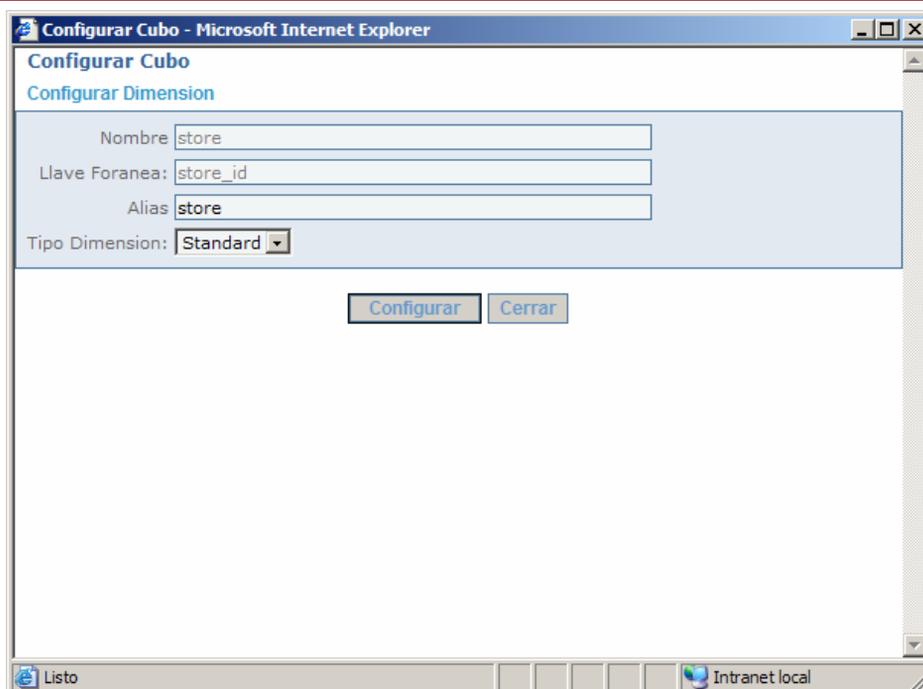


Figura 3.22 Configurando una Medida del cubo

Cabe mencionar que las características elegidas por el módulo de Análisis, tales como el nombre o los campos de la tabla de base de datos, de la cual la información es obtenida para el elemento del cubo, solo se podrán cambiar en el módulo de Análisis. Entonces en el módulo de Explotación solo se podrán configurar los elementos necesarios para las consultas multidimensionales (MDX). Esto último para mantener un orden en la configuración de los elementos en un módulo a otro.

3.6. Algoritmos Principales

En este punto se expone el algoritmo principal usado en la aplicación desarrollada por el módulo de Explotación. Este es representado de dos maneras:

- Pseudo código general indicando la funcionalidad general del algoritmo.
- Diagrama de Flujo.

3.6.1. Conversión de Consultas Multimensionales

Este algoritmo se utiliza para convertir la definición de los objetos a consultas multidimensionales (MDX). De esta manera, se puede realizar las consultas necesarias para los reportes, directamente en el motor OLAP.

Los objetos que pueden ser convertidos con este algoritmo son todos los que componen un reporte, esto es: filtros, rankings, etc. El proceso sigue el siguiente pseudo-código:

- Obtener lista de datos la columna del Reporte a procesar
- Obtener lista de datos de la Fila del Reporte a procesar
- Si la Columna tiene mas de un elemento
- Columnas=genera producto cartesiano(Columnas)
- En caso Contrario generar set de Datos.
- Si la Fila tiene mas de un elemento
- filas=genera producto cartesiano (Filas)
- caso contrario generar set de de Datos
- Consulta= Armar consulta(columnas, filas , Cubo)
- Devuelve la Consulta

El Diagrama de Flujo del pseudo-código arriba presentado, se puede observar en el diagrama de flujo de la figura 3.23 mostrada en la siguiente página.

