



# Universidad Austral de Chile

Facultad de Ciencias de la Ingeniería  
Escuela de Ingeniería Civil en Informática

## DESARROLLO DE SOFTWARE DE GESTIÓN PARA LA PRODUCCIÓN BOVINA EMPLEANDO AJAX.

Tesis para optar al Título de:  
Ingeniero Civil en Informática

Profesor Patrocinante:  
Sr. Luis Hernán Vidal Vidal  
Ingeniero Civil en Informática  
Licenciado en Ciencias de la Ingeniería

Profesor Co-Patrocinante:  
Sr. Juan Ricardo Lerdón Ferrada  
Ingeniero Agrónomo, D.E.A. en Economía Agraria,  
Doctor en Economía Agraria

Profesor Informante:  
Sra. Miguelina Vega Rosales  
Ingeniero Comercial  
Magíster en Estadística

CRISTIAN RODRIGO ROJAS PÉREZ  
VALDIVIA — CHILE  
2009



# Universidad Austral de Chile

Instituto de Informática

Valdivia, 4 de mayo de 2009.

De : Luis Hernán Vidal Vidal.

A : Sr. Juan Pablo Salazar F.

Director de Escuela de Ingeniería Civil en Informática.

Ref. : Informa Calificación Trabajo de Titulación.

MOTIVO: Informar revisión y calificación del Proyecto de Título "Desarrollo de software de gestión para la producción bovina empleando AJAX", presentado por el alumno CRISTIAN RODRIGO ROJAS PÉREZ, que refleja lo siguiente:

El esfuerzo constante, junto con la dedicación al tema son aspectos destacables durante el proceso del trabajo realizado.

Cumplimiento del objetivo propuesto.	7,0
Satisfacción de alguna necesidad.	6,5
Aplicación del método científico.	7,0
Interpretación de los datos y obtención de conclusiones.	7,0
Originalidad.	6,5
Aplicación de criterios de análisis y diseño.	7,0
Perspectivas del trabajo.	7,0
Coherencia y rigurosidad lógica.	7,0
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración.	7,0
	6,9

Por todo lo anterior expuesto califico el trabajo de titulación del Sr. CRISTIAN RODRIGO ROJAS PÉREZ con nota 6,9 (seis coma nueve).

Sin otro particular, se despide atentamente.

Ing. Luis Hernán Vidal Vidal.  
Profesor Instituto de Informática.  
Facultad de Ciencias de la Ingeniería.  
Universidad Austral de Chile.

Valdivia, 21 de abril de 2009

De: Juan Lerdón Ferrada  
Copatrocinante

A: Juan Pablo Salazar Fernández  
Director  
Escuela de Ingeniería Civil en Informática

Ref: Calificación proyecto de título

De mi consideración:

Habiendo revisado el trabajo de titulación "**Desarrollo de software de gestión para la producción bovina empleando AJAX**", presentado por el alumno Sr. **Cristian Rodrigo Rojas Pérez**, mi evaluación del mismo es la siguiente:

**Nota: 6,3** (Seis coma tres).

**Fundamento de la nota:** Interesante trabajo, enmarcado en un proyecto Fondef de I&D, cuyos resultados serán utilizados directa o indirectamente en futuros proyectos, debido a que el conocimiento desarrollado es aplicable en otros contextos. Debe mejorar, antes de ser publicado, la precisión del lenguaje técnico y la redacción. Observaciones y correcciones se incluyen en el documento adjunto.

Aspecto	Evaluación
Cumplimiento de objetivos	6,5
Satisfacción de alguna necesidad	6,5
Aplicación del método científico	6,0
Interpretación de los datos y obtención de conclusiones	6,2
Originalidad	6,5
Aplicación de criterios de análisis y diseño	6,0
Perspectivas del trabajo	7,0
Coherencia y rigurosidad lógica	6,0
Precisión del lenguaje técnico	5,7
Nota Final	<b>6,3</b>

Sin otro particular, saluda atentamente a usted,



UNIVERSIDAD AUSTRAL DE CHILE  
FACULTAD DE CIENCIAS AGRARIAS  
INSTITUTO DE ECONOMIA AGRARIA

---

Juan Lerdón Ferrada  
Director  
Instituto de Economía Agraria  
Universidad Austral de Chile

Valdivia, 14 Mayo del 2009

DE : Miguelina Vega Rosales

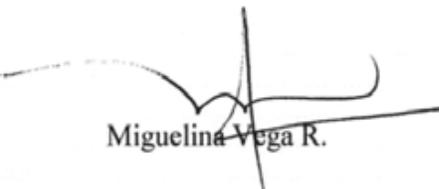
Profesor Instituto Informática

A: Dirección Escuela Ingeniería Civil en Informática

Informo a usted que el Proyecto de Título "Desarrollo de Software de Gestión para la Producción Bobina Empleando AJAX", presentado por el señor Cristian Rodrigo Rojas Pérez, tiene las siguientes calificaciones:

Ítem	Nota
Cumplimiento del objetivo propuesto	7
Satisfacción de alguna necesidad	6,8
Aplicación del método científico	6,5
Interpretación de los datos y obtención de conclusiones	6,8
Originalidad	6,2
Aplicación de criterios de análisis y diseño	6,5
Perspectivas del trabajo	6,5
Coherencia y rigurosidad lógica	6,8
Precisión del lenguaje técnico en la exposición, composición, redacción e ilustración	6,8
NOTA FINAL	6,7

Atentamente

  
Miguelina Vega R.  
Profesor Informante

# ÍNDICE DE CONTENIDO

ÍNDICE DE CONTENIDO.....	3
ÍNDICE DE TABLAS .....	5
ÍNDICE DE FIGURAS.....	6
RESUMEN.....	8
ABSTRACT.....	9
CAPÍTULO I.....	10
1. INTRODUCCIÓN.....	10
1.1. La Web 2.0.....	10
1.2. Rich Internet Applications.....	11
1.3. AJAX.....	12
1.4. Tecnologías de infocomunicación apoyando las decisiones de producción bovina.....	13
1.5. Software de gestión para la producción bovina.....	15
1.6. Antecedentes existentes al respecto.....	16
1.7. Importancia y naturaleza del estudio.....	17
1.8. Objetivos generales y específicos.....	17
1.8.1. Objetivos generales.....	17
1.8.2. Objetivos específicos.....	17
CAPÍTULO II.....	19
2. ASYNCHRONOUS JAVASCRIPT AND XML.....	19
2.1. Historia de AJAX.....	19
2.2. Estado actual.....	21
2.3. Tecnologías usadas en AJAX.....	23
2.3.1. HyperText Markup Language.....	25
2.3.2. Cascading Style Sheets.....	26
2.3.3. Extensible Markup Lenguaje.....	27
2.3.4. JavaScript.....	28
2.3.5. El objeto XMLHttpRequest.....	29
2.3.6. Web Services.....	32
2.4. Comparación del antiguo modelo Web con el Nuevo modelo AJAX.....	33
2.5. Tecnologías escogidas para el desarrollo del proyecto.....	35
2.5.1. Framework Google Web Toolkit (GWT).....	36
2.5.1.1. Arquitectura de Google Web Toolkit.....	36
2.5.1.2. Funciones del Framework GWT.....	38
2.5.2. Entorno de desarrollo integrado Eclipse.....	39
2.5.3. GWT Designer.....	39
2.5.4. Concurrent versions system (CVS).....	40
2.5.5. Base de datos PostgreSQL.....	41
2.5.6. pgAdmin.....	41
2.5.7. Hypertext Preprocessor (PHP).....	41
2.5.8. PHPEclipse.....	42
2.5.9. GWT Widget Library.....	42
2.6. Versiones de las herramientas utilizadas.....	43
2.7. Diagrama de tecnologías usadas en el software CONGABOV.....	43
CAPITULO III.....	45
3. SOFTWARE DE GESTIÓN PARA LA PRODUCCIÓN AGRÍCOLA Y BOVINA. (CONGABOV).....	45
3.1. Historia del Software.....	45
3.2. Conceptos básicos.....	47
3.3. Metodología usada para el desarrollo del software.....	49
3.4. Equipo del proyecto.....	51
3.5. Análisis de la versión de escritorio del software CONGABOV.....	52
3.6. Diseño del software.....	54
3.7. Configuración de la aplicación.....	58
3.7.1. Explotación.....	58
3.7.2. Plan de cuentas.....	59
3.7.3. Plan de actividades.....	63
3.7.4. Proveedores y clientes.....	64
3.8. Presupuestos.....	64
3.8.1. Presupuesto por actividades.....	65
3.8.2. Presupuesto de caja.....	67

3.9.	Informes del presupuesto.....	68
3.9.1.	Presupuesto técnico económico.....	68
3.9.2.	Estructura de costos.....	69
3.9.3.	Estructura de ingresos.....	72
3.9.4.	Resultado global de la explotación.....	73
3.9.5.	Presupuesto de caja.....	74
3.10.	Balance inicial.....	76
3.11.	Movimientos contables.....	77
3.12.	Libros diarios.....	82
3.12.1.	Diario de compras y ventas.....	83
3.12.2.	Diario de banco caja.....	83
3.12.3.	Resumen de compras y ventas.....	85
3.13.	Control de gestión.....	85
3.13.1.	Libro Mayor.....	86
3.13.2.	Gestión por actividad.....	87
3.13.3.	Balance general.....	88
3.13.4.	Resumen de márgenes.....	90
3.13.5.	Indicadores financieros.....	92
3.13.6.	Movimientos de caja.....	94
3.13.7.	Control de caja permanente.....	95
3.13.8.	Costo litro de leche.....	97
3.14.	Herramientas para cerrar el ejercicio.....	98
3.14.1.	Cálculo de revalorizaciones.....	98
3.14.2.	Cálculo de depreciaciones.....	99
3.14.3.	Cierre del ejercicio.....	100
3.14.4.	Reabrir ejercicio.....	100
CAPÍTULO IV.....		102
4. ANÁLISIS DE LAS TECNOLOGÍAS USADAS PARA LA CREACIÓN DE APLICACIONES		
AJAX.....		102
4.1.	Ventajas del uso de AJAX.....	102
4.1.1.	Usabilidad.....	102
4.1.2.	Basado en estándares.....	105
4.1.3.	No requiere plugins adicionales.....	106
4.1.4.	Reducción del tamaño de la información intercambiada.....	106
4.2.	Desventajas del uso de AJAX.....	106
4.2.1.	Incompatibilidades entre navegadores.....	107
4.2.2.	Pérdida de funcionalidades comunes de los navegadores.....	108
4.2.3.	Limitado por seguridad del navegador.....	110
4.2.4.	Aumento excesivo en el uso de JavaScript.....	111
4.2.5.	Aumento de solicitudes HTTP al servidor.....	112
4.3.	Recomendaciones de uso.....	113
4.3.1.	Utilizar Frameworks y librerías de componentes.....	113
4.3.2.	Disminuir el número de solicitudes al servidor.....	114
4.3.3.	Pronosticar la cantidad de datos a obtener por petición.....	115
4.3.4.	Indicar al usuario sobre el estado de la aplicación.....	117
4.3.5.	Separar las lógicas de usuario y del negocio.....	117
4.4.	Análisis del Framework GWT.....	119
4.4.1.	Ventajas de GWT.....	119
4.4.2.	Desventajas de GWT.....	120
4.4.3.	Recomendaciones para el uso de GWT.....	120
CAPÍTULO V.....		121
5. CONCLUSIONES.....		121
REFERENCIAS.....		123
LIBROS Y PUBLICACIONES.....		123
MATERIAL EN INTERNET.....		123

## ÍNDICE DE TABLAS

TABLA 1.	Ejemplo código CSS .....	26
TABLA 2.	Ejemplo documento XML.....	27
TABLA 3.	Ejemplo JavaScript.....	29
TABLA 4.	Dependencias del objeto XMLHttpRequest. ....	30
TABLA 5.	Estados del objeto XMLHttpRequest.....	30
TABLA 6.	Métodos del objeto XMLHttpRequest .....	31
TABLA 7.	Atributos del objeto XMLHttpRequest .....	31
TABLA 8.	Uso del objeto XMLHttpRequest .....	32
TABLA 9.	Versiones de las herramientas. ....	43
TABLA 10.	Módulos del sistema .....	50
TABLA 11.	Equipo de trabajo .....	52
TABLA 12.	Etapas de la gestión contable durante un ejercicio.....	58
TABLA 13.	Estructura de cuentas para la clase 6.....	59
TABLA 14.	Estructura del código de cuentas.....	60
TABLA 15.	Cuentas del activo inmovilizado .....	62
TABLA 16.	Actividades grupo bovinos de leche .....	63
TABLA 17.	Presupuesto caja.....	68
TABLA 18.	Presupuesto técnico económico .....	69
TABLA 19.	Estructura de costos (grupos de cuentas) .....	71
TABLA 20.	Estructura de costos (Cuentas).....	71
TABLA 21.	Estructura de ingresos .....	72
TABLA 22.	Resultado global de la explotación .....	74
TABLA 23.	Informe presupuesto caja .....	76
TABLA 24.	Balance inicial.....	77
TABLA 25.	Encabezado movimiento .....	78
TABLA 26.	Líneas de detalle .....	78
TABLA 27.	Ejemplo de movimiento contable.....	79
TABLA 28.	Tipos de movimientos.....	81
TABLA 29.	Diario banco caja .....	84
TABLA 30.	Gestión por actividad.....	87
TABLA 31.	Resumen de márgenes.....	91
TABLA 32.	Indicadores financieros .....	93
TABLA 33.	Movimientos de caja .....	94
TABLA 34.	Ejemplo Costos indirectos del rubro lechería .....	98
TABLA 35.	Ejemplo movimiento de revalorización .....	99
TABLA 36.	Ejemplo movimiento depreciaciones .....	99

## ÍNDICE DE FIGURAS

FIGURA 1.	Mapa meme Web 2.0 .....	11
FIGURA 2.	Esquema proyecto FONDEF .....	15
FIGURA 3.	Software de gestión para la producción bovina. ....	16
FIGURA 4.	Cronología de AJAX.....	21
FIGURA 5.	Mapa tecnologías para RIA's.....	22
FIGURA 6.	Arquitectura AJAX simplificada.....	24
FIGURA 7.	Ejemplo noticiero AJAX.....	25
FIGURA 8.	Modelo tradicional comparado con el modelo AJAX.....	34
FIGURA 9.	Comparación de la interacción producida en cada modelo.....	35
FIGURA 10.	Arquitectura GWT .....	37
FIGURA 11.	Plugin GWT Designer en Eclipse .....	40
FIGURA 12.	Diagrama tecnológico. ....	44
FIGURA 13.	Modelo de datos versión de escritorio .....	53
FIGURA 14.	Arquitectura lógica.....	54
FIGURA 15.	Modelo de datos.....	55
FIGURA 16.	Estructura pantallas .....	56
FIGURA 17.	Diagrama de flujo CONGABOV .....	57
FIGURA 18.	Explotación .....	59
FIGURA 19.	Administración de cuentas .....	62
FIGURA 20.	Actividades .....	63
FIGURA 21.	Proveedores y clientes.....	64
FIGURA 22.	Presupuesto por actividad. ....	65
FIGURA 23.	Formularios para creación de presupuestos .....	66
FIGURA 24.	Formas de ingreso de datos al presupuesto .....	67
FIGURA 25.	Presupuesto de caja.....	67
FIGURA 26.	Presupuesto técnico económico .....	69
FIGURA 27.	Estructura de costos .....	70
FIGURA 28.	Gráfico estructura de costos.....	71
FIGURA 29.	Estructura de ingresos.....	72
FIGURA 30.	Gráfico estructura de ingresos.....	72
FIGURA 31.	Resultado global de la explotación .....	73
FIGURA 32.	Gráfico resultado global de la explotación.....	74
FIGURA 33.	Informe presupuesto de caja .....	75
FIGURA 34.	Gráficos presupuesto de caja.....	75
FIGURA 35.	Balance inicial.....	76
FIGURA 36.	Administración movimientos contables.....	81
FIGURA 37.	Tipos de movimientos.....	82
FIGURA 38.	Pantalla para crear movimientos contables .....	82
FIGURA 39.	Diario compras ventas.....	83
FIGURA 40.	Diario banco caja .....	84
FIGURA 41.	Diario banco caja .....	85
FIGURA 42.	Resumen compras ventas .....	85
FIGURA 43.	Libro mayor .....	86
FIGURA 44.	Gestión por actividad .....	88
FIGURA 45.	Gráficos gestión por actividad .....	88
FIGURA 46.	Balance general.....	89
FIGURA 47.	Gráfico balance general .....	90
FIGURA 48.	Resumen de márgenes.....	91
FIGURA 49.	Gráficos resumen de márgenes .....	92
FIGURA 50.	Indicadores financieros .....	94
FIGURA 51.	Movimientos de caja .....	95
FIGURA 52.	Gráfico movimientos de caja .....	95
FIGURA 53.	Control de caja permanente.....	96
FIGURA 54.	Gráfico control de caja.....	96
FIGURA 55.	Cabeceras siempre visibles en grillas.....	103
FIGURA 56.	Paginación.....	103
FIGURA 57.	Ordenamiento de grillas .....	104
FIGURA 58.	Suggest list.....	104
FIGURA 59.	Control de ayuda .....	105
FIGURA 60.	Ejemplo de diferencias entre navegadores .....	108
FIGURA 61.	Widgets en una pantalla .....	109

FIGURA 62.	Versión imprimible .....	110
FIGURA 63.	Ejemplo solicitudes HTTP .....	112
FIGURA 64.	Ejemplo formulario .....	114
FIGURA 65.	Archivo XML y su representación gráfica.....	115
FIGURA 66.	Gráfico tiempos de procesamiento.....	116
FIGURA 67.	Indicadores de actividad.....	117
FIGURA 68.	Ejemplo separación de lógicas .....	118

## RESUMEN

Desde el surgimiento de la *World Wide Web*<sup>1</sup> a mediados de los noventa se ha visto como ha pasado de ser un simple visualizador de textos *online* a un mundo de aplicaciones interactivas de todo tipo como; *blogs*<sup>2</sup>, *Wikis*<sup>3</sup>, mapas, redes sociales, fotos, videos, noticias, lectores *RSS*<sup>4</sup>, y mucho más. Estas aplicaciones conforman las denominadas *RIA*'s<sup>5</sup> de las *WEB 2.0*<sup>6</sup>.

Son varios los motivos de esta proliferación de aplicaciones en *Internet* entre los cuales podemos destacar el surgimiento de nuevas tecnologías y técnicas que permiten a los desarrolladores lograr funcionalidades como solo antes se podían observar en aplicaciones de escritorio, entre las cuales cabe destacar *AJAX*<sup>7</sup>, técnica que será eje principal de esta tesis. Por otro lado tenemos la necesidad de las personas de organizar y digerir la creciente cantidad de información que las rodea ya sea en el ámbito personal como laboral, promoviendo el surgimiento de nuevo software para cumplir esta demanda.

Esta tesis analiza, evalúa y usa las nuevas tecnologías que se encuentran en *AJAX* para resolver un problema puntual. En los siguientes capítulos se describirá en detalle la migración de un *software* de gestión técnico-económica de explotaciones bovinas de leche y carne, construido para un ambiente de escritorio, a la *Internet*. Esta aplicación sirve para almacenar los datos de contabilidad y hacer posterior análisis de la información recolectada, ayudando así en la toma de decisiones de los empresarios del rubro agropecuario de la décima Región de los Lagos y decimocuarta Región de los Ríos.

Posteriormente, se entregan recomendaciones generadas a partir de la experiencia obtenida, las cuales pretenden ser una guía para el correcto uso de las tecnologías estudiadas en esta tesis.

---

<sup>1</sup> *World Wide Web* o Red Global Mundial es un sistema de documentos de hipertexto enlazados y accesibles a través de Internet.

<sup>2</sup> *Blog* o bitácora es un sitio *Web* periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores.

<sup>3</sup> Sitio *Web* cuyas páginas pueden ser editadas por múltiples usuarios voluntarios.

<sup>4</sup> Acrónimo de *Really Simple Syndication*. Este es un formato estándar para la sindicación de contenidos a los que un usuario puede suscribirse mediante un programa.

<sup>5</sup> Acrónimo de *Rich Internet Applications*.

<sup>6</sup> Término acuñado por Tim O'Reilly en 2004 para referirse a la segunda generación de aplicaciones en la historia de la Web.

<sup>7</sup> Acrónimo de *Asynchronous JavaScript And XML*. Es una técnica de desarrollo *Web* para crear aplicaciones interactivas o *RIA*.

## **ABSTRACT**

Since the emergence of the World Wide Web to the middle of 90's has been moved from being a simple viewer texts online to a world of interactive applications of all kinds; as blogs, wikis, maps, social networks, photos, videos, news, RSS readers, and more. These applications make up the so-called RIA's of the Web 2.0.

There are several reasons for this proliferation of applications on the Internet among which we can highlight the emergence of new technologies and techniques that allow developers to achieve functionalities as only before could be seen in desktop applications, among which include AJAX, technique that will be the main axis of this thesis. On the other hand we have the need of people to organize and digest the increasing amount of information that surrounds them, in the personal and working area, promoting the emergence of new software to meet this demand.

This thesis analyses, evaluates and use new technologies that are in Ajax to solve a specific problem. In the following chapters will be described in detail the migration of a software technical - economic management farms windings milk and meat built for a desktop environment to the Internet. This application works to store data accounting and do later an analysis of the information collected helping in the decision-making of entrepreneurs in the agriculture area of tenth region "de Los Lagos" and the fourteenth region "de Los Rios".

Subsequently are delivered recommendations generated from the experience, which are intended to be a guide for the proper use of the technologies studied in this thesis.

## **CAPÍTULO I.**

### **1. INTRODUCCIÓN.**

A continuación se presentan al lector en los conceptos básicos necesarios para comprender esta tesis, comenzando con los términos *Web 2.0*, *Rich Internet Applications* y finalmente la técnica *AJAX* con la cual se pueden crear las recién nombradas *RIA*'s.

Luego se expone el proyecto FONDEF “Tecnologías de infocomunicación apoyando las decisiones de producción bovina” para finalmente describir el modulo *CONGABOV*, *software* de gestión enfocado a los productores de leche y carne bovina el cual entregará informes, gráficos e indicadores financieros de los datos contables ingresados dando así mayor conocimiento sobre las empresas y ayudando en la toma de decisiones de las mismas. Todo esto se construirá usando la técnica *AJAX* y las tecnologías que esta involucra, para así poder dar todas las funcionalidades que requiere un sistema en Internet que manejará una cantidad considerable de datos.

