



**UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA  
COMPUTACIÓN**

**DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE  
DISTRIBUCIÓN DE INFORMACIÓN DE GESTIÓN**

**MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL  
EN COMPUTACIÓN**

**JOSÉ MIGUEL CASTRO GOROSTIZA**

**PROFESOR GUÍA:  
SR. HUGO ANDRÉS MORA RIQUELME.**

**MIEMBROS DE LA COMISIÓN:  
SR. JUAN FERNANDO ÁLVAREZ RUBIO.  
SR. GONZALO NAVARRO**

**SANTIAGO DE CHILE  
ABRIL 2007**

RESUMEN DE MEMORIA  
PARA OPTAR AL TÍTULO DE  
INGENIERO CIVIL EN COMPUTACIÓN  
POR: JOSÉ MIGUEL CASTRO G.  
FECHA: 28/06/2007  
PROF. GUÍA: Sr. HUGO ANDRÉS MORA R.

## **“DISEÑO E IMPLEMENTACIÓN DE UNA SOLUCIÓN DE DISTRIBUCIÓN DE INFORMACIÓN DE GESTIÓN”**

El objetivo general del presente trabajo de título es diseñar e implementar una solución de distribución de información de gestión, abordando el problema de modelamiento de sistemas de gestión para apoyar los procesos de inteligencia de negocios (Business Intelligence o BI) que las organizaciones requieren para poder entregar herramientas de monitoreo, análisis y evaluación de los procesos de negocio.

Esta memoria se basa en un trabajo para la empresa Correos Chile y busca resolver el principal problema que presentan las áreas de análisis de esta organización que consiste en una baja capacidad para acceder e interactuar con información de gestión almacenada en el datawarehouse corporativo.

Se determinaron los requerimientos de información de dos áreas comerciales encargadas de analizar las ventas de la compañía y definir la estrategia corporativa. Con estos requerimientos se determinaron las falencias del modelo previamente utilizado y las características necesarias del nuevo sistema para generar un cambio en los procesos de toma de decisiones.

El desarrollo final consistió en implementar dos datamarts. Uno para el análisis de ventas y otro para el análisis de ventas regional que, junto a un portal de gestión, permite tener acceso en tiempo real a toda la información requerida por los usuarios.

En base a los resultados y métricas que se definieron, se constató que, como resultado de la implementación de esta solución, Correos Chile logro reducir los tiempos de acceso a la información (desde horas a minutos) y simultáneamente mejoró la calidad de los reportes y la capacidad de análisis de las áreas involucradas.

Se concluye que la implementación de esta infraestructura y aplicaciones son capaces de mejorar sustancialmente los procesos de negocio de Correos Chile, mejorando el apoyo a los procesos de toma de decisiones y la calidad del análisis que se realiza. Ampliando la solución implementada al resto de la organización efectivamente mejorará la competitividad de la empresa.

# Índice general

1. Introducción .....	4
1.1 Justificación .....	5
1.2 Objetivo General .....	6
1.3 Objetivos Específicos.....	6
2. Definiciones previas.....	7
2.1 Definición de un proyecto de Inteligencia de Negocios. ....	7
2.2 Definición de un Datawarehouse. ....	8
2.3 Definición del modelo dimensional y análisis OLAP.....	8
3. Metodología .....	14
3.1 Kimball Bus Architecture .....	15
3.2 Corporate Information Factory .....	15
3.3 Componentes.....	15
3.4 Etapas del proyecto .....	17
4. Antecedentes .....	18
4.1 Antecedentes del negocio.....	18
4.2 Antecedentes tecnológicos.....	20
5. Análisis de requerimientos.....	21
6. Diseño .....	24
6.1 Diseño de la arquitectura.....	25
6.2 Diseño del modelo de datos. ....	29
6.3 Diseño de la aplicación. ....	33
6.3.1 Navegación de la Aplicación .....	35
6.3.2 Aplicación de Ventas Corporativas.....	36
6.3.3 Aplicación de Sucursales .....	39
6.3.4 Modelo de datos de las aplicaciones .....	41
6.3.5 Diseño de Alarmas y Suscripciones.....	46
6.3.6 Diseño de Mensajería.....	48
7. Implementación.....	50
7.1 Instalación y preparación de los equipos. ....	50

7.2 Implementación de las Bases de Datos .....	51
7.3 Implementación de las aplicaciones.....	53
7.3.1 Inicio y Navegación .....	54
7.3.2 Ventas Corporativas y Sucursales.....	56
7.3 Definición del proceso de mantención y crecimiento.....	72
8. Resultados .....	73
9. Conclusiones .....	75
9.1 Experiencias aprendidas y trabajo futuro.....	76
Bibliografía .....	78

## Índice de figuras

Figura 1 Modelo Multidimensional .....	10
Figura 2 Modelo Estrella.....	11
Figura 3 Modelo Snowflake.....	12
Figura 4 Organigrama Correos Chile.....	19
Figura 5 Relación Herramientas y Número de Usuarios .....	22
Figura 6 Relación entre usuarios y uso de procesador.....	29
Figura 7 Modelo de carga de datos .....	30
Figura 8 Modelo de creación de nuevos datos .....	31
Figura 9 Modelo de carga de datos para cubos .....	32
Figura 10 Áreas de la organización.....	33
Figura 11 Modelo de navegación.....	35
Figura 12 Diagrama de aplicación Ventas Corporativas.....	39
Figura 13 Diagrama de aplicación Sucursales .....	41
Figura 14 Modelo de datos del portal de ventas corporativas.....	42
Figura 15 Modelo de jerarquía de ejecutivos.....	42
Figura 16 Tabla ejemplo del modelo de sucursales .....	43
Figura 17 Modelo de datos de Cubo de Gastos, Cliente y Jefes Comerciales .....	43
Figura 18 Modelo de tablas asociadas .....	45
Figura 19 Modelo de tablas de seguridad .....	46
Figura 20 Modelo de aplicación suscripciones .....	47
Figura 21 Modelo de aplicación alarmas .....	47
Figura 22 Modelo de tablas de alarmas y suscripciones.....	48
Figura 23 Página de Aplicaciones de Alphablox .....	53
Figura 24 Página inicial del sistema.....	55
Figura 25 Submenú Postal .....	55
Figura 26 Query Builder .....	60
Figura 27 Panel de Control Ventas Corporativas.....	61
Figura 28 Ejemplo de cambio a gráfico .....	62
Figura 29 Ejemplo de selección de tipo de gráfico .....	62
Figura 30 Ejemplo de POPUP.....	63
Figura 31 Ejemplo de exportar a Excel.....	63
Figura 32 Ejemplo de reporte en Excel.....	64
Figura 33 Ejemplo de Cubo en Alphablox.....	65
Figura 34 Ejemplo 2 de Cubo en Alphablox.....	65
Figura 35 Ejemplo 3 de Cubo en Alphablox.....	66
Figura 36 Ejemplo 4 de Cubo en Alphablox.....	66
Figura 37 Ejemplo 5 de Cubo en Alphablox.....	67
Figura 38 Página de Presentaciones .....	68
Figura 39 Presentación Flash .....	68
Figura 40 Formulario 1 .....	69
Figura 41 Formulario 2 .....	69
Figura 42 Portal de Sucursales.....	70
Figura 43 Portal de Sucursales despliegue de datos en Flash .....	71
Figura 44 Portal de Sucursales Filtros .....	71
Figura 45 Resultados del proyecto.....	73

# Capítulo 1

## 1. Introducción

La administración y distribución de información es un problema en organizaciones de gran tamaño. Sus sistemas transaccionales almacenan varios Gigabytes<sup>1</sup> y en muchos casos Terabytes<sup>2</sup> de información en grandes bases de datos con el propósito de permitir el correcto funcionamiento de los procesos internos y asegurar la continuidad operativa.

Dar acceso a los sistemas de información a usuarios que no poseen los conocimientos técnicos necesarios para operar estos sistemas no es posible, sin embargo, es necesario que ellos tengan acceso a la información almacenada para poder realizar gestión sobre el negocio. Este acceso debe ser ágil, oportuno y asociado a información fidedigna.

En un mercado altamente competitivo y cambiante ésta agilidad en el análisis de información determinan el éxito y el fracaso de una organización. Los procesos demorosos para acceder a los datos se transforman en un gran problema para las empresas que buscan ser líderes de sus mercados. Sin capacidad de análisis, una empresa deja de ser dinámica y pierde la capacidad de adaptarse a nuevos escenarios simplemente porque no es capaz de anticiparse a ellos.

En Correos Chile, empresa Chilena dedicada al negocio Postal y de Paquetería, el área encargada de resolver este creciente problema y proveer de información en tiempos razonables a los distintos consumidores de información es llamada Área de Planificación Estratégica y Control de Gestión.

Dentro de las principales funciones de esta área está la de realizar análisis de datos de manera horizontal a todas las áreas de la institución, de manera de tener una visión global del negocio. Sin embargo, la complejidad para acceder a información de manera autónoma con la que se encuentran los usuarios de negocio ha obligado a esta área a destinar más recursos a la generación de reportes, abandonando su labor de proveer de una visión horizontal a la organización conllevando a una pérdida de competitividad en el mercado, eficiencia y la capacidad para adaptarse a nuevos escenarios.

---

<sup>1</sup> Gigabyte: Unidad de medida informática equivalente a mil millones de bytes.

<sup>2</sup> Terabyte: Unidad de medida informática equivalente a 2 elevado a 40 bytes.

## **1.1 Justificación**

Correos Chile necesita volver a tomar una posición de liderazgo en el mercado Postal y de Paquetería, para lo cual en conjunto con una serie de proyectos estratégicos, necesita de una plataforma que le permita agilizar la distribución de la información almacenada en sus sistemas.

Los proyectos de informática dentro de las grandes organizaciones como Correos Chile están enfocados principalmente a garantizar la continuidad operacional, dejando de lado proyectos estratégicos que les permitan diferenciarse de sus competidores

Este escenario ha cambiado lentamente con tecnologías como las de inteligencia de negocios, en el cual la integración de las áreas informáticas con las áreas comerciales y analíticas son capaces de generar valor y diferenciar a un competidor de otro.

Es esta importancia y el auge que está teniendo este tipo de tecnologías orientadas al negocio la principal motivación de esta memoria ya que permitirán a Correos Chile mejorar y agilizar sus procesos de acceso a la información.

Agilizar estos procesos en una empresa inmersa en un mercado altamente competitivo y cambiante es fundamental para poder sobrevivir y entregar un producto a la altura de lo que el mercado demanda.

## **1.2 Objetivo General**

El objetivo de esta memoria es realizar el diseño y la implementación de una solución de distribución de información de gestión, que permita el acceso a información crítica a todos los actores claves de Correos Chile y facilite la labor del área de Planificación Estratégica y Control de Gestión.

## **1.3 Objetivos Específicos**

Los objetivos específicos de este trabajo son:

- Proveer a Correos Chile de una infraestructura de inteligencia de negocios que responda adecuadamente a sus necesidades de acceso y distribución de información.
- Diseñar y desarrollar un conjunto de 2 sistemas utilizando la infraestructura anterior, que permitan el acceso a información crítica a toda la institución en torno a los siguientes temas:
  - Análisis de ventas corporativas y Balanced Scorecard<sup>1</sup>.
  - Gestión de sucursales y análisis geográfico.
- Desarrollar un set de herramientas que faciliten la interacción con la información disponible en los sistemas. Las herramientas propuestas son:
  - Alarmas: Permite suscribirse a una determinada Alarma y recibir notificaciones acerca del estado de un determinado indicador.
  - Suscripciones: Permite suscribirse a información de interés y poder revisarla de manera fácil y rápida.
  - Mensajería: Permite enviar y recibir mensajes a través del sistema para realizar un trabajo colaborativo sobre la información disponible.
- Proveer a correos chile de la metodología necesaria para expandir el sistema a otras áreas de la institución de manera eficiente y autónoma.

---

<sup>1</sup> Balanced Scorecard: Método para medir las actividades de una compañía en términos de su visión y estrategia.



# Capítulo 2

## 2. Definiciones previas

Para comprender de mejor manera el contenido de esta memoria, es necesario introducir al lector a un conjunto de definiciones que se utilizarán a lo largo del documento.

### ***2.1 Definición de un proyecto de Inteligencia de Negocios.***

Cuando hablamos de un proyecto de inteligencia de negocios, business intelligence o BI, hablamos de un proyecto estratégico dentro de las empresas que busca tener un impacto principalmente en las áreas comerciales y dentro de las gerencias.

Un proyecto de BI se diferencia de un proyecto de creación de infraestructura o de uno de mantenimiento ya que su aceptación es compleja, su costo es alto, tienen difícil seguimiento y por sobre todo tienen baja urgencia ya que el negocio no dejará de funcionar si no se realizan. Todos estos aspectos de un proyecto estratégico hacen que su definición y requerimientos deban estar plenamente alineados con la estrategia corporativa de manera de lograr venderse internamente como un proyecto necesario y que traerá efectivamente los beneficios que se prometen.

Cuando hablamos de un proyecto de BI, no hablamos de una tecnología en particular ni de un producto, hablamos de adoptar la capacidad de entregar información relevante a los principales actores de nuestra institución para su análisis y apoyo a la toma de decisiones, utilizando las herramientas tecnológicas necesarias para que la solución sea útil y de fácil adopción.

Una solución de BI puede ser desde un reporte en Excel a soluciones mucho más complejas como dashboards<sup>1</sup> corporativos que despliegan resúmenes de información consolidada de toda una institución.

---

<sup>1</sup> Dashboard: Panel de control que despliega una serie de indicadores de manera de permitir una fácil visualización.

## **2.2 Definición de un Datawarehouse.**

Un datawarehouse es un sistema de base de datos que busca almacenar y consolidar información de toda una institución a lo largo del tiempo, brindando la capacidad de realizar análisis complejo sobre él[2].

Un datawarehouse en general consta de los siguientes elementos[2]:

- Repositorio de datos.
- Modelo de datos orientado al negocio.
- Capacidad para realizar análisis OLAP.
- Procedimientos de extracción, transformación y carga.
- Algoritmos de limpieza de datos.

Otro concepto utilizado en el contexto de un datawarehouse es un datamart. Un datamart busca obtener el mismo resultado que un datawarehouse solo que orientado a un área o departamento específico dentro de una empresa.

En el ámbito de un proyecto de BI, un datawarehouse busca ser el repositorio central de datos que alimenta a todas las aplicaciones analíticas de la solución, presentando una sola versión de la verdad para todos los usuarios.

Dependiendo del modelo de datos, de su nivel de granularidad y de varios otros factores, el Datawarehouse nos permitirá tener mayores o menores alternativas para distribuir información de gestión y realizar análisis de datos en herramientas de alto nivel.

## **2.3 Definición del modelo dimensional y análisis OLAP.**

El modelamiento dimensional es una metodología de modelamiento de datos con una orientación hacia el negocio, esto quiere decir que la estructura de un modelo dimensional sale de las definiciones del negocio y no en base a lo óptimo en almacenamiento. Busca permitir a los usuarios interactuar con los datos de manera más intuitiva que lo que se obtiene con modelos relacionales tradicionales, los cuales están orientados a soportar las transacciones de los sistemas y no a responder preguntas de negocio.

Los principales factores que diferencian un modelo relacional de un modelo multidimensional son que el foco del primero es entregar escritura y lectura muy rápidas a nivel de una tabla, mientras que en el segundo caso, el cruce de muchas tablas y la posibilidad de responder a consultas del negocio debe ser posible con buenos tiempos de respuesta, relegando los tiempos de escritura a procesos ETL's<sup>1</sup> que se ejecutan con una periodicidad definida.

Los 2 conceptos asociados a estos modelos son[2]:

- OLTP (Online Transactional Processing) que se enfoca en sistemas transaccionales.
- OLAP (Online Analytical Processing) que está enfocado al análisis y cruce de información.

Un modelo dimensional, es decir, un modelo orientado a OLAP, esta construido sobre 2 componentes[4]:

- Dimensiones: Una dimensión define un área de análisis de la información, por ejemplo, el tiempo, geografía, sexo, etc. Una dimensión contiene una o más jerarquías que permiten navegar la información, por ejemplo, el tiempo contiene una jerarquía año-mes-día, de manera de que uno puede ver la información por año y luego navegar a un nivel inferior y revisar la información mes a mes, para luego llegar al detalle diario. Otra jerarquía podría ser año-semester y así sucesivamente. En cada dimensión debemos definir la operación que se utilizará para agregar la información, es decir, establecer si las ventas diarias al pasar por ejemplo a mes se deben sumar, contar, promediar u otra operación.
- Hechos: Son las medidas que vamos a poder ver en nuestro cubo, por ejemplo, las ventas, los costos, etc. Son los números que queremos analizar bajo las distintas dimensiones. Debemos tener en consideración la granularidad de la información que queremos analizar, es decir, si tenemos una granularidad diaria, podemos analizar estos datos por día, sin embargo, si se construyó el modelo con una granularidad mensual, el acceso a la información diaria no estará disponible. Por esta razón, en el modelamiento dimensional hay que tener en consideración las necesidades comerciales del cliente para poder llegar a un modelo que sea útil.

---

<sup>1</sup> ETL: Siglas en ingles para Extraer, Transformar y Cargar. Proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y luego cargarlos en otra base de datos.

Cuando tenemos un modelo dimensional, también hablaremos de un cubo principalmente porque la capacidad analítica que nos entrega este tipo de herramientas vista en un espacio tridimensional puede ser representada mediante esta estructura.

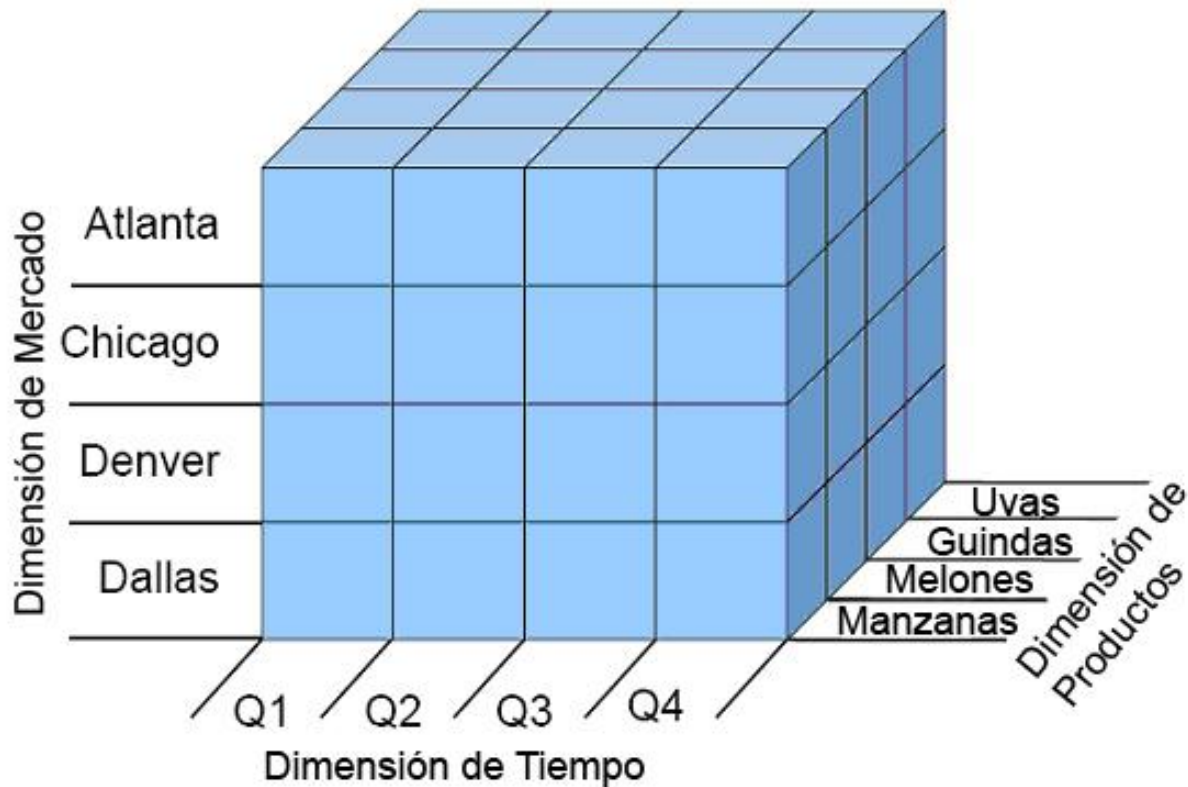


Figura 1 Modelo Multidimensional

En el cubo de la figura, se presentan 3 dimensiones, los mercados, los productos y el tiempo. En cada punto de intersección de estas 3 dimensiones encontramos los hechos.

Un Cubo además puede ser almacenado de 3 maneras. Cada una de ellas tiene sus ventajas y desventajas y deben considerarse al momento de diseñar una solución[6].

- **ROLAP (OLAP Relacional)** : Toda la información del cubo, sus datos, agregaciones y definición son almacenados en una base de datos relacional, es decir, ROLAP no almacena de manera separada los datos, si no que accede directamente a las tablas que lo definen y debe procesar la información en línea, haciéndola la alternativa más lenta en cuanto a tiempo de respuesta.

Las tablas que lo alimentan en general están en un modelo estrella que consta de una tabla de hechos y sus dimensiones conectadas por llaves foráneas. También se utiliza una extensión de este modelo llamado modelo Snowflake o copo de nieve.

En este caso, tenemos una tabla de hechos, tablas de dimensiones pero además cada dimensión puede tener tablas asociadas de manera de formar jerarquías más complejas.