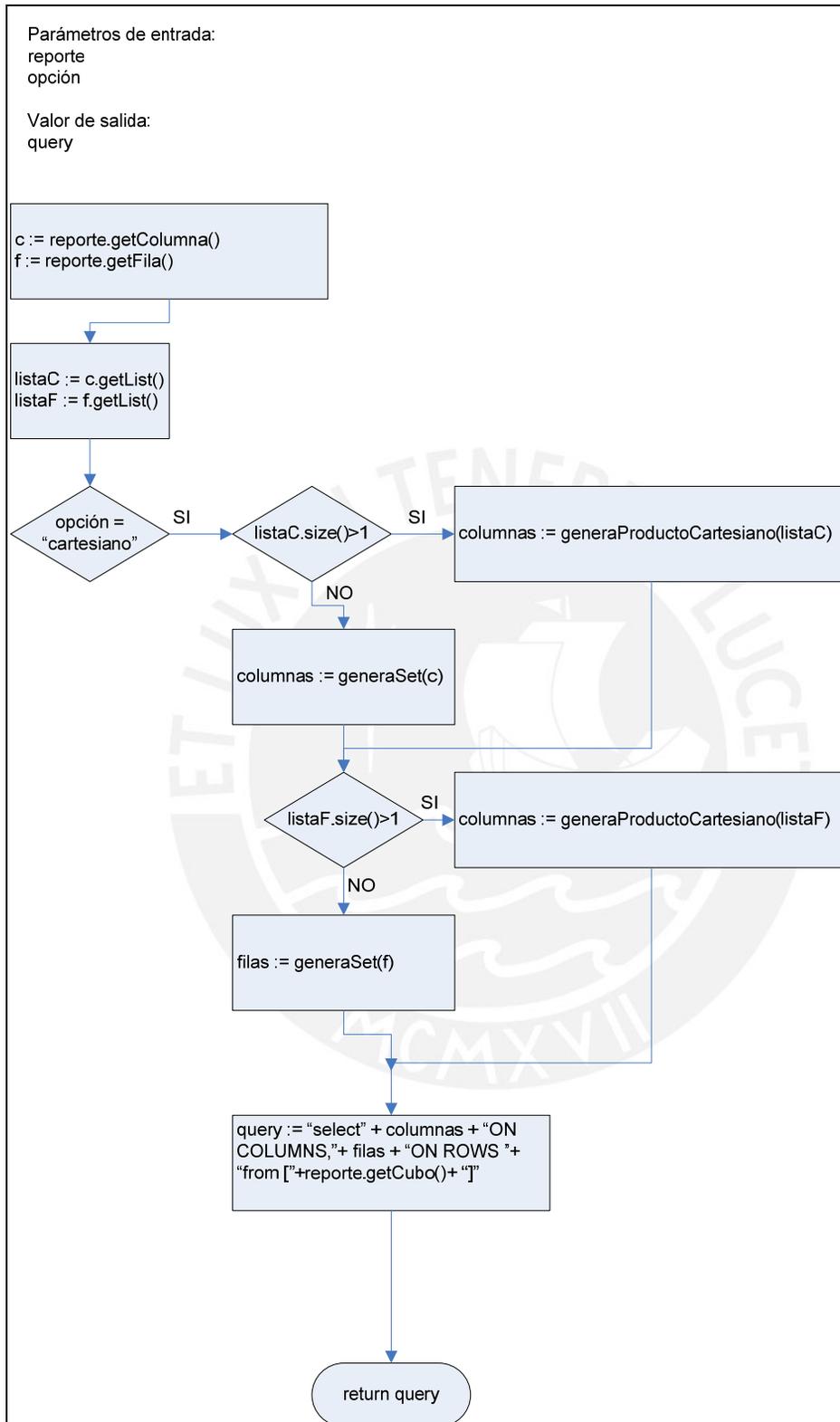


Figura 3.23 Diagrama de Flujo de Conversión de Consultas MDX

4. Construcción de la Plataforma de Integración

4.1. Consideraciones Preliminares

Al inicio de la segunda fase del proyecto, los tres módulos de este proyecto de tesis, plantearon de manera integrada el análisis y el diseño de la herramienta Axebit. Posteriormente en la fase de programación, se propuso la relación entre clases bases de cada módulo. Dado que se encontró que en muchas sino la mayoría de las clases tenían un uso compartido entre módulo y módulo, tuvieron que haber varias reestructuraciones al modelo original. Es por esta razón, que en la integración, el módulo de Explotación, tuvo que realizar un cambio bastante radical en los diagramas de clases (explicado en el punto 3.1), y con esto también un cambio en los casos de uso, diagramas de secuencia y en casi toda la documentación de este módulo. Pero finalmente se logró obtener un esquema bastante integrado y en el que cada módulo podía realizar el desarrollo de su herramienta, sin interferir unos con otros, excepto en aquellos puntos planificados con anterioridad, en los que hubo conflictos y retrasos, si, pero que encontraron finalmente una solución.