#### **1.1.La Web 2.0.**

El término *Web 2.0* fue acuñado en 2004 por Tim O'Reilly<sup>8</sup>, este observo que la *Web* era cada día más importante con apasionantes nuevas aplicaciones y con sitios apareciendo con sorprendente regularidad.

Lo que significa este término es muy amplio y no esta claramente definido, incluso llegándose a insinuar que es solo propaganda o la palabra de moda, pero si analizamos el universo de aplicaciones en *Internet* podemos apreciar que este concepto surge de los cambios que se están viendo en los últimos años, insinuando que la *Web* que se tiene hoy en día no es la misma de mediados de los noventa cuando el mundo vio aparecer las primeras páginas HTML. En la figura<sup>9</sup> 1 se ve el mapa meme<sup>10</sup> creado por O'Reilly.

---

<sup>8</sup> Nacido en 1954 en Cork (Irlanda) es fundador y presidente de O'Reilly Media (editorial), es un fuerte impulsor del software libre y uno de los autores del concepto *Web 2.0*.

<sup>9</sup> Elaborada por Markus Angermeier. [URL01]

<sup>10</sup> Neologismo de una unidad teórica de información cultural transmisible de un individuo a otro.



Son muchos los beneficios de este tipo de aplicaciones entre los cuales cabe destacar la mejora en la experiencia visual, no necesitan instalación salvo alguno que otro *plugin* en el navegador como Flash por ejemplo, las actualizaciones son automáticas sin necesidad de descargar paquetes para luego ser instalados, como es el caso de las aplicaciones de escritorio, se pueden ocupar desde cualquier computador sin importar el sistema operativo en que se este trabajando.

Existen muchas plataformas para desarrollar RIA`s como Adobe Flex, Silverlight de Microsoft, JavaFX Script de Sun Microsystems, Bindows de MB Technologies y AJAX la cual será la plataforma que se usará en esta tesis.

### **1.3.AJAX.**

AJAX, acrónimo de *Asynchronous JavaScript And XML*, es una técnica de desarrollo para aplicaciones *WEB* que se definen como *RIA`s (Rich Internet Applications)*. Desde hace unos pocos años han comenzado a aparecer en *Internet* aplicaciones novedosas e interactivas como lo son *GoogleMaps*<sup>11</sup>, *Gmail*<sup>12</sup>, *Calendar*<sup>13</sup>, *Docs & Spreadsheets*<sup>14</sup>, *Meebo*<sup>15</sup> y *YouOS*<sup>16</sup> por nombrar algunas.

Estas aplicaciones se han logrado crear, cambiando el viejo paradigma síncrono (*request/response*) por un modelo asíncrono, en el cual los datos que pide el usuario se solicitan al servidor y se cargan en segundo plano, sin interferir con la visualización y comportamiento de la página. En términos básicos, lo que se hace es cargar y renderizar una página, luego mantenerse en ese lugar, mientras rutinas y *scripts* comúnmente denominadas motor AJAX van al servidor buscando en *background* los datos que se solicitan para luego mostrarlos en la misma página inicial que se cargo [URL02].

Si bien AJAX se escucho por primera vez en un artículo publicado por Jesse James Garret en el sitio de *Adaptive Path* las tecnologías y conceptos se crearon hace ya varios

---

<sup>11</sup> Servicio proporcionado por *Google* el cual ofrece imágenes de mapas desplazables, así como fotos satelitales del mundo entero.

<sup>12</sup> Servicio de correo electrónico proporcionado por *Google*.

<sup>13</sup> Servicio de agenda proporcionado por *Google*.

<sup>14</sup> Procesador de textos y plantillas *online* proporcionado por *Google*.

<sup>15</sup> Programa de mensajería instantánea que se ejecuta en un navegador *Web*.

<sup>16</sup> Aplicación que se ejecuta en un navegador y simula un Sistema Operativo.

años y este es solo el término que las agrupa y explica como estas funcionan para lograr los efectos antes nombrados.

#### **1.4. Tecnologías de infocomunicación apoyando las decisiones de producción bovina.**

El objetivo principal de esta tesis es desarrollar un módulo para el proyecto FONDEF D03I1151 "Desarrollo e introducción de un sistema georreferenciado para apoyar en línea las decisiones de la producción bovina en la Décima Región". Con este proyecto se pretende suplir una falencia en el desarrollo actual de aplicaciones en Internet para el mundo agropecuario, potenciando las tecnologías de información aplicadas a necesidades concretas de la producción agropecuaria de la zona sur de Chile.

El proyecto pretende generar un paquete de seis servicios de apoyo a la toma de decisiones asociados a la producción bovina, que poseen como características el ser interactivos, de consulta en línea e implementados en base a una plataforma georreferenciada. Estos servicios, que al final del proyecto se encontrarán en una etapa pre-comercial, estarán disponibles en un sitio *Web* para la producción bovina. Este sitio tendrá a su vez una serie de aplicaciones destinadas a tener una constante retroalimentación con los usuarios finales del proyecto: los productores de carne y leche bovina, sus asesores técnicos, sus organizaciones gremiales y empresas lácteas y cárneas, asociadas a los mismos [URL03].

Los servicios que se generarán son los siguientes:

- **Software monitor de praderas (PRADMONITOR):** Este es un instrumento para ayudar al productor a manejar sus praderas con más eficiencia y obtener de ellas una mayor productividad. Está diseñado para planificar el manejo del pastoreo a través del año, considerando las diferencias en el crecimiento de la pradera, su contenido nutricional, el tamaño del rebaño y la producción de leche de las vacas.
- **Software de manejo de fertilización (RECOMFERT):** Este es un programa computacional para la toma de decisiones sobre el manejo de la fertilidad del suelo y la fertilización de los cultivos forrajeros y las praderas permanentes. Su objetivo es el cálculo de la dosis de fertilización óptima, considerando el

rendimiento alcanzable de un determinado agro ecosistema y para un potrero en específico que se desee fertilizar.

- **Software de gestión para el aseguramiento de calidad higiénica de leche y Producción Limpia (GESCALE):** Este es un software que permite evaluar la gestión lechera desde el punto de vista de la calidad higiénica de la producción, a través de la comparación de datos ingresados con estándares nacionales e internacionales relevantes.
- **Simulador de precios de la leche basados en pautas de pago (SIMUPLEC):** Este es un software con el cual el productor y/o asesor podrá obtener una simulación del pago máximo o mínimo por litro de leche a obtener en distintas plantas, digitando parámetros relacionados con su producción de leche y así poder efectuar comparaciones del precio obtenido si vendiera a una u otra planta.
- **Software de recomendaciones de alimentación (RECALIM):** Esta es una herramienta de apoyo en la toma de decisiones a nivel predial. El agricultor podrá formular una ración de mínimo costo *online* (a través de Internet) para la alimentación de bovinos de leche y carne.
- **Software de gestión para la producción bovina (CONGABOV):** Este es un software de apoyo a la gestión técnico-económica de explotaciones bovinas de leche y carne el cual sirve para planificar, controlar y evaluar permanentemente la empresa a partir de un presupuesto y de un control de gestión.

A continuación en la figura<sup>17</sup> 2 se muestra como esta conformado este proyecto, de que partes consta y como estas interactúan entre si. Destacamos el software CONGABOV ya que en el trabajará esta tesis.

---

<sup>17</sup> Creada por Cristian Rojas Pérez en base a la imagen de la Infografía del Proyecto FONDEF D03I1151. [URL03].

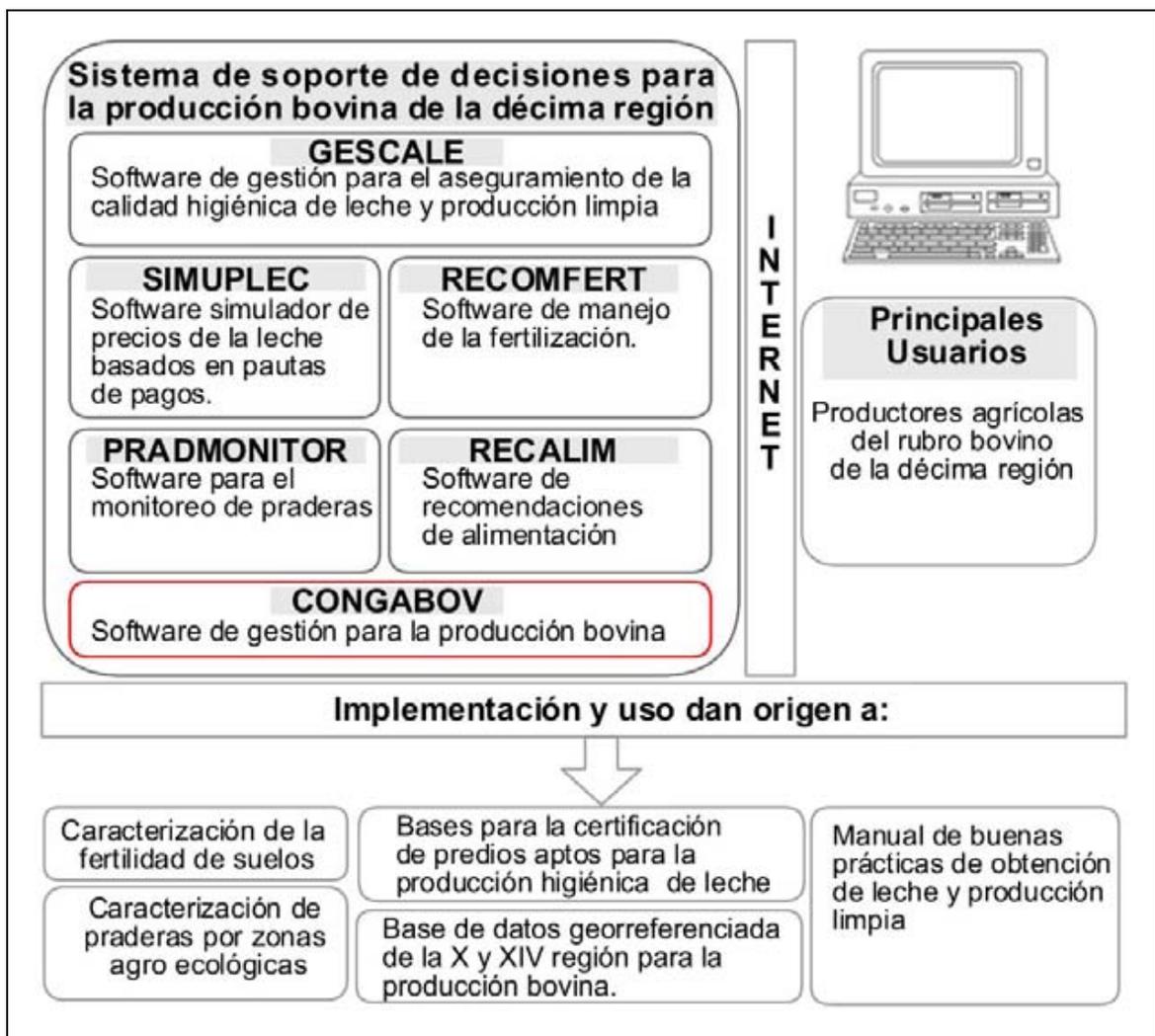


FIGURA 2. Esquema proyecto FONDEF

### 1.5. *Software* de gestión para la producción bovina.

CONGABOV nace de la necesidad de toda empresa de registrar, controlar y analizar la información contable que genera y así poder prever y crear estrategias eficaces para maximizar los beneficios de la misma. Uno de los objetivos de este software es ayudar en la necesidad que tiene todo empresario de establecer un presupuesto o plan de explotación y pronosticar su financiamiento. La elaboración del presupuesto permite plantear los problemas de equilibrio y expansión de la empresa. Los resultados que proporciona, permiten conocer si se dispondrá o no de ingresos suficientes para financiar el funcionamiento de su negocio, siendo alrededor de esta necesidad vital donde se centran los problemas decisivos de la mayoría de las explotaciones.

A continuación, el módulo contabilidad de gestión tiene como objetivo servir de apoyo a las decisiones de producción e inversión del empresario agrícola, mediante la entrega periódica de información relacionada con el desarrollo de cada uno de los rubros que

componen la empresa. Este módulo permite controlar, en forma permanente, como se esta llevando a cabo el plan de explotación formulado a comienzos del período contable.

Con toda la información que se ingresa al sistema se generan libros diarios, informes contables e indicadores financieros los cuales ayudan al agricultor a ver como se esta llevando a cabo lo planificado en el presupuesto y tomar decisiones puntuales en base a los datos entregados.

En la figura<sup>18</sup> 3 se aprecia de manera general las partes que componen este *software* y las funcionalidades que entrega.

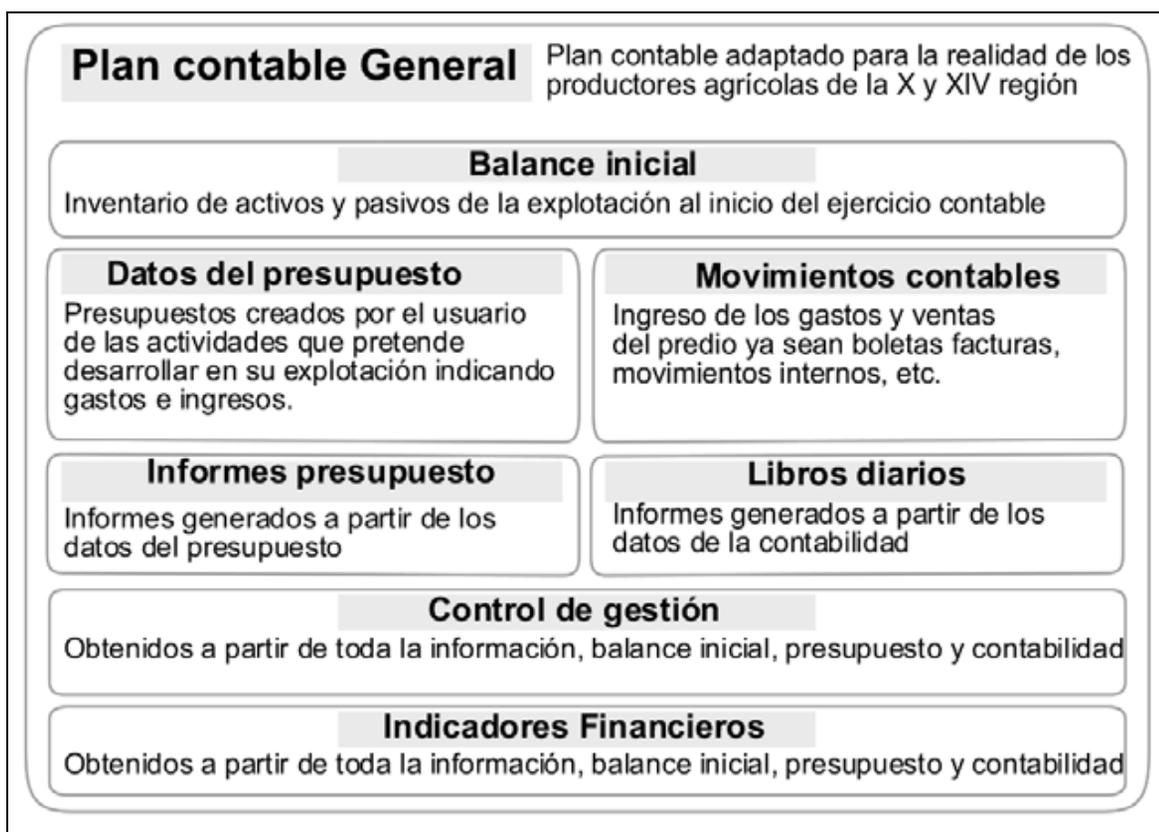


FIGURA 3. *Software* de gestión para la producción bovina.

### 1.6. Antecedentes existentes al respecto.

Se cuenta con la primera versión del *software* CONGABOV, esta fue creada por investigadores Instituto de Economía Agraria de la Universidad Austral de Chile, pasando por varias tecnologías, hasta que finalmente, su última versión fue desarrollada en Visual Basic 6.0 utilizando una base de datos ACCES. Se procederá a analizar la

<sup>18</sup> Creada por Cristian Rojas Pérez.

última versión de este *software*, para así poder extraer una idea básica de su diseño y desarrollar un sistema *online* multiusuario, que resuelva la misma problemática incluyendo mejoras y nuevas funcionalidades.

### **1.7.Importancia y naturaleza del estudio.**

Al estudiar las tecnologías de punta usadas para la creación de aplicaciones en *Internet*, se podrá desarrollar el *software* CONGABOV con todas las funcionalidades y capacidades que debería tener una herramienta contable, dando de esta manera una mejor experiencia a los usuarios de este sistema.

Dada la gran cantidad de tecnologías usadas en la técnica AJAX, es de gran importancia que en esta tesis se entreguen recomendaciones para poder hacer un buen uso de estas, todo esto mostrando ejemplos reales de uso, obtenidos al desarrollar la aplicación antes nombrada.

### **1.8.Objetivos generales y específicos.**

#### **1.8.1. Objetivos generales.**

- Desarrollar software de apoyo a la gestión técnico-económica de explotaciones bovinas de leche y carne (CONGABOV) para la decimocuarta región de Los Ríos y décima región de Los Lagos con la técnica de desarrollo WEB denominada AJAX utilizando el *framework*<sup>19</sup> GWT creado por Google.

#### **1.8.2. Objetivos específicos.**

- Identificar y evaluar tecnologías usadas para el desarrollo de RIA basado en AJAX utilizando el *framework* GWT.
- Analizar la aplicación de escritorio CONGABOV y diseñar su versión WEB multiusuario.

---

<sup>19</sup> Es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

- Implementar el Software CONGABOV (Contabilidad de Gestión Agrícola y Bovina) en un sistema en Internet multiusuario usando las tecnologías RIA's AJAX.
- Identificar ventajas y desventajas de AJAX y generar recomendaciones de trabajo al utilizar el *framework* GWT.

## CAPÍTULO II

### 2. ASYNCHRONOUS JAVASCRIPT AND XML.

En este capítulo se procederá a explicar el origen y funcionamiento de AJAX para que quede absolutamente claro la forma en que se desarrollo el Software CONGABOV y como se lograron todas sus funcionalidades.

También, se justifican las tecnologías escogidas para realizar este proyecto y la forma en que estas interactuarán entre si, ya que, como se puede observar mas adelante en este capítulo, la cantidad de tecnologías y *frameworks* existentes en la actualidad es bastante amplia, no habiendo mejores ni peores si no mas bien cada uno de estos ha sido creado para diferentes contextos y con distintos puntos de vista, siendo tarea del ingeniero el escoger el que mas se ajuste al problema que este desee resolver.

#### 2.1.Historia de AJAX.

Los orígenes de *Internet* se remontan a 1969 cuando se estableció la primera conexión de computadoras, conocida como ARPANET, entre tres universidades en California y una en Utah, EE. UU. En 1989 fue creada la WWW por Tim Berners-Lee y el belga Robert Cailliau, el primero de estos desarrolla en 1990 el primer navegador *Web* para recuperar y visualizar documentos de hipertexto y en 1991 el HTML, el cual es el lenguaje de marcado predominante en la construcción de páginas *Web*.

En 1994 el mismo Tim Berners-Lee crea el World Wide Web Consortium el cual produce estándares para la WWW. Los estándares que cabe destacar para el nacimiento de AJAX son el CSS en 1996, XML y DOM en 1998 y XHTML el 2000 entre otros.

En 1995 surge el lenguaje de programación interpretado JavaScript, el cual es soportado por todos los navegadores modernos, en 1997 la ECMA<sup>20</sup> lo adopta como estándar, poco después también se convierte en estándar ISO.

Las primeras técnicas de carga asíncrona de contenidos vienen desde el año 1996, cuando Internet Explorer 3 incluye el elemento *iframe*<sup>21</sup> y Netscape en el año 1997

---

<sup>20</sup> European Computer Manufacturers Association. organización internacional basada en membresías de estándares para la comunicación y la información.

introduce el elemento `layer`<sup>22</sup>. Ambos tipos de elementos podían llamar a una URL externa y con JavaScript tomar el contenido y cargarlo en la página paterna.

Luego aparece en 1998 el *Microsoft's Remote Scripting* o MSRS, el cual enviaba y recibía datos a través de un Applet<sup>23</sup> Java y se comunicaba con el cliente utilizando JavaScript.

Posteriormente las comunidades de desarrolladores *Web* crean una gama de técnicas de *Scripting* remoto para obtener los mismos resultados en diferentes navegadores. Con esto se crean las librerías JSRS<sup>24</sup> en el 2000, la introducción de la técnica imagen/cookie en el mismo año y la técnica JavaScript bajo demanda (*JavaScript on Demand*) en 2002. En este año también se sustituye el Applet Java por la interfaz XMLHttpRequest en el MSRS incorporado por primera vez en la versión 5.0 de Internet Explorer como objeto ActiveX<sup>25</sup>. Este objeto está implementado en la mayoría de los navegadores hoy en día, siendo una de las piedras angulares de la técnica AJAX, pero su implementación todavía no se ha hecho de manera estándar, habiendo diferencias entre los distintos navegadores. El World Wide Web Consortium<sup>26</sup> presentó el 27 de septiembre de 2006 el primer borrador de una especificación estándar de la interfaz. La versión actual del 15 de abril de 2008, etiquetada como borrador final (last call working draft), es el resultado de varias revisiones.

Mientras no se alcance una versión definitiva, los desarrolladores de aplicaciones *Web* deberán tener en cuenta las diferencias entre implementaciones o bien utilizar paquetes que realicen esta función. En estos años comienzan a proliferar soluciones para atender este problema, como Prototype, JQuery, DOJO por nombrar algunas, en un universo muy amplio de *frameworks*.

Finalmente en febrero 2005 Jesse James Garrett publica un artículo en el cual agrupa estas tecnologías en un solo término denominado AJAX, el cual es el acrónimo de

---

<sup>21</sup> Es un elemento HTML que permite insertar o incrustar otro documento HTML dentro de un documento HTML principal en Internet Explorer.

<sup>22</sup> Elemento que cumple la misma funcionalidad del `iframe` de Internet Explorer pero en Netscape.

<sup>23</sup> Un applet es un componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo un navegador web.

<sup>24</sup> Es una técnica de desarrollo Web para crear aplicaciones interactivas mediante el uso de una combinación de HTML, DOM y una capa de transporte con el `iframe`.

<sup>25</sup> Es un sistema de objeto distribuido y un protocolo desarrollado por Microsoft que permite a un editor encargar a otro la elaboración de parte de un documento y posteriormente volverlo a importar.

<sup>26</sup> Consorcio internacional que produce estándares para la World Wide Web.