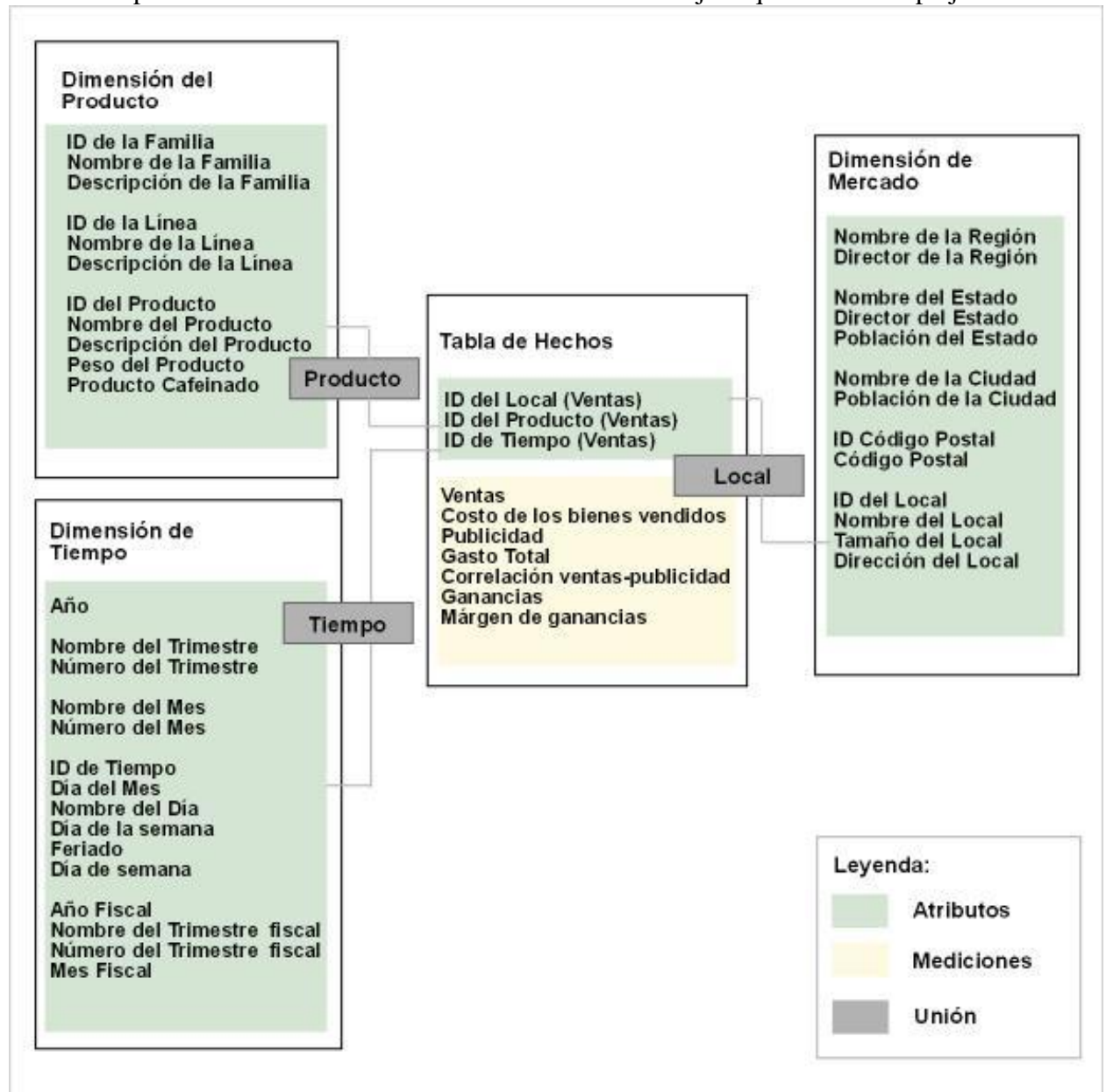
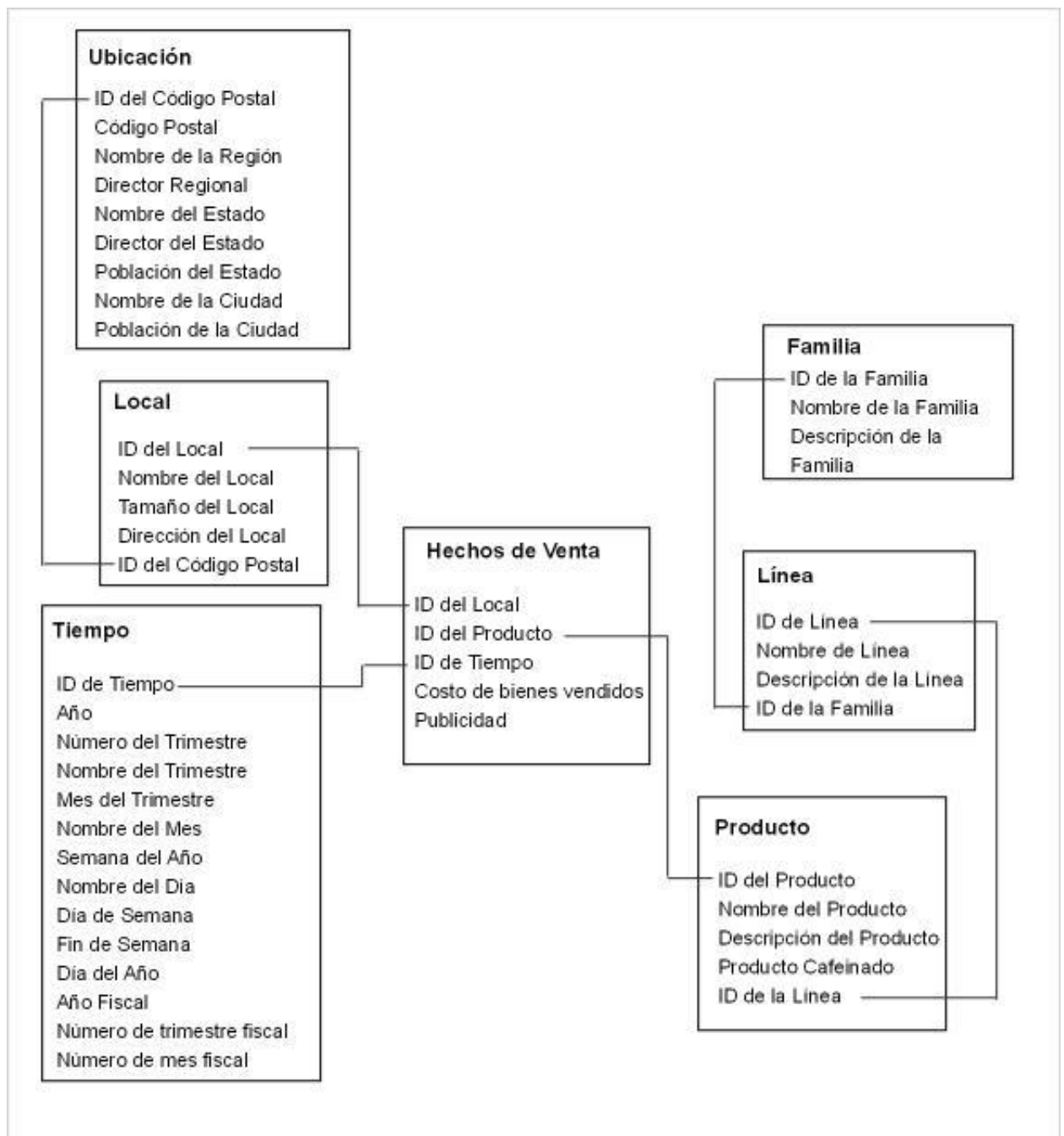


Figura 2 Modelo Estrella



**Figura 3 Modelo Snowflake**

- **MOLAP (OLAP Multidimensional):** En este caso, la información y agregaciones son almacenadas en una estructura multidimensional, separada de los datos de origen. Tiene un excelente tiempo de respuesta ya que las agregaciones se almacenan calculadas. Los puntos en contra de este tipo de solución es que dado que los datos se almacenan en forma separada, es necesario construir procedimientos de carga y refrescar el cubo ante cualquier cambio en los datos.

Además, es necesario definir que nivel de detalle queremos (granularidad de los datos) ya que si el detalle es muy alto, necesitaremos cargar la totalidad de los datos en el modelo y los procedimientos de carga y la navegación del cubo podrían ser muy lentos. Por el contrario si no tenemos cargado al nivel que necesitamos ya que nuestra granularidad fue muy poca para obtener mayor velocidad, tendremos que acceder a los datos en la base de datos de origen.

- **HOLAP (OLAP Híbrido):** Es un híbrido entre los dos casos anteriores. Las agregaciones se almacenan en un formato multidimensional, sin embargo al navegar el cubo a niveles de detalle más profundos, se utiliza la base de datos relacional.

## Capítulo 3

### 3. Metodología

A pesar de que los proyectos de Inteligencia de Negocios ya son soluciones maduras que llevan varios años en el mercado, aún están distantes de ser soluciones de escala corporativa como lo son herramientas de contabilidad, ERP y otros[8].

En la gran mayoría de los casos, los esfuerzos por construir una infraestructura de Inteligencia de Negocios son de nivel departamental, resultando en un conjunto de soluciones dispersas con altos costos de mantención.

Existen diferentes metodologías que buscan solucionar estos problemas y asegurar que la solución de Inteligencia de Negocios cumpla con los requerimientos de una solución de escala corporativa. Estas metodologías si bien difieren en muchos aspectos, tienen un marco común sobre el cual trabajan.

Este marco común incorpora los siguientes aspectos[8]:

- Las empresas se benefician de tener un datawarehouse y un ambiente analítico que ayude en los procesos de toma de decisiones.
- El objetivo principal de un datawarehouse es publicar información correcta, única y de fácil acceso a los usuarios tomadores de decisión mediante 2 etapas, una etapa de preparación de datos y una etapa de presentación de datos.
- Es importante tener una visión corporativa desde que se diseña una solución, sin la necesidad de tener que implementarlo todo en una primera etapa.
- Datamarts aislados y soluciones departamentales en el largo plazo son un gran problema, a pesar de solucionar problemas de información locales en el corto plazo.

Bajo este marco común, sobresalen 2 metodologías con visiones muy distintas la una de la otra.



### **3.1 Kimball Bus Architecture**

En el enfoque de Ralph Kimball, la etapa de preparación consiste en la extracción de los datos y en la transformación de estos datos en información presentable, mientras que la etapa de presentación esta estructurada de manera dimensional para tener un acceso rápido a la información[9].

Los modelos dimensionales en la etapa de presentación nacen de procesos del negocio y no de necesidades departamentales específicas, asegurando que todos los departamentos con los mismos procesos tendrán la misma visión de la información y una sola versión de la verdad.

En este modelo, la etapa de presentación almacena tanto detalles atómicos como información agregada, sobre los cuales al analista puede navegar y descender hasta los niveles más específicos de información sin la necesidad de reprogramar o utilizar diferentes herramientas.

### **3.2 Corporate Information Factory**

En el enfoque de Bill Inmon, Corporate Information Factory o CIF, la etapa de preparación de datos incorpora un proceso de extracción desde los sistemas operacionales para luego almacenar toda la información en un modelo relacional en tercera forma normal (3NF)[9].

A partir de este repositorio normalizado, la etapa de presentación extrae datos para construir otros repositorios de datos utilizados para exploración y datamining, y además se utiliza para proveer de información a data marts dimensionales departamentales.

A pesar de que la información en los data marts departamentales esta agregada, siempre existe la posibilidad de acceder al máximo nivel de detalle consultando el repositorio normalizado, sin embargo, se requerirá de otras herramientas y conocimientos para hacer esto.

### **3.3 Componentes**

Otro punto de consenso al momento de buscar herramientas para implementar un proyecto de inteligencia de negocios es que la real importancia de la herramienta radica en asistir a los usuarios en la realización de su trabajo. Más allá de la arquitectura y de los avances tecnológicos que una solución pueda incorporar, si no logra realizar las tareas que los usuarios necesitan, es una herramienta que no cumplirá con los objetivos.

Para cumplir con este propósito, existe una serie de componentes que la herramienta debe incorporar para tener una solución a nivel corporativo exitosa. Estos componentes son[8]:

- Una API completa que exponga el 100% de las funcionalidades de la herramienta de manera de poder integrar los sistemas completamente.
- Motor SQL poderoso que permita resolver consultas analíticas complejas en poco tiempo.
- Performance adecuado.
- Apertura de esquemas que le permita trabajar con modelos de diferentes bases de datos sin la necesidad de realizar modificaciones.
- Escalabilidad para pasar fácilmente de los cientos a los miles de usuarios y así poder extender el ámbito de operación de la aplicación.
- Repositorio de metadata único manteniendo la compatibilidad y la posibilidad de crecer sin problemas.
- Integración consistente con la metadata, previniendo que al borrar un componente con dependencias no exista conflicto.

Con estas consideraciones y modelos podemos asegurar de que la solución de inteligencia de negocios que se implementará estará al nivel de las expectativas del cliente y podrá crecer y abarcar a la compañía completa.

### **3.4 Etapas del proyecto**

1) Arquitectura de una solución de BI.

- En esta etapa se realizara una revisión bibliográfica de las principales tendencias y soluciones existentes en el mercado para aplicar, en la medida en que sea posible, estos conceptos en nuestro proyecto.

2) Análisis de requerimientos.

- En este punto buscamos tener una definición de los requerimientos necesarios para que la solución sea efectiva y cumpla con las expectativas de los usuarios finales.

3) Diseño de la solución.

- Con los requerimientos del cliente definidos, procedemos a realizar un diseño que nos permita presentar las ideas y estrategias que utilizaremos para alcanzar los objetivos.

4) Creación de la Infraestructura de BI.

- En esta etapa, realizaremos la instalación de los servidores, la configuración de las redes, la integración de los sistemas y la instalación de la plataforma seleccionada.

5) Implementación de las aplicaciones centrales.

- En esta etapa se desarrollará la aplicación de Ventas Corporativas y la aplicación de Sucursales.

6) Implementación de herramientas adicionales.

- En esta etapa se desarrollarán las aplicaciones de apoyo requeridas para mejorar la usabilidad de las aplicaciones centrales.

7) Resultados y conclusiones.

- Por ultimo, revisaremos los resultados obtenidos, el impacto del proyecto en la organización, el trabajo futuro que se debe realizar y las conclusiones generales.

# Capítulo 4

## 4. Antecedentes

### *4.1 Antecedentes del negocio.*

Correos Chile y en particular su área de Planificación Estratégica y Control de Gestión presentan un problema concreto que necesitan resolver para poder mantenerse competitivos en el mercado.

El problema principal es la incapacidad del área para responder adecuadamente a la creciente cantidad de requerimientos de información de la institución, por lo que buscan una solución concreta que resuelva y automatice estos procesos.

Profundizando un poco en el negocio de Correos Chile, nos damos cuenta de que los principales focos del negocio han ido cambiando conforme cambia la cultura del país. En un principio el negocio del envío postal de las personas era claramente el fuerte de la compañía y millones de cartas eran enviadas dentro y fuera del país cada día, sin embargo, la aparición de tecnologías como el e-mail han provocado grandes cambios en el mercado a los cuales Correos Chile a debido adaptarse.

Hoy, el transporte de paquetería, las alianzas estratégicas con empresas extranjeras y el envío de correo local de grandes empresas de retail y la banca son los principales focos.

Nuevas empresas nacionales y extranjeras han entrado a competir y la búsqueda por ofrecer un mejor servicio y precios más competitivos a pasado a ser de vital importancia para captar clientes en estos nuevos mercados.

Estos nuevos focos del negocio requieren de una visión y estrategia distintas a las utilizadas históricamente por la institución, es por esto que el análisis de información para entender este nuevo escenario y poder responder a los desafíos impuestos es muy importante. Todas las áreas de la organización requieren cada vez más acceso a datos y estudios para cumplir con las metas propuestas por la alta gerencia.

La estructura de correos consta básicamente de un directorio que dirige la empresa. Luego viene el gerente general y lo siguen 11 áreas que conforman la operación de la empresa.



**Figura 4 Organigrama Correos Chile**

La solución que se diseñará debe considerar la diversidad de los usuarios que la van a utilizar. Contamos con usuarios analistas que utilizarán cubos para hacer cruces de información, usuarios operativos que verán el cumplimiento de metas, usuarios gerenciales que desean tener una visión más general del estado del negocio, y usuarios expertos en planificación que buscan obtener mayor conocimiento de la información mediante modelos de minado de datos para poder detectar oportunidades en el mercado.

A pesar de que en el ámbito de esta memoria no se considera la construcción de todas las posibles vistas de información, buscamos establecer metodologías y practicas que permitan expandir el radio de influencia de la aplicación a la mayor cantidad de áreas de manera segura y rápida.

## **4.2 Antecedentes tecnológicos.**

Correos Chile ya ha realizado proyectos que buscan proveer de información a sus clientes internos. Ya posee un Datawarehouse en el cual se cargan datos desde los sistemas operacionales y se consolida información histórica. Sin embargo, a pesar de tener esta información disponible en sus sistemas, no poseen herramientas que permitan a usuarios sin conocimientos técnicos acceder autónomamente a la información.

Para suplantar esta brecha, el área de Planificación Estratégica suplanta esta funcionalidad creando reportes y planillas, pero con la creciente necesidad de información para adaptarse a los nuevos escenarios, el área no es capaz de brindar acceso a la información de manera ágil a todos los usuarios internos que lo requieren.

Mirando el problema desde una perspectiva de infraestructura técnica, la propuesta es implementar un servidor de Datawarehouse que se alimentará directamente del servidor de Datawarehouse corporativo existente en Correos Chile.

La razón de tener 2 Datawarehouse es que el primero ya posee toda la lógica de carga de datos de los sistemas operacionales, luego aprovecharemos este trabajo. Además, la información que contiene está consolidada y apoya a distintas áreas sin mayores problemas, sin embargo, ese Datawarehouse no dispone de herramientas sencillas para accederlo, y dado que buscamos una solución de alto desempeño en el despliegue de información y que sea accesible a una gran variedad de usuarios, se utilizará un segundo Datawarehouse dedicado a esta función.

Siendo consecuentes con las definiciones que hemos establecido, deberíamos decir que continuará existiendo solo un Datawarehouse, solo que su capacidad se expandirá y contará con un nuevo repositorio de datos y nuevos modelos para soportar las aplicaciones analíticas que se construirán.

## Capítulo 5

### 5. Análisis de requerimientos

Después de algunas reuniones con el cliente, se llegó a una definición de los requerimientos que la solución debe abarcar. Esta definición nace de las necesidades específicas del área, de los requerimientos de los usuarios finales, de las capacidades y conocimientos técnicos, de la arquitectura disponible, del presupuesto y de las limitaciones de las tecnologías de hoy en día.

Al internarse en las soluciones que el mercado ofrece, uno se encuentra con diversos participantes y visiones de lo que una solución de BI debe entregar. La primera decisión consiste en determinar cual será la manera de acceder a la información de gestión. Hay casos en que una simple planilla Excel puede solucionar la gran mayoría de los problemas, sin embargo, en otros casos, soluciones mucho más complejas son necesarias.

Hay 2 grandes grupos de soluciones que nos permiten tener un primer filtro para llegar a una solución adecuada[6].

- **Arquitectura Centralizada:** En este tipo de solución, tenemos que todos los elementos necesarios para el almacenamiento, el procesamiento y el despliegue de información se encuentran centralizados en una serie de servidores que responden a peticiones de los clientes.
- **Arquitectura Distribuida:** En estas soluciones, parte del proceso de BI se realiza en el cliente. Puede ser que el proceso de despliegue sea local o que se almacenen datos en el cliente.

En el primer caso tenemos beneficios de mantenibilidad y escalabilidad de la solución, sin embargo, su costo en servidores puede ser mayor y en el caso de aplicaciones que utilicen un browser para acceder a las aplicaciones, tenemos limitaciones en la usabilidad que podemos entregar a los usuarios.

Dependiendo de la cantidad de usuarios que accedan a la solución, este enfoque puede ser considerablemente menos costoso en términos de licenciamiento, ya que el costo del software se calcula en base a la cantidad de procesadores del servidor y no de la cantidad de usuarios que lo acceden.

En el segundo caso, tener que instalar una aplicación en cada cliente y mantenerlo eleva los costos de mantención. También el hecho de que el cliente tenga almacenados datos en su PC puede ser riesgoso para la institución en caso de información muy sensible. La ventaja de este

tipo de soluciones es que las interfaces pueden ser mucho más ricas que en el caso de soluciones que utilizan un browser como medio de acceso.

Un segundo parámetro para filtrar y encontrar la solución más adecuada es identificar los usuarios que utilizarán la herramienta junto con sus capacidades y conocimientos.

Para soluciones de BI, se suele hablar de una estructura piramidal de usuarios. El siguiente diagrama nos muestra como se relacionan la cantidad de usuarios y las diferentes necesidades de BI existentes. Además, se muestran algunas aplicaciones representativas para cada una de las necesidades[7].



Figura 5 Relación Herramientas y Número de Usuarios

En el caso de Correos Chile, realizando un pequeño catastro de las necesidades de los usuarios, nos damos cuenta de que se adecuan de manera bastante precisa al diagrama presentado y que además, todas las necesidades mencionadas son necesidades presentes en la organización.

Contamos con usuarios finales que requieren simplemente de reportes y datos, por ejemplo, de sus ventas. Contamos con usuarios analistas que necesitan realizar análisis complejo sobre los datos y también tenemos a la gerencia y ejecutivos que buscan tener una visión completa del negocio en un dashboard o panel de control.



Con esta clasificación en mente, comenzamos a darnos cuenta de que la necesidad de una herramienta customizable y que logre satisfacer en alguna medida todos los requerimientos de toda la gama de usuarios es fundamental.

Otros factores relevantes para determinar cual es la herramienta adecuada para nuestro caso de negocio son las siguientes:

- **Tiempos de respuesta adecuados:** El acceso a grandes cantidades de información y a capacidad de procesamiento son esenciales dadas las cantidades de datos que se manejan. Esto da un indicio de que necesitamos un buen servidor de base de datos que soporte la solución y que realice el procesamiento centralizadamente. Además, grandes cantidades de usuarios deben ser capaces de analizar datos concurrentemente sin tener problemas de performance.
- **Integración con las tecnologías existentes en la institución:** Dado que existen ya en producción soluciones de bases de datos y datamarts, necesitamos una solución que se integre con esa infraestructura y reutilice la mayor cantidad de recursos posible.
- **Alineada con la estrategia tecnológica de la empresa:** La empresa busca moverse en dirección a tener todos sus sistemas en la WEB a través de un portal corporativo. Dado esto, necesitamos que nuestra solución sea consecuente con esto y sea una solución WEB que se pueda entrelazar con el portal.
- **Libertad sobre el sistema operativo sobre el cual puede correr:** La diversidad de plataformas sobre la cual puede correr la solución también es un plus dado que se están estudiando diferentes alternativas para toda la infraestructura de Correos Chile. Por esta razón, la solución debe ser capaz de correr sobre la mayor cantidad de plataformas de manera de no quedar obsoleta en el corto plazo.
- **Modalidad de licenciamiento y costos asociados:** Con una cantidad no menor de usuarios, un modelo de licenciamiento por usuario puede ser muy costoso y complicado. Un licenciamiento por procesador en una arquitectura centralizada puede ser mucho más adecuada y de esta manera permitir el crecimiento de los usuarios finales sin restricciones comerciales.
- **Respaldo del proveedor, capacitación y soporte:** El respaldo de la empresa que produce el producto, la capacitación que pueda ofrecer, el soporte y nuevas versiones también son un punto a considerar.

Otro requerimiento importante es que la solución debe permitir crecer y generar nuevas herramientas de forma sencilla, de manera de extender las capacidades de la misma y abarcar requerimientos no formulados en la primera etapa. La razón de esto es que una vez que más y más áreas ingresen al sistema y conozcan las potencialidades de una herramienta como esta, sus requerimientos de seguro crecerán y más aún, se diversificarán abarcando cosas que pueden no estar contempladas en el diseño del producto. Por esta razón, la solución debe permitir extenderse y evolucionar.

# Capítulo 6

## 6. Diseño

La etapa de diseño de la solución busca asegurar que la aplicación y todos los componentes del proyecto cumplan efectivamente con los objetivos propuestos antes de realizar la implementación.

En un proyecto con requerimientos ambiguos y expectativas altas, esta etapa nos permite levantar ideas y aterrizar conceptos para llegar a una solución exitosa.

El diseño de la solución se divide en el diseño de los siguientes componentes:

- Arquitectura
  - Hardware
  - Software
  - Redes
  - Integración
  - Seguridad
- Aplicaciones
  - Modelo de datos
  - Funcionalidades
  - Navegación
  - Interacción

## 6.1 Diseño de la arquitectura

Lo primero a definir en el diseño de la arquitectura es la infraestructura tecnológica sobre la cual se ejecutará la solución. Esto implica definir que tipo de hardware utilizaremos, el software y como se integrarán todos estos sistemas con la infraestructura actual de Correos Chile.

Realizando un análisis sobre las posibles alternativas de software presentadas por diferentes proveedores, se opto por utilizar la plataforma DB2 de IBM por su versatilidad y posibilidades de crecimiento a bajo costo cumpliendo con la mayoría de los requerimientos.

DB2 es un motor de datos relacional que incorpora las últimas tecnologías. La experiencia de IBM en estos sistemas viene desde los comienzos de la computación y hace que este motor de datos sea tremendamente poderoso en cuanto a sus funcionalidades y rendimiento.

La versión Datawarehouse que utilizaremos en este proyecto incorpora además herramientas que nos ayudarán en un proyecto de BI.

Estas herramientas son[7]:

- Base de datos relacional (DB2 Universal DataBase) que actúa de repositorio principal de datos.
- Herramienta para crear procesos de carga (DB2 Datawarehouse Manager) y programarlos para que se ejecuten con cierta periodicidad.
- Motor de minado de datos (DB2 Intelligent Miner) con diversos algoritmos para la extracción de conocimiento y explotación de los datos.
- Aplicación para crear modelos dimensionales (DB2 Cube Views).
- Aplicación de visualización, que permite la navegación y presentación de los datos (DB2 QMF).

El motor de base de datos además incorpora Stored Procedures<sup>1</sup>, Triggers<sup>2</sup>, Transacciones, recuperación de fallas, balance de carga de trabajo y mucho más a un costo bajo comparado con otras alternativas de igual nivel.

---

<sup>1</sup> Stored Procedure: Es una subrutina disponible a aplicaciones que acceden a una base de datos.

<sup>2</sup> Triggers: Código procedural que es ejecutado automáticamente en respuesta a cierto evento en una tabla o base de datos en particular.

A pesar de que la manera en que almacenamos los datos, como los procesamos y como modelamos cubos es parte de este proyecto, nuestro foco principal es la distribución de la información, por lo que a pesar de que un motor de datos potente es importante, más importante es como podemos presentar y dar acceso a información de gestión a los usuarios de la organización.

Es aquí donde entra a participar DB2 Alphablox. Alphablox es un conjunto de objetos y servicios J2EE orientados al despliegue de información en un navegador. Alphablox y las aplicaciones que se creen utilizando sus objetos y servicios deben correr sobre un servidor de aplicaciones que sea robusto y permita la concurrencia de muchos usuarios navegando y manipulando la información.

El beneficio de trabajar sobre un servidor de aplicaciones y en ambiente Web además de robustecer la solución, nos permite ampliar nuestra gama de herramientas, permitiéndonos trabajar con componentes J2EE independientes de Alphablox, ampliar y modificar el comportamiento de los objetos en Alphablox y además trabajar en tecnologías que corren sobre un browser como Javascript<sup>1</sup>, AJAX<sup>2</sup>, Flash<sup>3</sup> y DHTML<sup>4</sup>.