Luego de definidas las clases y el como integrarlas y relacionarlas entre si, se definió el FrameWork Web.

Se eligió (JSF), que es un nuevo marco de trabajo para interfaces de usuario para aplicación

J2EE. Por diseño, es particularmente útil con aplicaciones basadas en la arquitectura MVC (*Model-View-Controller*), explicada ya en el punto 2.3 de esta documentación.

De acuerdo a un análisis con todos los integrantes del proyecto de tesis, este patrón se adecuaba perfectamente al análisis y al diseño previamente elaborado.

Para lograr una estandarización respecto a los mensajes a mostrar, y abrir también la posibilidad de mostrar estos mismos mensajes en algún otro idioma, se crearon los archivos de recursos (bundle): uno para los títulos de las páginas y otro para el contenido de estas.

4.2. Metodología de desarrollo

En este proyecto se decidió aplicar la metodología RUP con iteraciones iterativas.

RUP es una metodología que cuenta con 4 fases de ciclo de vida de *software*, los cuales se aplicaron a este proyecto de tesis de la siguiente manera:

- Concepción: En esta fase se definió el objetivo principal del proyecto, que se describe en el capítulo 1 de este documento.
- Elaboración: En esta fase se definieron los requerimientos del producto, descrito en detalle en el anexo A. Además se definió la arquitectura a usarse en la aplicación a desarrollar y se elaboró parte del análisis y diseño del software.
- Construcción: En esta fase se concentra la implementación de la aplicación. En esta fase se generaron cambios y mejoras en el diseño. Cambios que se hicieron necesarios en la integración de las aplicaciones desarrolladas por los tres módulos.
- Transición: En esta fase de RUP, se suele analizar cómo debe ser la introducción del producto a los nuevos usuarios y se realiza una planificación al respecto. Si bien en este proyecto se hace un somero análisis al respecto en el último capítulo de este documento, no se realizó en este proyecto una profundización respecto a esta fase RUP.

Este proyecto fue desarrollado en tres iteraciones en la fase de construcción. El número de iteraciones fue elegido de acuerdo al número de casos de uso, las dependencias entre estos y la complejidad que cada uno representaba en el desarrollo, haciendo un análisis rápido inicial. Cada iteración tenía como objetivo el desarrollo de diferentes funcionalidades de la aplicación del módulo de Explotación. Además al final de cada iteración, podían sugerirse mejoras u observaciones de las iteraciones anteriormente desarrolladas y presentadas.

La primera iteración del programa desarrollado por el módulo de Explotación, incluía la implementación de las clases bases que se usarían en este módulo, las cuales serían usadas en los componentes del cubo y del reporte. También en esta iteración se desarrollaron las

funcionalidades que crearían y configurarían los objetos pertenecientes al reporte.

La segunda iteración en cambio, de centró en el desarrollo de todas las funcionalidades para la administración y configuración de los cubos. Fue en esta iteración, cuando surgió la necesidad de un cambio bastante significativo en el diagrama de clases. Esto para lograr la integración y actualización de los objetos entre los módulos de Análisis y Explotación, explicada con más detalle en el punto 3.1 de esta documentación.

Finalmente, En la tercera iteración se desarrollaron todas las funcionalidades de administración y presentación de los reportes. En esta fase fue muy necesario, el uso de la herramienta ofrecida por JPivot, además de conocimientos bastante significativos en los que a consultas Multidimensionales (MDX) refiere.