*Asynchronous JavaScript And XML* y explica la forma en que trabajan estas para lograr las funcionalidades y aplicaciones de las que se ha estado hablando en este documento.

En la figura<sup>27</sup> 4 se muestra la cronología de eventos hasta que se crea el término AJAX.

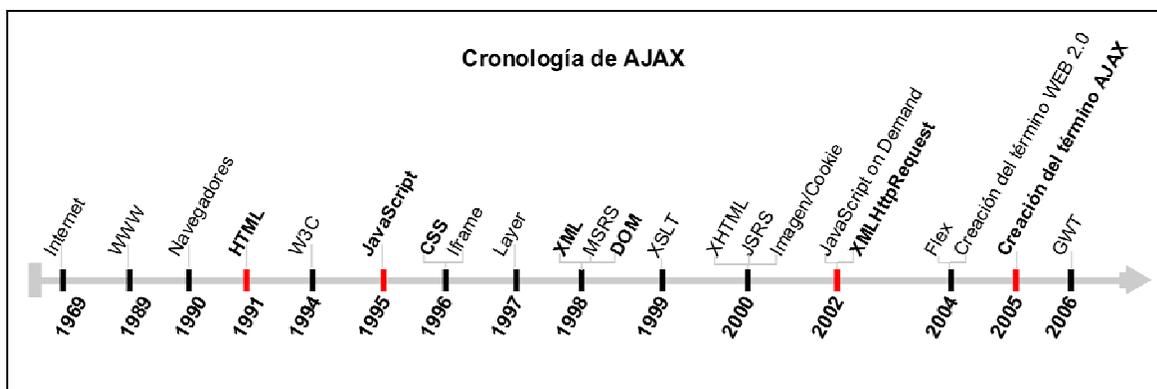


FIGURA 4. Cronología de AJAX.

## 2.2.Estado actual.

Existen muchas tecnologías y *frameworks* para desarrollar *Rich Internet Applications*, como Adobe Flex<sup>28</sup>, OpenLaszlo<sup>29</sup>, Atlas<sup>30</sup>, GWT, Echo2<sup>31</sup> por nombrar algunos entre muchos otros mas como se puede ver en la figura<sup>32</sup> 5.

<sup>27</sup> Imagen creada por Cristian Rojas Pérez.

<sup>28</sup> Conjunto de tecnologías para dar soporte al desarrollo de RIA's basadas en la plataforma propietaria Flash.

<sup>29</sup> Es una plataforma código abierto para el desarrollo y distribución de RIA's.

<sup>30</sup> Es un conjunto de extensiones para ASP.NET desarrollado por Microsoft para implementar la funcionalidad de Ajax.

<sup>31</sup> Echo es un *framework* para el desarrollo de aplicaciones Web orientado a objetos y basado en eventos.

<sup>32</sup> Imagen creada por Cristian Rojas Pérez.

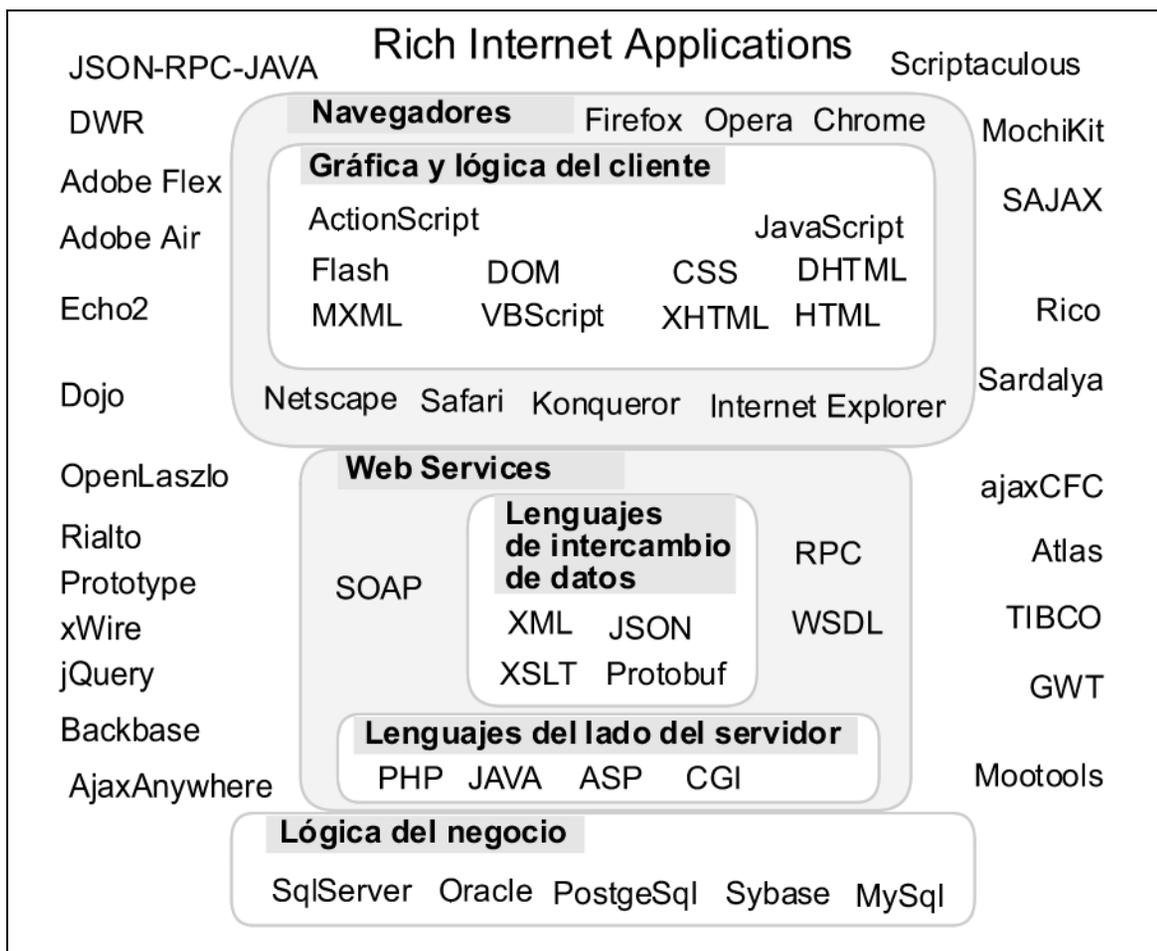


FIGURA 5. Mapa tecnologías para RIA's

En esta imagen vemos las tecnologías usadas, divididas en el lado del cliente (navegadores), *web services*<sup>33</sup> y la lógica del negocio, fuera de esta división una pequeña muestra de todo el universo de *frameworks* existentes que usan estas tecnologías, lenguajes, protocolos y estándares para crear las *Rich Internet Applications*.

Cabe destacar que algunos de los *frameworks* mostrados en la imagen anterior no se consideran parte de AJAX, ya que no usan JavaScript ni tampoco HTML como es el caso de Adobe Flex que usa el lenguaje ActionScript creado por Macromedia y el *plugin*<sup>34</sup> Flash para generar la lógica de usuario y gráfica, pero estos si crean aplicaciones asíncronas y no necesitan recargar la página para actualizar contenido. En este documento nos centraremos solo en los componentes de AJAX, citando el resto de tecnologías para hacer comparaciones cuando corresponda.

<sup>33</sup> Es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones.

<sup>34</sup> Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de la API.

Para explicar AJAX utilizaremos las tecnologías más básicas y esenciales como los son el HTML, CSS, JavaScript entre otras, simplificando así esta tarea y se recurrirá constantemente a la comparación del nuevo modelo con el antiguo basado en navegación por páginas, logrando comparar de esta manera los paradigmas síncrono y asíncrono.

### **2.3. Tecnologías usadas en AJAX.**

Luego de años de avances y pruebas se han creado estándares y tecnologías que facilitan el desarrollo de este tipo de aplicaciones, así al hablar de AJAX nos estamos refiriendo al término que las agrupa. Las tecnologías esenciales usadas se pueden dividir de la siguiente manera:

- HTML, XHTML y CSS para la presentación gráfica. Todo esto podría ser reemplazado por MXML, lenguaje con el cual la herramienta FLEX genera sus componentes gráficos, usando el *plugin* Flash en el navegador.
- DOM para la manipulación dinámica de los elementos HTML o XML.
- XML, XSLT, HTML o texto plano, para el intercambio de información con el servidor. Estos lenguajes podrían ser reemplazados por otros mas óptimos para la aplicación que se desee desarrollar, como JSON, o incluso usar formatos binarios para disminuir el flujo de datos como es el caso de Protobuf de Google.
- JavaScript para manipular todos los elementos antes mencionados y realizar peticiones HTTP asíncronas con el objeto XMLHttpRequest. En el caso de que se use el framework FLEX, el lenguaje de *Scripting* hubiera sido ActionScript o también se podría utilizar VBScript de Microsoft para estas tareas.

En la figura<sup>35</sup> 6 se simplifica el modelo, dejando solamente las partes de este que se necesitan para explicar AJAX y las tecnologías básicas para su implementación.

---

<sup>35</sup> Imagen creada por Cristian Rojas Pérez.

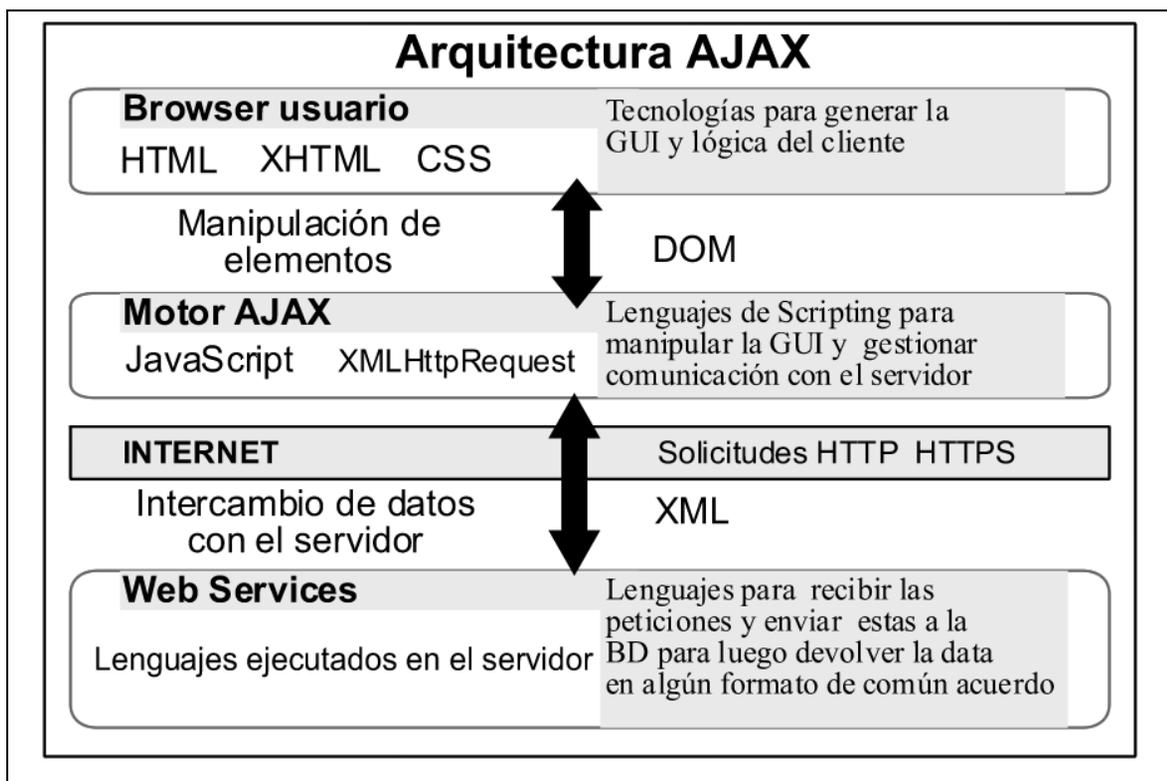


FIGURA 6. Arquitectura AJAX simplificada.

Lo que se quiere dejar en claro es que ya existen muchas tecnologías para crear RIA`s, pero todas estas aplicaciones deben tener al menos los siguientes elementos:

- Lenguajes para generar elementos gráficos en el lado del cliente.
- Lenguajes de *Scripting* que se ejecuten en el lado del cliente, para así manipular los elementos y actualizar los datos sin que se recargue la página mediante peticiones asíncronas.
- Lenguajes para intercambiar información con el servidor.
- Lenguajes que se ejecuten en el lado del servidor para generar *Web Services*.

A continuación se procede a exponer cada una de las tecnologías usadas en esta técnica, para después explicar como estas interactúan entre si. Para esto usaremos un ejemplo muy simple, el cual consiste en una página *Web*, en la que el usuario puede seleccionar de una lista de noticias la que desee leer y esta se cargará de manera asíncrona a continuación en la misma página. Cabe destacar que los códigos que veremos a continuación se alejan de la realidad de una aplicación en producción, pretendiendo ser solo de carácter educativo. En la figura 7 se muestra la página de noticias.

<b>Noticias:</b>
<a href="#">1 - Concluye la cuarta y última salida espacial de los astronautas del Endeavour.</a> <a href="#">2 - Presentan WikiEduca, nueva plataforma educativa.</a> <a href="#">3 - Glaciares inmensos bajo la superficie de Marte.</a>
<b>Noticia seleccionada:</b>
<p>JUEVES, NOVIEMBRE 20, 2008</p> <h2>Glaciares inmensos bajo la superficie de Marte</h2> <p><i>La sonda Mars Reconnaissance Orbiter (MRO) de la NASA ha detectado glaciares que cubren una superficie de decenas de miles de kilómetros cuadrados y se extienden desde las montañas marcianas con un espesor en algunos casos de unos 800 metros.</i></p> <p>"Se trata de un descubrimiento muy importante porque no sólo constata la existencia de agua en Marte sino que cubre la necesidad del líquido elemento que tendrán las futuras misiones interplanetarias", asegura Ali Safaeinili, científico del Laboratorio de Propulsión a Chorro (JPL) de la NASA. "El hallazgo también es importante porque el agua de esos glaciares se encuentra en latitudes bajas del planeta, lejos de los polos, y en algunos casos a sólo unos tres metros de profundidad", indicó. Debido a que el agua es uno de los ingredientes esenciales de la vida tal como la conocemos en la Tierra, la existencia de grandes reservas del líquido congelado es un signo alentador para los científicos que buscan vida más allá de nuestro planeta, según señala el estudio publicado por 'Science'.</p> <p>Cristian Rojas Pérez <span style="float: right;">crojasperez@gmail.com</span></p>

FIGURA 7. Ejemplo noticiero AJAX.

Este ejemplo está compuesto por una página llamada `index.html`, en la cual se encuentran el código HTML, CSS y JavaScript necesarios para dar la estructura, apariencia y funcionalidad a la aplicación y otro archivo llamado `noticias.xml`, en el cual se encuentra la información de la noticia solicitada.

### 2.3.1. HyperText Markup Language

*HyperText Markup Language* o HTML por sus siglas en inglés, puede ser usado para representar noticias en hipertexto, documentación *online* e *hipermedia* colaborativa, resultados de consultas a bases de datos, documentos de estructura simple con gráficos, etc.

La *World Wide Web* (W3) relaciona información vinculada a través del mundo. HTML provee un formato simple para representar esta información, y todos los programas compatibles con la W3 deben ser capaces de manejar documentos con esta estructura. [URL04]

Se escribe en forma de "etiquetas", rodeadas por corchetes angulares (<,>). Este puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un script (por ejemplo Javascript), el cual puede afectar el comportamiento de navegadores Web y otros procesadores de HTML.

No se verá en mayor detalle este código, por ser muy extenso y fácil de comprender por el lector, en caso de que desee ahondar mas en el tema usando las referencias de Internet aquí citadas.

### **2.3.2. Cascading Style Sheets**

*Cascading Style Sheets* o CSS por sus siglas en Ingles, es un mecanismo para dar estilo a documentos HTML y XML, que consiste en reglas simples, a través de las cuales se establece cómo se va a mostrar un documento en pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento, a través de un dispositivo de lectura. Esta forma de descripción ofrece a los desarrolladores control total sobre el estilo y formato de sus documentos, separando contenido y presentación. [URL05]

CSS funciona a base de reglas, es decir, declaraciones sobre el estilo de uno o más elementos. La regla tiene dos partes; un selector y la declaración. A su vez la declaración está compuesta por una propiedad y el valor que se le asigne.

A continuación, se ve en la tabla 1 el código CSS del noticiero AJAX que ha sido usado como ejemplo, en este se aprecia como se definen las reglas de presentación para los titulo, resúmenes, fecha, etc.

TABLA 1. Ejemplo código CSS

```
<style type="text/css">
  div#encabezado{
    font-family: Arial; font-size: 12px; font-weight bold;
  }
  div#fecha{
    font-family: Arial; font-size: 10px;
  }
  div#titulo{
    font-family: Arial; font-size: 20px;
  }
  div#resumen{
    font-family: Arial; font-size: 12px; font-style: italic;
  }
  div#cuerpo{
    font-family: Arial; font-size: 12px; text-align: justify;
  }
  div#creador{
    font-family: Arial; font-size: 10px;
  }
  div#contacto{
    font-family: Arial; font-size: 10px;
  }
</style>
```

### 2.3.3. Extensible Markup Lenguaje

*Extensible Markup Language* o XML por sus siglas en Inglés, es un simple y muy flexible formato de texto derivado del SGML (ISO 8879). Originalmente diseñado para enfrentarse al reto de la publicación electrónica a gran escala, XML también esta jugando un rol cada vez mas importante en el intercambio de una gran variedad de información a través de la *Web* y cualquier otro lugar. [URL06]

Los documentos XML están conformados por unidades de almacenamiento llamadas entidades que pueden o no tener información parseada. La información parseada esta compuesta por caracteres, de los cuales algunos son de información y otros son de marcas. Las marcas codifican la descripción del documento, diseño de almacenamiento y la estructura lógica. XML provee un mecanismo para imponer restricciones en el diseño de almacenamiento y estructura lógica. [URL07]

A continuación en la tabla 2 se presenta una parte del documento XML usado en el noticiero AJAX, este almacena la información correspondiente a la noticia seleccionada por el usuario.

TABLA 2. Ejemplo documento XML

```
<?xml version="1.0" encoding="utf-8"?>
<datos>
  <noticia>
    <fecha>
      JUEVES, NOVIEMBRE 20, 2008
    </fecha>
    <titulo>
      Glaciares inmensos bajo la superficie de Marte
    </titulo>
    <resumen>
      La sonda Mars Reconnaissance Orbiter (MRO) de la NASA ha
      .....
    </resumen>
    <cuerpo>
      "Se trata de un descubrimiento muy importante porque no sólo
      .....
    </cuerpo>
    <creador>
      Cristian Rojas Pérez
    </creador>
    <contacto>
      crrojasperez@gmail.com
    </contacto>
  </noticia>
</datos>
```

### 2.3.4. JavaScript

JavaScript es un lenguaje de scripts, interpretado, multiplataforma y parcialmente orientado a objetos. Fue creado por Netscape específicamente para su uso en el desarrollo de sitios *Web*. Actualmente es un estándar mantenido por el ECMA.

El código JavaScript puede enlazarse o añadirse a las páginas Web proporcionando un control total y dinámico sobre ellas. Además, también permite controlar hasta cierto punto las aplicaciones que lo ejecutan, habitualmente navegadores. [URL09]

El lenguaje fue inventado por Brendan Eich en la empresa Netscape Communications, que es la que desarrolló los primeros navegadores *Web* comerciales. Apareció por primera vez en el producto de Netscape llamado Netscape Navigator 2.0.

Tradicionalmente, se venía utilizando en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación únicamente cliente, sin acceso a funciones del

servidor. JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Los autores inicialmente lo llamaron Mocha y más tarde LiveScript, pero fue rebautizado como JavaScript en un anuncio conjunto entre Sun Microsystems y Netscape, el 4 de diciembre de 1995. [URL10]

En el contexto de AJAX, JavaScript sirve para darle la funcionalidad en el lado del cliente a las aplicaciones, ya sea validación de formularios, gestionar las peticiones al servidor, cambiar dinámicamente el HTML dependiendo de lo que requiera el usuario, etc. Por ejemplo, en el noticiero AJAX cuando el usuario selecciona la noticia que desea leer JavaScript toma este evento y ejecuta la función

En la tabla 3 se expone la función usada en el noticiero AJAX, que toma la respuesta en formato XML de la solicitud XMLHttpRequest y despliega los datos de la noticia seleccionada.

TABLA 3. Ejemplo JavaScript

```
<script type="text/javascript">
function enviaSolicitud () {
    // Se envia la solicitud
    solicitudAjax('noticias.xml','','solicitudRecibida');
}
function solicitudRecibida () {
    //Si es que se ha hecho correctamente la solicitud se prosigue.
    if(req.readyState==4) {
        var xml = req.responseXML;
        // Se obtiene el elemento XML fecha
        var fecha =
xml.getElementsByTagName("fecha")[0].childNodes[0].nodeValue;
        // Se modifica el contenido de la capa fecha para que muestre la
        // fecha obtenida.
        document.getElementById('fecha').innerHTML = fecha;
        .....
        .....
    }
}
</script>
```

### 2.3.5. El objeto XMLHttpRequest.

Como se menciona anteriormente, este objeto creado por Microsoft ha sido incorporado en la mayoría de los navegadores, habiendo diferencias en su implementación

provocando que al momento de desarrollar una aplicación que invoque esta interface haya que tener en cuenta en que o en cuales navegadores se ejecutará esta.

Para resolver este problema la W3C presento el 27 de diciembre de 2006 el primer borrador de una especificación estándar para esta interfaz. Con fines de explicar como funciona este objeto y evitar el problema de la compatibilidad entre navegadores, se usará la última revisión de esta especificación, entregada por la W3C el 15 de abril de 2008. [URL11]

La especificación del objeto XMLHttpRequest, define una API que provee *Scripts* para dar la funcionalidad al cliente de enviar o cargar información desde un servidor. El nombre de este objeto fue puesto pensando en mantener compatibilidad con las primeras versiones que aparecieron en la *Web*, pero puede ser muy engañoso debido a varias cosas. Primero el objeto soporta cualquier formato basado en texto, no solo el XML. Segundo, puede ser usado para efectuar peticiones HTTP y HTTPS, incluso en algunas implementaciones se pueden realizar mas tipos de peticiones, aparte de las nombradas recientemente, pero estas no serán expuestas en este documento debido a que no forman parte de la última especificación de la W3C. Finalmente soporta todo tipo de peticiones definidas en el protocolo HTTP.

Este objeto depende directamente de las siguientes especificaciones que se pueden ver en la tabla 4.