Alphablox nos permitirá desplegar gráficos, tablas dinámicas o cubos, tablas y reportes, además de permitirnos manejar usuarios y otras facilidades en una arquitectura centralizada y segura, en la cual el cliente solo debe poseer un navegador para ingresar al sistema mientras que todo el procesamiento se realizara en el servidor central.

Una plataforma como esta requiere en términos de infraestructura lo siguiente:

- Servidor de Datos: Un servidor que tendrá el propósito de almacenar y responder a las solicitudes de información de otros servicios. Además, tendrá que procesar la información para tenerla en niveles de agregación adecuados para responder con la mayor rapidez a consultas complejas de los usuarios.
- Servidor de Aplicaciones: Servicio que contiene todas las aplicaciones del sistema y que responde a las solicitudes y pedidos de los usuarios. Es el servidor de aplicaciones el que se conecta con el servidor de datos para obtener la información.
- Servidor de Usuarios: Es el servidor que contiene todos los usuarios del sistema y sus niveles de acceso. Con esto tenemos una solución segura que desplegará la información correcta según sea el perfil del usuario.

---

<sup>1</sup> Javascript: Lenguaje que permite ejecutar código en un navegador en el computador cliente.

<sup>2</sup> AJAX: Sigla para Asynchronous JavaScript and XML. Técnica de desarrollo Web que busca entregar mayor interactividad a aplicaciones Web intercambiando pequeñas cantidades de datos con un servidor sin la necesidad de refrescar la pagina Web completa.

<sup>3</sup> Flash: Lenguaje que permite crear animaciones basadas en vectores.

<sup>4</sup> DHTML: Sigla para Dynamic HTML. Termino usado para referirse a un conjunto de tecnologías utilizadas para crear paginas Web interactivas y animadas.

- Computadores Cliente: Cada cliente accederá a la solución a través de su navegador, lo que implica que cada computador efectivamente debe tener instalado un navegador compatible con la solución y tener acceso al servidor de aplicaciones.
- Red TCP-IP: Mediante la Red TCP-IP cada cliente se conectará al servidor de aplicaciones. Dependiendo de la complejidad de las aplicaciones y la cantidad de usuarios, necesitaremos mayor ancho de banda para tener una interacción fluida.

Luego de reiteradas reuniones con el cliente y con las áreas involucradas en este proyecto, se determino que la infraestructura tecnológica consistirá en los siguientes componentes:

- Hardware:
  - Servidor
    - Servidor HP Proliant con procesador Intel Xeon a 3,4 GHz.
    - Capacidad para 4 procesadores.
    - Dos discos duros de 10.000 RPM en RAID 0 para el sistema y aplicaciones más dos discos duros de 15.000 RPM para datos.
    - 2 GB de memoria ram.
    - Es un servidor de bajo costo y gran capacidad de crecimiento.
  - Redes
    - El sistema se integrará a la red corporativa de Correos Chile.
    - Red de 100Mbit/seg en base a protocolo TCP/IP.
  - Clientes
    - Cada cliente utiliza un computador conectado a la red corporativa y con capacidad para ejecutar Microsoft Internet Explorer 5 o superior o en su reemplazo, Mozilla Firefox.

Dado que el software que se utilizará tiene un licenciamiento por Procesador, es importante dimensionar correctamente la necesidad de procesamiento para no incurrir en gastos innecesarios. Un procesador extra significa prácticamente duplicar el costo en licencias de software.

Para este proyecto, el dimensionamiento del hardware se basó en 2 puntos:

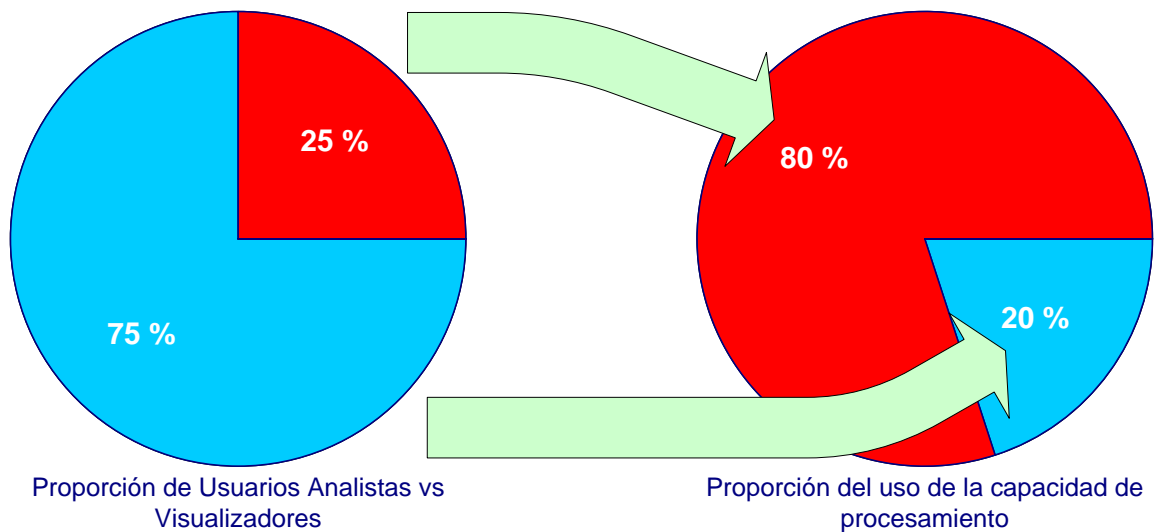
- Recomendación de expertos
  - Expertos de IBM apoyaron el proceso de selección del hardware y realizaron sus recomendaciones en base a la cantidad de datos y de usuarios que se proyectaban.
- Benchmark en sistemas similares
  - Se realizó una prueba de concepto en un sistema similar y se midió el tiempo de respuesta con una cantidad de usuarios concurrentes para verificar que estuviera en niveles adecuados.

El sistema se pensó en una primera etapa para atender a 250 usuarios, de los cuales se espera una concurrencia del 30%, es decir, el hardware y redes deben soportar el acceso concurrente de 75 usuarios.

En una segunda etapa, y dependiendo del éxito del sistema y del porcentaje de penetración que tenga la tecnología, se busca llegar a 500 usuarios con 150 de ellos concurrentes.

Es necesario destacar que no todos los usuarios son iguales. Existen principalmente 2 tipos de usuarios en el sistema, el primero es el usuario que visualiza información y el segundo es el usuario que interactúa y analiza información.

En términos de proporciones, el primer tipo corresponde a un 75% de los usuarios totales y el segundo al 25% restante aproximadamente, sin embargo, el segundo grupo es responsable del 80% del uso de la capacidad de procesamiento, por lo que el crecimiento en usuarios debe ser medido en términos de cómo progresa esta distribución.



**Figura 6 Relación entre usuarios y uso de procesador**

La infraestructura seleccionada permite crecer de manera sencilla en el futuro, por lo que el crecimiento de usuarios es solo un problema de costos en licenciamiento del software.

Otro factor importante en esta decisión es que ya se contaba con hardware similar dentro de la organización, lo que favorecía a estos productos frente a otros de la competencia ya que su mantención e integración son más sencillas.

En términos de redes y del software en los clientes, se reutilizó la arquitectura existente. En esta primera etapa se consideró, en base a las pruebas de performance realizadas, que la red de 100 Mbit/seg<sup>1</sup> con la que cuenta Correos Chile es más que suficiente dado que su uso actual es bastante reducido.

## **6.2 Diseño del modelo de datos.**

El diseño del modelo de datos tiene por objetivo definir de qué manera nos alimentaremos de los sistemas actuales y como le entregaremos datos a nuestras aplicaciones pensando en que debemos cumplir con un nivel de servicio adecuado a las expectativas de los usuarios. Además de esta definición, cada aplicación que se implementará tendrá un modelo de datos propio.

<sup>1</sup> Mbit/seg: Sigla para referirse a Megabits por segundo.

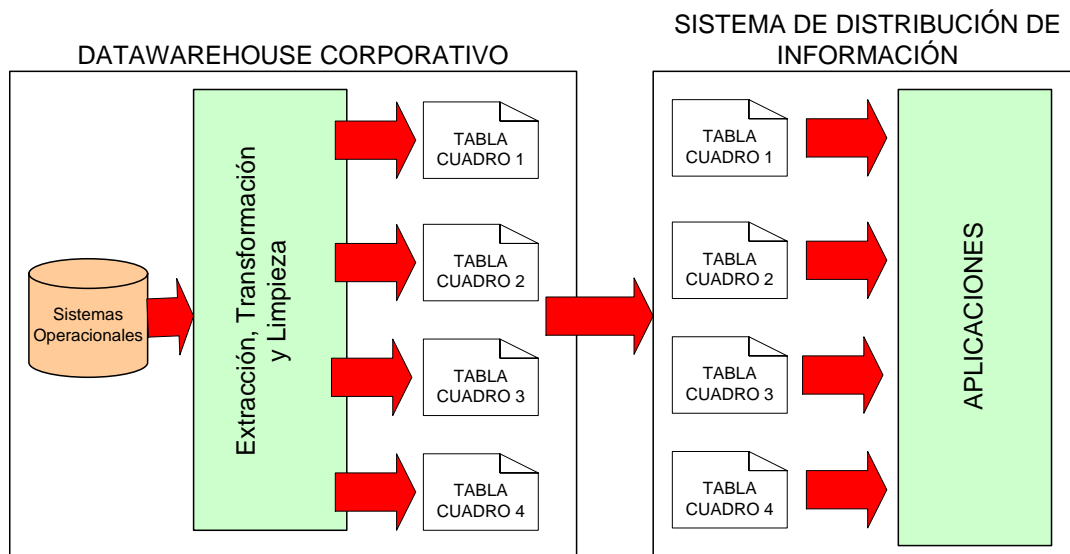
Con las proyecciones de cantidad de usuarios y como se componen entre analistas y visualizadores, más las restricciones de hardware y costos que ya tenemos, necesitamos modelar los datos de manera de poder cumplir las expectativas de los dos grupos de usuarios y mantener un nivel de respuesta adecuado.

Considerando que no poseemos la infraestructura y el poder de procesamiento suficientes para tareas complejas sobre grandes cantidades de datos, lo que haremos será modelar los datos pensando en utilizar la menor capacidad del servidor de gestión y transferir esa complejidad al servidor de Datawarehouse de la compañía donde se generaban los reportes anteriores. De esta manera nos beneficiamos en 2 formas, la primera es que efectivamente no necesitaremos de un gran servidor para cumplir con los requerimientos de los usuarios ya que la información llegara agregada en los niveles necesarios a nuestros sistemas y la segunda es que reutilizaremos tecnología ya incorporada en la empresa, lo que implica que el área de Planificación Estratégica podrá seguir trabajando en las plataformas que ya están acostumbrados y podrán seguir generando los datos ahí.

El lado negativo es que para poder responder a todos los requerimientos, necesitaremos replicar parte de los datos en varias tablas resumen similares, pero que tendrán distintos niveles de agregación o atenderán áreas diferentes.

También el proceso para genera nuevos reportes implicará la generación de tablas en el Datawarehouse existente, crear las tablas equivalentes en el sistema de gestión, crear un procedimiento de la carga, un procedimiento de actualización y luego la visualización.

El modelo queda resumido en el siguiente diagrama:

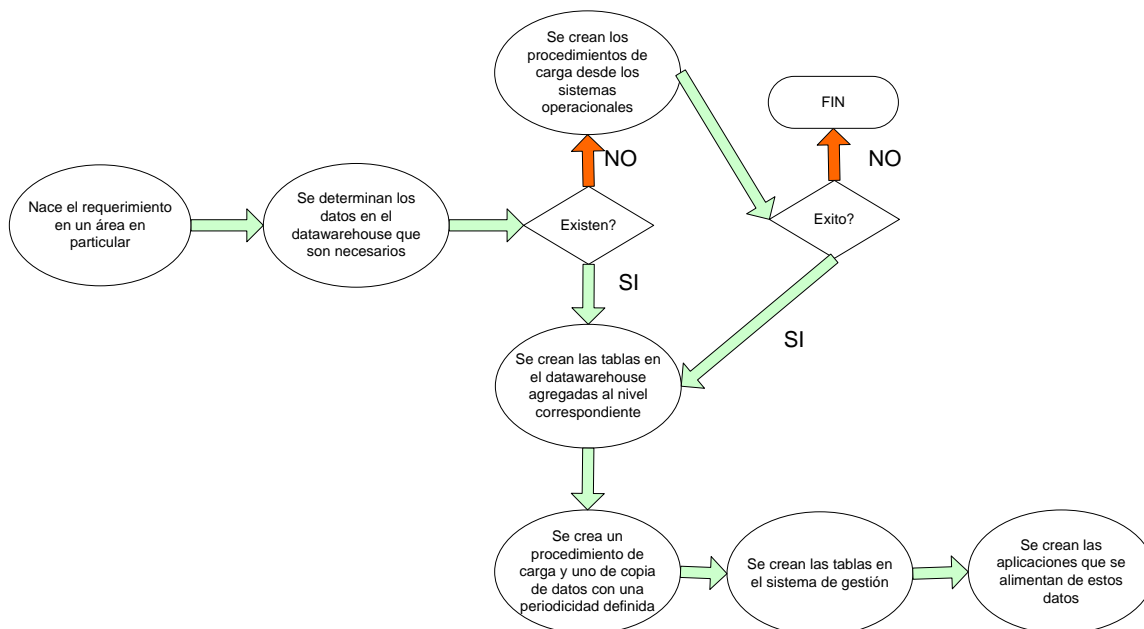


**Figura 7 Modelo de carga de datos**



Cada una de las tablas alimentará de datos a un elemento en la aplicación, de manera de tener la menor cantidad de datos posibles y necesitar de la menor capacidad de procesamiento, transfiriendo el costo al Datawarehouse y al procedimiento de actualización.

El procedimiento de creación de nuevos datos y reportes es el siguiente:



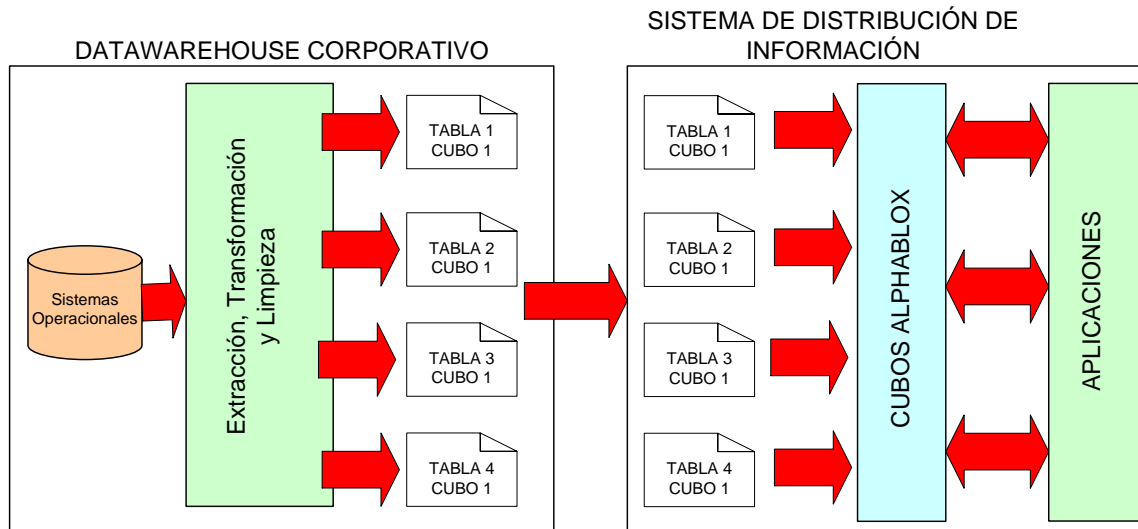
**Figura 8 Modelo de creación de nuevos datos**

Una vez que los datos están en nuestro sistema, tenemos 2 tipos de datos, los datos para tablas y gráficos y los datos para cubos. La diferencia es que dado que en un cubo la capacidad de interacción es mucho mayor que los filtros que podemos definir en una aplicación normal, la cantidad de datos también es mucho mayor y la capacidad de procesamiento necesaria también. Sin embargo, parte de nuestros objetivos es entregar herramientas al porcentaje de analistas dentro de los usuarios, por lo que es necesario contar con este tipo de solución y no solo contar con un dashboard informativo.

El modelo multidimensional, al contrario del modelo relacional, busca entregar herramientas de análisis más complejas que permitan a los usuarios navegar la información y descubrir nuevo conocimiento. Para ello, necesitamos construir cubos que resuelvan problemáticas de negocios particulares.

Bajo el modelo sobre el que estamos trabajando y utilizando de mejor manera las herramientas que tenemos, los pasos para construir los cubos es similar a la establecida anteriormente solo que incorporamos una capa anterior a la capa de las aplicaciones en donde definiremos y guardaremos los cubos construidos.

Estos cubos son cubos ROLAP, luego se alimentan de información en tablas relacionales, por lo que el procedimiento de generación de datos es exactamente el mismo.



**Figura 9 Modelo de carga de datos para cubos**

Los procesos de copia desde el datawarehouse corporativo al sistema de distribución de información de gestión pueden realizarse de diversas maneras con distintos costos asociados.

Entre las alternativas más caras se encuentran la compra de conectores especializados y herramientas de ETL que realizan la copia y carga de tablas de manera automática. Los ambientes de desarrollo de estos procesos en este tipo de herramientas son ambientes gráficos en los cuales uno puede extraer, transformar y cargar información de manera muy sencilla.

El requerimiento en el caso de este proyecto es simplemente la copia de tablas de una base de datos a otra, por lo que la opción anterior resulta sobredimensionada para el problema en particular, y dada la restricción de presupuesto impuesta, se descarta automáticamente.

Otra opción es realizar un procedimiento manual, en donde se realice la extracción de los datos de las tablas a archivos planos, luego se transfieran por FTP de una maquina a otra y luego se carguen. Esta opción es bastante razonable cuando la cantidad de tablas es pequeña y no hay muchos cambios, lo cual no es el caso en este proyecto.

La opción que se adoptó es aprovechar la facilidad que tiene DB2 para crear Stored Procedures en JAVA. Esto quiere decir que podemos en un Stored Procedure utilizar todas las funcionalidades de Java y de la maquina virtual en la cual esté corriendo. En el modelo normal de Stored Procedures, uno no puede interactuar con elementos fuera del motor de la base de datos, sin embargo, con Java Stored Procedures, podemos hacerlo e incluso podemos utilizar clases externas como cualquier programa Java.

Con esta funcionalidad, lo que hacemos es crear un Java Stored Procedure en DB2 que utilice la clase JDBC de Oracle para acceder al Datawarehouse y de esta manera copie todas las tablas necesarias de un punto a otro. Para saber que tablas debe copiar, creamos una tabla de configuración, en la cual se listan todas las tablas junto con la periodicidad en que se deben copiar. En caso de que la tabla de destino no exista, la creamos utilizando la descripción de la tabla en Oracle.

De esta manera, el procedimiento para transferir una tabla será simplemente crear la tabla en el Datawarehouse, cargar los datos, definir cada cuanto tiempo queremos copiarla y llenar la tabla de configuración.

Los modelos de datos particulares de cada aplicación se revisarán en los siguientes capítulos, pero la metodología de carga y actualización será común a todos ellos.

### 6.3 Diseño de la aplicación.

La aplicación debe considerar en esta primera etapa 2 áreas de negocio con las cuales se trabajará conjuntamente para lograr los objetivos propuestos, sin embargo, las áreas que serán incorporadas en una segunda etapa son muchas más.

El siguiente diagrama refleja los ámbitos de la empresa que se nutren de los reportes generados en el área de Planificación Estratégica y que son el primer target de este proyecto.

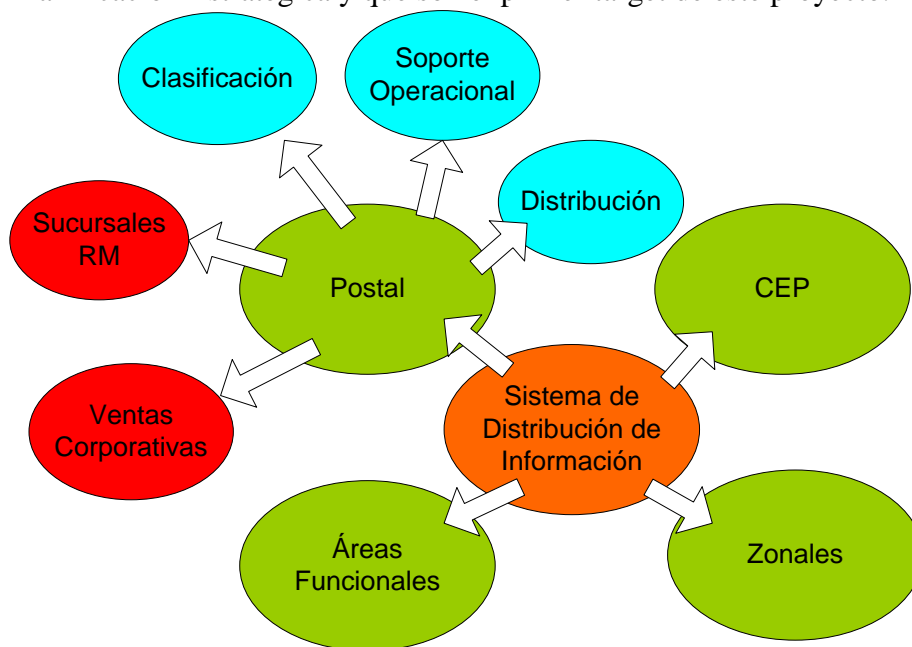


Figura 10 Áreas de la organización

De este conjunto de áreas, esta primera etapa del proyecto considera el área de Ventas Corporativas y el Área de Sucursales RM.

Si el proyecto es exitoso, se ampliará a toda el área Postal para luego continuar con el resto de la organización.

Independiente del área y de los requerimientos particulares de cada una de ellas, nuestra solución debe tener lineamientos generales definidos para todas ellas de manera de estandarizar los procesos y optimizar los tiempos de desarrollo.

A pesar de ser una solución interna de la empresa, nuestra aplicación es una aplicación Web en la cual aplican los principales principios de usabilidad y de arquitectura de la información.

Existen una serie de principios sobre lo que se debe y lo que no se debe hacer tanto en una aplicación Web, en diseño de dashboards y en aplicaciones de distribución de información. Los principales principios que se aplicaron en el diseño de esta aplicación son los siguientes[1]:

- Principios que se deben aplicar
  - Mantener los niveles de navegación al mínimo.
  - Toda la información no puede estar a más de 3 o 4 clicks de distancia.
  - El despliegue de la información y los elementos que se usen deben ser adecuados para la información que se quiere mostrar.
  - La información más importante debe destacarse del resto.
  - Es necesario organizar la información en grupos basados en las funciones de negocio, entidades y su uso.
  - Las comparaciones de valores deben tener sentido y mantener relación en escalas y unidades.
  - Los indicadores deben ser elegidos con cuidado y deben ser significativos para el negocio.
  - La interfaz debe ser atractiva para los usuarios.
- Principios sobre lo que no se debe aplicar
  - No se debe exceder los límites de una pantalla.

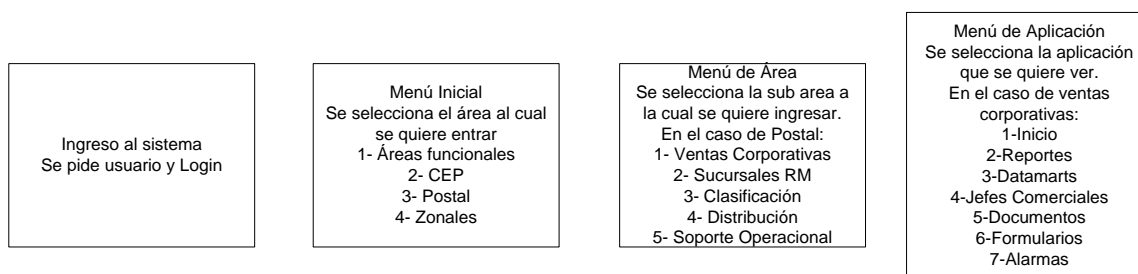
- No se debe entregar un contexto equivocado a los datos.
- No se debe mostrar excesivo nivel de detalles si no es necesario.
- No se debe incorporar mucha variedad de información en una misma pantalla.
- No se debe decorar la pantalla con decoraciones inútiles.
- No se debe abusar de los colores.

Tomando en cuenta estos principios y los requerimientos e ideas del cliente, se llegó al diseño de la aplicación. Este diseño lo separamos en un diseño general que enmarca todas las aplicaciones incluidas las aplicaciones que se desarrollarán a futuro y el diseño particular de cada una de las aplicaciones que sí se abarcaron en este proyecto.