4.3. Estimación de Esfuerzo para el proyecto

Para asegurar la inversión del tiempo y planeamiento adecuado necesario en plataforma Web en la segunda parte del proyecto, se usaron las técnicas siguientes:

- Cocomo II [COCOMO, 2002]
- Puntos de Función [IFP, 1999].

Con estas se logró obtener la medida de meses por persona, lo cual estima cuánto esfuerzo se requiere por cada integrante para poder cubrir el desarrollo de la herramienta.

La estimación fue mucho más fácil de realizar en la segunda etapa, dado que el equipo de desarrollo tenía conocimientos más claros en lo que respecta a funcionalidad de la herramienta y conocimientos de estructuras usadas en Inteligencia de Negocios.

Además, el equipo de trabajo al estar más cohesionado entre sí, conocía mejor cual eran las debilidades y fuerzas de los integrantes, lo que daba una mejor visión de cuanto podía necesitarse en tiempo y costos el desarrollo del proyecto. El documento de estimación de esfuerzo elaborado para el proyecto se encuentra en el anexo K.

4.4. Distribución de las tareas en el equipo

La distribución de las tareas de desarrollo de este proyecto de tesis, se tomaron en cuenta varios criterios.

- Cuales eral las tareas que tomó cada miembro en la primera parte del proyecto (en el curso de Desarrollo de Programas 1), ya que tenía experiencia previa de las funcionalidades necesaria en esa parte del proyecto.
- El tiempo disponible de cada miembro, y a que si bien ya no existía carga académica de los estudios universitaria para algunos de los miembros, existía la carga del trabajo.

La implementación de los casos de uso se distribuyó según la tabla 4.1:

Casos de Uso	Persona Responsable
Administrar Filtros	Sergio Falcón
Administrar Ranking	Rosanna Palma
Administrar Cubo	Sergio y Rosanna
Administrar Reportes	Sergio Falcón
Ver Reporte	Raúl Celi
Exportar Reporte	Raúl Celi y Sergio Falcón

Tabla 4.1 Distribución de Responsabilidades de Implementación de Casos de Uso

4.5. Consideraciones para la Implementación de la Herramienta

Para evitar la sobrecarga innecesaria en el servidor se decidió completar algunas validaciones usando código Javascript.

Para también reducir tiempo en el desarrollo de la interfaz gráfica de la herramienta se adicionó en la programación, la librería JPivot, que agrega a la herramienta más opciones gráficas en la presentación de los reportes, así como en la configuración y exportación de los reportes y los datos de estos. JPivot ofrecía funcionalidades adicionales a la herramienta como por ejemplo, el poder graficar y configurar un reporte luego de recibido un cubo formateado en el esquema XML Mondrian y una consulta MDX.

4.6. Pruebas de Desempeño

Realizar pruebas involucra la ejecución de un sistema o aplicación bajo condiciones controladas y la evaluación de los resultados obtenidos (por ejemplo: si el usuario está en la interfaz A de la aplicación utilizando el hardware B, haciendo C entonces debe resultar D). Las condiciones controladas incluyen tanto condiciones normales como anormales. Las pruebas

deben buscar intencionalmente que las cosas salgan mal de tal manera que se pueda saber si se obtienen ciertos resultados que no se deberían obtener o si no se obtienen los resultados que se deberían obtener. Las pruebas están orientadas a la “detección” de defectos.

[SW QA/TEST RESOURCE CENTER, 2002]

Se realizaron diversas pruebas en la herramienta, para asegurar el correcto funcionamiento y las virtudes de esta. Entre los objetivos de las pruebas, se buscaba comprobar la correcta creación de los cubos y de la estructura XML. Esta estructura es muy importante, ya que posteriormente esta serviría como fuente de datos para el servidor OLAP y las consultas MDX.

En la etapa de pruebas se usó diversos motores de base de datos para comprobar entre otras cosas la compatibilidad de estas con el sistema en desarrollo. Estos motores de base de datos que contenían información como origen de datos fueron: SQL Server 2000, MySQL y Access 2000; pero la lista de bases de datos que se puede usar con la herramienta, puede fácilmente ser ampliado por la compatibilidad del servidor OLAP usado. Esto se aplica no solo para la aplicación del módulo de Explotación, sino también el módulo de Análisis y Extracción.