TABLA 4. Dependencias del objeto XMLHttpRequest.

Especificación	Descripción
DOM	Un agente que implemente este objeto debe soportar las funcionalidades definidas en DOM2Events [URL12] y DOM3Core [URL13].
HTML5	Esta especificación depende del HTML5 [URL14] para implementar el objeto Window y encontrar la codificación de caracteres de un recurso text/html.
HTTP	El agente que use este objeto debe soportar algún protocolo HTTP [URL15], que al menos soporte los siguientes métodos: GET, POST, HEAD, PUT, DELETE, OPTIONS.

Este se puede encontrar en cinco estados distintos como se puede apreciar en la tabla 5.

TABLA 5. Estados del objeto XMLHttpRequest

Estado	Valor	Descripción
UNSENT	0	Estado con el que se inicializa el objeto.
OPENED	1	Estado del objeto cuando el método open() fue satisfactoriamente invocado.
HEADRES_RECEIVED	2	Estado del objeto cuando todas las cabeceras de respuesta han sido recibidas.
LOADING	3	Estado del objeto cuando el cuerpo de la respuesta ha sido recibido.
DONE	4	Estado del objeto cuando toda la información ha sido transmitida o ha ocurrido un error en el proceso

Como todo objeto posee varios métodos y atributos los cuales se muestran en la tabla 6 y 7.

TABLA 6. Métodos del objeto XMLHttpRequest

Método	Descripción
open(method, url, async, user, password)	Especifica el método, URL y otros atributos opcionales de una petición.
setRequestHeader(header, value)	Añade un par etiqueta/valor a la cabecera HTTP a enviar.
send(data)	Envía la petición.
abort()	Cancela la petición en curso deteniendo el algoritmo de la función send().
getAllResponseHeaders()	Devuelve el conjunto de cabeceras HTTP como una cadena.
getResponseHeader(header)	Devuelve el valor de la cabecera HTTP especificada.

TABLA 7. Atributos del objeto XMLHttpRequest

Atributo	Tipo	Descripción
readyState	unsigned short	Entrega la constante correspondiente al actual estado del objeto.
responseText	DOMString	Si el estado actual del objeto es LOADING o DONE retorna la respuesta de la petición, en caso contrario retorna una cadena de texto vacía.
responseXML	Document	Si el estado actual del objeto es DONE, retorna la respuesta XML, en caso contrario retorna <i>null</i> .
status	unsigned short	Devuelve el código de estado HTTP enviado por el servidor, de no poder obtenerla lanza la excepción INVALID_STATE_ERR.
statusText	DOMString	Devuelve el texto asociado al estado HTTP enviado por el servidor.

En la tabla 8 se muestra el código JavaScript usado en el noticiero AJAX que utiliza este objeto para hacer una solicitud HTTP y obtener el fichero noticias.xml, en el cual se encuentran los datos de la noticia seleccionada por el usuario.

TABLA 8. Uso del objeto XMLHttpRequest.

```
var req = false;
//Se verifica el navegador
if (window.XMLHttpRequest) {
  req = new XMLHttpRequest();
} else if (window.ActiveXObject) {
  req = new ActiveXObject("Microsoft.XMLHTTP");
}
//Función que asigna los parámetros al objeto.
function solicitudAjax(url, parametros, accion) {
  req.open('GET', url, true);
  req.onreadystatechange = accion;
  req.send(parametros);
}
```

Como en este ejemplo no se han usado librerías externas se ha tenido que verificar de una manera muy básica que sentencia usar al crear el objeto, en desarrollos reales es mucho el tiempo que se tiene que invertir en asegurar que la aplicaciones sean compatibles con la mayor cantidad de navegadores posibles, por lo que es altamente recomendable usar *frameworks* que faciliten esta tarea.

### 2.3.6. Web Services

Un servicio *Web* es un sistema de *software* diseñado para apoyar la interacción de interoperabilidad entre aplicaciones a través de una red. Este posee una interfaz que lo describe en un formato procesable por maquinas (específicamente WSDL<sup>36</sup>). Otros sistemas interactúan con el servicio Web en la manera prescrita por su descripción, usando mensajes SOAP, usualmente transportados vía HTTP en formato XML y en conjunto con otros estándares *Web* relacionados. [URL16]

Los servicios *Web* son independientes de los lenguajes de programación y plataformas. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios *Web*.

---

<sup>36</sup> Web Services Description Language, formato XML que se utiliza para describir servicios Web

como PHP, con el cual se puede crear un servicio Web para que este sea llamado por nuestro noticiero o alguna otra aplicación desarrollada por un tercero.

Si bien los servicios *Web* no forman parte de AJAX, se nombran aquí ya que serán necesarios al momento de desarrollar la aplicación CONGABOV, siendo recomendable que el lector cuente con una noción básica de estos.

## **2.4.Comparación del antiguo modelo *Web* con el Nuevo modelo AJAX.**

Se acaban de analizar las tecnologías necesarias para crear aplicaciones AJAX, ahora que ya se cuenta con los elementos básicos se procederá a explicar como estas interactúan entre si para lograr el efecto de asincronía que se puede apreciar en aplicaciones como Flickr y Google Maps por nombrar algunas.

El modelo clásico de aplicaciones *Web* trabaja de la siguiente forma: la mayoría de las acciones de usuario en la interface producen una solicitud HTTP al servidor. Este realiza procesamientos, recupera información, calcula algunos números, conversa con varios sistemas y luego retorna una página HTML al cliente. Este es un modelo adaptado de la *Web* original, usada como un medio de hipertexto, pero lo que hace a la *Web* buena para el hipertexto no necesariamente la hace buena para las aplicaciones de *software*. [URL02]

Una de las grandes desventajas de este modelo es que por casi cada acción que realice el usuario este tiene que esperar a que el servidor le envíe de vuelta lo solicitado, produciendo muchos tiempos muertos al dejar la página en blanco mientras se carga el nuevo contenido.

Una aplicación AJAX elimina la dinámica “arrancar-frenar-arrancar-frenar” del modelo Web tradicional, introduciendo un intermediario –un motor AJAX- entre el usuario y el servidor. Pareciera que sumar una capa a la aplicación la haría menos reactiva, pero la verdad es lo contrario. [URL02]

En la figura<sup>37</sup> 8 se muestra una comparación de estos dos modelos.

---

<sup>37</sup> Obtenida del artículo publicado por Jesse James Garret “Ajax: A New Approach to Web Applications”



FIGURA 8. Modelo tradicional comparado con el modelo AJAX.

La diferencia que se puede apreciar aquí, es que en vez de enviarse directamente las peticiones del usuario al servidor, estas son enviadas a un motor AJAX, el cual se encarga de gestionar el tráfico entre el servidor y la interfaz. Este motor AJAX se carga al inicio de la sesión y permite que la comunicación entre el usuario y el servidor se produzca asincrónicamente, de esta manera el usuario nunca verá una pagina en blanco esperando a que los datos sean cargados, esto debido a que las acciones que comúnmente generaban una solicitud http en el antiguo modelo ahora son transformadas en solicitudes a funciones normalmente en JavaScript que conforman el motor AJAX como se puede ver en la figura<sup>38</sup> 9.

<sup>38</sup> Obtenida del artículo publicado por Jesse James Garret "Ajax: A New Approach to Web Applications"

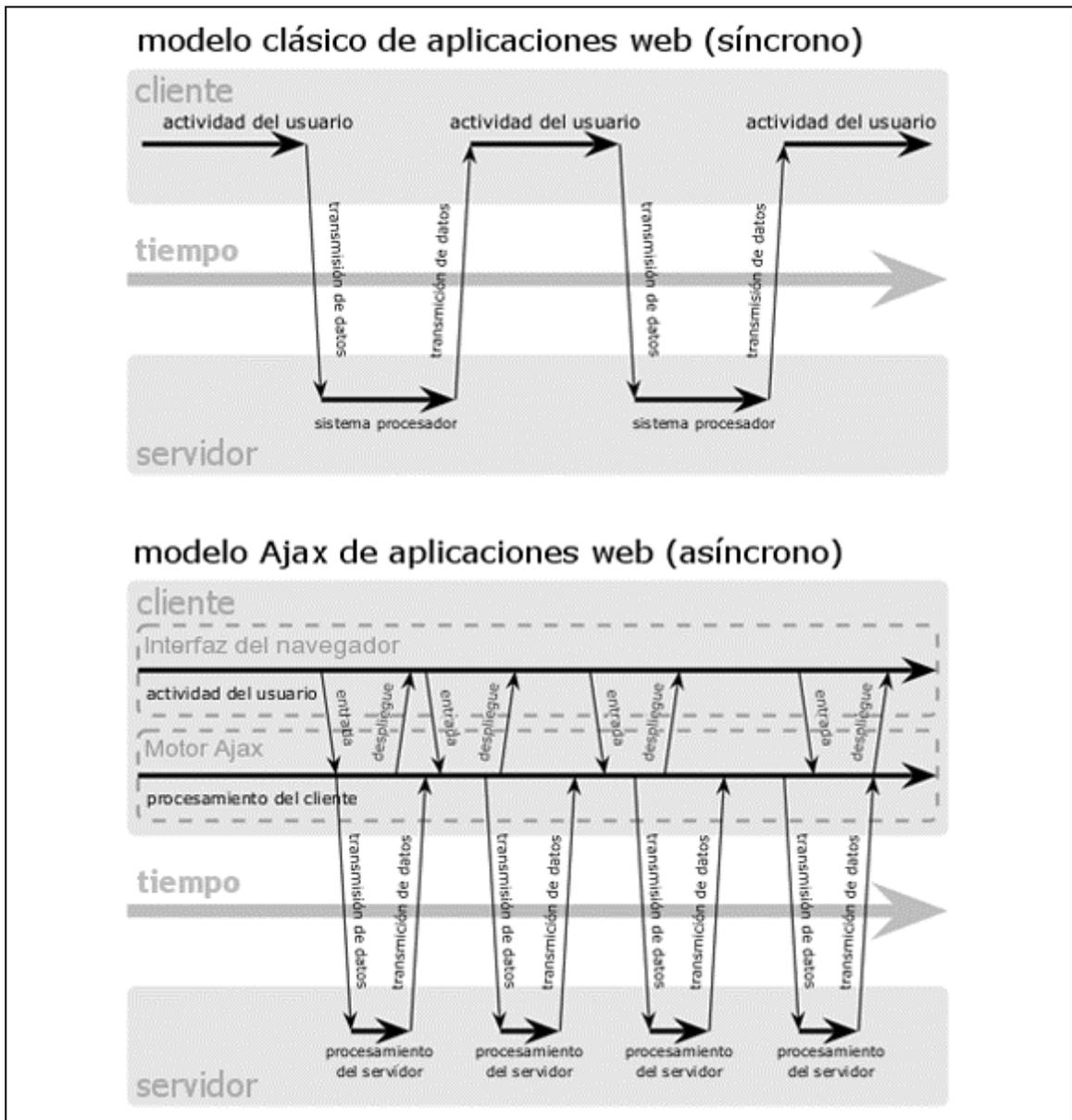


FIGURA 9. Comparación de la interacción producida en cada modelo.

Ahora ya no todas las acciones del usuario necesitan una llamada al servidor, por ejemplo una simple validación de información, editar datos en memoria, incluso navegación pueden ser hechas por rutinas JavaScript o alguna otra tecnología de manera local. En caso de necesitar información adicional del servidor se envían solicitudes XMLHttpRequest al servidor y estos se traen en algún formato, por lo general XML, sin necesidad de detener la interacción entre la interfaz y el usuario.

## 2.5. Tecnologías escogidas para el desarrollo del proyecto.

Ahora que ya se cuenta con una comprensión general de que es y como funciona AJAX, se introducirá al *framework* y tecnologías escogidas para el desarrollo del *software* CONGABOV.

Como se comento anteriormente las diferentes implementaciones del objeto XMLHttpRequest, producen que al momento de desarrollar una aplicación AJAX, se tenga que tener en cuenta este hecho y programar para que esta funcione en los distintos navegadores. Para evitar este inconveniente se han desarrollado una gran cantidad de *frameworks* que resuelven este problema y además incorporan *Widgets*<sup>39</sup> para agregar mayor funcionalidad disminuyendo considerablemente el tiempo de desarrollo.

Dentro de todo este gran espectro de *frameworks*, al cual pudimos dar una pequeña mirada al principio de este capítulo, se ha optado por la solución desarrollada por Google, llamada Google Web Toolkit, herramienta con la cual esta empresa ha desarrollado sus ya muy populares aplicaciones. A continuación se exploran sus características y se justifica su elección.

### **2.5.1. Framework Google Web Toolkit (GWT).**

Google Web Toolkit es un *framework* de *software* libre con el cual se pueden crear aplicaciones AJAX en lenguaje JAVA, permitiendo usar entornos integrados de desarrollo (IDE) como Eclipse, IntelliJ o NetBeans, los cuales proporcionan completas herramientas de depuración con inspecciones y puntos de interrupción variables, facilitando enormemente el desarrollo de aplicaciones y evitando tener que acometer la ardua tarea de aprendizaje de las peculiaridades del JavaScript y CSS en los navegadores.

Una vez depurado el código, GWT se encarga de compilar el código fuente y generar su equivalente en JavaScript y HTML, asegurando su compatibilidad con los navegadores Internet Explorer, Firefox, Mozilla, Safari y Opera.

#### **2.5.1.1. Arquitectura de Google Web Toolkit**

En la figura<sup>40</sup> 10 vemos los cuatro componentes principales de Google Web Toolkit.

---

<sup>39</sup> Un Widget, también conocido como artilugio o control, es un componente gráfico, con el cual el usuario interactúa, como por ejemplo, una ventana, un botón, una barra de tareas o una caja de texto.

<sup>40</sup> Obtenida del sitio oficial de GWT [URL17]

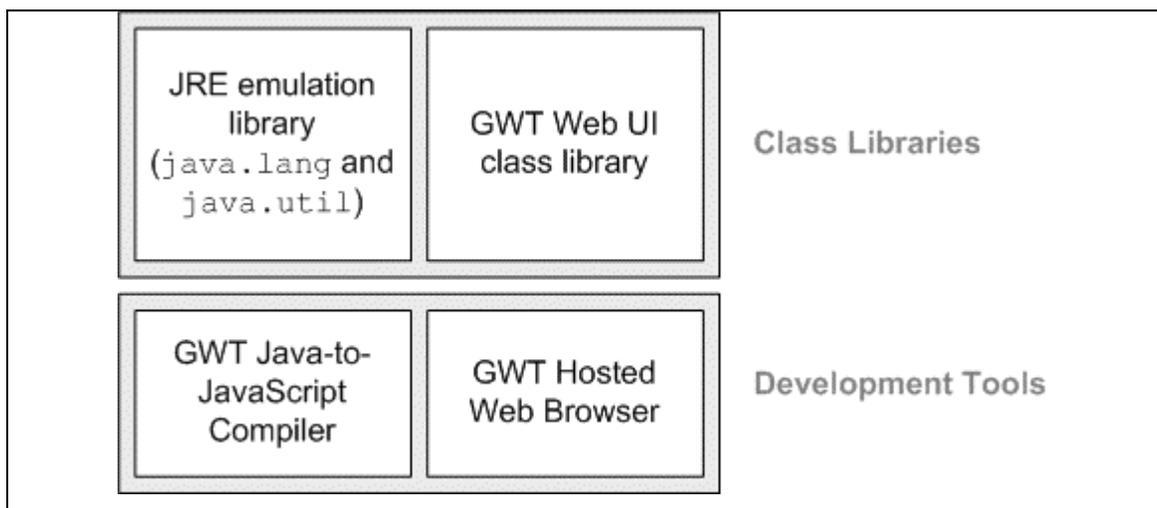


FIGURA 10. Arquitectura GWT

- **Compilador de Java a JavaScript:** Este traduce del lenguaje de programación JAVA a JavaScript, GWT produce descargas más pequeñas para el usuario final que las aplicaciones escritas a mano. Al contrario que el método de reutilización de bibliotecas JavaScript tradicionales, el compilador de GWT analiza el código fuente para incluir nada más que el código que va a necesitar.
- **Navegador Web alojado en GWT:** En el modo alojado, la aplicación se ejecuta como código de bytes dentro de la máquina virtual de Java (JVM). La mayor parte del tiempo de desarrollo se suele pasar en este modo, ya que la ejecución en el la máquina virtual permite aprovechar las ventajas de las funciones de depuración y permanecer dentro de un entorno integrado de desarrollo como Eclipse. Para conseguir esto, GWT inserta un control de navegador especial (Internet Explorer en Windows o un control de Gecko/Mozilla en Linux) que se conectará a la máquina virtual de Java.
- **Biblioteca de emulación de entorno de ejecución JAVA (JRE):** GWT contiene implementaciones JavaScript de las clases utilizadas con más frecuencia en la biblioteca de clases estándar de Java, incluida la mayoría de las clases de paquetes java.lang y un subconjunto de las clases de paquetes java.util. El resto de la biblioteca estándar de Java no se admite de forma nativa en GWT. Por ejemplo, paquetes como java.io no se aplican a las aplicaciones *Web*, ya que acceden a la red y al sistema local de archivos.
- **Biblioteca de clases de interfaz de usuario Web de GWT:** La biblioteca de clases de interfaz de usuario *Web* GWT es un conjunto de interfaces y clases personalizadas que permiten crear *Widgets* como botones, cuadros de texto, imágenes y texto en los navegadores *Web*. Ésta es la biblioteca de interfaz principal que se utiliza para crear aplicaciones GWT.

### 2.5.1.2. Funciones del Framework GWT.

A continuación se listan algunas de las funciones más importantes de GWT las cuales fundamentan la elección de este *framework* para el desarrollo de la aplicación CONGABOV.

- **Widgets dinámicos y reutilizables:** GWT cuenta con un repertorio básico de componentes gráficos como listas, cajas de texto, botones, etc. Se pueden crear *Widgets* mediante la composición de los proporcionados inicialmente por el *framework*, distribuirlos automáticamente en paneles e intercambiarlos con otros desarrolladores en un archivo JAR.
- **Interfaz nativa JavaScript (JSNI):** En caso de que la biblioteca base de GWT no cumpla con todas las necesidades requeridas por el desarrollador, este puede mezclar código JavaScript normal junto con el código JAVA, pudiéndose agregar funciones de otros *frameworks* hechos en JavaScript como por ejemplo Prototype, Script.aculo.us o manipular directamente con funciones JavaScript los elementos del navegador. Esta funcionalidad fue clave al optar por esta herramienta ya que permite incorporar otros componentes ampliando considerablemente las posibilidades en la creación de aplicaciones AJAX.
- **Administración del historial del navegador:** A diferencia de otros *frameworks* AJAX en los cuales para obtener esta funcionalidad se tiene que implementar *Frames* y otras técnicas complejas GWT trae funciones de fácil uso para implementar el historial del navegador.
- **Depuración del código:** GWT al permitir desarrollar las aplicaciones AJAX completamente en código JAVA y con el modo *Web* alojado usando la JVM probar las aplicaciones sin necesidad de convertir este código a JavaScript. Eso significa que cuando el código realice una acción como gestionar un evento de ratón, se obtendrá una depuración JAVA completa, con excepciones y funciones de depuración avanzadas de entornos integrados de desarrollo como Eclipse.
- **Compatibilidad con navegadores:** GWT libra al desarrollador de la tarea de hacer el código compatible con los navegadores mas usados del mercado, este *framework* genera código JavaScript compatible para Internet Explorer, Firefox y Safari. Cabe destacar que no presenta una compatibilidad completa con el

navegador Opera, ni tampoco con las versiones más antiguas de los navegadores antes nombrados.

- **Software completamente libre:** Todo el código de GWT está disponible bajo la licencia de Apache 2.0.

### 2.5.2. Entorno de desarrollo integrado Eclipse

Eclipse es una comunidad de código abierto, cuyos proyectos están enfocados a construir una plataforma de desarrollo, a la cual se le pueden agregar *frameworks*, herramientas y entornos de compilación en tiempo real para la construcción, depuración y administración de *software* a través de su ciclo de vida. [URL18]

Este entorno se ha escogido para el desarrollo de la aplicación CONGABOV por ser de código abierto y contar con una muy buena reputación a nivel de desarrolladores, además de ser el IDE recomendado por la gente de Google para usar el *framework* GWT. Las principales funcionalidades que posee se listan a continuación.

- Editor de texto
- Resaltado de sintaxis
- Compilación en tiempo real
- Control de versiones con CVS
- Asistentes (*wizards*): para creación de proyectos, clases, test, etc.

### 2.5.3. GWT Designer

Este es un *plugin* para Eclipse, el cual permite crear interfaces gráficas de usuario usando los *widets* de GWT. Mediante el uso de elementos gráficos y wizards, genera automáticamente el código JAVA, pudiéndose editar este posteriormente. [URL19] A continuación en la figura 11 se muestra este *plugin* instalado en Eclipse.

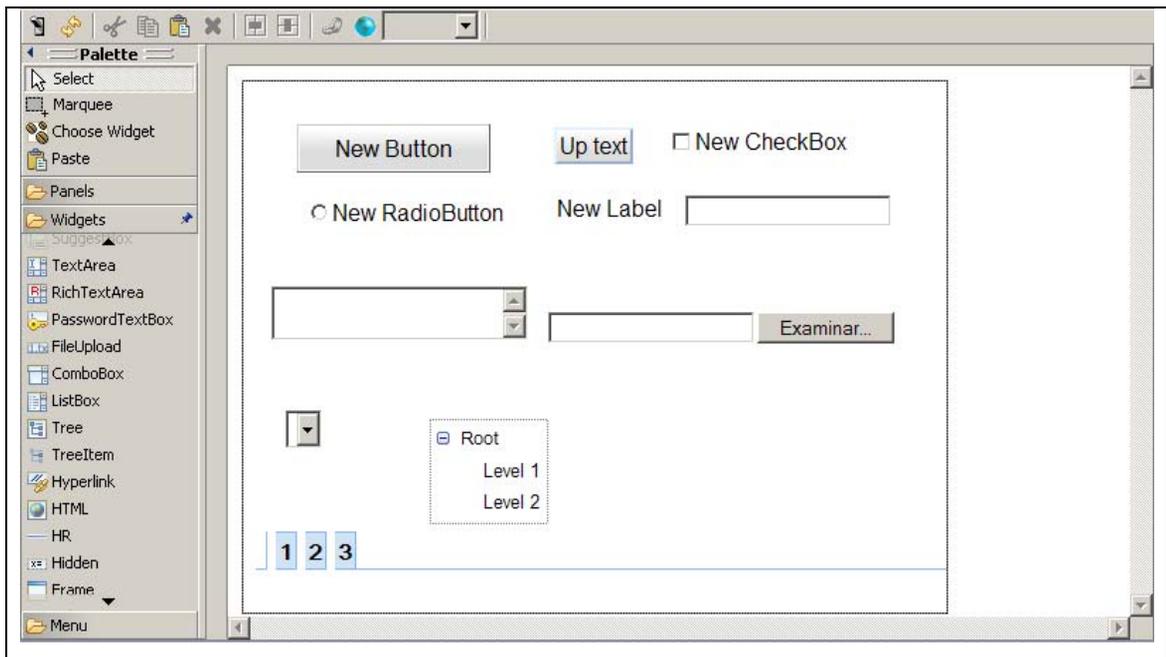


FIGURA 11. Plugin GWT Designer en Eclipse

Como se puede observar, este *plugin* nos entrega una paleta al lado izquierdo en la cual se presentan los *widgets* básicos con los que cuenta GWT y al lado derecho un área en la cual podemos ir insertando estos y comprobando inmediatamente como queda gráficamente la aplicación.