### 6.3.1 Navegación de la Aplicación

La navegación de la aplicación busca integrar de manera sencilla pero visualmente atractiva, el conjunto total de áreas de la organización. El objetivo es presentar una navegación por áreas que valla del nivel más genérico al más específico, permitiéndonos además incorporar información en áreas intermedias con algún nivel de agregación.

El siguiente diagrama muestra la manera de acceder a la información.



**Figura 11 Modelo de navegación**

Esta es la estructura base que se utilizará en todas las aplicaciones. Solo en el último paso nos encontraremos con opciones diferentes para cada caso en particular, pero la lógica para acceder la información será la misma.

### 6.3.2 Aplicación de Ventas Corporativas

Para acceder a la aplicación de ventas corporativas, es necesario pertenecer al área de ventas de la compañía. Esta área presenta 2 tipos de usuarios, un grupo de analistas que analizan e interactúan con la información en busca de oportunidades de negocio, y un segundo grupo más operacional distribuidos en una jerarquía de 2 niveles, los ejecutivos y los Jefes Comerciales a cargo de un grupo de ejecutivos. La proporción de usuarios visualizadores y analistas siguen la proporción presentada anteriormente, es decir, tenemos un 25% de usuarios analistas y un 75% de usuarios visualizadores de información.

Lo primero que necesitamos en esta aplicación es presentarle al usuario un panel de control con la evolución de sus ventas, ya sea a nivel de ejecutivo o jefe comercial.

Los componentes que necesitamos son:

- Resumen de totales
- Variación de la cartera propia
- Principales Alzas
- Principales Bajas
- Evolución por producto
- Evolución acumulada por producto
- Distribución de ventas

Esta información necesitamos poder verla en base a dos filtros, el primero determina cual es la dimensión de los valores que veremos, sus opciones son:

- Mes Nominal
- Acumulado Nominal
- Mes Real
- Acumulado Real

Dependiendo de la selección de este filtro, se desplegará la información correspondiente. El segundo filtro es el mes, en donde podemos seleccionar que mes queremos visualizar.

Dado que esta información es evolutiva, necesitamos poder verla de manera tanto grafica como en números, y así poder tener claro con un solo vistazo cuales son las áreas en donde deberíamos poner mayor atención.

Para tener mayor flexibilidad, necesitamos exportar los datos a una planilla Excel o a un reporte en PDF, de manera de poder compartirlos o profundizar el análisis en herramientas comunes a los usuarios.

Otra sección de la aplicación de ventas corporativas es la sección de reportes. La idea detrás de esta sección es permitir tener una infinidad de reportes preestablecidos con un determinado nivel de interacción para que los usuarios tengan acceso a información que necesitan recurrentemente. En esta sección tenemos el reporte de Mejores Clientes, en el cual obtenemos para nuestro nivel cuales son nuestros mejores clientes y su historia. Los filtros son por año, mes y podemos filtrar por un cliente específico.

La tercera sección son los datamarts. El nombre de esta área es una decisión que se tomo en base a utilizar conceptos conocidos por los usuarios, a pesar de no ser técnicamente correctos.

En esta área tenemos 3 cubos:

- Clientes
- Tráfico
- Jefes Comerciales y ejecutivos

El primero nos permite ver las ventas netas y el presupuesto de este año y el anterior. Las dimensiones que tenemos para navegar son los clientes, el tiempo, el tipo de moneda y los productos.

Con esta herramienta los analistas pueden tener una idea de que productos se mueven en que temporadas del año y que clientes son los que mueven estas transacciones.

El segundo cubo, el de tráfico, nos entrega información similar a la del cubo de clientes solo que esta vez el hecho es el numero de piezas o cartas que se enviaron. También incluye las dimensiones de tiempo, producto y clientes.

El tercer cubo es el de Jefes Comerciales y Ejecutivos. En él se pueden analizar las ventas netas y presupuesto de este año y del anterior por jefe comercial y abrirlo en sus ejecutivos. De esta manera podemos identificar que Jefes Comerciales y Ejecutivos están cumpliendo las metas definidas. Junto con esto tenemos las dimensiones de tiempo, producto y tipo de moneda, de

manera de que podemos saber exactamente para un Jefe Comercial, que producto esta bajo el presupuesto y en que temporalidades del año.

La cuarta sección es la de Jefes Comerciales. En esta área presentamos el mismo panel de control inicial, solo que ahora nos enfocamos en los Jefes Comerciales, permitiéndoles filtrar por sus ejecutivos la evolución de los indicadores.

La quinta sección es la de Documentos. Esta sección es un repositorio de documentos y presentaciones que son interesantes para los usuarios. Es aquí donde se pondrán archivos importantes y comunicaciones que puedan ser de utilidad para un área en particular.

Los documentos pueden estar en un de las siguientes sub secciones:

- Presentaciones
- Histórico de Presentaciones
- Archivos
- Noticias
- Documentos Ejecutivos

La sexta sección es la de formularios. En esta sección se subirán formularios de uso típico por los usuarios como por ejemplo las solicitudes de feriados legales o las fichas de los clientes.

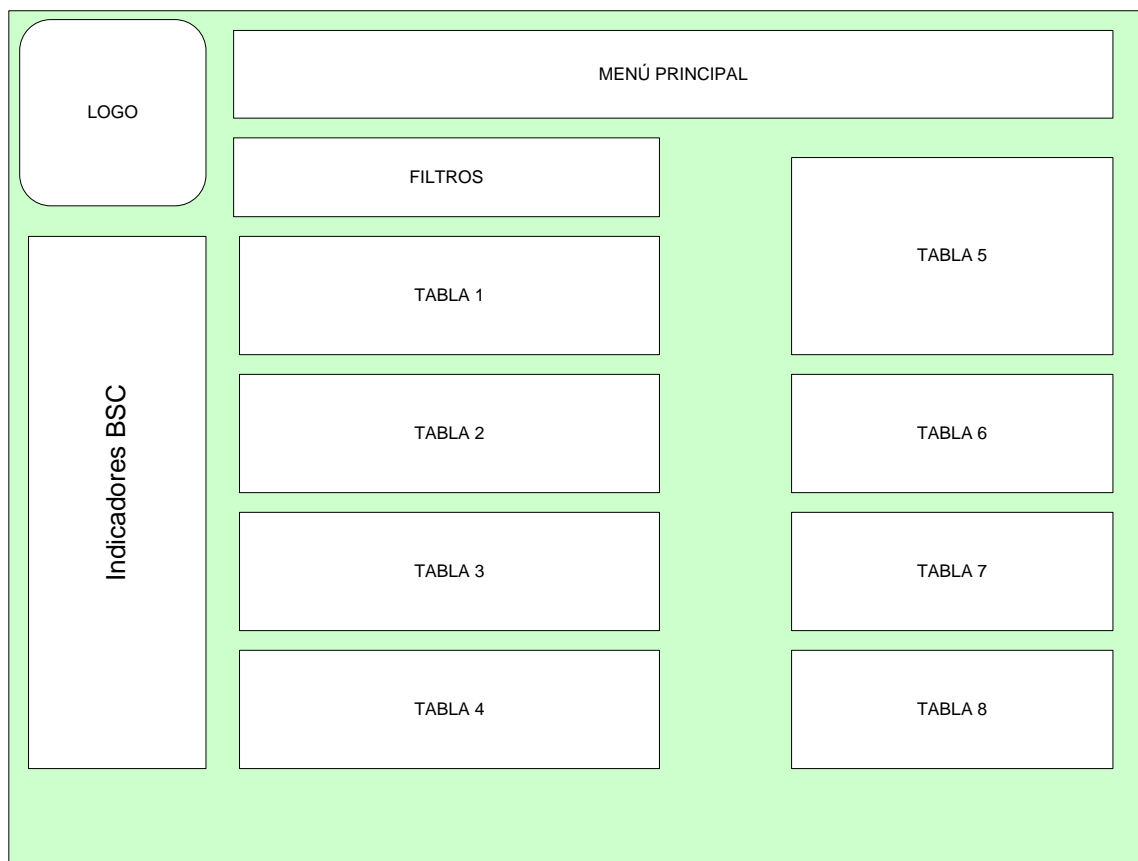
La séptima y última sección es la de alarmas que nos permite administrar nuestras suscripciones a los distintos indicadores. Aquí puedo suscribirme a nuevas alarmas como también puedo revisar el estado actual de los indicadores definidos.

Junto a todas estas alternativas, queremos también tener de manera grafica la evolución de los indicadores del Balanced Score Card corporativo. Estos indicadores para el área de ventas son:

- Ventas Netas
- Segmentos Grandes
- Segmentos Pequeños
- Con Seguimiento
- Sin Seguimiento



Estos indicadores deben estar presentes siempre y deben permitir ver el detalle que lo compone. El siguiente diagrama muestra la estructura básica del panel de control.



**Figura 12 Diagrama de aplicación Ventas Corporativas**

### **6.3.3 Aplicación de Sucursales**

La aplicación de sucursales es muy similar a la aplicación de ventas corporativas en términos de los datos que queremos mostrar al usuario. La gran diferencia con la aplicación anterior es que nos interesa tener una visualización gráfica del estado de los indicadores de gestión en el país.

La definición de cómo se estructura y divide el país es la siguiente:

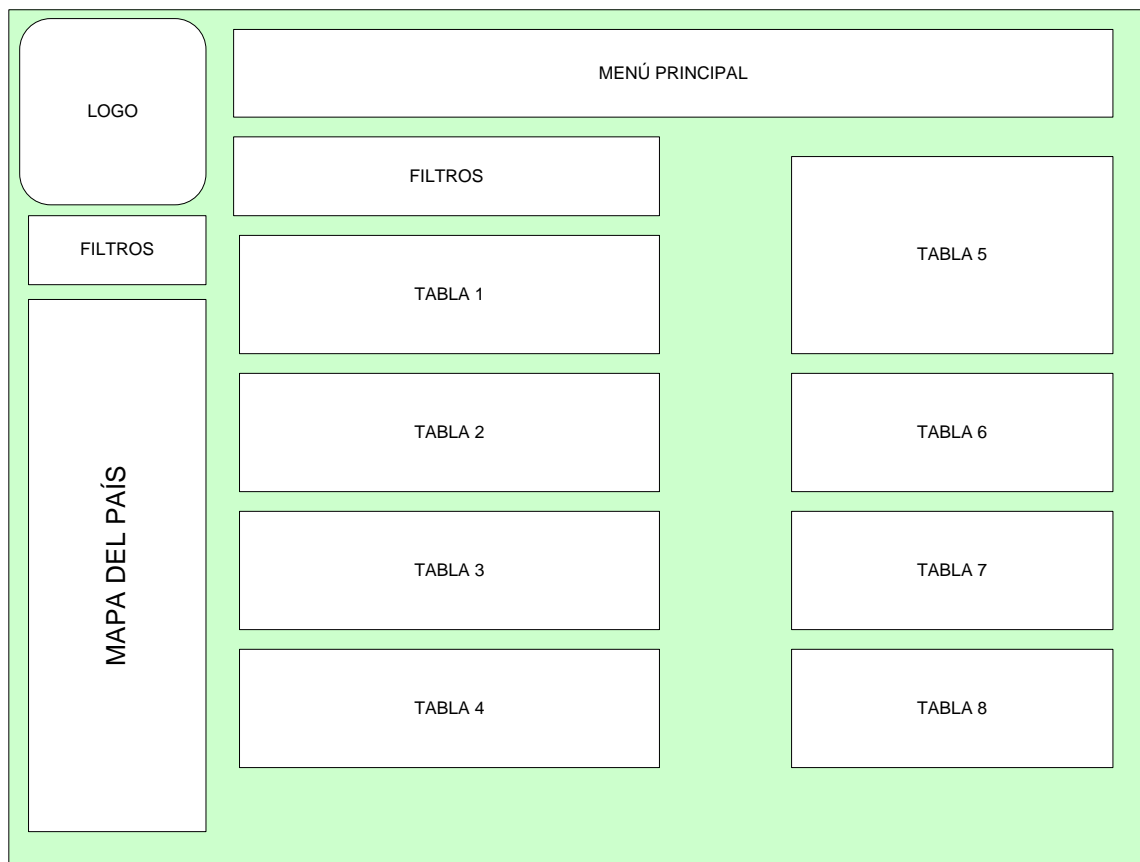
- Regional
  - División en las regiones del país.
- Zonal
  - División en las siguiente agrupaciones:
    - Zona Norte (I-II-III-IV)
    - Zona Centro (V-VI-RM)
    - Zona Sur (VII-VIII-IX)
    - Zona Austral (X-XI-XII)

Los filtros definidos son:

- División: Zonal o Regional
- Indicador: Ingresos, Control Presupuestario o BSC
- Línea: Postal, CEP o Total

Estos filtros se aplicarán sobre un mapa del país que mostrará en colores el estado de los indicadores. Además, al hacer clic en una zona o región, los datos presentados en el panel de control se filtrarán según esa selección.

El resto de las secciones de esta aplicación mantienen la lógica presentada en la aplicación de ventas corporativas, solo que todo será filtrado según la selección que se realice en el mapa. El siguiente diagrama muestra la estructura básica del panel de control de sucursales.



**Figura 13 Diagrama de aplicación Sucursales**

### **6.3.4 Modelo de datos de las aplicaciones**

Siguiendo la definición de cómo trabajaríamos los datos en este proyecto, debemos construir una tabla para cada uno de los cuadros en el panel de control, de esta manera, el acceso a los datos será muy rápido ya que solo se necesitarán aplicar los filtros seleccionados, sin mayores cruces de tablas ni agregaciones ya que la información habrá sido procesada y actualizada en el Datawarehouse.

En la aplicación de ventas, tenemos que podemos filtrar por mes, por tipo de información y por ejecutivo en el caso de la sección de Jefes Comerciales. De esta manera, cada tabla queda de la siguiente manera:

PV_CUADRO1	PV_CUADRO2	PV_CUADRO3	PV_CUADRO4
FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO	FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO	FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO	FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO

PV_CUADRO5	PV_CUADRO6	PV_CUADRO7
FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO	FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO	FILTRO PRCN_MES CDG_EJECUTIVO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR FLAG_AGRUPADO FLAG_EJECUTIVO

**Figura 14 Modelo de datos del portal de ventas corporativas**

En estas tablas almacenamos de manera resuelta el valor de las ventas netas del mes actual, las ventas netas del mes año anterior, las ventas netas del presupuesto y la variación porcentual. Esta información esta almacenada mensualmente, junto con una descripción y al filtro que corresponda.

Además de esto, existe un campo que almacena el código del ejecutivo, de manera de que podemos saber a quien corresponden las ventas. Cuando no se trata de un ejecutivo, el Flag Ejecutivo se apaga de manera de que podemos sumar los datos sin problemas.

La estructura de los ejecutivos y sus jefes se almacena en las siguientes tablas:

PV_ESTRUCTURA	PV_EJECUTIVOS
CDG_JCOM RUT_JCOM NMB_JCOM CARTERA_COMERCIAL	CDG_EJECUTIVO RUT_EJECUTIVO NMB_EJECUTIVO CDG_JCOM

**Figura 15 Modelo de jerarquía de ejecutivos**

Simplemente es una tabla con los Jefes Comerciales y otra tabla con los Ejecutivos, asociados cada uno a un Jefe Comercial.

La aplicación de sucursales sigue una lógica similar, solo que agregamos en las tablas la región. Con este dato, somos capaces de filtrar por región o por zona agrupando según sea necesario. Cabe notar que en algunos casos el resultado de una zona será la suma de sus regiones, pero en otros será el promedio.

PS_CUADRO1_R	
	PRCN_AÑO PRCN_MES REGION NRO DESCRIPCION NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO VAR_POR

**Figura 16** Tabla ejemplo del modelo de sucursales

Los cubos también tienen sus tablas resultas de manera de necesitar la menor cantidad de cruces posibles entre ellas. En el siguiente diagrama se presentan los diagramas de las tablas que alimentan los cubos de Clientes, Jefes Comerciales y Gastos.

PV_IVC_GASTOS	PV_IVC_CLIENTE	PV_IVC_JEFES_COMERCIALES
PRCN_AÑO PRCN_MES CUENTA_CONTABLE CENTRO_COSTO TIPO_DOCUMENTO DESC_DOCUMENTO NOMBRE_PROVEEDOR RUT_PROVEEDOR NRO_FACTURA FECHA_CHEQUE ESTADO_DE_PAGO NRO_DE_CHEQUE NOMBRE_BANCO TIPO_MONEDA NETO_AÑO_ACTUAL NETO_AÑO_ANTERIOR NETO_AÑO_PPTO NETO_MES_ACTUAL NETO_MES_ANTERIOR NETO_MES_PPTO	PRCN_AÑO PRCN_MES CLIENTE PRODUCTO CLIENTE_CENTRALIZADO TIPO_MONEDA NETO_AÑO_PPTO NETO_AÑO_ACTUAL NETO_AÑO_ANTERIOR NETO_MES_ACTUAL NETO_MES_PPTO NETO_MES_ANTERIOR	PRCN_AÑO PRCN_MES PRODUCTO EJECUTIVO JEFE_COMERCIAL TIPO_MONEDA NETO_AÑO_PPTO NETO_AÑO_ACTUAL NETO_AÑO_ANTERIOR NETO_MES_ACTUAL NETO_MES_PPTO NETO_MES_ANTERIOR

**Figura 17** Modelo de datos de Cubo de Gastos, Cliente y Jefes Comerciales

Como vemos, tenemos resuelto en una sola tabla los hechos y dimensiones. Esto es exactamente igual a tener las tablas separadas en términos del modelamiento.

Otras 2 tablas relevantes son las que contienen los usuarios y los productos. Estas tablas se utilizan para obtener mayor información al momento de querer profundizar.

CLIENTES	PRODUCTOS_CONTABLES
CLNT_CODIGO	CDG
CLNT_RUT_CLIENTE	NOMBRE
CLNT_SUCURSAL	GRUPO
CLNT_DV_CODIGO	CODIGO_AGRUPAMIENTO_IVC
CLNT_DV_CLIENTE	NOMBRE_AGRUPAMIENTO_IVC
CLNT_NOMBRE_CLIENTE	MERCADO
CLNT_HOLDING	CODIGO_AGRUPAMIENTO_PPTO
CLNT_CBDR_CDG	NOMBRE_AGRUPAMIENTO_PPTO
CLNT_CLLC_FAX	CODIGO_SUBGRUPO
CLNT_CLLC_DRC	NOMBRE_SUBGRUPO
CLNT_CLLC_CDG_REF	TIPO_INGRESO
CLNT_CDDS_CDG	SUB_TIPO_INGRESO
CLNT_CLLC_FONO	LINEA
CLNT_CLLC_CONTACTO	AMBITO
CLNT_CLLC_MAIL	DESC_GENERO
CLNT_CMNS_CDG	CDG_GENERO
CLNT_CNCS_CDG	TIPO_ESPECIE
CLNT_CNVN_CDG	CDG_TIPO_ESPECIE
CLNT_CTGR_CDG	GRUPO_TRAFICO
CLNT_EMPR_CDG	DIVISION
CLNT_ESCL_CDG	TIPO_CLASIFICACION
CLNT_GCLL_CDG	AGRUPAMIENTO_HOLDING
CLNT_MDLP_CDG	
CLNT_PSTL_CDG	
CLNT_RPLG_RUT	
CLNT_RPLG_RUT_DV	
CLNT_RPLG_NMB	
CLNT_RPLG_CRG	
CLNT_RGNS_CDG	
CLNT_SGMN_CDG	
CLNT_SCVN_CDG	
CLNT_TPCL_CDG	
CLNT_ZOAC_CDG	
CLNT_CREATION_DATE	
CLNT_CREATED_BY	
CLNT_LAST_UPDATE_DATE	
CLNT_LAST_UPDATED_BY	
CARTERA_FINAL	
FECHA_VIGENCIA_CARTERA	
CLIENTES_CENTRALIZADOS	
INDUSTRIA	
CDG_CARTERA_FINAL	
SEGMENTO_D_POSTAL	
SEGMENTO_D_CEP	
CLNT_NEW_OLD	
CLNT_CDDS_NOMBRE	
CLNT_CMNS_NOMBRE	
CLNT_CNCS_NOMBRE	
CLNT_CTGR_NOMBRE	
CLNT_ESCL_NOMBRE	
CLNT_GCLL_NOMBRE	
CLNT_RGNS_NOMBRE	
CLNT_SGMN_NOMBRE	
CLNT_SCVN_NOMBRE	
CLNT_TPCL_NOMBRE	
CLNT_ZOAC_NOMBRE	
CLNT_DSTC_CDG	
CLNT_DSTC_NOMBRE	
CLNT_SSGM_CDG	
CLNT_SSGM_NOMBRE	
CLNT_FCH_INGRESO	
CLNT_FACTURA_UNICA	
CLNT_CREATED_BY_NOMBRE	
CLNT_LAST_UPDATED_NOMBRE	

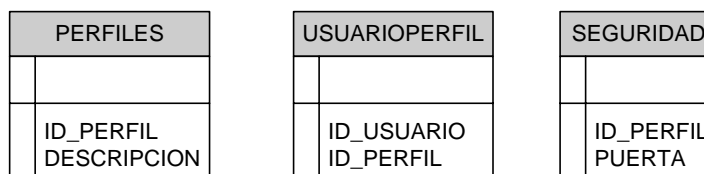
**Figura 18 Modelo de tablas asociadas**

Para hacernos cargo de la seguridad, manejamos 3 tablas muy sencillas que hacen referencia a Perfiles de usuario, usuarios y puertas. El concepto es que cada objeto o conjunto de ellos puede

ser definido como una puerta, luego, el acceso a la información es a nivel de objeto, los cuales pueden ser objetos de Alphablox, páginas Web o incluso datos específicos en una tabla.

La validación de la seguridad se hace mediante una función creada en DB2 que devuelve un 1 si tiene acceso o un 0 si no tiene acceso buscando en las tablas el nombre de la puerta y el usuario.

Las tablas asociadas son las siguientes:



**Figura 19 Modelo de tablas de seguridad**

### 6.3.5 Diseño de Alarmas y Suscripciones.

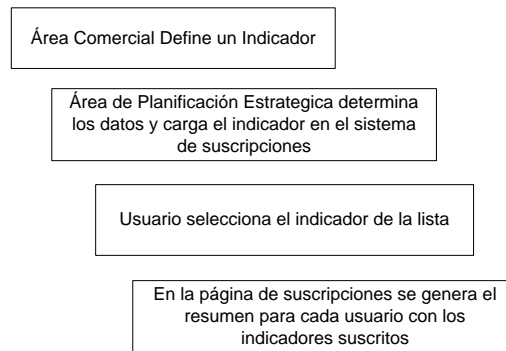
Alarmas y Suscripciones es un sistema que busca potenciar y dar mayor usabilidad al sistema.

Principalmente nuestro proyecto distribuye información y permite a distintos usuarios interactuar con ella, sin embargo, para indicadores críticos que requieren una revisión constante, es muy molesto necesitar ingresar y navegar hasta la información reiteradamente solo para encontrarse que un indicador esta dentro de sus niveles aceptables y que no se requiere ninguna acción.

El sistema de suscripciones lo que hace es presentar una página resumen, con indicadores definidos por el usuario y con links a donde el usuario puede encontrar más información relacionada con él.

El flujo de información para lograr esto está expuesto en el siguiente diagrama:





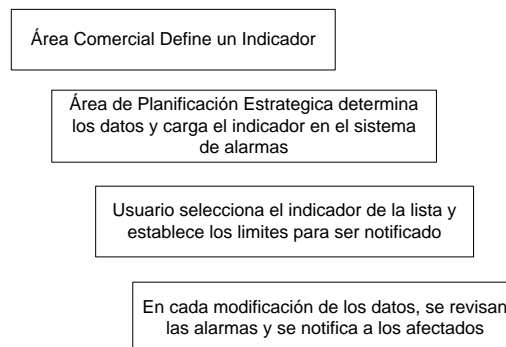
**Figura 20 Modelo de aplicación suscripciones**

La definición de un indicador es simplemente una query que se ejecuta para cada uno de los usuarios que se suscriben de manera de obtener los datos correspondientes a su perfil.

El sistema de Alarmas es muy parecido al sistema de suscripciones y utiliza la misma lógica. La diferencia es que en Alarmas, lo que buscamos es alertar al usuario cuando un indicador sobrepasa cierto nivel superior o inferior.

Esta herramienta además puede utilizarse como medio de control ya que un usuario con gente a su cargo puede definir alarmas a su equipo, ya sean visibles o no visibles para ellos, y de esta manera saber por ejemplo si las ventas de un ejecutivo han bajado mucho o si por el contrario, han subido sobre las expectativas y se desea investigar las acciones que influenciaron esto.

El siguiente diagrama muestra el flujo de información de este sistema.