4.6.1. Casos de Prueba

En este punto se presentan algunos casos de pruebas aplicados al flujo de creación de los casos de uso Administrar Reporte (Figura 4.2) y Administrar Cubo (Figura 4.3). Como se mencionó en los puntos anteriores, estos dos casos de uso los que son considerados dentro del proyecto de tesis realizado por este módulo de Explotación, como los casos de uso principales. Cada caso de prueba que se desarrolló en el presente documento y además los presentados en el anexo de Documentos de Prueba, incluyen el objetivo del caso de prueba, la precondition que deben tener estos casos de prueba antes de su ejecución, las clases implicadas en el flujo del caso de prueba, el proceso para la realización de la prueba correspondiente y finalmente el resultado esperado para cada caso de prueba.

Si al realizarse la prueba con el caso de prueba no se obtiene el resultado propuesto por este, se documentará el resultado. De esta manera, finalizadas todas las pruebas se podrá analizar los casos de prueba que no cumplen, las razones del porqué y se podrá realizar las correcciones necesarias.

La lista completa de casos de prueba se encuentra en el anexo L (Documentos de Pruebas) de la documentación de este módulo de Explotación.

Administrar Reporte – Crear Reporte	
ID Prueba:	BX01
Objetivo :	Crear un reporte mediante la selección e ingreso de elementos válidos.
Precondición:	Existen los atributos, jerarquías, medidas, indicadores, filtros y rankings necesarios para crear los reportes.
Clases :	Reporte, ControlReporte, GestorReporte, ReporteBean, AdministrarReporteBean, CrearReportesBean, AdministrarReportes, CrearReporte
Proceso:	<ol style="list-style-type: none"> 1. Seleccione la opción Nuevo Reporte. 2. Seleccione una o más jerarquías y colóquelas en las filas 3. Seleccione una o más jerarquías y colóquelas en las columnas. 4. Seleccione una o más jerarquías y colóquelas en las páginas. 5. Seleccione una o más medidas (o indicadores) y colóquelas en filas o columnas. 6. Seleccione los filtros que desee añadir al reporte. 7. Seleccione los rankings que desee añadir al reporte. 8. Seleccione la opción Guardar. 9. Ingrese un nombre para el reporte.
Resultado Esperado:	Se muestra el mensaje “El reporte ha sido guardado con éxito”.

Tabla 4.2 Caso de Prueba del Caso de uso Administrar Reporte

Administrar Cubos – Crear Cubo	
ID Prueba:	BX09
Objetivo :	Crear un cubo con datos válidos.
Precondición:	La conexión de base de datos del proyecto está configurada.
Clases :	Cubo, AdministrarCubo, AdministrarCuboBean, CrearCuboBean, Dimension, BXDimension, BXIndicador, BXMedida, Medida, Indicador, BXAtributo, Atributo, Jerarquia ,BXJerarquia, Fact, Tema,
Proceso:	<ol style="list-style-type: none"> 1. Seleccione Crear Cubo. 2. Seleccionar el Tema al que el cubo pertenecerá. 3. Ingrese el nombre y descripción del cubo. 4. Seleccione añadir Dimensión y seleccione una de las dimensiones junto con las jerarquías y atributos que se quiera usar y que estén disponibles. Seleccione la Fact table en la que se centrará el cubo y las medidas que se usarán de esta. 5. Ingrese el nombre para el cubo (tema de explotación). 6. Seleccione Finalizar.
Resultado Esperado:	Se muestra el mensaje “El cubo se ha creado con éxito”.

Tabla 4.3 Caso de Prueba del Caso de uso Administrar Cubos

4.6.2. Reporte de Pruebas

En este punto se detallan los reportes de las pruebas realizadas tanto para las pruebas de casos de uso de la aplicación al ser realizadas.

Los escenarios para los casos de prueba realizados en los cuales se prueba la correcta ejecución de una funcionalidad, se detallan en el siguiente punto. Cada reporte contiene un caso de prueba, el resultado obtenido y su verificación del resultado correspondiente.