#### 2.5.4. Concurrent versions system (CVS)

CVS es un sistema de mantenimiento de código fuente (grupos de archivos en general), útil para grupos de desarrolladores que trabajan cooperativamente usando alguna clase de red.

Para ser más concreto, CVS permite a un grupo de desarrolladores trabajar y modificar concurrentemente ficheros organizados en proyectos. Esto significa que dos o más personas pueden modificar un mismo fichero sin que se pierdan los trabajos de ninguna.

CVS utiliza una arquitectura cliente-servidor: un servidor guarda la(s) versión(es) actual(es) del proyecto y su historia, y los clientes conectan al servidor para sacar una copia completa del proyecto, trabajar en esa copia y entonces ingresar sus cambios. El servidor normalmente utiliza un sistema operativo similar a Unix, mientras que los clientes CVS pueden funcionar en cualquier de los sistemas operativos más difundidos.

Los clientes pueden también comparar diferentes versiones de ficheros, solicitar una historia completa de los cambios, o sacar una "foto" histórica del proyecto tal como se encontraba en una fecha determinada o en un número de revisión determinado. [URL20]

### **2.5.5. Base de datos PostgreSQL**

PostgreSQL es un sistema de base de datos relacional orientada a objeto de código abierto, la cual tiene quince años de trayectoria. Funciona en la mayoría de los sistemas operativos, incluido Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), y Windows. Es compatible con ACID<sup>41</sup>, posee completo soporte para llaves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en múltiples lenguajes). Incluye la mayoría de los tipos definidos en SQL92 y SQL99, incluyendo INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de largos objetos binarios, incluyendo imágenes, sonido y video. Posee interfaces nativas de programación para C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, entre otros. [URL21]

Se usará esta base de datos para el desarrollo del software CONGABOV, debido a que todos los módulos del proyecto FONDEF D03I1151 almacenan los datos en esta.

### **2.5.6. pgAdmin**

Herramienta de administración de código abierto para la base de datos PostgreSQL. Incluye interfase gráfica de administración y una herramienta para consultas SQL entre otras prestaciones. [URL22]

Con esta herramienta se accederá al servidor donde se encuentra alojada la Base de datos y se crearan las tablas, procedimientos y todo lo que tiene que ver con la lógica del negocio de la aplicación CONGABOV.

### **2.5.7. Hypertext Preprocessor (PHP)**

---

<sup>41</sup> Conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

El servidor donde se encuentra alojado el proyecto FONDEF D03I1151 tiene instalados el servidor Apache y el lenguaje del lado del servidor PHP, el cual es un lenguaje de código abierto interpretado de alto nivel, especialmente pensado para desarrollos *Web* y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas *Web*, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. [URL23]

Este lenguaje se usará para crear los Web Services en el lado del servidor, para así generar consultas a la base de datos PostgreSQL y enviar la información a la aplicación AJAX para ser desplegada al usuario.

Cabe destacar que el resto de módulos del proyecto FONDEF D03I1151 están agrupados en un software llamado EGroupware, el cual es una herramienta para trabajo en grupo, que permite administrar contactos, enviar mensajes a los usuarios y más funcionalidades. Esta aplicación de software libre fue desarrollada en PHP, así que al momento de considerar la integración del módulo CONGA a esta plataforma, conviene escoger este lenguaje para facilitar el proceso de integración.

### 2.5.8. PHPEclipse

PHPEclipse es un *plugin* para Eclipse el cual facilita el desarrollo de programas en PHP en este *framework* agregando funcionalidades tales como: [URL24]

- **Características básicas:** Coloreado de sintaxis, concordancia de paréntesis, plegamiento de código, completado de código, sugerencia de parámetros, marca de ocurrencias, sugerencias de funciones, integración de manual PHP, plantillas de código.
- **Depuración:** Soporte para XDebug y DBG.
- **Control de versiones:** CVS y SVN.
- **Conectividad:** SSH/SFTP, FTPS.

### 2.5.9. GWT Widget Library

Esta es una librería de *widgets* para agregar mayor funcionalidad al *framework* GWT, cuenta con soporte para paginación, calendarios, calculadoras, etc. La idea es ahorrar el trabajo de desarrollar *widgets* más complejos y usar los de esta librería de código abierto.

Fue creada Robert Hanson's y además de proveer controles gráficos, cuenta con una librería para el lado del servidor, dando soporte a la creación de los RPC<sup>42</sup>. Para mayor información de esta librería revisar sitio *Web* del proyecto. [25]

## 2.6. Versiones de las herramientas utilizadas.

En la tabla 9 se presentan las versiones para las herramientas utilizadas en la realización del proyecto.

TABLA 9. Versiones de las herramientas.

Herramienta	Versión utilizada
Google Web Toolkit	1.3.3
Eclipse	3.2
GWT Designer	1.7
Concurrent versions system	Versión incorporada en Eclipse 3.2
PostgreSql	8.04
pgAdmin	1.6.2
PHP	5.1.4
PHPEclipse	1.1.X
GWT Widget Library	0.1.5

## 2.7. Diagrama de tecnologías usadas en el software CONGABOV

En la figura 12 se muestran como están estructuradas las tecnologías en el *software* CONGABOV, tanto al momento de su desarrollo como en su resultado final en producción.

<sup>42</sup> Remote Procedure Call, protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.

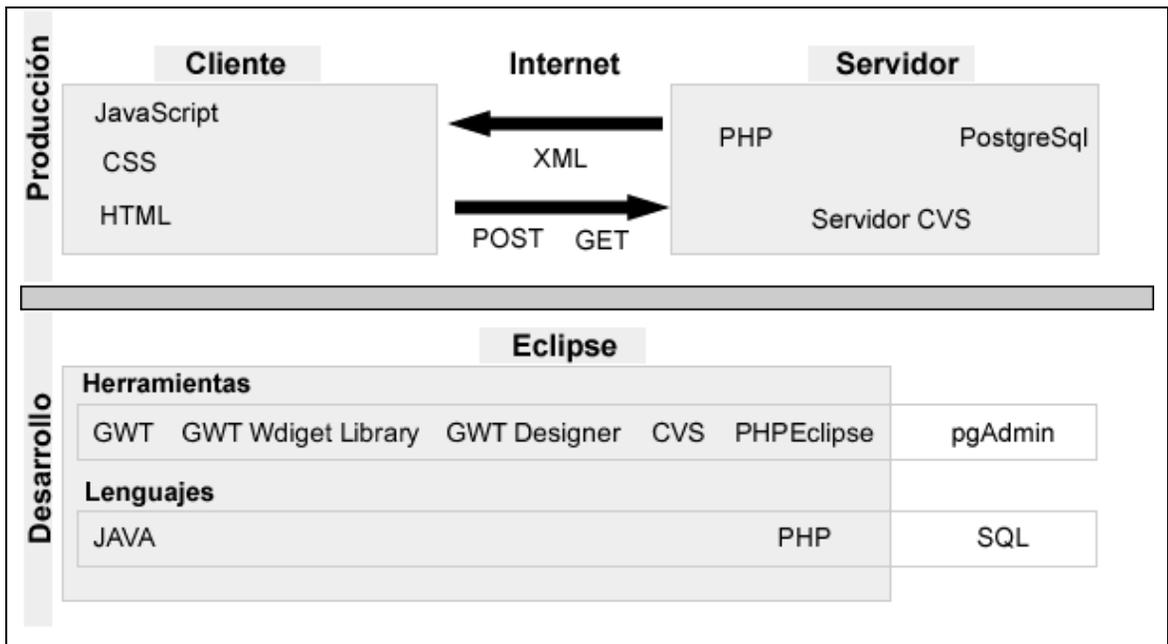


FIGURA 12. Diagrama tecnológico.

## **CAPITULO III**

### **3. SOFTWARE DE GESTIÓN PARA LA PRODUCCIÓN AGRÍCOLA Y BOVINA. (CONGABOV)**

En este capítulo se dejará de lado el aspecto técnico de las tecnologías vistas, para revisar lo desarrollado con ellas y así exponer las características, funcionalidades y base teórica del módulo CONGABOV a desarrollar para el proyecto FONDEF D03I1151.

Toda información para su desarrollo, ha sido extraída de las reuniones mantenidas con el grupo de trabajo de este proyecto y de la última versión de este software desarrollada por investigadores del Instituto de Economía Agraria de la Universidad Austral de Chile.

Primero, se hará una breve historia de este software, conceptos básicos de contabilidad, diseño de la aplicación, para luego proceder a exponer el sistema completo, módulo a módulo, según el orden de precedencia de estos.

#### **3.1.Historia del Software**

En los siguientes párrafos se revisa la historia de esta aplicación, obtenida del libro “CONGA, UN SISTEMA PARA LA PLANIFICACIÓN Y EL CONTROL DE GESTIÓN DE EMPRESAS AGRICOLAS”. Esto, para entender el contexto y las necesidades que impulsaron la creación de este software. [Ler07]

En 1978 la Universidad Austral de Chile, a través del Instituto de Economía Agraria, decide abordar seriamente el tema de la Gestión Empresarial Agrícola, recabando información sobre el desarrollo de esta área en diversos países del mundo y concluyendo que Francia era uno de los más avanzados.

En efecto, ya en 1966 el Consejo de Administración del Instituto de Gestión y de Economía Rural (I.G.E.R.) de ese país, creó una comisión la cual permitió establecer el “Plan Contable Agrícola 1967”, que fue sometido a la aprobación del Consejo Nacional

de la Contabilidad y que posteriormente sería implementado a nivel de la Comunidad Económica Europea (C.E.E).

La Comisión de la C.E.E. precisaba disponer de información fidedigna sobre la situación económica y sobre las rentas, en los distintos tipos de explotaciones agrícolas, no pudiéndose disponer de dicha información sin la existencia de registros contables. Por tal motivo, el Consejo de Ministros de la C.E.E. aprobó un reglamento, el que permitió crear una red de información contable agrícola sobre las rentas y la economía de las explotaciones agrícolas en el seno de la Comunidad Económica Europea y perfeccionar un sistema de planificación y control de gestión de dichas explotaciones, tomando como base el plan contable mencionado anteriormente.

Ante esta realidad, la Universidad Austral de Chile gestionó en 1979, un convenio de cooperación científica y tecnológica con el gobierno de Francia con el objetivo de adecuar a la realidad nacional tanto el plan de cuentas como el sistema de planificación y control de gestión ya mencionados, así como implementar un Centro de Gestión Agrícola en Valdivia.

Así nace la primera versión del sistema computacional CONGA (Contabilidad de Gestión Agrícola), diseñado para planificar, controlar y evaluar permanentemente el negocio agrícola a través de la contabilidad y como respuesta a la necesidad del empresario agrícola, de contar con un sistema que le permita obtener información útil y oportuna para la toma de decisiones. Dicha versión 1.0, escrita en lenguaje RPGII, comenzó a funcionar en la Universidad Austral en 1981, en un antiguo computador con tarjetas, el cual posteriormente fue reemplazado por otro más moderno, debiéndose reescribir el sistema en lenguaje COBOL, dando origen a la versión 2.0.

Paralelamente, en 1982, con la ayuda de especialistas franceses y con la participación de agricultores desde Los Ángeles a Osorno, se cumple el anhelo de crear en la Universidad Austral de Chile, un Centro de Gestión Agrícola, único en el país. En 1984, luego de un par de años de operación, a solicitud de agricultores usuarios del Centro y producto de la experiencia del trabajo en terreno, se crea la versión 3.0 del CONGA, que incorpora una serie de ajustes tendientes a simplificar su manejo y hacerlo más eficiente.

En 1988, producto del advenimiento de la microcomputación, los usuarios comienzan a adquirir sus propios equipos, siendo necesaria una nueva versión 4.0 del CONGA para microcomputadores, la cual incorporo nuevos ajustes, algunos demandados por los agricultores y otros por la nueva realidad, en que la gestión se haría en computadores personales y a nivel predial. Esta situación hizo no recomendable seguir financiando el funcionamiento del Centro de Gestión, manteniéndose el vínculo con los agricultores mediante la actualización permanente del sistema, a demanda de ellos y estudios específicos sobre la situación técnico – económica de sus empresas.

En 1994, producto del desarrollo de nuevas tecnologías de programación, más eficientes, se implementa la versión 5.0 del CONGA escrita en lenguaje Clipper , la que más tarde evolucionaría a la versión 6.0 que incorpora ajustes requeridos específicamente por la pequeña agricultura, tanto a nivel predial como a nivel de organizaciones de productores.

En el año 2004, se decide desarrollar una nueva versión 7.0 en ambiente Windows con el lenguaje de programación Visual Basic 6.0 y una base de datos ACCESS, con el principal objetivo de hacer el sistema más interactivo y fácil de utilizar. Esta nueva versión no cambia, en lo medular, la estructura que el sistema CONGA tenía en su versión anterior y que ha sido el resultado de más de 20 años de trabajo; solo incorpora ajustes provenientes de algunos Centros de Gestión Agrícola existentes a nivel nacional y que utilizan el sistema.

### **3.2. Conceptos básicos**

A continuación, se dará una breve explicación de los términos y conceptos básicos sobre la teoría contable usada en este software necesaria para comprender los siguientes capítulos de esta tesis.

- **Activo:** Todo objeto o bien natural que posee una persona natural o jurídica, tales como maquinarias, equipos, edificios, muebles, vehículos, materias primas, productos en proceso, herramientas, etc. [Sep95]
- **Pasivo:** Conjunto de deudas con terceras personas que tiene una empresa en un momento dado. [Sep95]

- **Débito y crédito:** El débito contable es una de las dos características de toda aplicación a los libros y siendo la otra el crédito. Un débito se refleja en el "debe" y son cuentas por naturaleza del activo contable y debe de estar siempre acompañado de un crédito reflejado en el "haber" que puede ser o en el pasivo o en el patrimonio. Existen dos tipos de naturalezas contables en las cuentas, las cuentas de naturaleza deudora (todas aquellas que van en el débito) y las cuentas de naturaleza acreedora (todas aquellas que van en el crédito). [URL27]
- **Predio:** es una propiedad inmueble que se compone de una porción delimitada de terreno.
- **Explotación agraria:** Esta es la unidad técnico-económica propia de la base del sector primario, equivalente a la empresa en otros sectores económicos, y cuya producción son los productos agropecuarios (agrícolas o ganaderos). [URL26] Para el software CONGABOV una explotación puede componerse de varios predios.
- **El plan contable general:** Es un conjunto de clasificaciones, nomenclaturas y reglas metodológicas de evaluación y operación que permiten recolectar uniforme y universalmente las informaciones de base, efectuar su recolección, transmisión, procesamiento y entrega de los resultados e informaciones para los fines micro y macroeconómicos. [CON78]
- **Plan de cuentas y actividades:** Documento que se caracteriza por reunir en un leguaje claro y sencillo los diversos elementos de una empresa; el sistema empleado permite mecanizar su uso a través del computador. El plan de actividades considera las actividades o rubros desarrollados normalmente en una explotación. El plan de cuentas es una lista de conceptos que permiten describir la situación de la empresa en un momento determinado. [Ler07]
- **Clase:** Agrupación de carácter económico y financiero destinada a ordenar homogéneamente los componentes del plan contable general. [CON78]
- **Ejercicio contable:** Período de tiempo en que se mide la actividad económica o financiera de una empresa. Los ejercicios contables comprenderán períodos de 1 año calendario, de acuerdo a las disposiciones técnicas y legislativas vigentes. [CON78]
- **Balance inicial:** Tiene por objeto mostrar, a la fecha de inicio de un ejercicio contable, la estructura patrimonial y financiera de la empresa. [Ler07]
- **Depreciación:** Distribución en el tiempo de los cargos por disminución de valor de las inmovilizaciones físicas, derivada de su utilización en el proceso

productivo, de las condiciones de explotación o en razón de adelantos tecnológicos. [CON78]

- **Revalorización:** Ajuste en el valor de un instrumento financiero para mantenerlo sin cambios reales respecto a una variable (EJ. IPC). [URL28]
- **Liquidez:** Atributo de un activo financiero de ser comprado o vendido en el mercado sin provocar cambios considerables en el precio. Un activo es considerado líquido cuando es fácilmente convertible en efectivo. [URL28]

### 3.3. Metodología usada para el desarrollo del software

La versión de escritorio del CONGABOV, entrega información sobre las funcionalidades y una clara división en módulos de la aplicación a desarrollar, reduciéndose con esto considerablemente el tiempo en la toma de requerimientos y modularización. Cabe destacar que la nueva versión en Internet tiene que ser multiusuario y agregar nuevas mejoras en la interfaz gráfica como por ejemplo ordenamiento por columnas, paginación, listas que filtren opciones, etc.

Se ha decidido usar el diseño estructurado como metodología para desarrollar este *software*. En esta se busca dividir el programa en módulos, siguiendo los principios de diseño de descomposición por refinamientos sucesivos, creación de una jerarquía modular y elaboración de módulos independientes.

A continuación se listan las etapas en las que se dividió el desarrollo.

- **Estudio de la versión de escritorio del software CONGABOV:** En esta etapa se recopilaron los requerimientos de la aplicación obtenidos a partir del *software* recién nombrado y reuniones con los integrantes del proyecto FONDEF D03I1151.
- **Estudio, comparación y elección de tecnologías:** En el capítulo anterior se trata este punto con mayor detalle pero a modo de resumen las tecnologías escogidas fueron la técnica de programación AJAX, el *framework* GWT y el lenguaje XML para el intercambio de datos. El resto de tecnologías son las que residen en el servidor del proyecto y con las que se desarrollaron el resto de módulos justificando su uso por este motivo.

- **Estudio del marco teórico:** Se profundizaron conceptos tales como explotación, ejercicio contable, plan de cuentas y actividades, presupuestos, movimientos contables, balance inicial, etc.
- **Modularización del software:** En la tabla 10 se observan los primeros niveles de la modularización obtenida a partir del estudio de la versión de escritorio del *software* CONGABOV.

TABLA 10. Módulos del sistema

Módulo	Descripción
Administración del sistema	Creación de usuarios, explotaciones, predios, categorías de cuentas y actividades, tipos de movimientos, etc. En general todo lo referente a administración del software a nivel administradores del sistema.
Administración parámetros de usuario	Administración de los datos de la explotación, proveedores, clientes, cuentas y actividades. Todo lo concerniente con los parámetros del usuario.
Presupuestos	Planificación del ejercicio contable el cual esta subdividido en el presupuesto por actividades y el presupuesto de caja.
Informes del presupuesto	Varios informes obtenidos de los datos ingresados en los presupuestos.
Balance inicial	Inventario de los activos y pasivos de la explotación.
Movimientos contables	Módulo en el que se ingresa la contabilidad diaria.
Libros diarios	Informes obtenidos de los datos ingresados en la contabilidad.
Informes de gestión	Generados a partir de los datos del presupuesto y contabilidad. Estos indican si es que lo presupuestado se esta cumpliendo.
Indicadores financieros	Resultado, productividad, liquidez, solvencia, rentabilidad.
Generación de gráficos	Todos los informes tienen una representación gráfica la cual facilita su comprensión. Este módulo es un conjunto de clases encargado de generar estos gráficos a partir de los informes.
Estimación de costo litro de leche	A partir de la contabilidad se obtiene el costo de producción de cada litro de leche.
Cierre anual	Módulo que al término del año contable calcula los resultados finales y balance inicial para el próximo ejercicio preparando el sistema para comenzar todo nuevamente.

- **Creación de documentos de diseño:** Una vez obtenidos los módulos del sistema y las funcionalidades se crean los modelos de datos, arquitectura del sistema, diagrama de interfaces de usuario, descripción procedimientos almacenados, etc.
- **Desarrollo del sistema:** Se definen funciones globales del sistema como solicitudes HTTP al servidor, creación de grillas paginadas, calendarios para llenar los campos tipo fecha, etc. A continuación a los módulos antes nombrados en la tabla 10 se les aplica descomposición por refinamientos sucesivos. El orden de desarrollo fue el mismo que se presenta en la tabla 10. Esta etapa fue la que tomo mas tiempo en completarse dividiéndose en muchas subetapas más.
- **Poblado de la base de datos:** Esta aplicación para funcionar requiere que el plan de cuentas y actividades estén previamente cargados en el sistema, así como también los tipos de impuestos y sus correspondientes valores, precio del IPC y

tipos de movimientos contables. Todo esto fue ingresado con el módulo de administración del sistema.

- **Evaluación de escalabilidad:** Para asegurar la escalabilidad del sistema a medida que se vayan agregando usuarios se crearon explotaciones con ejercicios contables y datos generados de forma aleatoria. La idea de esto es asegurar un mínimo de usuarios y explotaciones soportados. Esta prueba consto de 1000 explotaciones con un ejercicio contable cada una. Se optimizaron los procedimientos para tener un rendimiento aceptable. La idea es que cada informe demore menos de 1 segundo en generarse.
- **Ingreso de predio con un ejercicio contable modelo:** Esta etapa consistió en tomar datos reales de una explotación e ingresarlos al sistema para luego verificar los resultados comparando la antigua versión de escritorio con la nueva versión *Web*.
- **Validación de los resultados entregados por los informes:** Usando los datos reales ingresados en el punto anterior se comparan los informes creados en la nueva versión *online* con los informes de la versión de escritorio verificando todas las diferencias y analizando si se deben a fallas en la digitación de los datos o a los cálculos de los procedimientos.
- **Corrección de informes:** Con la información del punto anterior se procede a corregir los procedimientos en los cuales se encontraron defectos en los cálculos.
- **Correcciones a nivel de pantallas:** Se hacen arreglos y mejoras en la forma de ingresar los datos, la manera en que estos se muestran, se miden tiempos de carga, se incorporan todas las validaciones necesarias para que no se provoquen errores y todos lo arreglos a nivel de interfaz de usuario que se necesiten para hacer la aplicación agradable y fácil de usar para el público objetivo.
- **Correcciones generales:** En esta última etapa se verifican constantemente las pantallas y cálculos del sistema, realizando correcciones y mejoras hasta llegar a un punto de estabilidad del software y satisfacción por parte de los encargados del proyecto FONDEF D03I1151.