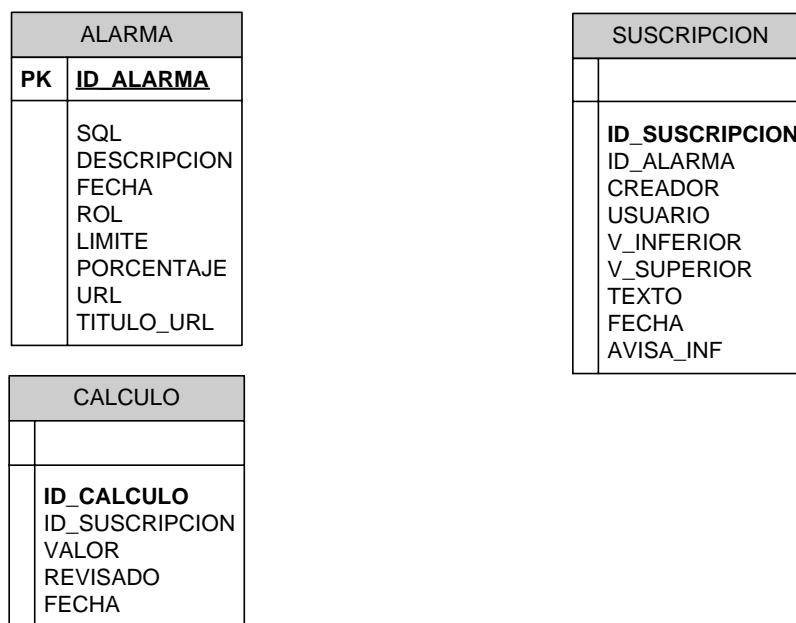
**Figura 21 Modelo de aplicación alarmas**

El sistema de alarmas avisará por correo electrónico a los interesados de un determinado indicador, teniendo en cuenta además que por ejemplo un indicador de ventas debe considerarse al nivel correcto del suscriptor, es decir, un ejecutivo verá sus ventas, mientras que un supervisor de sucursal verá las ventas agregadas de todos los ejecutivos a su cargo.

El proceso de definición de una alarma es similar al de suscripciones, solo que esta vez el administrador define si tendrá cotas superiores o inferiores o si representará un porcentaje de otro valor.

Nuevamente la clave en una alarma esta en la Query que se defina ya que esta es la que se ejecutará y evaluará, de manera de que si retorna 1 debemos enviar un aviso al suscriptor con la información relevante.

Además, en las suscripciones y alarmas debemos guardar un link a la pagina que puede ser de ayuda para encontrar más información respecto al valor de ese indicador par poder tomar acción.



**Figura 22 Modelo de tablas de alarmas y suscripciones**

Contamos con 3 tablas para alarmas, ALARMA, CÁLCULO y SUSCRIPCIÓN. La primera guarda la definición de la alarma, el SQL o la Query que se debe ejecutar, una descripción, el ROL que tendrá acceso a ella, que tipo de limite se usará y si es un porcentaje de otro valor. También guardamos la URL a la que re direccionaremos en caso de la activación. La tabla de CALCULO guarda para una alarma y una suscripción, el valor de la evaluación de la query. Por ultimo, la tabla de suscripción define quien esta suscrito a que alarma, los limites inferiores y superiores y si es a si mismo a quien debe revisar o a algún dependiente de el.

### 6.3.6 Diseño de Mensajería.

Una de las principales actividades de un analista es compartir y colaborar sobre los resultados y datos que maneja. Es por esta razón que se incluyo dentro del sistema un área de mensajería simple.

El diseño de este sub sistema en esta etapa es solo una aplicación que permita enviar mensajes a los pares y al supervisor directo, de manera de poder comentar la información que puede estar viendo en pantalla y el plan de acciones que se ejecutarán.

En esta primera etapa, este sistema es bastante simple y no incorpora mayores funcionalidades que las ya mencionadas, pero es importante destacar que se busco incorporar desde el comienzo la intención de que el análisis de información y las comunicaciones que esto genera estuvieran integrados en el mismo sistema.

# Capítulo 7

## 7. Implementación

### *7.1 Instalación y preparación de los equipos.*

Los equipos y herramientas que se utilizaron en este proyecto permiten una instalación fácil y rápida. La integración con sistemas de autenticación de usuarios LDAP esta completamente resuelta y no se requiere mayor trabajo.

El software instalado en el servidor es el siguiente:

- Microsoft Windows Server 2003 (Sistema Operativo)
- DB2 UDB DWE V8.1 (Base de datos)
- DB2 Cube Views V8.1 (Herramienta para modelar Cubos ROLAP)
- IBM Websphere 5.1 (Servidor de Aplicaciones)
- DB2 Alphablox V8.1 (Framework BI)
- DB2 Intelligent Miner V8.1 (Herramienta para minado de datos)
- Macromedia Dreamweaver (Editor HTML)

Una vez que tenemos el software instalado y el equipo conectado a la red, procedemos a realizar pruebas de conectividad intentando acceder al servidor de aplicaciones y a la base de datos utilizando equipos remotos conectados a la red.

En esta primera etapa, la solución que estamos construyendo aún no es un sistema crítico para la organización, por lo que no es necesario contar con ambientes separados de desarrollo y producción y cumplir con niveles de servicio exigentes. Sin embargo, para realizar modificaciones, experimentos y desarrollos futuros con tranquilidad, se creó un ambiente de desarrollo paralelo al ambiente de producción.

Este ambiente tiene exactamente la misma configuración de software que el ambiente en producción pero se ejecuta sobre un hardware menos poderoso y de menor costo,

específicamente en un pc Pentium 4 con 1 GB en memoria Ram. Las pruebas realizadas en este ambiente de desarrollo muestran un desempeño muy bueno mientras la cantidad de usuarios que la acceden sean entre 2 y 4 usuarios concurrentes, degradándose el performance rápidamente cuando este número se sobrepasa.

Nuestro ambiente de producción no tiene un contrato de nivel de servicio (Service level agreement), por lo que no es necesario preocuparse por el uptime del sistema ni mucho menos por el performance de las aplicaciones en función de cumplir con este contrato, sin embargo, uno de los objetivos del proyecto es que los usuarios adopten esta tecnología, lo que nos obliga a prestar un nivel de servicio acorde a las expectativas y necesidades de los clientes, independiente de si estos requerimientos están o no en un contrato.

## **7.2 Implementación de las Bases de Datos**

Un vez que tenemos la infraestructura de software y hardware instalada y funcionando, procedemos a la implementación de las bases de datos. Este procedimiento implica los siguientes pasos:

- Definición de Table Spaces<sup>1</sup>.
- Creación de la Base de datos.
- Creación de las tablas.
- Creación de Stored Procedures de carga.
- Primera carga de datos.

Nuestro ambiente de producción cuenta con un disco duro dedicado para los datos de esta aplicación, por lo que crearemos 3 table spaces en este disco:

- Table space temporal: Table space de 5 GB en el cual se almacenan tablas temporales, tablas precalculadas y otros elementos temporales de la base de datos.
- Table space del sistema: Table space de 5 GB en donde se almacenan logs y definiciones de stored procedures y cubos.
- Table space de datos: Table space de 30 GB en donde almacenaremos todos nuestros datos.

---

<sup>1</sup> TableSpace: Lugar de almacenamiento que utiliza una Base de Datos.

Con la descripción de las tablas en lenguaje DDL generado por los programas de diseño de las bases de datos, generamos todas las tablas asociadas al proyecto. DDL es el lenguaje utilizado en SQL para definir los campos y tipos de datos de una tabla y es estándar en la industria. Lo único que cambia de uno a otro motor de base de datos son los nombres de los tipos de datos o los tipos de datos particulares de un motor en particular, por lo que al generar el DDL hay que especificar que se utilizará un motor DB2.

La creación de los stored procedures se realiza en la aplicación de desarrollo de DB2. Esta aplicación nos permite definir un proyecto en el contexto de una base de datos para construir stored procedures en SQL o JAVA, Triggers y otros elementos de DB2.

El stored procedure de carga de datos esta construido en JAVA, por lo que se estructura muy similar a un programa en este lenguaje. A continuación se presenta un ejemplo del código de un Stored Procedure en este lenguaje.

```
/**
 * JDBC Stored Procedure DB2ADMIN.carga1
 */
package PKG61102081357420;

import java.sql.*;          // JDBC classes

public class Carga1
{
    public static void carga1 ( ResultSet[] rs1 ) throws SQLException, Exception
    {
        // Get connection to the database
        Connection con = DriverManager.getConnection("jdbc:default:connection");
        PreparedStatement stmt = null;
        boolean bFlag;
        String sql;

        sql = "SELECT PROCSHEMA, PROCNAME FROM SYSCAT.PROCEDURES";
        stmt = con.prepareStatement( sql );
        bFlag = stmt.execute();
        rs1[0] = stmt.getResultSet();
    }
}
```

La ventaja de poder construir stored procedures en Java que se ejecuten dentro de la Máquina Virtual donde se está ejecutando DB2 es que podemos acceder clases externas al igual que lo haríamos en un programa en Java. Esta funcionalidad no presente en otros lenguajes para crear Stored Procedures nos permitira tener 2 funcionalidades esenciales de nuestro sistema. La primera es la capacidad de acceder a una base de datos Oracle utilizando la clase JDBC correspondiente y la segunda es poder enviar correo electronico para nuestro sistema de alarmas.

## 7.3 Implementación de las aplicaciones

El primer paso para construir una aplicación en Alphablox es definirla en el panel de administración de Alphablox y definir su fuente de datos.

Para esto, ingresamos a la pantalla de administración de Alphablox y en el menú de aplicaciones creamos una nueva aplicación. Le llamaremos “Sistema de Distribución de Información”. Luego, en la pagina de fuentes de datos, creamos una nueva conexión a DB2 a la base de datos “correos”. A esta nueva fuente de datos le llamaremos de igual manera, “correos”.

Una fuente de datos en Alphablox es el nexa con una base de datos. Con ella podemos acceder a la base de datos si tenemos el perfil para hacerlo desde cualquier aplicación.

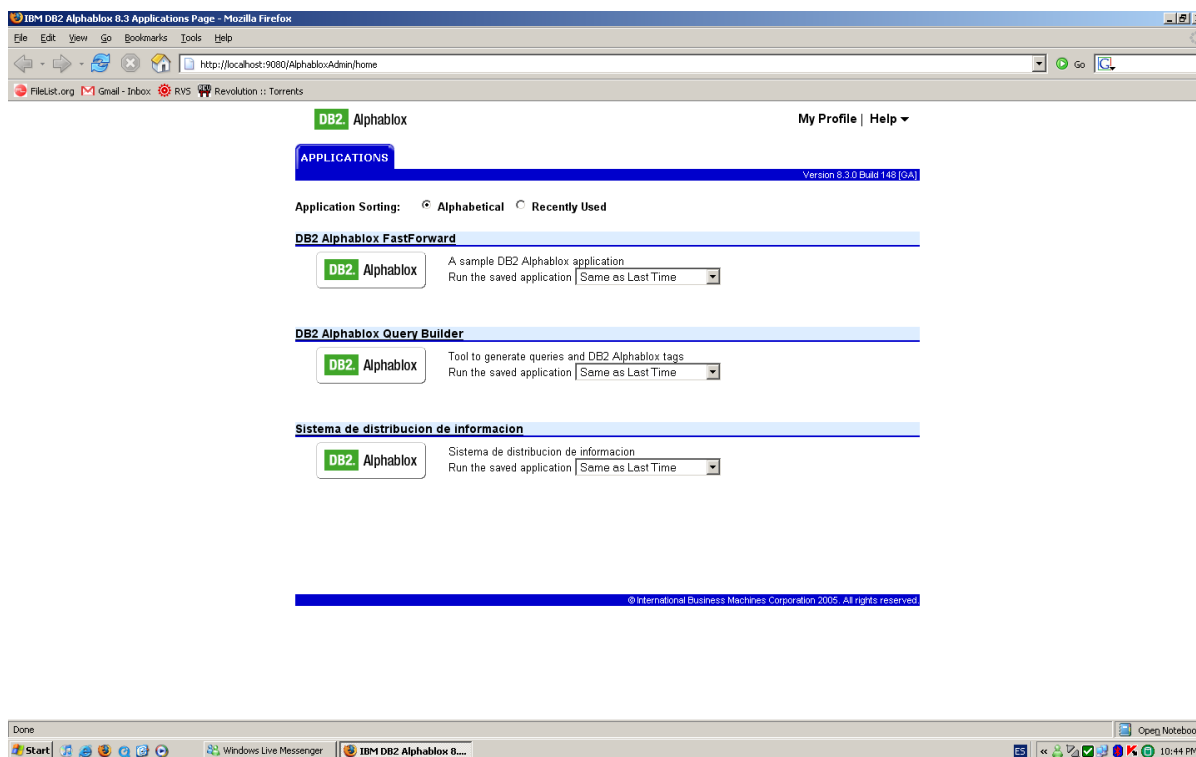
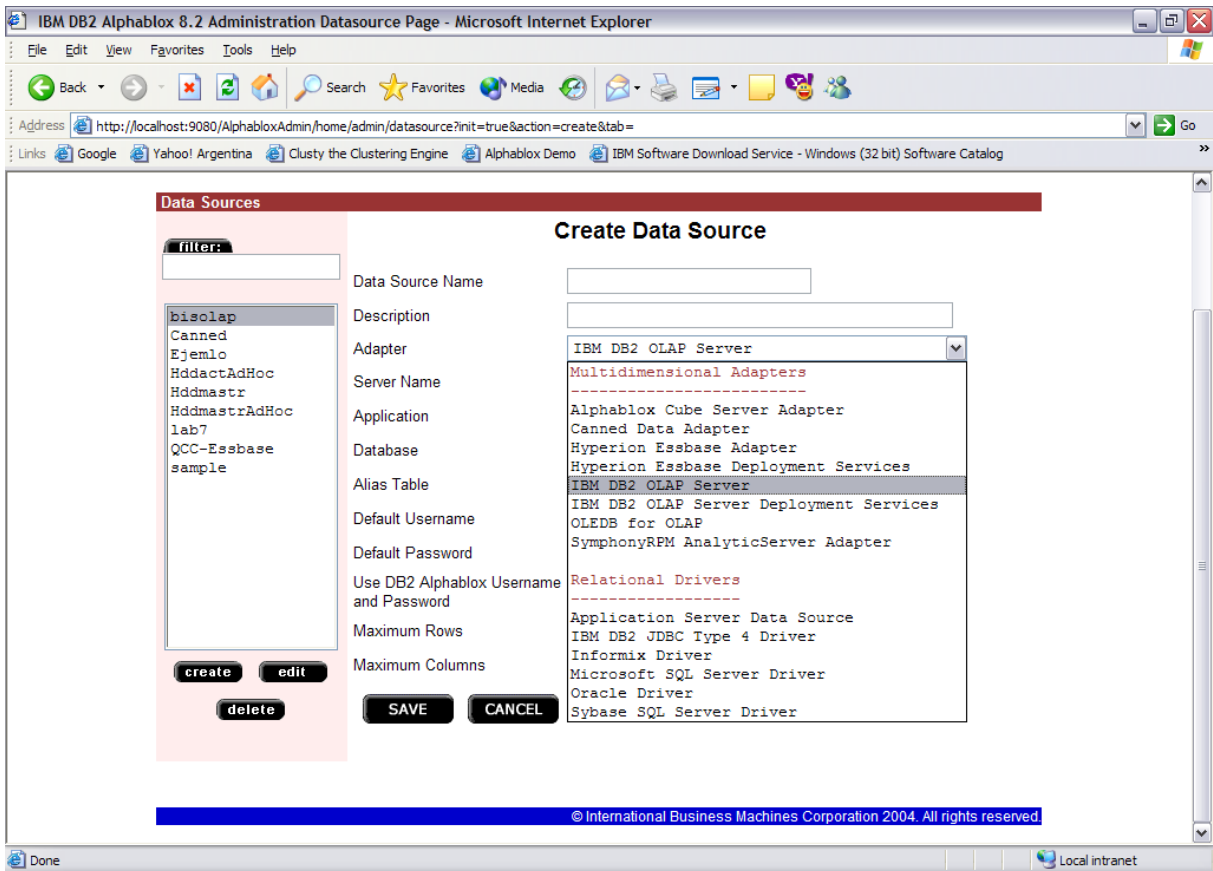


Figura 23 Página de Aplicaciones de Alphablox



Cuando creamos una nueva aplicación, Alphablox lo que hace internamente es construir una serie de archivos y estructuras de directorios dentro del servidor de aplicaciones, de manera de que ahora tenemos una nueva aplicación corriendo en el servidor que además ya tiene incorporadas todas las librerías y configuraciones necesarias para acceder a los datos que se definieron en la etapa anterior. Esta nueva aplicación J2EE en el servidor de aplicaciones solo puede correr si existe una instancia de Alphablox corriendo en el sistema ya que hace uso de sus servicios, por lo que no es posible construir una aplicación y copiarla a otro servidor de aplicaciones. Es necesario realizar todo el proceso de creado de una aplicación nuevamente.

### 7.3.1 Inicio y Navegación

El ingreso y la navegación para acceder a la información de gestión deben ser simples y atractivos. Es importante que el usuario se sienta cómodo en el sistema, logre llegar a la información que necesita de manera rápida y que visualmente la solución sea atractiva.

Para lograr este propósito, se opto por construir un menú circular en Flash. Este menú despliega mediante animaciones circulares los diferentes niveles de navegación, a los cuales el usuario accede haciendo clic en cada uno de los cuadrantes.



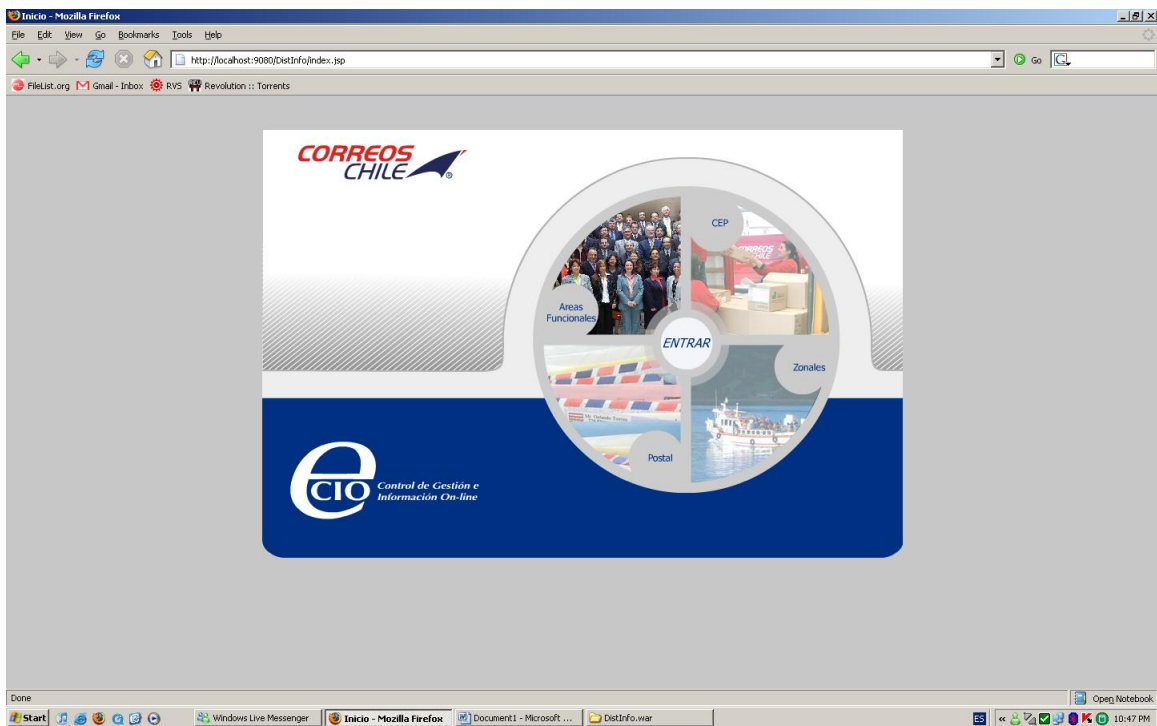


Figura 24 Página inicial del sistema



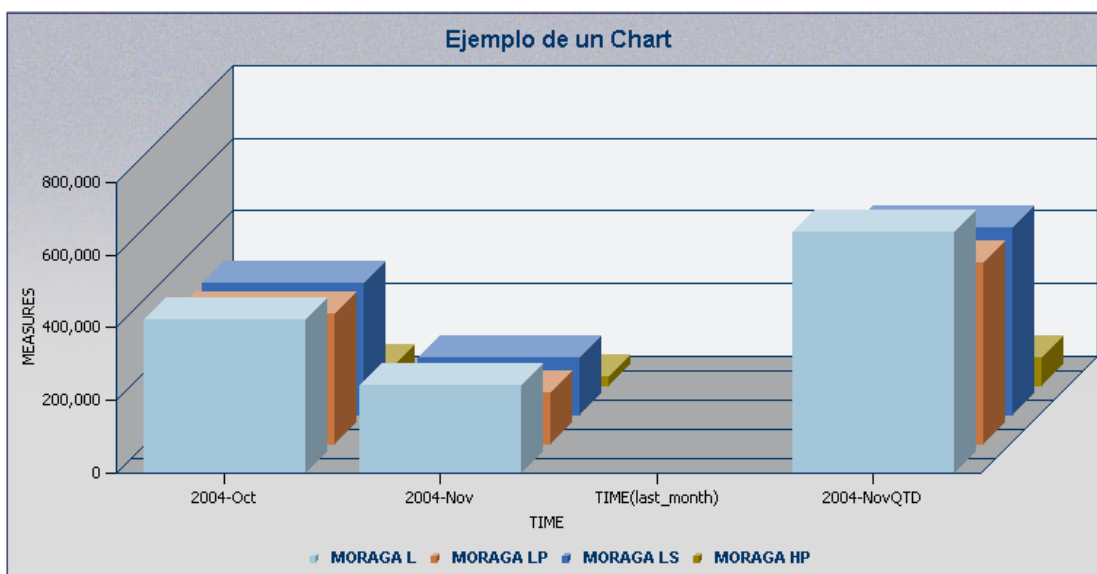
Figura 25 Submenú Postal

### 7.3.2 Ventas Corporativas y Sucursales

Alphablox nos permite utilizar una serie diversa de objetos llamados bloxs para el despliegue y acceso a la información. Los blox son componentes reusables que se combinan y ensamblan entre si en una pagina JSP. El resultado es una pagina Web interactiva (DHTML).

Los principales Blox provistos por Alphablox son los siguientes:

- DataBlox
  - Provee acceso a base de datos multidimensionales.
  - Provee acceso a base de datos relacionales.
  - Ejecuta consultas a la base de datos en SQL.
  - Provee resultsets con el resultado de los queries a otros componentes blox como así también a los programadores para utilizarlos en sus propias aplicaciones.
- ChartBlox
  - Presenta una vista grafica de los datos.
  - Permite a los usuarios la manipulación de los datos en diferentes formatos y tipos de gráficos.
  - Permite a los usuarios realizar drill down siguiendo la jerarquía de datos y también hacer pivot de los mismos.



- DataLayoutBlox

- Muestra las diferentes dimensiones.
- Permite a los usuarios mover las dimensiones en la página, filas, columnas, etc.

- GridBlox

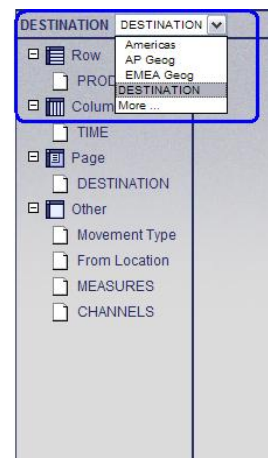
- Muestra los datos de una base Multidimensional o Relacional en forma de grilla.
- Permite a los usuarios analizar y manipular los datos.
- Permite a los usuarios realizar drill down y pivotear la data.