4.6.2.1. Pruebas de Caso de Uso

Conforme a lo acordado por los tres módulos que desarrollaron la aplicación de este proyecto de tesis, se desarrollaron pruebas de casos de uso, las que a su vez al ser ejecutadas lograron obtener los siguientes reportes de casos de prueba, expuestos en este punto.

En la tabla 4.9 se presenta una lista de reportes de prueba realizados para algunos de los casos de prueba definidos para la herramienta desarrollada por este módulo de Explotación. En esta tabla se detalla no solo los nombres de los casos de prueba y su código, sino además el escenario en el que se realizó el desarrollo de la prueba. En el contenido de estos reportes también se incluye el resultado obtenido para las pruebas al ser realizadas y la manera en como se puede realizar la verificación del resultado, es decir, como puede el usuario de manera tangible verificar si el resultado obtenido es el mismo que el caso de prueba propone, y de esta manera poder definir si la herramienta tiene un funcionamiento correcto para ese punto.

En la presente documentación se detalla únicamente los reportes de casos de prueba para el caso de uso Administrar Reporte (Flujo de creación, modificación y eliminación), los reportes de casos de prueba para el caso de uso Crear Cubo (Flujo de creación y eliminación) y también para un reporte de caso de prueba para el caso de uso Configurar Cubo.

La lista completa de reportes de casos de prueba se encuentra en el anexo L (Documento de Pruebas). En el anexo indicado, se puede ver la ejecución de cada uno de los casos de prueba diseñados para la aplicación desarrollada por este módulo de Explotación y el resultado obtenido u esperado por obtener para cada caso.

4.6.2.2. Pruebas de Otras funcionalidades

En la tabla 4.5 se detalla una lista de reportes de prueba para otras funcionalidades de la herramienta que no encajan directamente en un caso de uso. El listado completo se encuentra en el Anexo M.

Caso de prueba	Escenario de prueba	Resultado obtenido	Verificación del resultado
BX01- Administrar Reporte – Crear Reporte	Existen objetos creados en el módulo de Análisis. Se crearon también con anticipación filtros o un ranking para aplicar en el reporte.	obtiene un nuevo elemento XML para que contiene la estructura de datos del reporte	Se observa la estructura en el árbol de navegación de objetos con el nuevo reporte creado.
BX02- Administrar Reporte – Crear Reporte	Existen objetos creados en el módulo de Análisis. Se crearon también con anticipación filtros o un ranking para aplicar en el reporte.	Sale un mensaje de error que indica que el reporte creado no tiene	No se observa la estructura en el árbol de navegación de objetos con el nuevo reporte creado.
BX04- Administrar Reporte – Modificar Reporte	Existe un reporte previamente creado.	Sale un mensaje de que la modificación ha sido realizada con éxito.	En el reporte modificado se pueden observar los cambios.
BX05- Administrar Reporte – Eliminar Reporte	Existen reportes creados previamente.	Se obtiene que confirma que se borró el elemento	Cuando se listan todos los reportes existentes, no aparece más el reporte eliminado.
BX09- Administrar Cubo - Crear Cubo	Existen un Tema determinado y los elementos creados previamente para este tema en el módulo de análisis.	Se obtiene un cubo nuevo y dos archivos XML con la configuración del cubo	En el árbol de objetos se observa el nuevo cubo creado. Y dentro del mismo formulario de creación del cubo, se puede ver otro árbol que contiene todos los objetos del cubo.
BX12- Administrar Cubo - Eliminar Cubo	Existe un cubo previamente creado.	El sistema muestra un mensaje de confirmación.	En el árbol de objetos no se puede ubicar más el cubo eliminado.
BX13- Configurar Cubo	Existe un cubo previamente creado	El sistema envía un mensaje de confirmación, que indica que el cubo se guardó con los cambios	En el árbol de objetos se encuentra ya otro objeto con el mismo nombre que el que se quiere usar.

Tabla 4.4 Reporte de Pruebas de Caso de Uso

Caso de prueba	Escenario de prueba	Resultado obtenido	Verificación del resultado
BX37- Exportar Reportes	El sistema muestra un mensaje de error solicitando el nombre del reporte a exportar.	El sistema muestra un mensaje de error solicitando el nombre del reporte a exportar.	No se pueden encontrar los archivos generados en la ubicación seleccionada.