### 3.4. Equipo del proyecto

En la tabla 11 se ve el equipo de trabajo del proyecto, resaltados están los nombres de las personas que tuvieron que ver directamente con la creación del módulo CONGABOV siendo Juan Lerdón el investigador responsable de este *software*. Con

estas personas se llevaron a cabo las reuniones de toma de requerimientos y revisiones de la aplicación a medida que se terminaban sus funcionalidades.

TABLA 11. Equipo de trabajo

<b>Equipo de investigación</b>	<b>Equipo de gestión y transferencia tecnológica</b>
<b>Juan Lerdon F., Ing. Agr., D.E.A., Dr. (Director)</b>	Marcela Amtmann R., Méd. Vet. M.Sc.
René Anrique G., Ing.Agr. Mg.Sc. Ph.D. (Director alterno)	Maruja Cortés B., Ing.Agr. M.Sc.
Bernardo Carrillo L., Ing .Agr. M.Sc.	Mª Angélica Carrasco G., Ing.Agr. M.Sc.(c)
Dante Pinochet T., Ing. Agr. M.Sc. Ph.D.	Mireya Briones N. Ing. Agr. Mg.Sc.
Dagoberto Villarroel, Ing. Agr.	Daniela Durán I. Ing. Agr.
Eliana Scheihing G., Ing. Civil Mat.	<b>Cristian Muñoz B. Ing. Agr</b>
Humberto Navarro D., Ing. Agr. M.Sc.	
<b>Juan Pablo Salazar F., Ing. Civil Inf., M.B.A.</b>	
Laura Nahuelhual M., Ing. Agr. M.Sc. Ph.D.	
Oscar Balocchi L., Ing. Agr. M.Sc. Ph.D.	
<b>Patricia Rosales P., Ing. Civil Inf.</b>	
Víctor Moreira L., Ing. Agr. M.Sc. Ph.D.	

Para el desarrollo del software se contó con dos programadores; Cristian Rojas Pérez y Manuel Pérez Escalante, siendo el primero de estos el encargado del diseño y gestión informática.

Cabe destacar la labor de Leandro Ahumada y Jorge Vera, quienes hicieron el poblado de la base de datos y creación de la explotación con la cual se probó el *software* ayudando también en la validación de informes y pantallas de la interfaz de usuario.

### **3.5.Análisis de la versión de escritorio del software CONGABOV**

Como se menciono anteriormente este software esta desarrollado usando un motor de base de datos ACCESS de Microsoft. En la figura 13 se muestra el modelo de datos obtenido, usando ingeniería inversa de la herramienta VISIO.

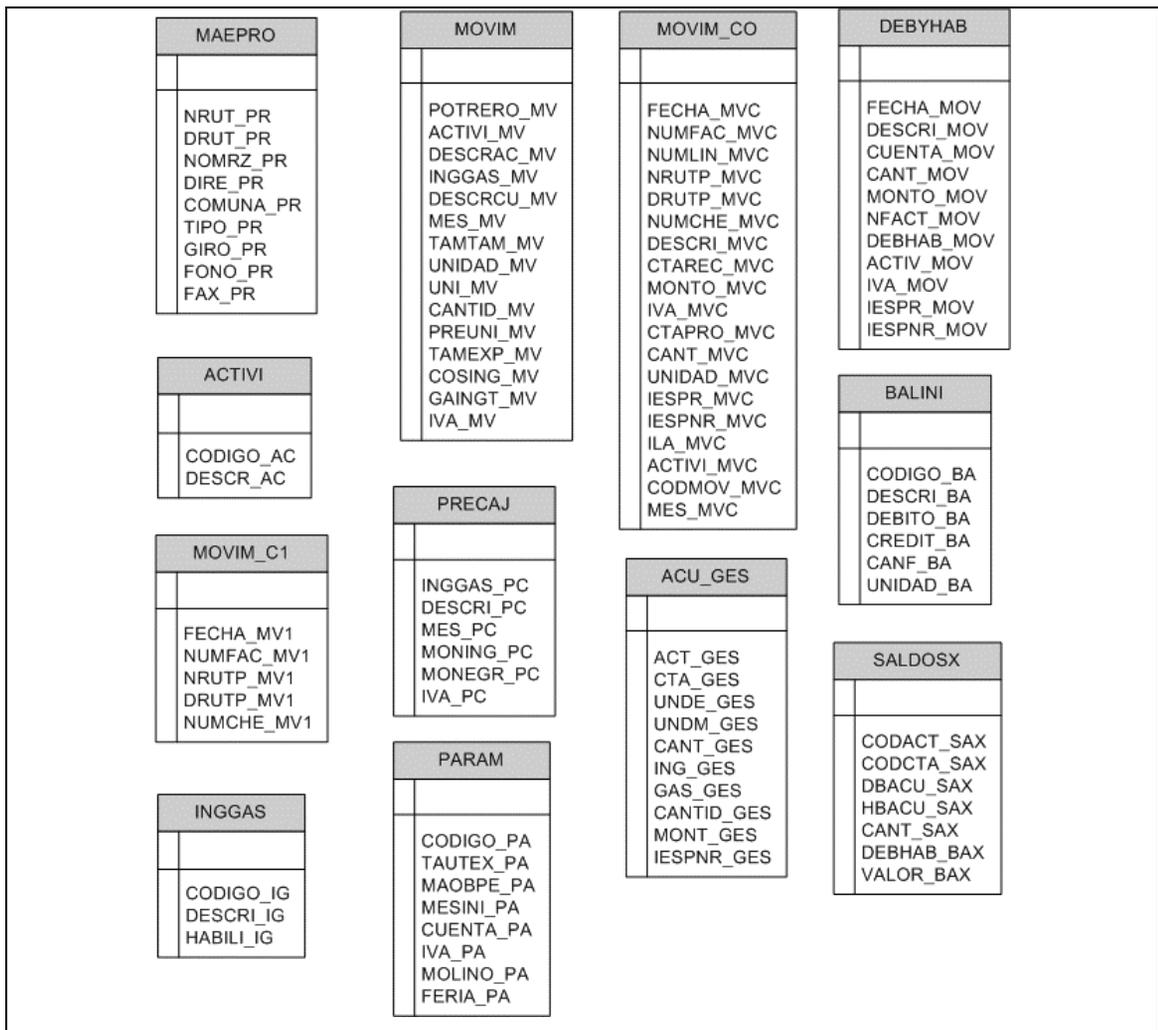


FIGURA 13. Modelo de datos versión de escritorio

La última versión de CONGABOV fue desarrollada el año 2004 y al ver el modelo de datos se puede observar claramente que no posee integridad referencial ni cumple con las normas de normalización estándares.

Para el desarrollo de la nueva versión de este *software* se optó por el motor de base de datos PostgreSQL que cumple con los requisitos para brindar este servicio *online*.

Se ha optado por confeccionar nuevamente el modelo de datos, ya que este tiene que cumplir con la integridad referencial y un cierto grado de normalización. Además se deberán agregar más tablas para poder soportar múltiples usuarios, guardar el historial de los ejercicios e incorporar nuevas funcionalidades creciendo en gran medida el nuevo modelo de datos.

El lenguaje de programación usado para la antigua versión de escritorio fue Visual Basic 6.0. Al analizar los códigos se puede observar que toda la lógica de los informes

esta programada en las pantallas. El problema de esto es que se mezclan la lógica de la interfaz de usuario junto con la lógica del negocio, haciendo más difícil la mantención del código.

En el CONGABOV *online* se ha optado por separar estas lógicas dejando todo lo que tiene que ver con la contabilidad y generación de informes en la base de datos en procedimientos almacenados así los fuentes en JavaScript que tenga que descargar el cliente en su computadora solo contendrán la lógica de usuario siendo estos archivos de menor tamaño. Esto también provoca que el código sea más fácil de mantener y migrar a otras plataformas si es que se conserva la base de datos, ya que solo se tiene que cambiar la interfaz de usuario.

Por los motivos antes nombrados se ha decidido usar la antigua versión de escritorio solo para ver el funcionamiento de las pantallas y de vez en cuando revisar el código y ver la forma en que se calculan los informes.

### 3.6. Diseño del software

Como se menciona anteriormente al desarrollar la versión *online* de esta aplicación se han separado las lógicas en las distintas capas del modelo, dejando la lógica de interfaz de usuario en el navegador y todo lo referente a los cálculos y generación de informes de contabilidad en el servidor. Se puede observar esta distribución en la figura 14.

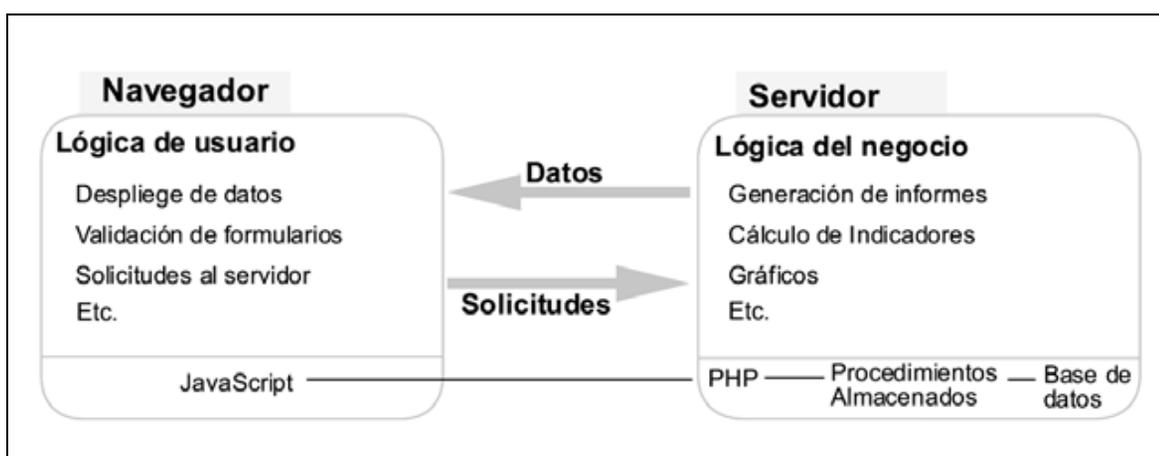


FIGURA 14. Arquitectura lógica

Aquí se puede observar que las funciones en JavaScript solo se comunican con el servidor mediante los servicios *Web* creados en PHP y que estos solo acceden a los datos

a través de procedimientos almacenados, los cuales son los únicos componentes que tienen contacto directo con la base de datos. En algunos casos, para no agregar complejidad al sistema, se hicieron consultas directamente de los archivos en PHP, pero estas solo son para visualizar la información.

En la figura 15 se aprecia el modelo de datos simplificado en el cual solo aparecen los nombres de las tablas y entidades más importantes de esta aplicación.

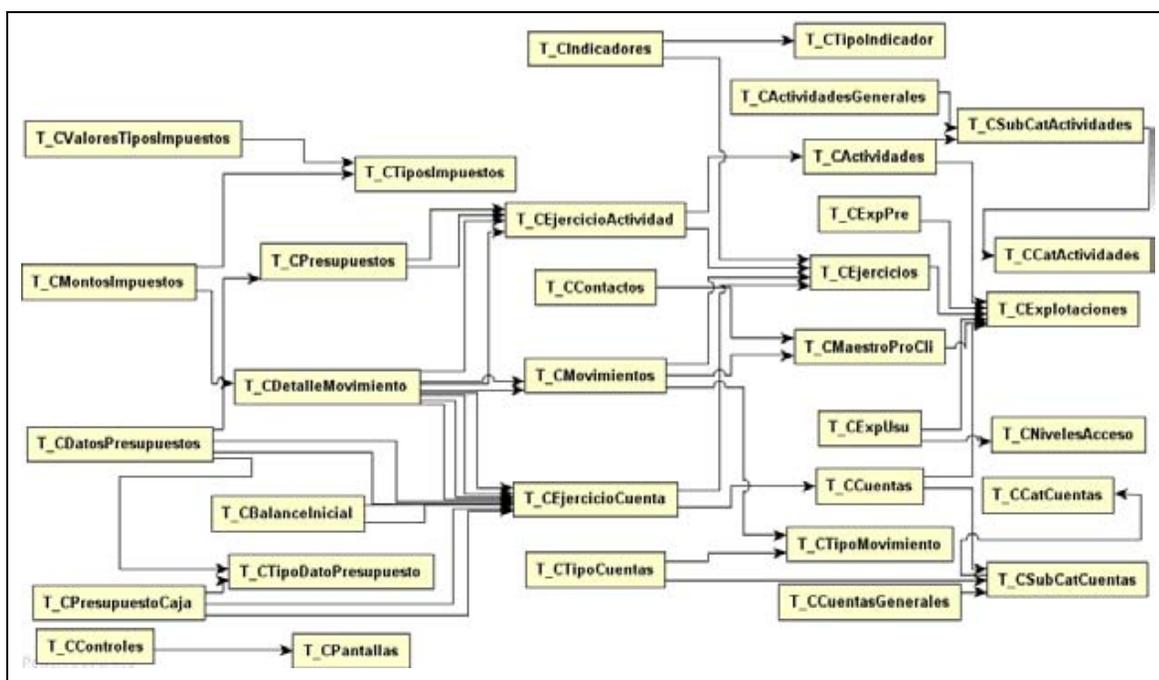


FIGURA 15. Modelo de datos

Las tablas más críticas en este modelo son T\_CDetalleMovimiento y T\_CMovimientos, siendo estas donde se guarda la información de contabilidad y las que más crecen a medida que se usa la aplicación.

En la figura 16 se muestra la estructura de pantallas en las que se han distribuido las funcionalidades del CONGABOV. Solo se han considerado las pantallas que ven los clientes que usen la aplicación, excluyendo el módulo de administración del sistema.

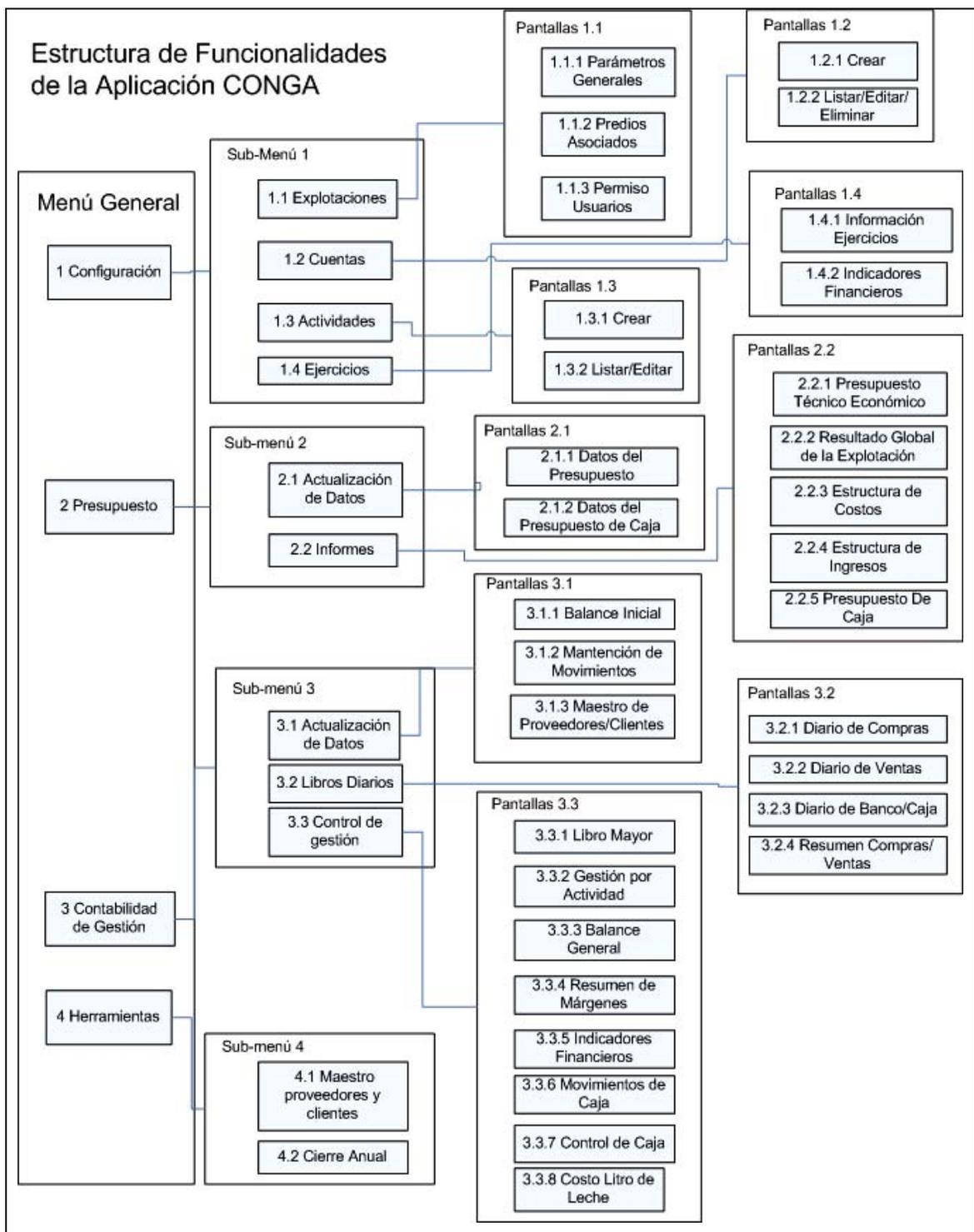


FIGURA 16. Estructura pantallas

Al comenzar a usar la aplicación, el usuario debe partir en el módulo de configuración, ingresando los datos de su explotación, preparando sus listas de cuentas y actividades, agregando sus proveedores y clientes y creando su inventario en el balance inicial. A continuación, crear los presupuestos y revisar los informes de este módulo para así constatar que las actividades que efectuará en el año estarán bien financiadas y rendirán lo esperado. Durante el año contable, se irán ingresando periódicamente los gastos e ingresos en el módulo de contabilidad, pudiéndose comprobar en los informes de gestión



TABLA 12. Etapas de la gestión contable durante un ejercicio.

<b>Etapa 1</b>	<b>Etapa 2</b>	<b>Etapa 3</b>
Inicio del periodo contable.	Durante el ejercicio.	Al cierre del ejercicio.
-Inventario y balance. -Elaboración de presupuesto. -Plan contable de la empresa.	-Registro de las operaciones contables diarias. -Emisión y análisis de la información.	-Nuevos inventarios. -Ajuste del ejercicio contable anterior. -Nuevos presupuestos para el próximo ejercicio.

No se verán los demás documentos de diseño como las interfaces de usuario, diagrama de clases, etc., ya que considerando que se crearon 82 archivos fuente de JAVA, 167 servicios *Web* en PHP, 364 procedimientos almacenados, 46 tipos de datos en PostgreSQL, 8 vistas y 52 tablas, entre otras cosas, estos documentos serian demasiado extensos.

### **3.7. Configuración de la aplicación.**

Para que CONGABOV funcione correctamente debe estar previamente configurado, esto implica ingresar los parámetros de la explotación, actividades, cuentas, proveedores y clientes.

#### **3.7.1. Explotación**

Una explotación agropecuaria, concepto explicado anteriormente, consta de los siguientes parámetros.

- **Superficie agrícola útil:** Esta es la superficie en los predios de los que consta la explotación en la cual se están desarrollando actividades agropecuarias. Esta se mide en hectáreas y es necesaria para calcular indicadores financieros e informes.
- **Mano de obra permanente:** Cantidad de personas que trabajan en la explotación, necesaria para hacer cálculos en los informes.
- **Mes inicio de ejercicio:** El usuario puede seleccionar cualquier mes y el sistema automáticamente calcula la fecha de inicio y término del ejercicio, dando una duración exacta de un año contable. Estos valores se usan de manera interna para asegurar que no se ingresen datos fuera de estas fechas.

El software CONGABOV es capaz de gestionar N explotaciones por usuario, limitándose este número solo por conceptos de contrato mediante el administrador del sistema.

En la figura 18 se observan las pantallas donde se despliegan y modifican estos valores.

FIGURA 18. Explotación

### 3.7.2. Plan de cuentas

Este esta estructurado en clases numeradas de 0 a 9 y cada clase conforma subclases numeradas también de 0 a 9, las que a su vez, se dividen en cuentas principales y en cuentas divisionarias, cada una de ellas numeradas de 0 a 9. Así por ejemplo, el orden sucesivo para la clase 6 se puede ver en la tabla 13.

TABLA 13. Estructura de cuentas para la clase 6

Subclases	Cuentas principales	Cuentas divisorias
60 de la clase 6;	620 dentro de la subclase 62;	6220 dentro de la cuenta principal 622;
61 de la clase 6;	621 dentro de la subclase 62;	6221 dentro de la cuenta principal 622;
62 de la clase 6;etc.	622 dentro de la subclase 62; etc.	6222 dentro de la cuenta principal 622; etc.

El carácter divisorio decimal de la clasificación, permite continuar la apertura hacia la derecha, de acuerdo con las necesidades de la implementación. Las clases están definidas de manera tal que permiten distinguir claramente la estructura financiera y económica de una empresa. [CON78]

La implementación de este plan contable en el sistema CONGABOV esta dividida en clases de cuentas identificándose cada una de ellas por un código de 6 dígitos

correspondiendo las 5 primeras a cuentas de balance, las dos siguientes a cuentas de gestión y la última a cuentas de pérdidas y ganancias. Esto se aprecia en la tabla<sup>44</sup> 14.

TABLA 14. Estructura del código de cuentas

Código	Nombre	Tipo de cuenta
100000 200000 300000 400000 500000	Capitales Permanentes. Activo Inmovilizado. Cuentas de Stock. Cuentas a Terceros. Cuentas Financieras.	Cuentas de balance.
600000 700000	Gastos por Naturaleza. Ingresos por Naturaleza.	Cuentas de gestión.
800000	Cuentas de Resultado.	Pérdidas y ganancias.

El objetivo del plan de cuentas es poder describir el estado de una empresa en un momento determinado.

Este se basa en el plan contable general de la comunidad económica europea y ha sido diseñado específicamente para empresas agrícolas y establecido a partir de los actos esenciales de la vida de una empresa. [Ler07]

Para entender este plan de cuentas procedemos a describir cada una de sus clases especificando la función de cada una de ellas.