PRODUCT	2004-Oct	2004-Nov	TIME(last_month)	2004-NovQTD	TIME(last_qtr)	TIME(last_year)	TIME
CASCADE							
FIRECREST	107861	56228		164089	164089	164089	
FRESNO	143136	53694		196830	196830	196830	
GOLDCREST	27813	11512		39325	39325	39325	
HAKONE							
MORAGA L	420737	240828		661565	661565	661565	
MORAGA LP	357335	140278		497613	497613	497613	
MORAGA LS	360649	155312		515961	515961	515961	
MORAGA HP	52452	23047		75499	75499	75499	
MORAGA	1191173	559465		1750638	1750638	1750638	
MORAGA PLUS	184175	99141		283316	283316	283316	
Mobile Other							
Mobile Current	1654158	780040		2434198	2434198	2434198	
Mobile EOL	200	50		250	250	250	
MOBILE	1654358	780090		2434448	2434448	2434448	
DESKTOP	924353	436458		1360811	1360811	1360811	
SERVER	228923	105917		334840	334840	334840	
EMERGING	668439	302097		970536	970536	970536	
No BU							
Product Views							
PRODUCT	3476073	1624562		5100635	5100635	5100635	

- PageBlox

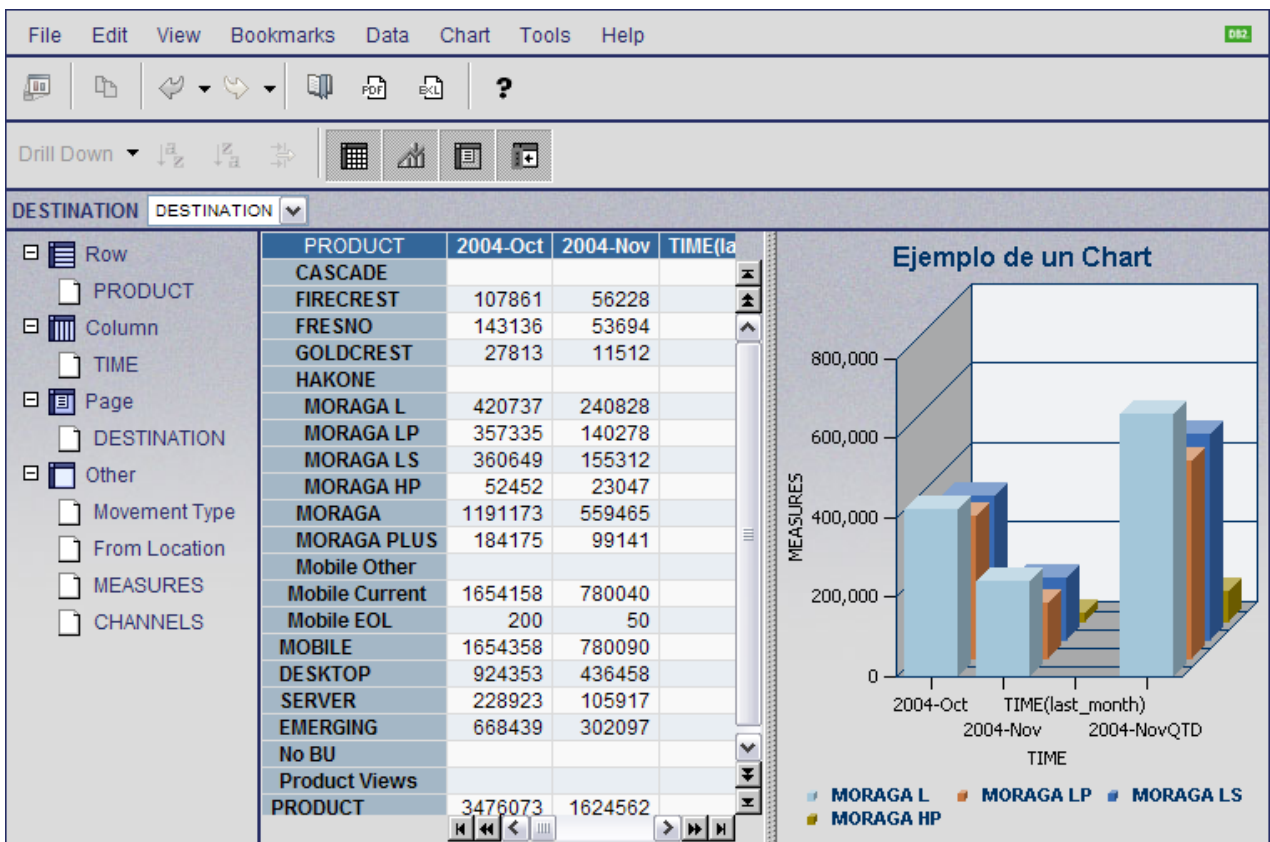
- Presenta una lista de las diferentes dimensiones que hay en la página. Esto se usa para permitir a los usuarios filtrar los datos.
- Los datos que se filtran mediante el PageBlox afectan al gridblox y el charblox.



- **ToolbarBlox**
  - Muestra los botones que le permite a los usuarios acceder a las diferentes funcionalidades provistas
    - Ocultar o mostrar el gridBlox y el ChartBlox.
    - Seleccionar el tipo de grafico.
    - Grabar o cargar vistas.
    - Abrir un blox en una ventana nueva.
    - Pivot de filas y columnas.
  - El programador mediante blox puede habilitar y deshabilitar los botones mostrados en el toolbar



- **PresentBlox'**
  - Combina las funcionalidades de todos los blox mencionados anteriormente en un solo blox, simplificando el desarrollo de aplicaciones y conservando el estado real de la página.



Con estos elementos podemos construir diferentes formas de visualizar los datos, con gráficos, tablas y elementos de interactividad que permitirán a los usuarios analizar de la mejor manera la información.

Alphablox además nos entrega una herramienta que nos permitirá construir la base de la visualización y su código de manera semiautomática. Esta herramienta se llama Query Builder y su función es, a partir de una fuente de datos y una query, generar un Present Blox estándar para ser customizado mediante el menú. De esta manera, podemos definir que componentes queremos mostrar, que tipo de gráficos, construir columnas precalculadas y mucho más. Una vez que tenemos la customización lista, exportamos el código de ese blox para agregarlo a nuestra aplicación.

Este método nos permite crear visualizaciones simples de manera muy rápida, ya que podemos publicar en Internet el resultado de cualquier consulta a la base de datos con un nivel alto de interactividad sin mucho trabajo.

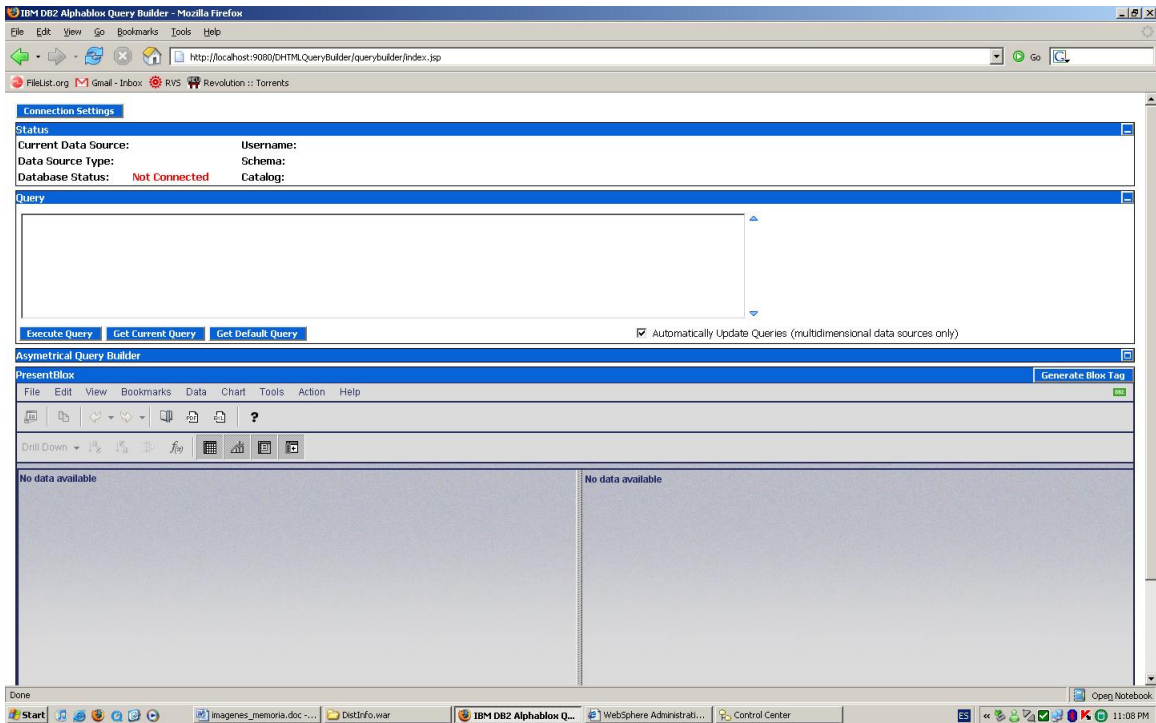
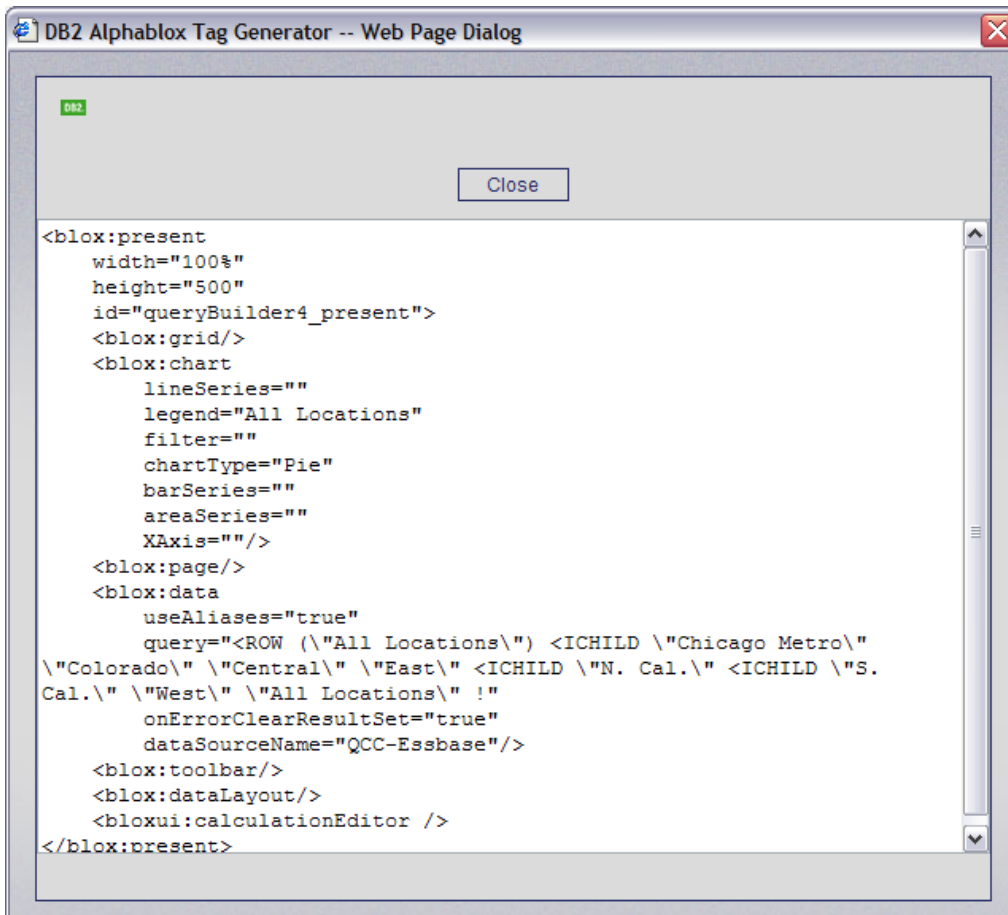


Figura 26 Query Builder



Para el panel de control de Ventas Corporativas utilizamos esta metodología para construir la base y poder presentar al cliente una idea de cómo se verá el resultado final. Esto es muy importante ya que podemos tener feedback del cliente antes de realizar los desarrollos complejos.

Luego de tener la estructura base, se realizó el desarrollo para agregar los filtros e interacciones entre los componentes Web y Alphablox para que todos interactúen entre sí. Un ejemplo de esto es que si selecciono un mes en la página, todas las tablas y gráficos se refrescan para reflejar los datos del mes seleccionado.

El resultado de este desarrollo es la aplicación que se muestra a continuación.

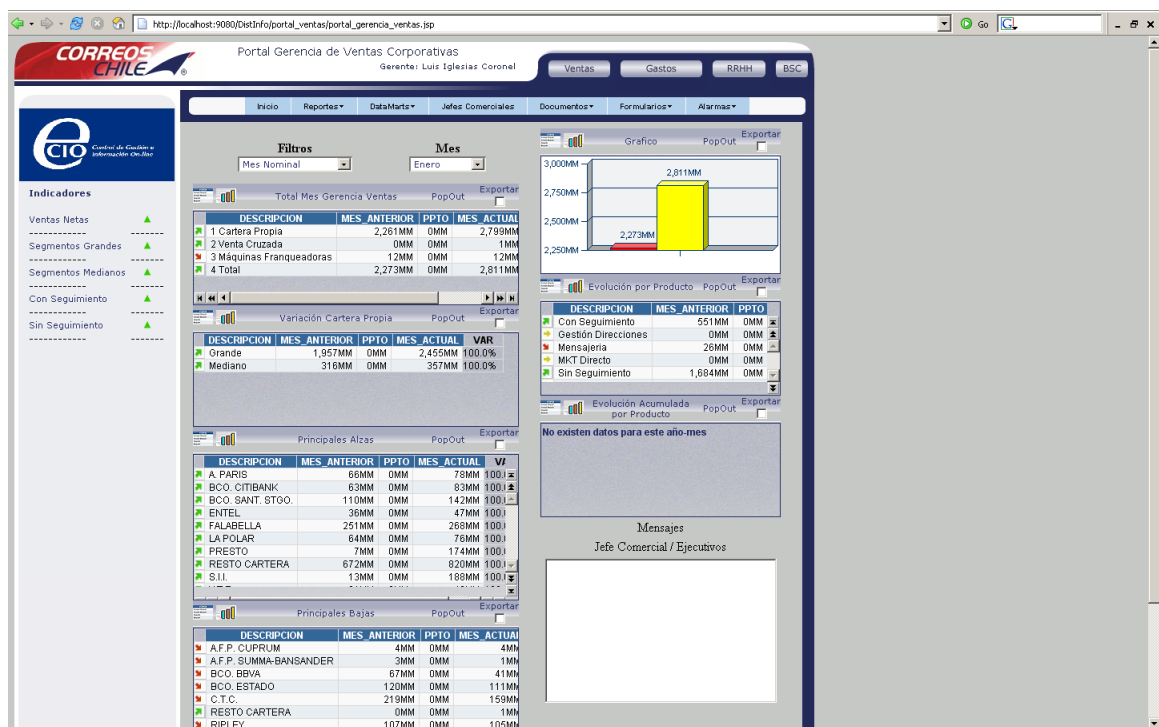


Figura 27 Panel de Control Ventas Corporativas

A la izquierda tenemos los elementos del Balanced Score Card y a la derecha tenemos los cuadros que muestran la evolución de distintos indicadores. Cada uno de los cuadros además incorpora una columna que nos indica si su evolución en el tiempo ha sido positiva, neutra o negativa mediante un indicador gráfico.

Además de las funcionalidades que entregan los bloxs, esta aplicación incorpora dos iconos en la parte superior de cada caja, estos iconos nos permiten cambiar la visualización entre datos y gráficos. También mediante el botón izquierdo sobre un blox podemos acceder a un menú que nos permitirá cambiar el tipo de gráfico, los ejes y otras opciones que personalizan la aplicación.

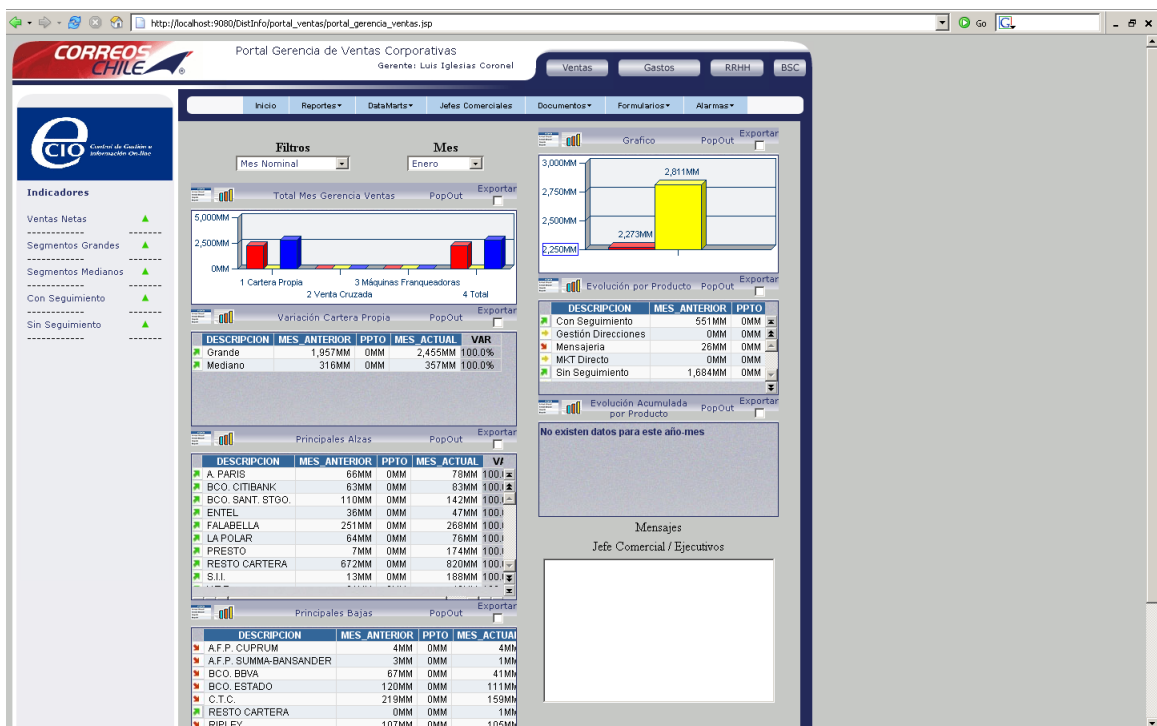


Figura 28 Ejemplo de cambio a gráfico

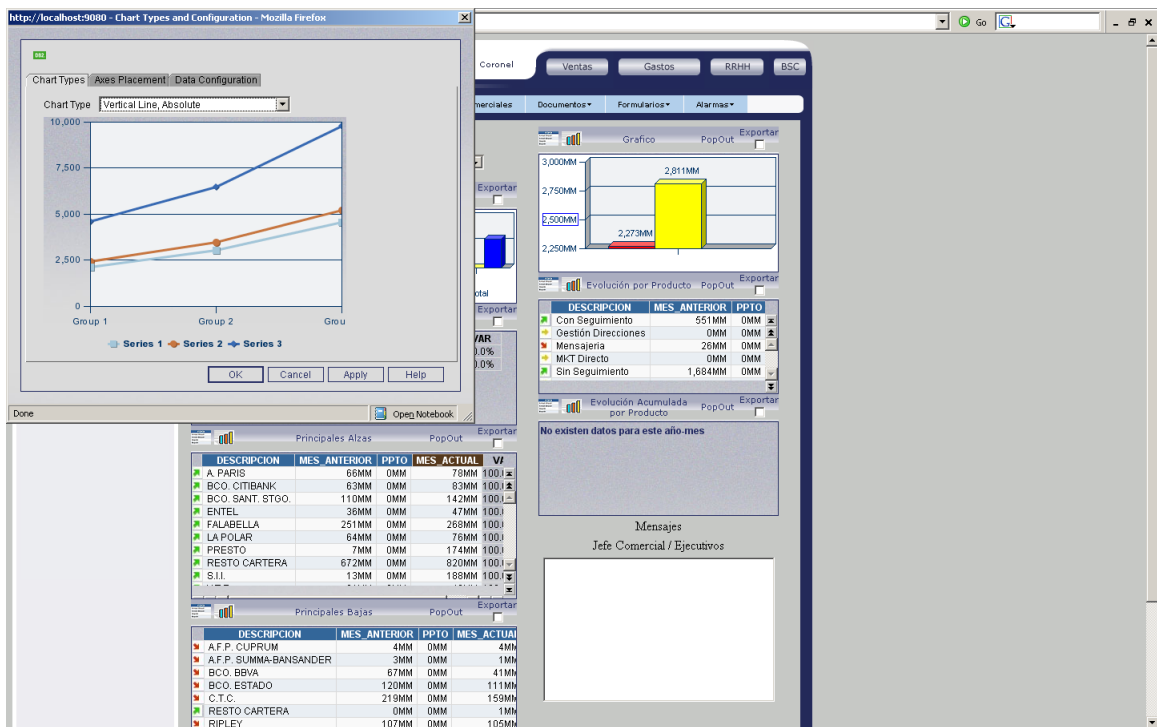


Figura 29 Ejemplo de selección de tipo de gráfico

Otra funcionalidad que se agregó es un botón de pop-up en cada una de las cajas. Este botón permite que si el usuario desea analizar de mejor manera un grafico y sus datos, pueda separar este cuadro del resto de manera de verlo en una ventana de mayor tamaño, en la cual podrá ver mayor nivel de detalle.



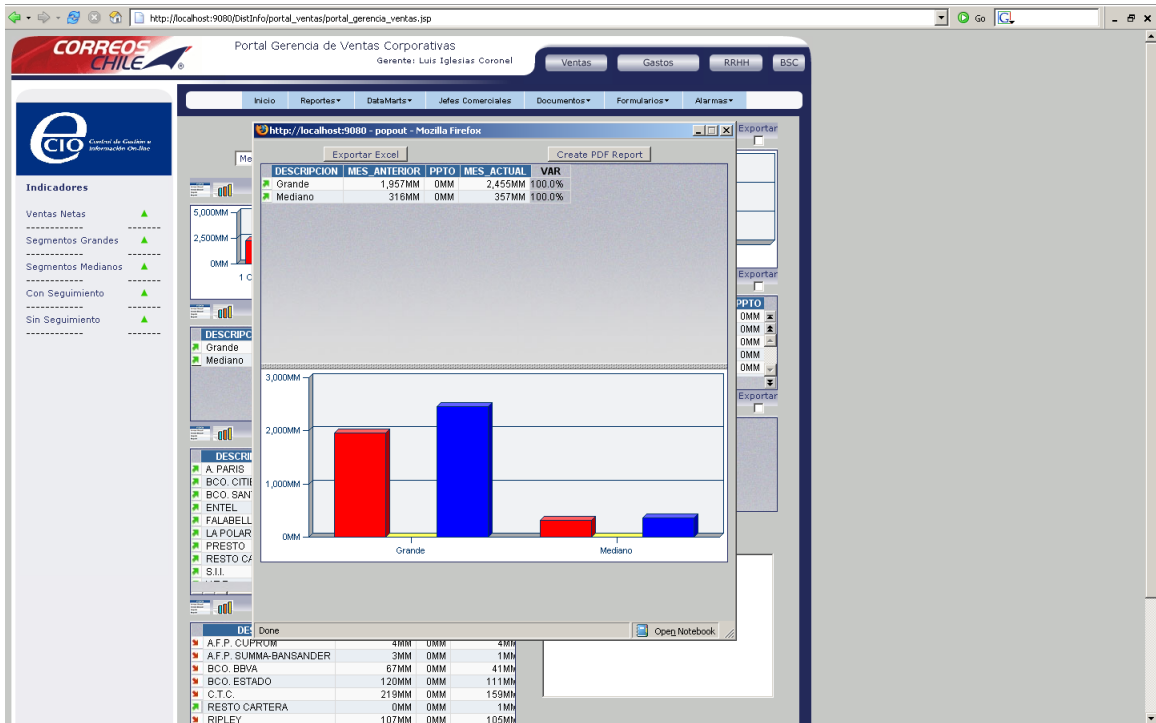


Figura 30 Ejemplo de POPUP

Por ultimo, cada una de estas cajas permite exportar a Excel o PDF, ya sea individualmente o en conjunto seleccionando el check box en la esquina de la caja y seleccionando exportar al pie de la página. Esta funcionalidad también fue construida adhoc para las necesidades del cliente.