Tabla 4.5 Reporte de Pruebas de Caso de Uso de otras funcionalidades

Se considera este último reporte de prueba en esta categoría, ya que la verificación de la generación de archivos es externa propiamente de la aplicación.



5. Observaciones, Conclusiones y recomendaciones

5.1. Observaciones

- Para emprender un proyecto de tesis que desarrolle un tema similar a este, es necesario además de la experiencia y habilidades en el desarrollo Web, un sólido conocimiento en conceptos de Inteligencia de Negocios y los elementos que conforman una base de datos dimensional.
- Se realizó una investigación y capacitación previa al desarrollo de la herramienta, pero estos conocimientos tuvieron que ser mucho más ampliados conforme ésta se fue diseñando y construyendo.
- La investigación realizada para poder desarrollar de este módulo abarcó, los elementos que componen un cubo y las configuraciones que pueden realizarse en estos para consultas MDX (Multidimensionales). Como nuestro servidor OLAP es Mondrian, se tuvo que realizar una investigación de cómo esta herramienta manejaba las estructuras de los cubos.
- Adicionalmente fue necesario un conocimiento bastante amplio sobre consultas multidimensionales, y dado que el tema es aún en parte nuevo y difiere un poco entre Mondrian y Microsoft (creador de este lenguaje), se tuvo que determinar claramente como íbamos a trabajar.

- Si bien el resultado fue una herramienta integrada con los tres módulos necesarios para un ETL de una base de datos, el costo en tiempo fue también bastante amplio, mucho más de lo planeado al inicio de la segunda fase del proyecto (posterior al curso de Desarrollo de Programas 1).

5.2. Conclusiones

- Se logró la integración completa de los objetos entre módulos. Es decir el usuario está en la capacidad de crear un objeto base en el módulo de análisis y configurar este mismo objeto en el módulo de explotación, según sus necesidades.
- Se logró el objetivo principal del proyecto de Tesis: Se completó el desarrollo de la herramienta de explotación OLAP, la cual puede ser accedida desde una interfaz Web, y que permite a los usuarios utilizar la metadata creada en la etapa de Análisis, para definir las estructuras de datos que utilizarán los cubos y en base a estas 'últimas definir las estructuras que se mostrarán en los reportes.
- La herramienta de explotación desarrollada funciona como un generador de reportes OLAP. Es decir esta herramienta permite al usuario definir y personalizar sus propios reportes de manera sencilla indicando que elementos (atributos o métricas) se colocarán en filas y en columnas.
- La herramienta desarrollada posee un alto grado de escalabilidad al permitir conexiones a distintos motores de bases de datos. Esto se debe a que se utilizó un esquema flexible que permite ampliar de manera sencilla los tipos de motores soportados. Para agregar nuevos motores de bases de datos solo es necesario agregar el *driver jdbc* adecuado, y realizar pequeños cambios en el Control de Conexiones.
- Para cumplir con los objetivos del proyecto de *software*, éstos deben estar claramente definidos y especificados. Se debe conocer y definir cada detalle de los requerimientos y modelarlos adecuadamente en todas las etapas para su posterior desarrollo. Para lograrlo se ha utilizado la metodología de desarrollo de RUP y la notación UML.
- Para el éxito del proyecto de desarrollo de *software*, es necesario la comunicación y el compromiso de cada uno de los integrantes de un equipo. En caso alguno de estos elementos falte, el resultado puede ser el retraso del proyecto o el fracaso total del mismo.

5.3. Recomendaciones

Considerando futuras ampliaciones del sistema, se sugiere tomar en cuenta los siguientes puntos:

Internacionalización y soporte de más idiomas

La aplicación desarrollada en este proyecto de tesis tiene una interfaz Web, y por lo tanto puede ser accedida desde cualquier lugar del mundo. Debido a que en este proyecto se utilizó archivos de tipo *bundle* para los textos, mensajes, menús y demás interfaces con el usuario, se puede considerar ampliar la herramienta con la posibilidad de cambio de lenguajes de la interfaz mostrada al usuario.