- **Capitales permanentes:** Esta clase muestra la composición de los recursos financieros, de carácter permanente o duradero, con los que cuenta una empresa. Comprende la participación de los propietarios o socios, las reservas, los créditos de terceros exigibles y provisiones a más de un año de plazo. (100000:capital propio, 100910:cuenta socio-1) [CON78]
- **Activo inmovilizado:** Muestra el conjunto de bienes y derechos, tangibles e intangibles, destinados a un uso duradero y a permanecer en la empresa, básicamente como medios de producción. (210000: suelo agrícola, 219100: vacas crianza) [CON78]
- **Cuentas de Stock:** Bienes materiales de la empresa, no incorporados a sus inmovilizaciones, que serán objeto de transformación, cuando se trata de una empresa industrial o agrícola, o de venta cuando se trata de una empresa comercial. (310100: Stock salitre sódico, 313100: Concentrados) [CON78]

<sup>44</sup> Tabla obtenida de la guía práctica “Sistema para la planificación y el control de gestión agrícola, CONGA”.

- **Cuentas de terceros:** Describe las relaciones de la empresa con otros agentes económicos. Refleja las deudas o créditos con terceros - clientes, proveedores, Estado, socios, personal u otros – originados por las operaciones corrientes, tales como compras o ventas de bienes y servicios realizadas a menos de un año. (410100: Feria ganadera, 412100: Colun) [CON78]
- **Cuentas financieras:** Muestra el conjunto de valores y de transacciones financieras del ejercicio (menos de un año), utilizados como capital de operación y de desarrollo de las actividades de la empresa. (500100: Crédito BCI, 570000: Caja) [CON78]
- **Gastos por naturaleza:** Muestra los gastos del ejercicio, clasificados por su naturaleza económica, lo que facilita el desarrollo de métodos de control interno de gestión empresarial. (601050: Superfosfato triple, 630110: Arriendo maquinaria suelos) [CON78]
- **Ingresos por naturaleza:** Muestra los ingresos del ejercicio, clasificados genéricamente por la naturaleza económica de las operaciones que las originan. (702210: Venta Papas, 708430: Venta de vaquillas)[CON78]
- **Cuentas de resultado:** Esta clase esta destinada a mostrar en forma resumida y global, los resultados de gestión del ejercicio financiero de la empresa. (870100: Resultados del Ejercicio) [CON78]

Cada una de estas clases principales se divide en subclases dependiendo del nivel de detalle que se requiera. No se analizaran todas estas debido a que no es objetivo de este documento exponer el plan de cuentas completo si no más bien explicar como funciona este. Para que el lector tenga una idea de cuan extenso es el plan de cuentas este consta de los 8 niveles superiores recién nombrados, 112 subniveles y un total de 461 cuentas en el plan general pudiéndose extender este número dependiendo de las necesidades de cada usuario.

A nivel de sistema este plan cuenta con tres niveles siendo los dos primeros manipulados solamente por el administrador del *software* y solo el nivel mas bajo puede ser modificado por los usuarios. Esto debido a que se necesita mantener la estructura general de este plan para así poder calcular indicadores financieros y generar informes contables.

Al crearse una explotación esta comienza con un conjunto de cuentas por defecto ajustado a la realidad de las explotaciones agropecuarias de la región. La figura 19 muestra la pantalla de administración que tienen los usuarios para el manejo de las cuentas.

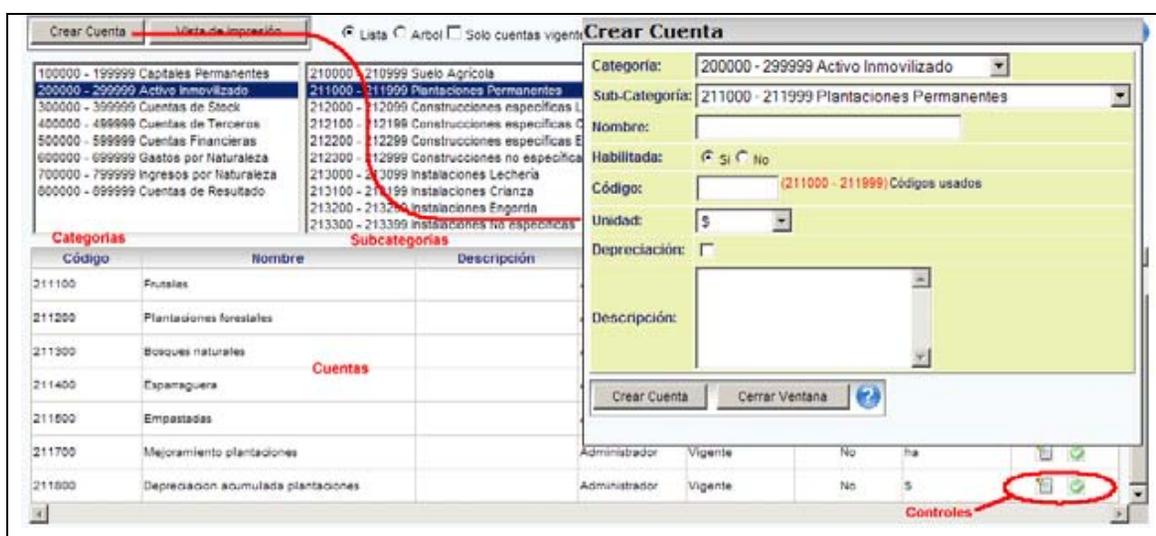


FIGURA 19. Administración de cuentas

En esta nueva versión fue necesario agregar un parámetro para fijar la unidad en la que se contabiliza la cuenta, asegurando así la constancia de esta durante el ejercicio contable permitiendo la correcta generación de los informes. También se incorporó un campo a las cuentas del activo inmovilizado indicando el índice de depreciación lineal de los activos para poder calcular su depreciación al término del ejercicio.

La tabla 15 muestra un grupo de cuentas del activo inmovilizado en el que se aprecian todos los campos.

TABLA 15. Cuentas del activo inmovilizado

Código	Nombre	Unidad	Depreciación (años)
211100	Plantaciones frutales	ha	20
212000	Construcciones Lechería	\$	30
217100	Vacas en producción de leche	Cabezas	no

El plan de cuentas es el pilar fundamental de esta aplicación, en el residen todos los conceptos necesarios para poder efectuar correctamente la contabilidad y la gestión, por este motivo a medida que se avance en este capítulo se seguirá recurriendo a el constantemente.

### 3.7.3. Plan de actividades

El funcionamiento de este es igual al del plan de cuentas con la diferencia de que describe las actividades desarrolladas en una explotación, cada una de las cuales se representa con un código de 4 dígitos.

Este consta de 3 niveles y en el primero encontramos los gastos e ingresos generales, producción vegetal y producción animal, englobando de esta manera las actividades comúnmente desarrolladas en una explotación agropecuaria. En la tabla 16 se ve un ejemplo de estas y a continuación en la figura 20 sus pantallas de administración.

TABLA 16. Actividades grupo bovinos de leche

código	Nombre
8420	Lechería
8430	Vaquillas
4845	Terneros

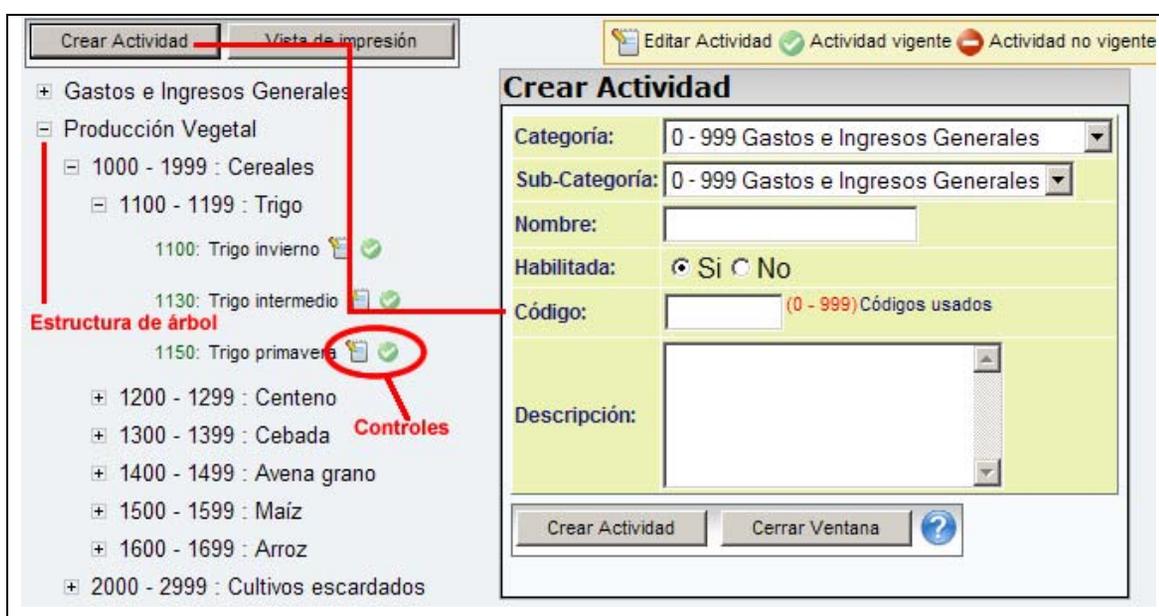


FIGURA 20. Actividades

Al igual que con las cuentas, las explotaciones al crearse vienen con un plan de actividades pre cargado al cual el usuario puede agregar o modificar valores dependiendo de la realidad de su negocio.

### 3.7.4. Proveedores y clientes

Los proveedores son terceros que abastecen a la empresa de los bienes o servicios necesarios para su operación y los clientes son a los que la empresa vende los bienes y servicios que produce. Los primeros se encuentran dentro del rango de cuentas 400000 – 409999 y los segundos en el 410000 – 419999 existiendo además una clase de cuentas llamadas proveedores y clientes recíprocos los cuales son entidades que actúan de las dos formas dependiendo de la situación y se encuentran en la clase 412000 – 419999. En la figura 21 se muestra la pantalla de administración apreciándose los parámetros de estas entidades.

	RUT	Razón Social
8	97006000 - 6	Bci
9	79891680 - 7	Carlos Angulo
10	79891680 - 7	Carlos Angulo
11	03576613 - 8	Casa del Perno
12	08170618 - 2	Clinica movil
13	81094100 - 6	Colun
14	82392800 - 6	Cooprinsem
15	89528800 - 4	Copec
16	98545490 - k	Cto
17	96714340 - 5	Dass computacion
18	00000000 - 7	Empresad

**Crear Entidad**

RUT: [ ] - [ ]

Razón Social: [ ]

Comuna: Aisén

Dirección: [ ]

Giro: [ ]

Relación Prioritaria: Proveedor y Cliente

Dirección de Despacho: [ ]

Vigente:

Contacto: [ ]

Crear Cerrar Ventana ?

**Controles**

FIGURA 21. Proveedores y clientes

El sistema crea automáticamente una cuenta en el plan contable al crearse un proveedor o cliente. Estas se usan al ingresar los movimientos contables en el módulo de contabilidad.

### 3.8.Presupuestos

Existen dos tipos de presupuestos en el sistema, estos son el presupuesto por actividad y el presupuesto de caja, de estas dos fuentes de datos ingresados por el usuario el sistema genera una serie de informes técnicos y económicos que permitirán presupuestar y simular el resultado de cada rubro y del predio en su conjunto.

### 3.8.1. Presupuesto por actividades

El presupuesto o plan de explotación sirve para analizar si la empresa continuará con los rubros o actividades desarrolladas hasta el momento o si algunas deberán cambiarse o modificarse en su proceso. Estos comprenden una lista detallada de gastos e ingresos que el empresario espera tener durante el ejercicio, datos obtenidos a partir de la experiencia que ha tenido en años anteriores o estudios de mercado realizados para evaluar la factibilidad de una actividad. Después del primer ejercicio el sistema mantiene los presupuestos creados por el usuario para que este no tenga que volver a crearlos desde cero, sino simplemente hacer las modificaciones que crea pertinente. A continuación en la figura 22 se muestra la pantalla donde se visualizan los presupuestos.

Presupuesto seleccionado

Presupuesto: 2210: Papas Tamaño: 10 Unidad: ha Actualizar Editar Datos del presupuesto

Cuenta	Descripción	Mes	Unidad	Cantidad / ha	Precio Unitario	Tamaño	Total / ha	Costo/Ingreso Total	IVA
1 601050	Superfosfato triple	Agosto	kg	960	170	10	163.200	1.632.000	No
2 601083	Supermitro	Agosto	kg	660	166	10	90.760	907.600	No
3 601084	Muriato de potasio	Agosto	kg	240	150	10	36.400	364.000	No
4 601104	papa semilla	Agosto	kg	3.000	200	10	600.000	6.000.000	No
5 601220	Herbicida	Agosto	lt	3	4.200	10	12.600	126.000	No
6 601220	Herbicida	Septiembre	lt	1	26.200	10	26.200	262.000	No
7 610200	Mano de obra ocasional	Marzo	\$	300	550	10	165.000	1.650.000	No
8 610200	Mano de obra ocasional	Septiembre	\$	0	40.000	10	40.000	400.000	No
9 630100	Arriendo de maquinaria	Agosto	hr	12	10.000	10	120.000	1.200.000	No
10 630140	Arriendo maquinaria fumigación	Agosto	\$	0	5.000	10	5.000	50.000	No
11 630140	Arriendo maquinaria fumigación	Septiembre	\$	0	5.000	10	5.000	50.000	No
12 702210	Venta Papa	Abril	qsm	270	6.000	10	1.620.000	16.200.000	No

Mostrar: 30 Total: 13 Desde: 1 Hasta: 13 Controles

FIGURA 22. Presupuesto por actividad.

Cada presupuesto esta asociado a una actividad y requiere que el usuario ingrese los parámetros de tamaño y unidad en que se trabajara el rubro, por ejemplo 30 hectáreas, en el caso de trabajarse en pesos (\$) el sistema no necesita el campo tamaño. Estos valores serán usados por el sistema para realizar cálculos en los informes. En la figura 23 pantalla 1 se crean los presupuestos y en la pantalla 2 se le pueden agregar datos.

The image displays two software windows. The first window, titled 'Crear presupuesto', contains a 'Crear Actividad' button, a dropdown menu for 'Actividad' (selected as '4110: Manzanos'), a text input for 'Tamaño' (value '5'), a red note 'Ingrese el tamaño de ha', a dropdown for 'Unidad' (selected as 'ha'), and buttons for 'Crear Presupuesto' and 'Cerrar Ventana'. A red circle highlights the 'Actividad' dropdown, with a red arrow pointing to the text 'Actividad presupuestada' above it. The second window, titled 'Agregar Datos al Presupuesto', contains a 'Crear Cuenta' button, a dropdown for 'Cuenta' (selected as '631100: Mantencion plantaciones'), a dropdown for 'Mes' (selected as '13: Anual: Enero a Diciembre'), a dropdown for 'Unidad' (selected as '\$'), a dropdown for 'IVA' (selected as 'SI'), radio buttons for calculation methods ('Total' is selected), and a 'Total' input field showing '30000'. It also has 'Agregar' and 'Cerrar Ventana' buttons.

FIGURA 23. Formularios para creación de presupuestos

Una vez creado el presupuesto se procede a ingresar datos que estén directamente asociados con la actividad, como por ejemplo, el costo anual de la mantención de las plantaciones, la cuenta de electricidad de un trimestre, honorarios de los empleados de la empresa en un determinado mes, etc. Todos estos datos están asociados a una cuenta del plan contable previamente configurado por el usuario, estas cuentas pueden ser de gastos (600000) o de ingresos (700000). En el campo mes se decide si se quiere que el total se prorratee durante todo el año, se asigne a un determinado trimestre o semestre, se cargue a un mes específico o se considere como un movimiento interno, el sistema se encarga de dividir los valores correctamente durante el ejercicio.

Además existen tres formas de ingresar la información dependiendo de con que datos se cuente, estas son las siguientes:

- **(A) Total:** Esta alternativa se selecciona cuando no se dispone de información de detalle o la estimación de gasto o ingreso resulte difícil, consiste en presupuestar en forma global el costo total.
- **(B) Tamaño por Costo unitario:** Si sólo se tiene información respecto a número de unidades involucradas y el costo o ingreso global por unidad de referencia. El sistema calcula el total al multiplicar estos dos valores.
- **(C) Cantidad por unidad:** Cuando se dispone de información de Cantidad de insumo o producto a utilizar u obtener por unidad, el precio unitario del insumo o producto y el número de unidades involucradas. El sistema calcula el total al multiplicar estos valores.

A continuación en la figura 24 las tres formas de ingreso de datos a los presupuestos.



FIGURA 24. Formas de ingreso de datos al presupuesto

### 3.8.2. Presupuesto de caja

Todo plan de explotación debe prever, además de los ingresos y costos del periodo, las inversiones a realizar durante el ejercicio, las amortizaciones de deudas cuyos vencimientos ocurrirán dentro del periodo analizado, los reembolsos de dinero por concepto de créditos otorgados y los saldos de cuentas corrientes si existieran. Estos datos están asociados a cuentas de balance (100000 a 599999) y pueden ser asignados a un año completo, semestre, trimestre, mes específico o movimientos internos. Los datos ingresados aquí se ven reflejados en el informe del presupuesto de caja. En la figura 25 se ve la pantalla donde estos se crean.



FIGURA 25. Presupuesto de caja

A continuación vemos la tabla 17 que contiene los datos de un presupuesto de caja típico, en este se ven pagos de prestamos, solicitudes y pagos de créditos de cuentas corrientes e ingresos a caja.

TABLA 17. Presupuesto caja

Cuenta	Descripción	Ingresos	Egresos	Mes	IVA
140100	Préstamo 1		7.102.804	Diciembre	No
500100	Crédito BCI		7.998.000	Segundo Trimestre: Abril - Mayo - Junio	No
500200	Crédito Banco Estado	8.000.000		Diciembre	No
500200	Crédito Banco Estado		3.360.000	Anual: Enero a Diciembre	No
500300	Crédito Banco Santander	10.000.000		Enero	No
500300	Crédito Banco Santander		10.000.000	Abril	No
500300	Crédito Banco Santander		720.000	Anual: Enero a Diciembre	No
570000	Caja	3.150.000		Agosto	No

### 3.9. Informes del presupuesto

La elaboración del presupuesto permite plantear los problemas de equilibrio y de expansión de la empresa. Los resultados que proporciona, permiten conocer si el agricultor dispondrá o no de ingresos suficientes para financiar su funcionamiento, siendo alrededor de esta necesidad vital donde se centran los problemas decisivos de la mayoría de las explotaciones.

El *software* CONGABOV genera a partir de los datos ingresados por el usuario una serie de informes que permiten conocer cuales serán los egresos e ingresos totales presupuestados para una actividad en específico o toda la explotación, flujos de caja para el ejercicio contable y resultados globales entre otras cosas. Todo esto apoyado con gráficos y contenido de ayuda por informe para facilitar la comprensión de la información entregada.

#### 3.9.1. Presupuesto técnico económico

En este informe, se describe el programa técnico desarrollado por el agricultor para cada predio y las actividades vinculadas a éstos, entregando los resultados económicos correspondientes al programa financiero de cada actividad planificada, como el detalle de costos o ingresos de cada cuenta y sus montos acumulados. Finalmente, se determina

el ingreso, el costo y el margen tanto por unidad como por actividad. En la figura 26 se ve la pantalla donde se genera este informe. [Ler07]

Cuenta	Descripción	Mes	Unidad	Cantidad / Unidad	Precio Unitario	Tamaño	Monto / ha	Total	Acumulado Cuenta
601082	Fertilizante Mezcla 2	Segundo Trimestre - Abril, Mayo y Junio	kg	420	170	170	52.774	12.138.000	12.138.000
601083	Supernitro	Diciembre	kg	71	165	170	8.659	1.991.550	1.991.550
601200	Pesticidas	Agosto	\$	0	3.200	170	2.366	544.000	544.000
601820	Polietileno para silo	Diciembre	\$	0	5.300	87	2.005	461.100	
601820	Polietileno para silo	Diciembre	\$	0	5.300	87	2.005	461.100	
630100	Arriendo de maquinaria	Diciembre	\$	0	0	0	6.522	1.500.000	1.500.000
703300	Produccion ensilaje	Movimiento Interno	ton	22	5.000	87	41.609	9.570.000	9.570.000
703500	Pastoreo	Movimiento Interno	\$	0	0	0	32.721	7.525.750	7.525.750

Totales		Ingreso / ha	74.329	Total Ingreso	17.095.750
		Costo / ha	74.329	Total Costo	17.095.750
		Margen / ha	0	Margen Actividad	0

FIGURA 26. Presupuesto técnico económico

Este informe se genera a partir solo de los datos ingresados en el módulo del presupuesto por actividad. En la tabla 18 se describen todas las columnas y cálculos que muestra este informe.

TABLA 18. Presupuesto técnico económico

Columna	Descripción
Cuenta	Código de la cuenta usada.
Descripción	Nombre de la cuenta usada.
Mes	Mes al que fue cargado ya sea uno específico o trimestral, semestral, etc.
Unidad	La unidad en la que se trabaja la cuenta.
Cantidad / Unidad	Es el campo cantidad por unidad en el caso de que se haya usado el tipo de ingreso C.
Precio unitario	Es el campo costo unitario en el caso en que se hubiera usado el tipo de ingreso B o C.
Tamaño	Es el campo tamaño en el caso de que se hubiera usado el tipo de ingreso B o C.
Monto / Unidad	Es el total del dato dividido en la unidad en que se presupuesto la actividad.
Total	Total del dato
Acumulado cuenta	Suma todos lo montos asociados a la cuenta y a la actividad seleccionada.

Luego el informe muestra una tabla con los totales de gastos e ingresos y el margen de la actividad obtenida de la suma aritmética de estos valores. También muestra estos totales divididos por el tamaño del presupuesto.

### 3.9.2. Estructura de costos.

Este informe tiene como objetivo reagrupar los egresos presupuestados, indicando el costo total y por unidad que representa cada ítem o grupo de cuentas, determinando los

gastos totales y la medida a nivel de cada empresa considerada en su conjunto; información entregada al final del informe. [Ler07]

Se genera a partir de los datos presupuestados de todas las actividades tomando en consideración únicamente las cuentas 600000 correspondientes a los gastos y calculando los totales por grupos de cuentas. Un grupo de cuentas es por ejemplo el grupo gastos en pesticidas (601200) en el se encuentran las cuentas; Fungicidas (601210), Herbicidas (601220), Insecticidas (601230) y Raticidas (601240).