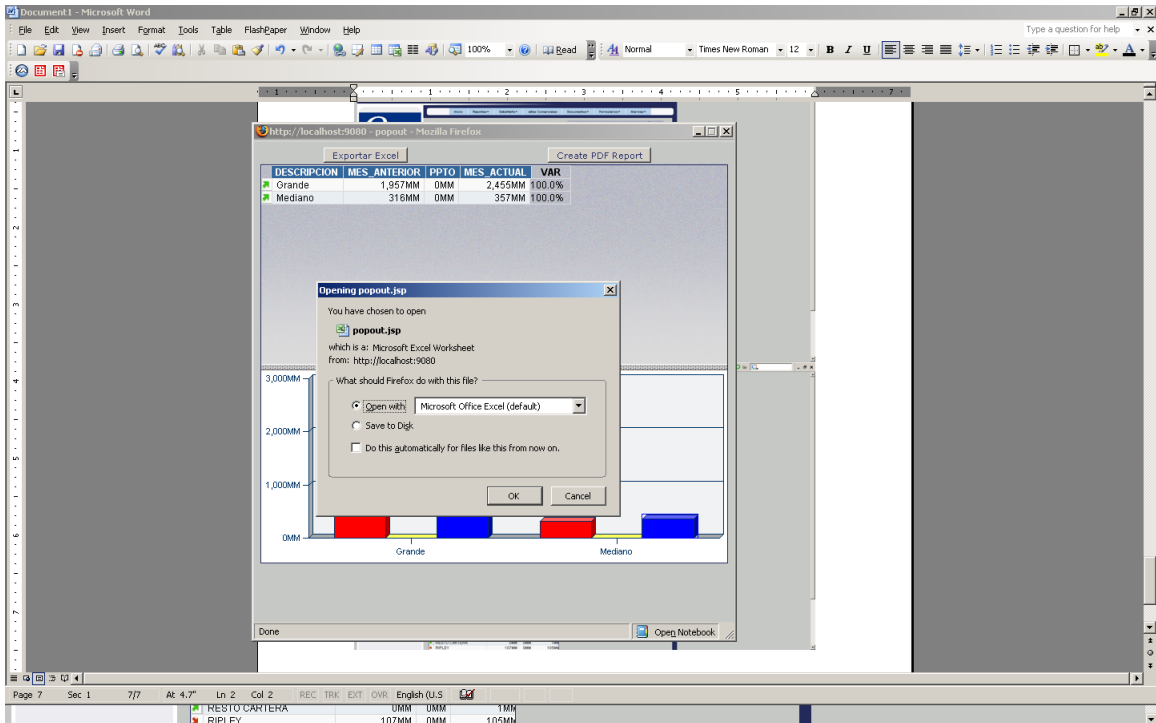


Figura 31 Ejemplo de exportar a Excel

DESCRIPCION	MES_ANTERIOR	PPTO	MES_ACTUAL	FLAG_AGRUPADO	
Grande		1,957MM	0MM	2,455MM	1
Mediano		316MM	0MM	357MM	1

**Figura 32 Ejemplo de reporte en Excel**

A pesar de todos los cambios que uno realiza mientras se interactúa con los bloxs y con el sitio en general, nunca se realiza una recarga de la página, solo se refrescan los componentes y se piden los datos al servidor de ser necesario. Esto se logra ya que Alphablox nos permite exportar métodos a Javascript permitiéndonos una interactividad más fluida y transparente para el usuario.

Para los cubos el procedimiento de construcción es distinto. Alphablox puede conectarse y presentar cubos almacenados de manera dimensional sin problemas. Esto incluye cubos almacenados en servidores multidimensionales como el servicio Analysis Services de SQL Server. En nuestro caso, no poseemos un servidor multidimensional ya que nuestra solución utiliza cubos almacenados de manera relacional en DB2, por lo que Alphablox al no tener un servicio al cual conectarse utiliza un servicio propio en el cual carga la metadata de los cubos relacionales y entrega de esta manera acceso a la información.

El primer paso para la visualización de cubos relacionales es construir la meta data en la herramienta Cube Views de DB2. En esta herramienta definimos las tablas involucradas, los hechos, las dimensiones y las jerarquías que nos interesan.

El segundo paso es exportar esa meta data e importarla en Alphablox Cubes, lo que se realiza en el área de administración de Cubos.

Con esto construido, solo nos basta agregar una fuente de datos al servidor de cubos de Alphablox y luego agregar un Present Blox en nuestra aplicación para permitir el acceso a la información del cubo.

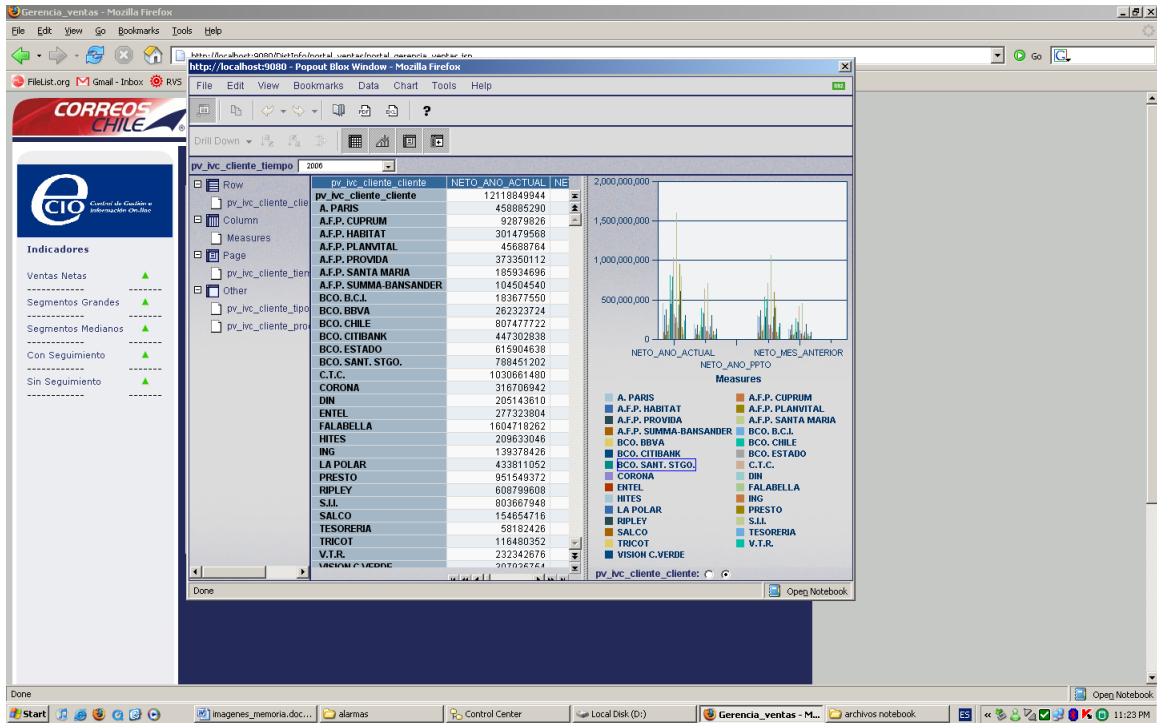


Figura 33 Ejemplo de Cubo en Alphablox

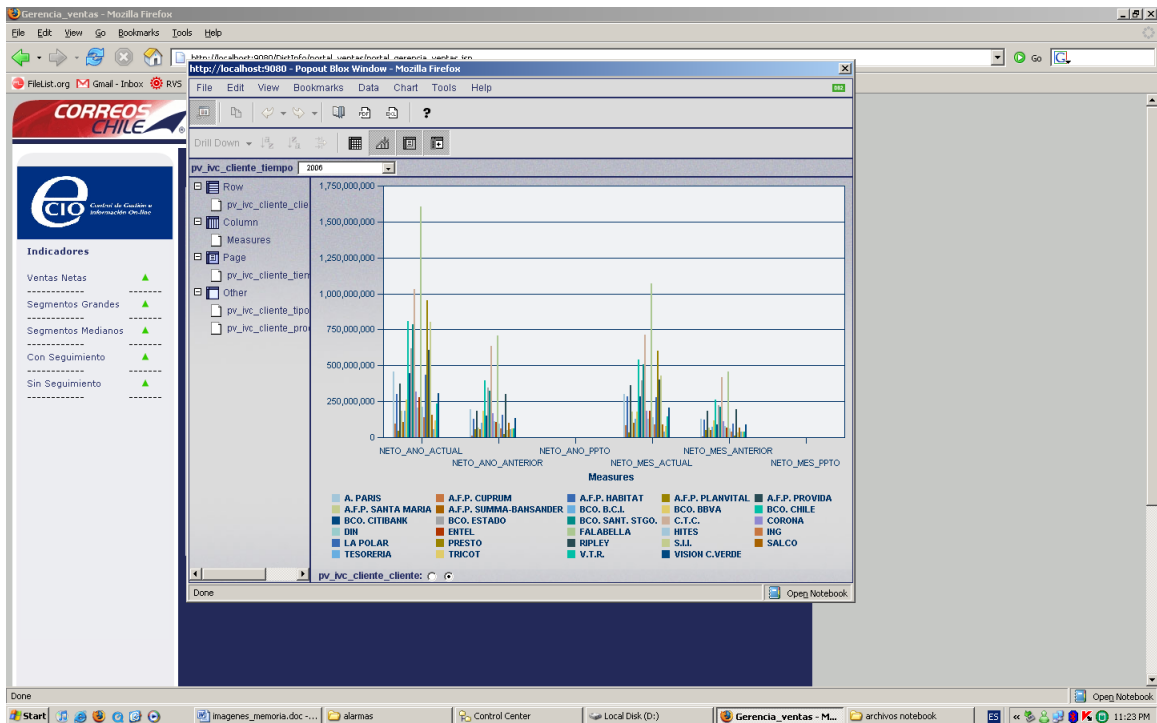


Figura 34 Ejemplo 2 de Cubo en Alphablox

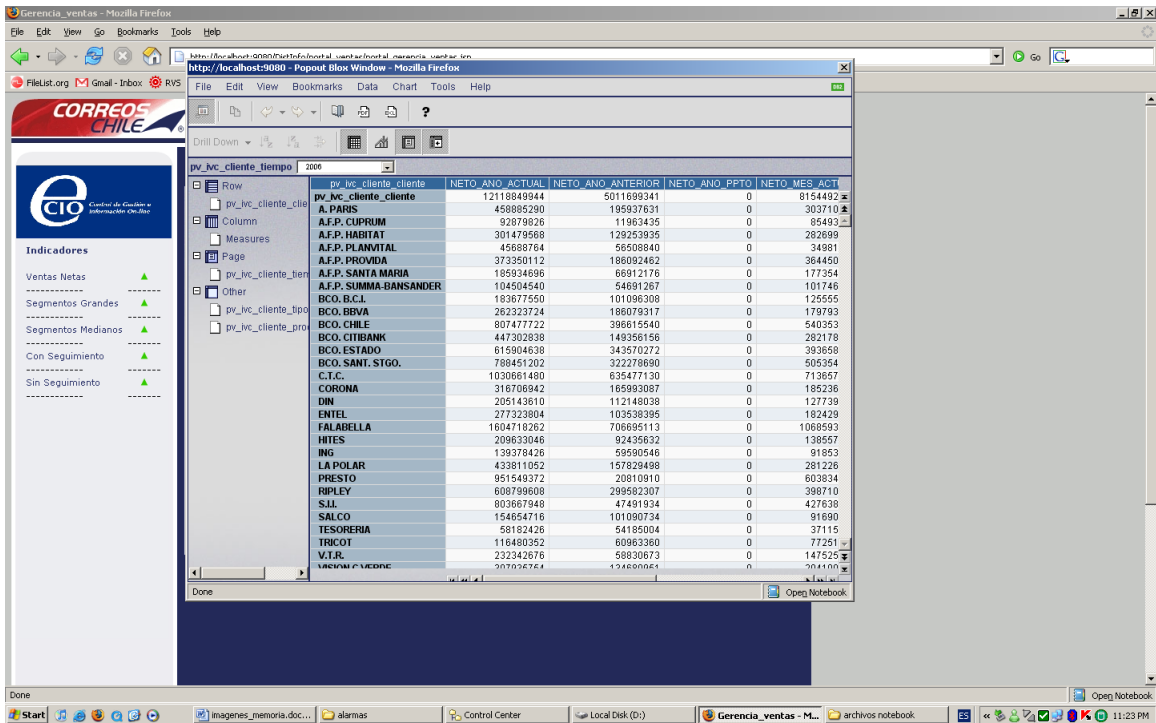


Figura 35 Ejemplo 3 de Cubo en Alphablox

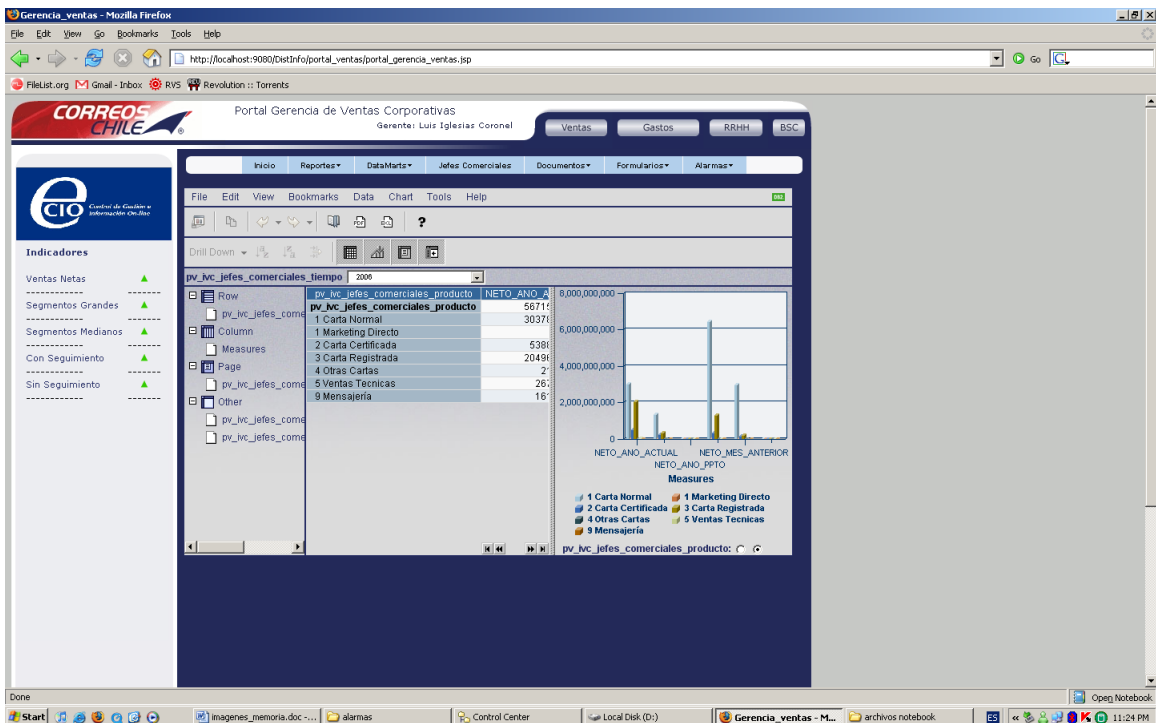


Figura 36 Ejemplo 4 de Cubo en Alphablox

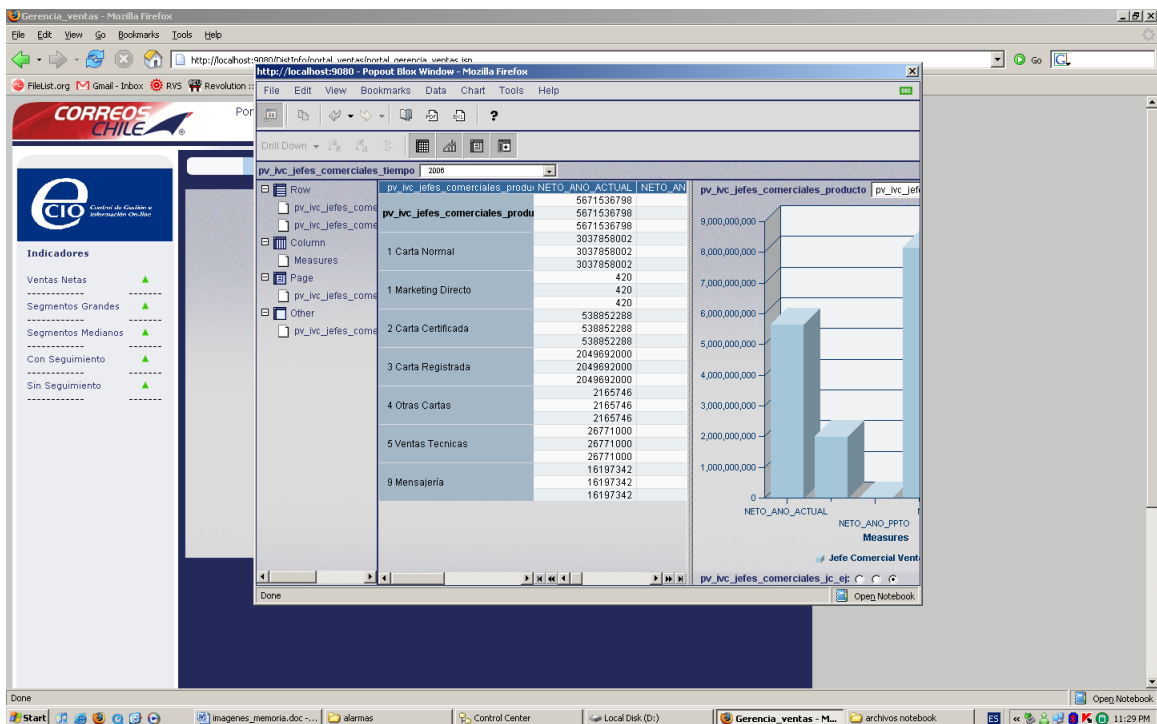


Figura 37 Ejemplo 5 de Cubo en Alphablox

En el caso de los cubos, toda la lógica está en la metadata, por lo que la presentación de la información solo se limita a definir el estado inicial en que se desplegará, es decir, que dimensiones estarán en que ejes de un grafico, que dimensión serán filtro y que jerarquías deben estar expandidas.

Otra área de la aplicación es el área de Documentos. En ella, presentaremos una serie de documentos y presentaciones en formato Flash para ser consultadas por los usuarios.

Nuevamente se creo un pequeño sistema de manera de que las publicaciones se agreguen de manera ordenada y las presentaciones tengan un formato estándar.

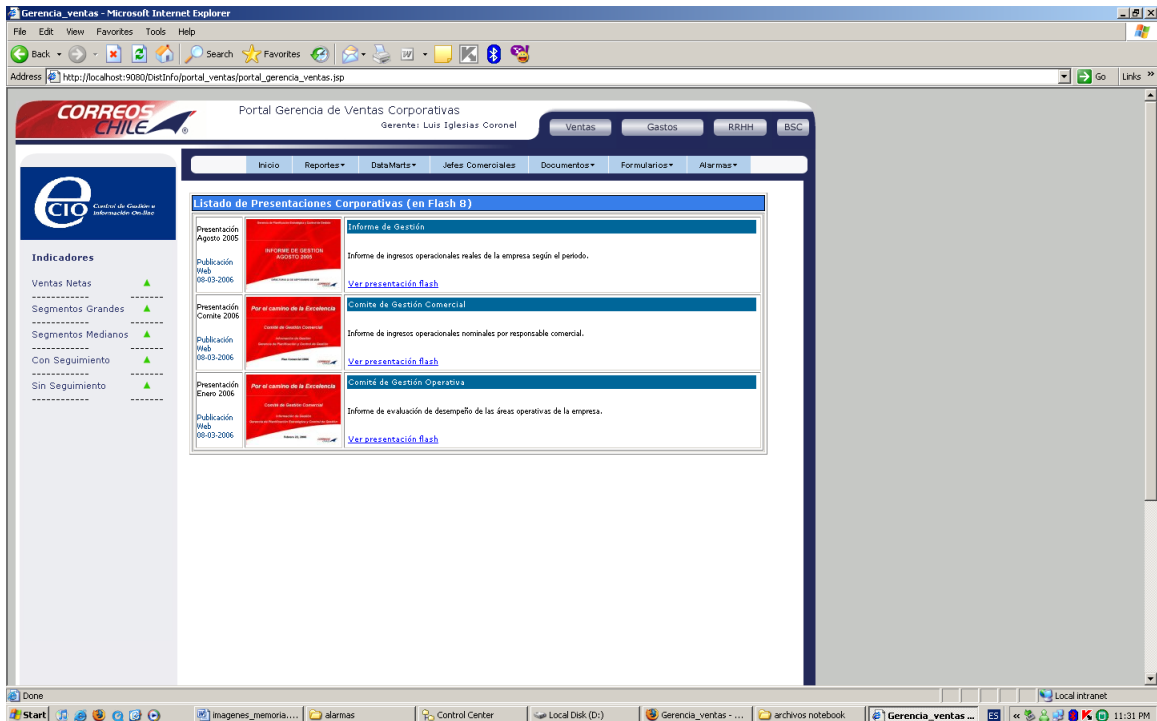


Figura 38 Página de Presentaciones

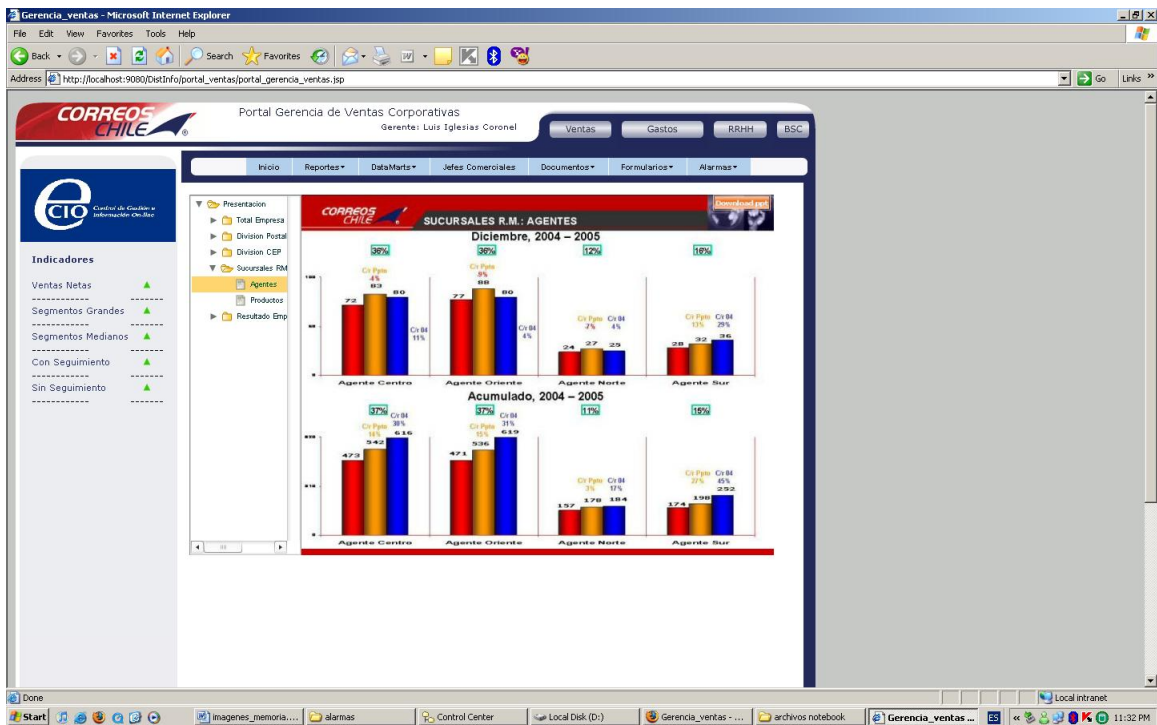


Figura 39 Presentación Flash

Para los formularios que se agregaron en el sistema, creamos un set de plantillas en JSP que simplemente aceptan una serie de campos y los guardan en la base de datos. Esta aplicación no tiene mayor complejidad y solo busca agregar funcionalidades requeridas por el cliente al sistema.