Aumentar la facilidad de uso de la herramienta

La aplicación tiene una interfaz grafica intuitiva y fácil de usar, sin embargo, se puede considerar la creación de asistentes o *wizards* para hacer aún más fácil y rápido el uso de la herramienta para los usuarios.

Además, es necesario contar con la ayuda en línea completa de la herramienta, y documentación detallada de la misma para que se pueda usar al máximo sus potencialidades.

Extender el alcance la Aplicación

La aplicación permite definir los objetos básicos que componen un Cubo, sin embargo se puede considerar aumentar los tipos de objeto que éste soporta, como tablas agregadas que permitan mejorar el desempeño de consultas complejas.

Ofrecer capacitaciones previas a los usuarios respecto a los elementos de una base de datos dimensional

Especialmente el usuario que cumplirá el rol de analista tiene que tener conocimientos sólidos respecto al uso de elementos en estructuras multidimensionales. De esta manera el analista tendrá mayores posibilidades para una correcta configuración del cubo, además de una optimización en los resultados de los reportes.

Aumentar la seguridad de la aplicación

Si bien ofrece una ventaja para el acceso de usuarios externos que la aplicación esté desarrollada en plataforma Web, es importante sin embargo evitar que este acceso esté mal manejado. Para lograr dicho objetivo, se recomienda proteger los servidores que contienen la información con las aplicaciones o herramientas que el usuario final desee y tenga disponibles, por ejemplo: firewalls, sistemas anti-hackers routers, etc. Esto debido a que la aplicación accede a información importante para una o varias empresas, según sea el caso, lo que representa un importante activo para estas.



ANEXOS

Los anexos a referenciados en esta documentación son los mencionados en la siguiente lista:

ANEXO A. PROPUESTA DEL PROYECTO BI-PUCP

ANEXO B. LISTA DE EXIGENCIAS

ANEXO C. ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

ANEXO D. DOCUMENTO DE ANÁLISIS

ANEXO E. DOCUMENTO DE DISEÑO

ANEXO F. DOCUMENTO DE ARQUITECTURA

ANEXO G. DOCUMENTO DE XML

ANEXO H. DIAGRAMAS DE SECUENCIAS

ANEXO I. ESTÁNDARES DE PROGRAMACIÓN

ANEXO J. DISEÑO DE PANTALLAS

ANEXO K. DOCUMENTO DE ESTIMACIÓN

ANEXO L. DOCUMENTO DE PRUEBAS

ANEXO M. REPORTES DE PRUEBAS

BIBLIOGRAFÍA

[SW QA/TEST RESOURCE CENTER, 2006]

Adaptado de:

Software Quality Assurance Resource Center

<http://www.softwareqatest.com>

[PENTAHO, 2006]

Adaptado de:

Pentaho, Open Source Business Intelligence

<http://mondrian.pentaho.org/>

[JPIVOT, 2006]

Adaptado de:

SourceForge projects, JPivot- a JSP Based Olap Client

<http://jpivot.sourceforge.net/>

[MyFaces, 2007]

Adaptado de:

Apache MyFaces Project

<http://myfaces.apache.org/>

[ORACLE, 2006]

Adaptado de Oracle Technology Network

www.oracle.com

[ORACLE, 2005]

Introduction to Javasever Faces - What is JSF?

<http://www.oracle.com/technology/tech/java/newsletter/articles/introjsf/index.html>

[Microsoft, 2005]

Adaptado de

Microsoft Corporation

<http://www.microsoft.com/latam/sql/>

[COGNOS, 2007]

Adaptado de Cognos Corporation

<http://www.cognos.com>

[Boasson, 95]

Adaptado de Maarten Boasson, *The Artistry of Software Architecture*. IEEE Software, Vol. 12, No. 6, Noviembre 1995 (pp. 13-16).

[Apache 2005]

Adaptado de "Apache Software Foundation"

<http://tomcat.apache.org/>

[IFP, 1999] The International Function Point User Group (IFPUG). 1999. Function Point Counting Practices Manual Release 4.1. USA.

[COCOMO, 2002]

Adaptado de Cocomo II.

<http://sunset.usc.edu/research/COCOMOII/>.