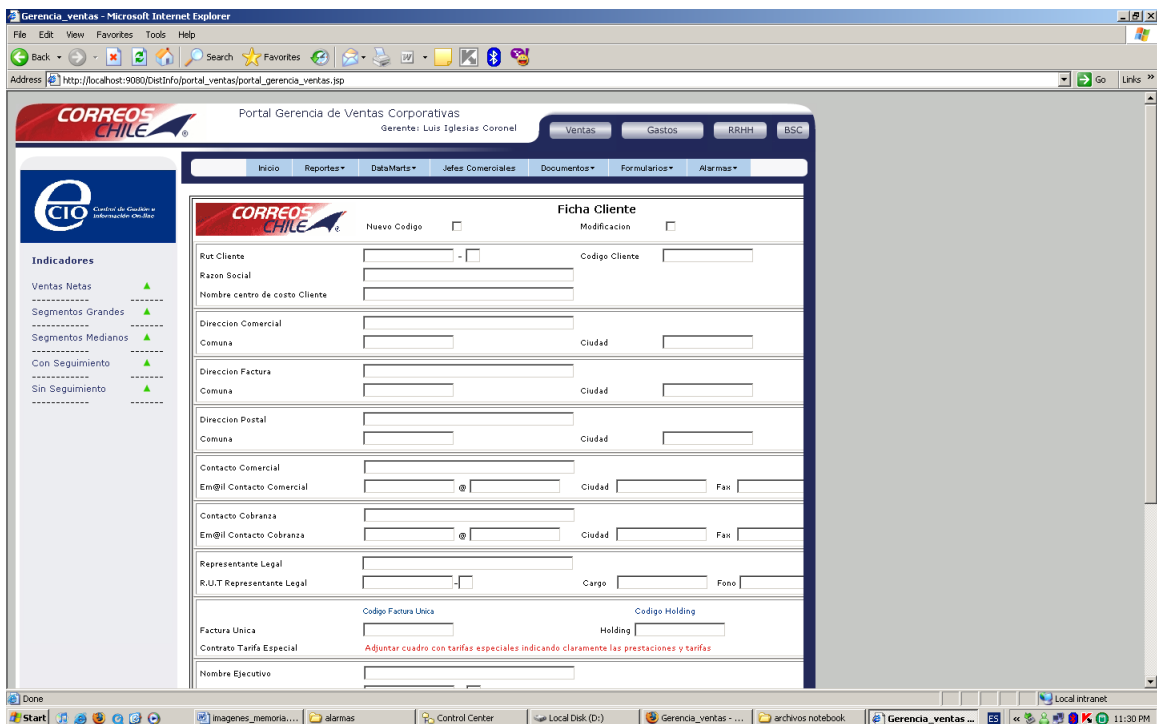


Figura 40 Formulario 1

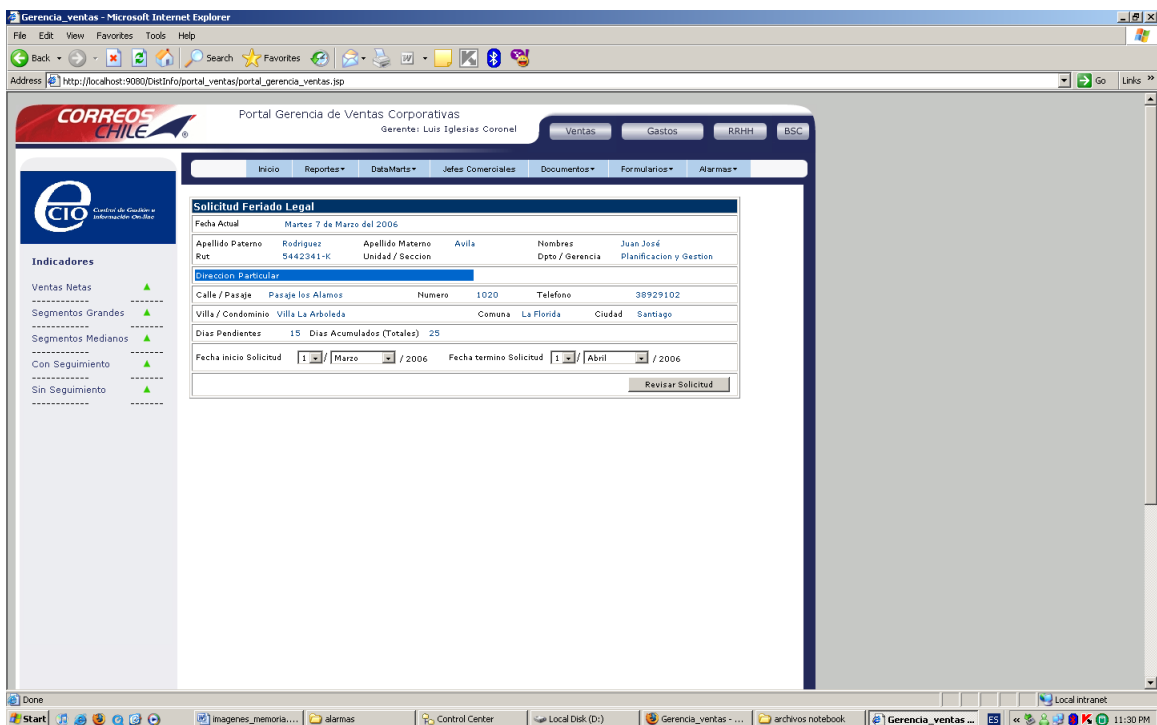


Figura 41 Formulario 2

La aplicación de sucursales es muy similar a la aplicación de ventas corporativas, con la gran diferencia de que cuenta con un mapa de Chile que permite filtrar la información ya sea por región o por zona. El desarrollo de este mapa está realizado sobre Flash y se actualiza leyendo un archivo de texto que es escrito, ante cambios en los datos, por un stored procedure.

Al presionar en cualquier zona o región, todos los bloxs del panel se actualizan para reflejar la selección realizada, entregando al usuario la posibilidad de realizar análisis por zonas o regiones de manera muy sencilla.

Además, el mapa incorpora colores por región o zona para representar el estado del indicador que este seleccionado en el área de filtros del mapa.

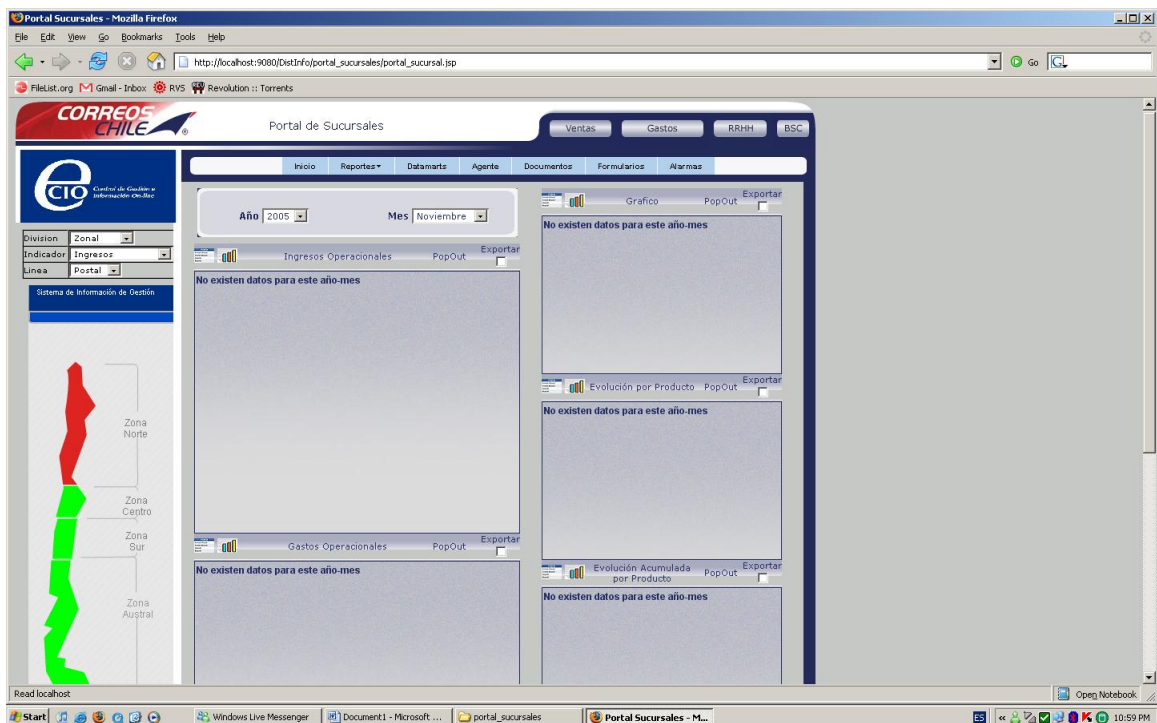
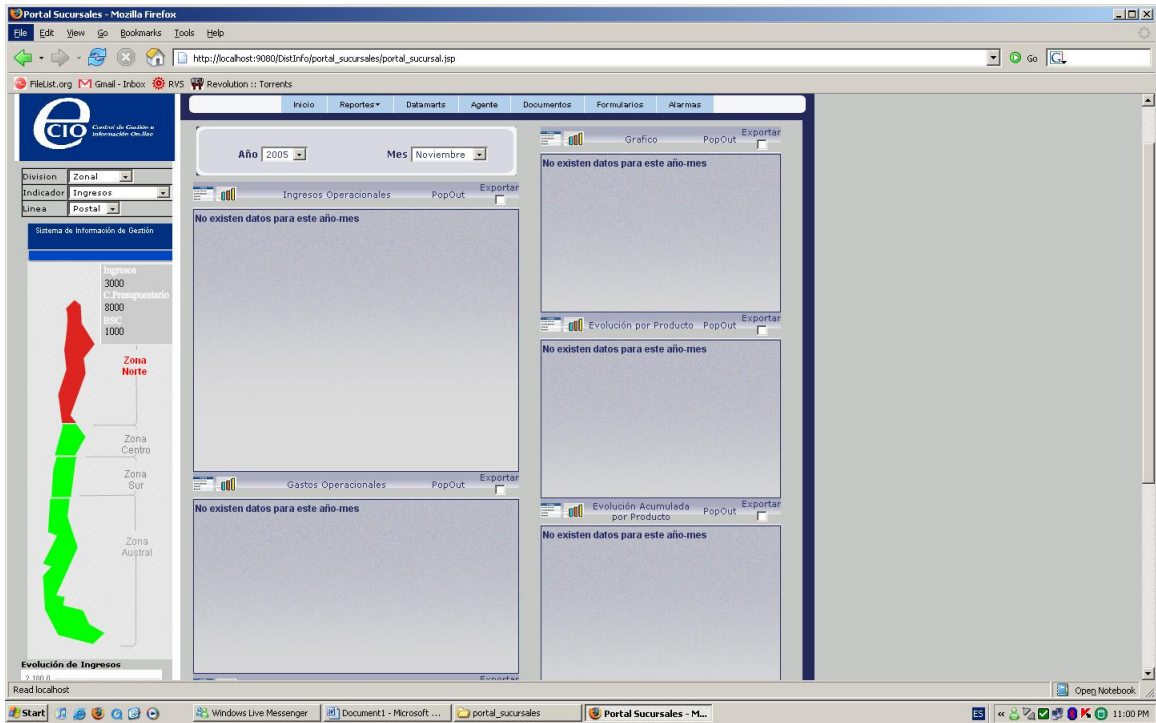
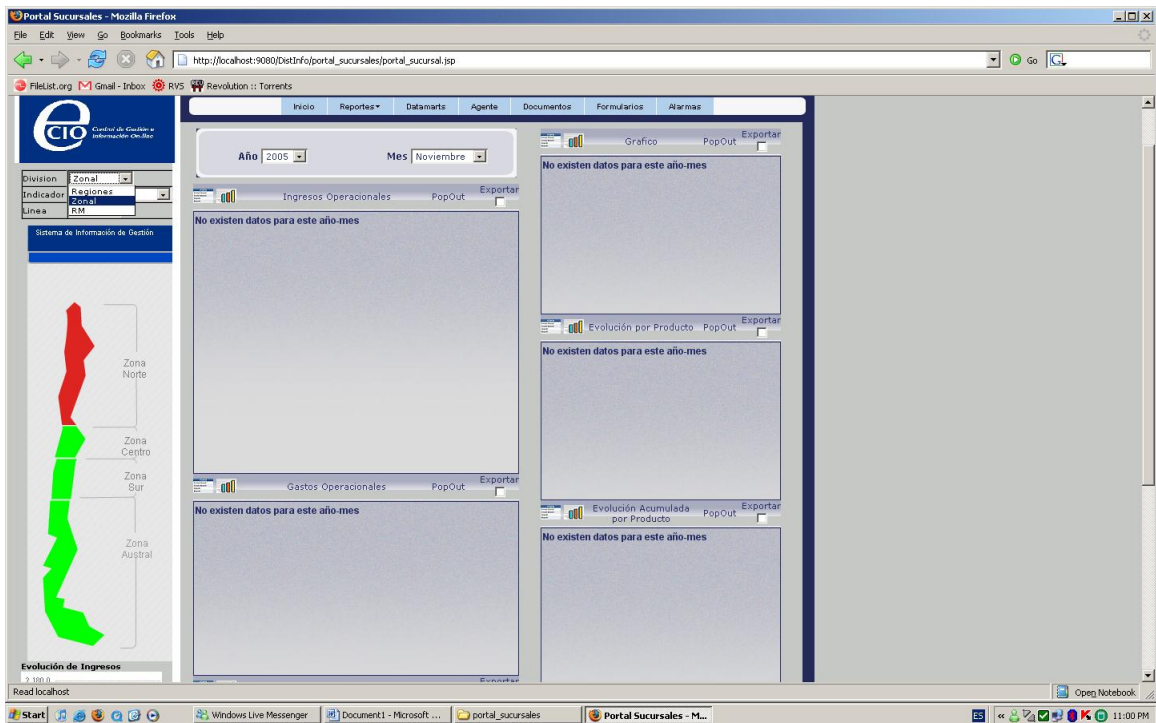


Figura 42 Portal de Sucursales





**Figura 43 Portal de Sucursales despliegue de datos en Flash**



**Figura 44 Portal de Sucursales Filtros**

### **7.3 Definición del proceso de mantención y crecimiento**

Parte importante del proyecto es definir un proceso para que el área de Planificación Estratégica y Control de Gestión sea capaz de generar nuevo contenido de manera rápida y sin los conocimientos técnicos de un experto, de manera de incorporar al sistema a nuevas áreas de la organización.

La generación de los datos, desde las fuentes operacionales al Datawarehouse corporativo y luego al sistema de distribución de información está bastante resuelto y no genera mayores problemas para el área, sin embargo, la creación de aplicaciones J2EE, JSP, Javascript y otras tecnologías asociadas, si escapa el expertise que tienen.

Para resolver esto, el área utilizará de manera mucho más intensiva de lo que lo hicimos en este proyecto, las aplicaciones provistas por Alphablox para la generación automática de bloxs, a pesar de que esto implique no tener las interacciones que se crearon acá.

El procedimiento que se definió es utilizar una página JSP estándar que se guardo en un directorio particular. Esta página contiene todas las librerías necesarias y además contiene la navegación del sitio. En esta página se pegarán los códigos generados por la aplicación Query Builder, de manera de tener la posibilidad de presentar estos datos de manera rápida e interactiva, pero de alguna manera, aislados del resto de la página.

Alphablox no entrega herramientas de desarrollo más simples que permitan crear páginas realmente interactivas sin la intervención del código, por lo que el área estará limitada a prescindir de estas funcionalidades mientras no tengan los conocimientos necesarios para construirlos.

## Capítulo 8

### 8. Resultados

Para poder medir los resultados y el impacto de este proyecto, es necesario definir métricas que se adecuen a los objetivos definidos.

La principal motivación de este proyecto era la de proveer a Correos Chile de la infraestructura necesaria para proveer de información a sus distintas áreas internas. Esto involucra hardware, software, desarrollos y personas orientadas a mejorar los procesos de toma de decisiones.

Bajo este objetivo, debemos definir métricas que nos permitan establecer si el proyecto fue exitoso o no en sus distintas líneas de impacto y nos permitan definir el trabajo futuro necesario para que el proyecto sea rentable en el largo plazo.

La primera métrica que definimos busca medir el impacto que ha provocado el proyecto en las dos áreas donde se realizó el proyecto, esto es el área de ventas corporativas y el área de sucursales. Este impacto tiene que ver con la cantidad, calidad y velocidad en que estas áreas obtienen información.

Para obtener esta información recurrimos a mediciones reales y a encuestas en las áreas involucradas. Los resultados fueron los siguientes:

Ítem/Tiempo	Reportes 2 a 3 horas	Después proyecto instantáneo	del Proyectado Futuro instantáneo	a Beneficio 2 a 3 horas
Acceso a un reporte existente	2 a 3 horas	instantáneo	instantáneo	2 a 3 horas
Obtención de un reporte nuevo	72 horas	12 horas	3 a 4 horas	60 horas
Calidad de los reportes (1 a 7)	5	6	6	
Arreglo de reportes defectuosos	30 horas	6 horas	2 horas	24 horas
Calidad del análisis posible (1 a 7)	5	5	6	
Capacidad de trabajo en equipo (1 a 7)	3	4	6	
Velocidad en encontrar la información necesitada (1 a 7)	3	5	6	
Facilidad para utilizar la herramienta	5	4	6	

**Figura 45 Resultados del proyecto**

En esta tabla se pueden obtener varias conclusiones que se analizaran en la sección siguiente, sin embargo, es interesante ver como hemos logrado uno de los principales objetivos que era liberar recursos del área de Planificación Estratégica.

Dentro del mismo proyecto se buscaba dejar implementado un procedimiento para crecer y agregar nuevas áreas al sistema. Este procedimiento si bien quedo definido y el área de

planificación estratégica está preparada para recibir nuevos usuarios, la dificultad técnica de la herramienta puede ser una barrera para crear aplicaciones que realmente se hagan cargo de las necesidades de los usuarios.

Otras métricas interesantes son la cantidad de usuarios que utilizan el sistema, el tiempo que permanecen conectados, cuantos pedidos de cosas nuevas se realizan, cuales fueron los cambios en el negocio y muchas otras, sin embargo, necesitamos de un tiempo de prueba mayor al disponible para medir realmente alguno de estos datos y poder concluir algo con un nivel de certeza razonable.

## Capítulo 9

### 9. Conclusiones

La primera conclusión de este proyecto es que efectivamente las empresas pueden beneficiarse de los proyectos de Inteligencia de Negocios y distribución de información. Durante este proyecto no se logró verificar si efectivamente un proyecto de este tipo es rentable o no debido a los plazos que se manejaron, pero sí efectivamente se verificó que la capacidad de acceso a la información y la posibilidad de realizar análisis mejoró sustancialmente los procesos de toma de decisiones.

Evaluar efectivamente la rentabilidad de este proyecto es una tarea a largo plazo que irá de la mano de la utilización y crecimiento que tenga el sistema, así como de la capacidad analítica del personal que tendrá acceso a él.

De la tabla de evaluación del proyecto podemos concluir que para los usuarios finales no es una tarea sencilla adoptar una nueva tecnología. Si bien los tiempos de respuesta en la entrega de información mejoraron, es difícil para los usuarios modificar su manera tradicional de trabajar, a pesar de valorar el beneficio obtenido. Mediante capacitaciones y trabajo en equipo con ellos, es posible mejorar este indicador y por ello a futuro se esperan mejores resultados.

Otro punto interesante es que la capacidad de análisis no mejoro mucho. Esto se debe principalmente a que los usuarios no conocen todas las funcionalidades del sistema y tienden a exportar los datos a Excel para realizar un análisis parecido al que realizaban antes del proyecto. Si bien este comportamiento no es malo, la idea es incorporar los análisis más recurrentes a la aplicación de manera de que todos tengan acceso a esa información y no se mantenga como conocimiento específico de una sola persona.

También en el área de trabajo en equipo el proyecto no alcanzo los niveles que nos gustaría. La razón de esto es que en esta etapa se implementó solo una parte básica de las posibilidades que nos ofrece un sistema Web para comunicarnos y compartir información. Claramente un sistema de mensajería como el implementado en esta primera etapa es insuficiente para generar un producto potente que incentive el trabajo colaborativo.

Los modelos y metodologías que se utilizaron obtenidos de la revisión bibliográfica realizada, efectivamente sirven de guía para realizar un proyecto de inteligencia de negocios. En muchas instancias se evaluaron alternativas que se alejaban de los modelos definidos y que en el largo plazo hubiesen terminado por ser un gran problema para el éxito del proyecto. Sin embargo, lograr hacer ver el valor al cliente de seguir una metodología establecida aún cuando los tiempos de implementación son mayores, es un tema que requiere trabajo y no debe subestimarse.

En términos de la herramienta, esta cumplió con los objetivos planteados. Si bien el modelo de datos y todo el conjunto de aplicaciones fue construido pensando en generar el menor costo de procesamiento al servidor, el rendimiento fue el adecuado y existe el espacio para seguir creciendo. El gran problema de esta plataforma, en particular de Alphablox, es que requiere de un nivel de conocimiento alto en programación para poder obtener su máximo potencial. El no incorporar una herramienta de diseño intuitiva y sencilla deja fuera a un gran grupo de usuarios que les gustaría crear sus propias interfaces y aplicaciones. Sin embargo, gracias a que Alphablox está escrito en JAVA y bien documentado, es posible hacer prácticamente cualquier cosa que se desee ya que sus objetos pueden ser extendidos y modificados a cualquier nivel.

Una última conclusión tiene que ver con el costo de la solución. Si bien es posible con creatividad e ingenio reducir los costos asociados a un proyecto de este tipo, solo el hardware y licenciamiento del software ya corresponden a un porcentaje importante del costo total. Si buscamos replicar este tipo de proyecto en empresas más pequeñas, encontramos que existen alternativas Open Source que pueden lograr efectos similares al de esta experiencia pero el esfuerzo requerido para obtener la misma interactividad sería mucho mayor. El costo de una plataforma como la de IBM se justifica cuando está dentro de los planes crecer y llegar a gran diversidad de usuarios simultáneamente.

## **9.1 Experiencias aprendidas y trabajo futuro.**

- Es importante para el éxito del proyecto, lograr hacer conversar los requerimientos del cliente con las posibilidades que nos da la herramienta.
- Los tiempos de desarrollo son mucho mayores que los tiempos que uno está acostumbrado y que el cliente espera, pero es necesario realizar el trabajo en los tiempos requeridos para obtener una solución robusta.
- En la planificación del proyecto es necesario considerar tiempos para solucionar problemas de integración, seguridad y capacitaciones de usuarios que utilizarán la plataforma.
- Es necesario trabajar con los usuarios finales del producto, de manera de asegurar que la aplicación será adoptada sin problemas. Hay que tener en cuenta que la visión de los usuarios cambia de manera muy fácil por lo que es necesario hacerse cargo de los procesos más que de las personas.
- La componente gráfica de las aplicaciones y las interfaces atractivas ayudan mucho a que una aplicación sea aceptada, sin embargo, la funcionalidad es la que le dará la vida en el largo plazo.

- Las agendas de las áreas comerciales e informáticas no siempre están alineadas y es necesario en un proyecto como este en que ambas partes son importantes, lograr atender los requerimientos de ambas partes.
- Un gran problema es la baja motivación de los usuarios por adaptarse a una nueva tecnología, por lo que mantener un plan de capacitación en donde los usuarios puedan conocer los nuevos sistemas es fundamental para el éxito de la plataforma. De igual manera, integrar nuevas soluciones a ambientes lo más parecidos a los ambientes típicos de trabajo de los usuarios provoca que el cambio sea mucho más aceptado.
- En un segundo proyecto, se propone trabajar más en profundidad los temas de colaboración del sistema. Esta funcionalidad agrega una dimensión nueva a las funcionalidades implementadas y agregaría mucho valor. Lo desarrollado en esta primera etapa solo es una prueba de concepto y debe profundizarse y potenciarse.
- Agregar conceptos de la Web 2.0 también es muy interesante. La posibilidad de que existan blogs, rankings, tags y todas las nuevas tendencias permitirán interacciones con el sistema muy potentes. Dado que estamos en una plataforma J2EE, podemos incorporar todas estas nuevas tecnologías sin problemas.
- En una segunda etapa también sería interesante construir una aplicación que permita la autogeneración de reportes, es decir, que los propios usuarios sean capaces a partir de unas tablas generar su reporte y dejarlo publicado en la Web. Alphablox incorpora un primer acercamiento a este tipo de funcionalidades con su Alphablox Portal, sin embargo, debe adaptarse y hacerse crecer para resolver los requerimientos de los usuarios que no tienen un manejo técnico.
- Por último, para utilizar la totalidad de las herramientas incorporadas, se recomienda explotar la herramienta DB2 Intelligent Miner. Con esta herramienta es posible realizar procesos de minado de datos como la creación de clusters, reglas de asociación, redes Bayesianas y otros de manera muy intuitiva y permitirían obtener mucho conocimiento a partir de los datos que ya se manejan en el sistema.

## Bibliografía

- [1] Few, Stephen. Information Dashboard Design: The Effective Visual Communication of Data, O'Reilly, 2006. 223p.
- [2] Moss, Larissa T. y Atre, Shaku. Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support Applications, Addison-Wesley Professional, 2003. 576p.
- [3] Inmon, W.H. Building the Data Warehouse. 2<sup>nd</sup> Edition, Wiley, 1996. 401p.
- [4] Kimball, Ralph. y Ross, Margy. The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling. 2<sup>nd</sup> Edition, Wiley, 2002. 464p.
- [5] Bloch, Joshua. Effective Java Programming Language Guide. Addison-Wesley Professional, 2001. 252p.
- [6] Getting Started with Data Warehouse and Business Intelligence por Maria Sueli Almeida "et al". IBM Redbooks, 1999. 260p.
- [7] Reinschmidt, Joerg. y Francoise, Allison. Business Intelligence Certification Guide. IBM Redbooks, 2000. 166p.
- [8] Raden, Neil. Stepping up BI expectations. [en línea] <[http://www.intelligententerprise.com/030901/614feat1\\_1.jhtml](http://www.intelligententerprise.com/030901/614feat1_1.jhtml)>[consulta: 20 enero 2007].
- [9] Kimball, Ralph. y Ross, Margy. Difference of opinion. [en línea] <<http://www.intelligenteai.com/showArticle.jhtml?articleID=17800088>>[consulta: 20 enero 2007].