También da la opción de seleccionar un grupo de cuentas y mostrar el total para cada una de las cuentas que le pertenezca y haya sido considerada en algún presupuesto. En la figura 27 se ve la pantalla del informe.

Vista de impresión Ver Gráfico ?

Totales por Categoría:

Grupo Cuentas	Descripción	Total	/Ha	%
1.- 601000	Gastos en Fertilizantes y Abonos	18.553.050	80.665	9,01
2.- 601100	Gastos en Semillas y Plantas	6.000.000	26.087	2,91
3.- 601200	Gastos en Pesticidas	1.759.500	7.650	0,85
4.- 601300	Gastos en Alimentos de Ganado	76.066.319	330.723	36,82
5.- 601400	Gastos en Productos Veterinarios	3.720.000	16.174	1,81
6.- 601500	Gastos en Combustibles y Lubricantes	3.640.000	15.826	1,77
7.- 601800	Gastos en Insumos Varios	922.200	4.010	0,46

601000 : Gastos en Fertilizantes y Abonos

Cuenta	Descripción	Cantidad	Unidad	Total	/Ha	%
1.- 601050	Superfosfato triple	9.600	kg	1.632.000	7.096	0,79
2.- 601082	Fertilizante Mezcla 2	71.400	kg	12.138.000	52.774	5,89
3.- 601084	Muriato de potasio	2.400	kg	384.000	1.670	0,19
4.- 601083	Supernitro	17.570	kg	2.899.050	12.605	1,41
5.- 601000	Aplicacion de fertil	0	\$	1.500.000	6.522	0,73

Total Costos 206.015.966 Total / Ha 895.722

FIGURA 27. Estructura de costos

El gráfico de torta generado solo muestra la información de los grupos de cuentas en caso de que su participación en el total de gastos sea mayor al 2 %, todos los grupos que no superen este porcentaje se agrupan en la categoría otros. Esto se ve en la figura 28.

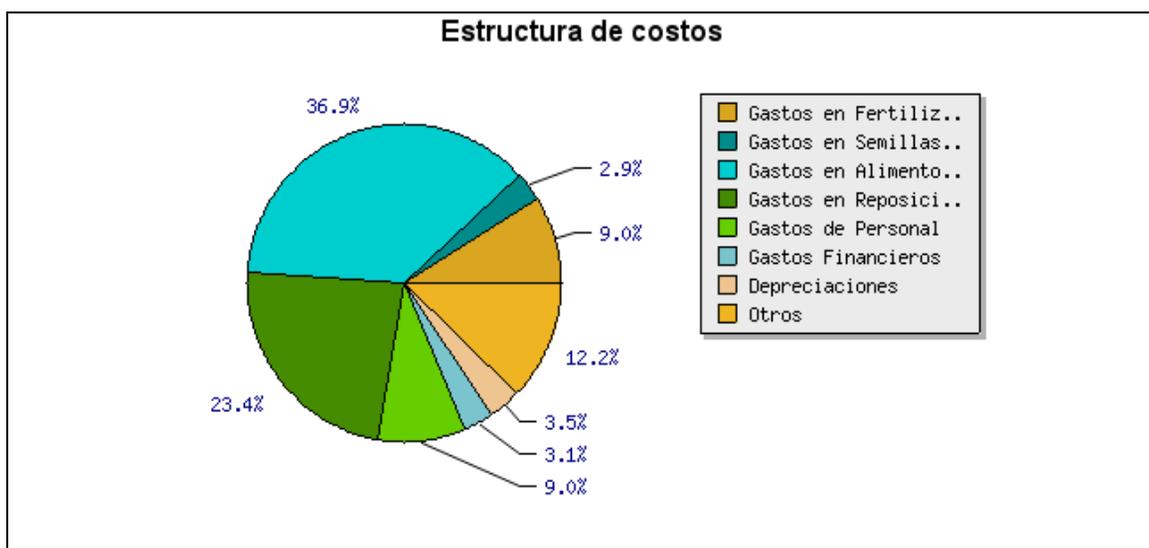


FIGURA 28. Gráfico estructura de costos

En las tablas 19 y 20 se explican cada uno de los campos generados en este informe. La primera corresponde al grupo de cuentas y la segunda a las cuentas del grupo seleccionado.

TABLA 19. Estructura de costos (grupos de cuentas)

Columna	Descripción
Grupo de cuentas	Código del grupo de cuentas
Descripción	Nombre del grupo
Total	Se suman los montos de todos los datos de los presupuestos que estén asociados a alguna cuenta del grupo.
/Ha	El monto total lo divide por la superficie agrícola útil de la explotación.
%	Indica el porcentaje de participación del grupo en el total de gastos presupuestados para todas las actividades (No incluye el presupuesto de caja).

Al seleccionarse un grupo de cuentas se muestra la información de sus cuentas. La definición de los campos mostrados se ven en la siguiente tabla.

TABLA 20. Estructura de costos (Cuentas)

Columnas	Descripción
Cuenta	Código de la cuenta.
Descripción	Nombre de la cuenta.
Cantidad	Cantidad total presupuestada para la cuenta.
Unidad	Unidad en la que se mide la cuenta.
Total	Monto total presupuestado para la cuenta.
/Ha	Monto total dividido por la superficie agrícola útil de la explotación.
%	Indica el porcentaje de participación de la cuenta en el total de gastos presupuestados para todas las actividades.

Finalmente se muestran el total de costos presupuestados para la explotación en el desarrollo de sus actividades y este monto dividido por la superficie agrícola útil.

### 3.9.3. Estructura de ingresos

Este informe tiene como objetivo reagrupar los ingresos presupuestados por cada cuenta, para esto revisa lo presupuestado para todas las actividades y obtiene los datos que estén asociados a cuentas de ingresos (700000). En las figuras 29 y 30 se ven el informe y su respectivo gráfico.

Cuenta	Descripción	Cantidad	Unidad	Total	/Ha	%
702210	Venta Papa	3.000	qqm	18.000.000	78.281	5,95
703300	Produccion ensilaje	1.914	ton	9.570.000	41.609	3,16
703500	Pastoreo	0	\$	7.525.750	32.721	2,49
704000	Venta de Manzanas	300	ton	15.735.000	68.413	5,2
708420	Venta de vacas desecho	15.000	kg	5.400.000	23.478	1,79
708430	Venta de vaquillas	74.800	\$	65.100.000	283.043	21,53
708440	Venta de terneros	11.000	kg	5.500.000	23.913	1,82
708491	Venta de leche	1.309.400	lt	163.755.345	711.980	54,15
708550	Venta de novillos	21.270	kg	11.824.100	51.409	3,91
				<b>Total</b>	<b>Total / Ha</b>	
<b>Total Ingresos</b>				302.410.195	1.314.827	
<b>Total Costos</b>				206.015.966	895.722	
<b>Beneficio Neto</b>				96.394.229	419.105	

FIGURA 29. Estructura de ingresos

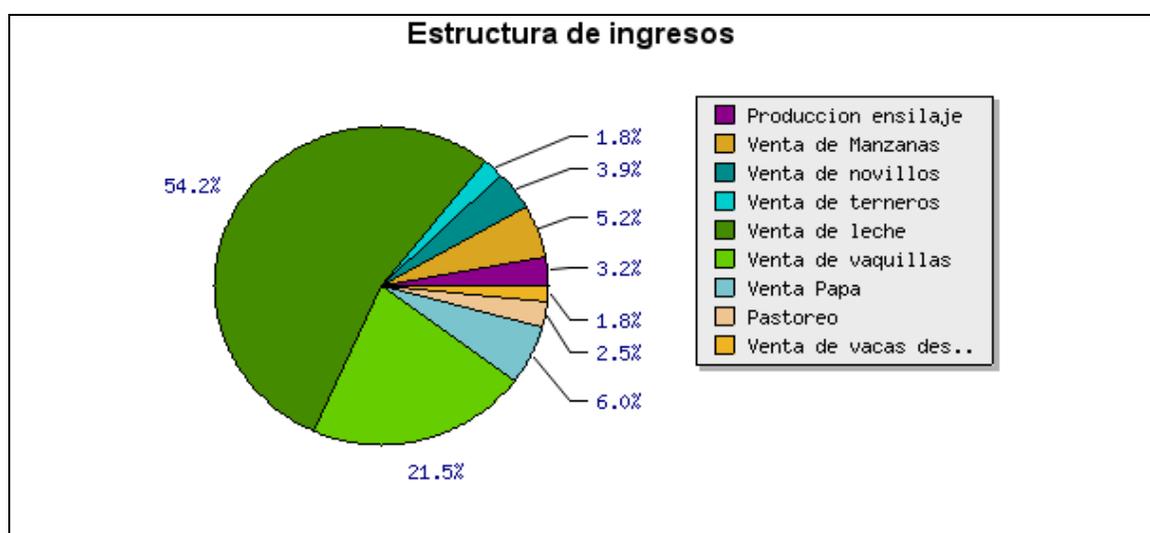


FIGURA 30. Gráfico estructura de ingresos

En la tabla 21 se describen cada uno de los campos que componen este informe.

TABLA 21. Estructura de ingresos

Columna	Descripción
Cuenta	Código de la cuenta.
Descripción	Nombre de la cuenta.
Cantidad	Cantidad total presupuestada para la cuenta.
Unidad	Unidad en que se mide la cuenta.
Total	Monto total presupuestado en la explotación para la cuenta.
/Ha	Monto total dividido en la superficie agrícola útil.
%	Porcentaje de participación de la cuenta en los ingresos totales de la explotación.

Luego se calculan el **total de ingresos** y el **total de costos** de la explotación, la resta de estos valores nos da el **beneficio neto**. Estos valores también se presentan divididos por la superficie agrícola útil.

### 3.9.4. Resultado global de la explotación

En este informe se identifican las actividades de cada empresa, agrupadas por tipo de producción, de acuerdo al plan de cuentas elaborado previamente, indicándose el tamaño y los márgenes presupuestados por actividad como por grupo de producción (Ej. en el caso de las explotaciones individuales, cereales, producción animal, etc.). Su objetivo, será poner en evidencia el resultado de cada empresa, el que se obtendrá restando del margen total los gastos generales. Así, este documento permitirá al usuario verificar si los presupuestos de las empresas están equilibrados y, de acuerdo a ello, analizar los ajustes necesarios.

Codigo Actividad	Descripción	Tamaño Del Presupuesto	Total	Total / Unidad
<b>CULTIVOS ESCARDADOS</b>				
2210	Papas	10	5.348.500	534.850
	<b>Total</b>	<b>10</b>	<b>5.348.500</b>	<b>534.850</b>
<b>FORRAJERAS</b>				
3410	Pradera mejorada	230	0	0
	<b>Total</b>	<b>230</b>	<b>0</b>	<b>0</b>
<b>FRUTICULTURA</b>				
4100	Manzanos	5	7.757.125	1.551.425
	<b>Total</b>	<b>5</b>	<b>7.757.125</b>	<b>1.551.425</b>
<b>EXPLOTACIÓN BOVINA DE LECHE</b>				
8441	Crianza	220	6.915.037	31.432
8420	Lechería	237	84.246.689	355.471
8430	Crianza de vaquillas	135	22.839.400	169.181
	<b>Total</b>	<b>592</b>	<b>114.001.126</b>	<b>556.084</b>

	Total	Total/Ha
Margen Total Explotación	127.106.751	552.638
Gastos Generales	-30.712.522	-133.532
Beneficio Neto	96.394.229	419.105

FIGURA 31. Resultado global de la explotación

Como ejemplo, en la figura 31 se ve el grupo de la explotación bovina de leche (8400) y sus respectivas actividades; crianza (8441), lechería (8420) y crianza de vaquillas (8430). El informe calcula los márgenes totales para cada actividad y luego entrega el margen total del grupo. El gráfico de este informe ejemplificado en la figura 32 muestra el margen total de cada actividad y luego el margen total dividido por el tamaño del presupuesto.

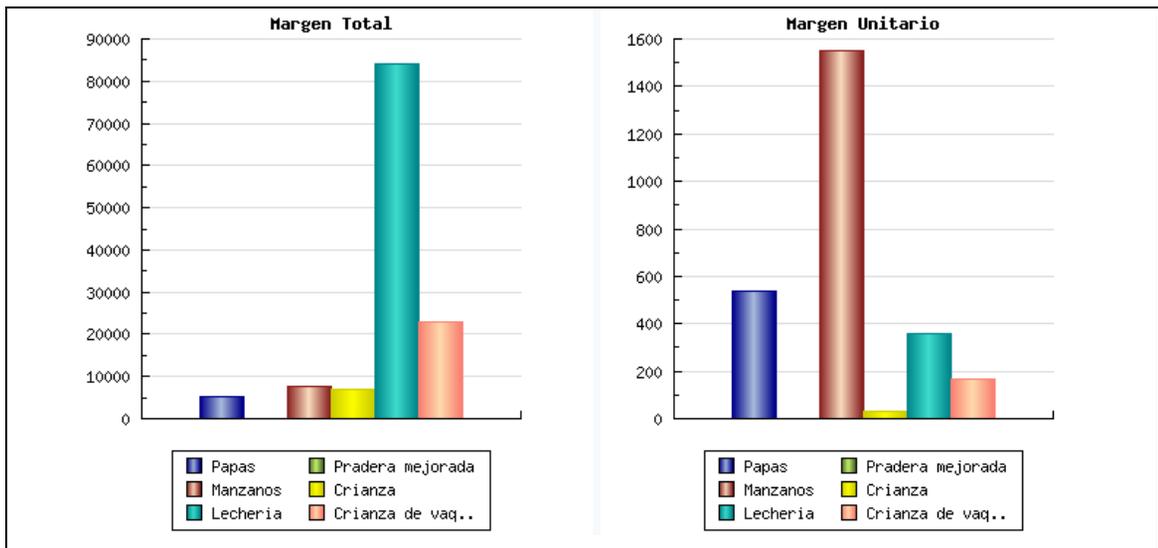


FIGURA 32. Gráfico resultado global de la explotación

En la tabla 22 se detallan cada una de las columnas mostradas por este informe.

TABLA 22. Resultado global de la explotación

Columna	Descripción
Código actividad	Código de la actividad.
Descripción	Nombre de la actividad.
Tamaño del presupuesto	Tamaño ingresado por el usuario al crear el presupuesto par la actividad.
Total	Margen total de la actividad
Total / Unidad	Margen total dividido por el tamaño del presupuesto.

Luego, este informe entrega el margen total de la explotación que corresponde a la suma de todos los márgenes de las actividades, para después sumarle el resultado (normalmente negativo) de la actividad gastos e ingresos generales (0000), la cual hace referencia a los gastos comunes de todas las actividades como la electricidad, teléfono, depreciaciones de maquinarias, combustibles y lubricantes, etc. A esta actividad también se le asocian ingresos de ventas o servicios que no corresponden con ningún rubro de la empresa. Con esto se obtiene el **beneficio neto de la explotación**.

### 3.9.5. Presupuesto de caja.

En este informe se detallan las cantidades de dinero que se estima ingresen o egresen de caja cada mes durante el periodo contable analizado, de acuerdo a las informaciones proporcionadas en el presupuesto. Es decir, se desglosan tanto los egresos como los

ingresos, indicando las últimas filas el total de ingresos, el total de egresos, el saldo mensual y el saldo acumulado mes a mes.

Si el presupuesto de caja presenta saldos acumulados negativos, significa que la empresa no puede autofinanciarse durante esos meses. Ante esta situación, se deberán tomar las medidas de corrección pertinentes; por ejemplo, analizar la posibilidad de sacar antes al mercado algún tipo de producto, cancelar más tarde ciertos insumos, solicitar a alguna entidad financiera un crédito operacional para solventar los déficits de caja previstos, etc. [Ler07] En la figura 33 un ejemplo de este informe.

	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul
<b>Egresos</b>												
Aplicacion de fertil	750.000			750.000								
Ariendo de maquinaria	1.200.000				1.500.000							
Ariendo maquinaria fumigacion	50.000	65.000	15.000		15.000	15.000						
Concentrados	3.125.000	785.000	2.889.000	2.655.000	2.502.000	1.491.000	1.452.000	1.875.000	3.181.666	3.391.666	3.721.666	3.425.000
Credito BCI									2.666.000	2.666.000	2.666.000	
Credito Banco Santander	60.000	60.000	60.000	60.000	60.000	60.000	60.000	60.000	10.060.000	60.000	60.000	60.000
Credito BancoEstado	280.000	280.000	280.000	280.000	280.000	280.000	280.000	280.000	280.000	280.000	280.000	280.000
Electricidad	130.780	130.780	130.780	130.780	130.780	130.780	130.780	130.780	130.780	130.780	130.780	130.780
Fertilizante Mezcla 2									4.046.000	4.046.000	4.046.000	
Gastos Inseminacion											420.000	
<b>Total</b>												
Total Ingresos	21.263.800	19.849.625	15.914.880	20.558.020	22.780.120	26.394.800	14.660.800	18.488.500	53.961.100	19.724.250	14.586.000	20.308.200
Total Costos	21.256.665	11.186.940	9.817.199	11.693.824	21.296.753	5.015.095	4.616.720	7.925.720	26.723.915	14.923.915	15.673.915	9.400.665
Saldo Mensual	7.235	8.662.685	6.097.681	8.864.196	1.463.367	21.379.705	10.044.080	10.562.780	27.237.185	4.800.335	-1.087.915	10.907.635
Saldo Acumulado	7.235	8.669.920	14.767.601	23.631.797	25.095.164	46.474.869	56.518.949	67.081.729	94.318.914	99.119.249	98.031.334	108.938.969

FIGURA 33. Informe presupuesto de caja

Este informe tiene asociado gráficos, mostrados en la figura 34, que permiten visualizar la evolución del total de ingresos y de egresos (gráfico de líneas), así como los saldos mensuales y acumulados (gráfico de área).

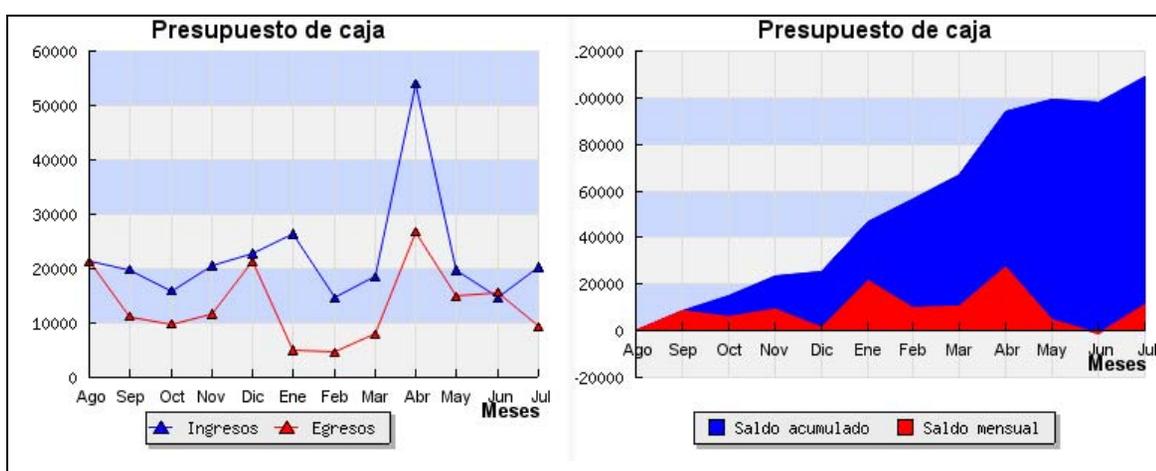


FIGURA 34. Gráficos presupuesto de caja.

En la tabla 23 se explican los campos y funcionamiento de este informe.

TABLA 23. Informe presupuesto caja

Columna	Descripción
Cuentas	En la primera columna se listan las cuentas separadas en las de gastos (600000) y luego las de ingresos (700000). Solo se muestran las cuentas que hayan sido usadas en el presupuesto.
Meses	Las siguientes 12 columnas representan los meses del ejercicio en curso, comenzando por el mes de inicio, parámetro ingresado en la configuración de la explotación. En estas se muestran los montos totales de cada cuenta para ese mes en específico.

### 3.10. Balance inicial

El balance es una presentación estática y ordenada del inventario, a una fecha determinada, de la situación de los recursos financieros, de los cambios de patrimonio y de la situación de los empleos de un agente económico. [CON78]

El balance inicial es un inventario de los bienes (activos), y deudas (pasivos) que posee la empresa al comienzo del ejercicio contable. La primera vez que el usuario use la herramienta CONGABOV tendrá que crear este inventario, de ahí en adelante en cada nuevo ejercicio el balance inicial es generado automáticamente por el sistema en base a la información del ejercicio anterior. En la figura 35 se muestran las pantallas de administración.

Solo se admiten cuentas del balance

Cuenta	Descripción	Débito	Crédito	Cantidad	Unidad
1 100000	Capital Propio		741.287.000	0,5	
2 140110	Crédito Bhif				
3 210000	Suelo Agrícola				
4 211000	plantaciones permanentes				
5 212000	construcciones				
6 213000	Instalaciones				
7 214000	Maquinarias y Equipo				
8 214200	Equipo de riego				
9 217100	Vacas de producción				
10 217140	Vaquillas preñadas				
11 217145	vaquillas encaste				
12 217180	Toros				
13 217200	Caballos de trabajo	720.000		6	Cabezas
14 313230	Paja	300.000		1.000	Fardos

Total Débitos: 884.935.000    Total Créditos: 884.935.000

FIGURA 35. Balance inicial

Este solo admite cuentas de balance (100000 - 599999) siendo esta una de las validaciones de la pantalla de ingreso. El monto total de créditos debe ser igual al de débitos, esto se justifica debido a que la diferencia entre estos dos constituye el capital propio de la empresa (cuenta 100000) siendo este calculado automáticamente por el sistema y motivo por el cual el usuario no puede modificar manualmente esta cuenta. En la tabla 24 se muestra un ejemplo de balance inicial.

TABLA 24. Balance inicial

Cuenta	Descripción	Débito	Crédito	Cantidad	Unidad
100000	Capital Propio		741.297.000		\$
140110	Crédito Bhif		117.000.000		\$
210000	Suelo Agrícola	631.800.000		351	ha
211000	plantaciones permanentes	22.320.000		5	ha
212000	construcciones	44.365.000			\$
213000	Instalaciones	10.735.000			\$
214000	Maquinarias y Equipo	28.100.000			\$
214200	Equipo de riego	4.350.000		5	ha
217100	Vacas de producción	89.600.000		224	Cabezas
217140	Vaquillas preñadas	13.500.000		27	Cabezas
217145	vaquillas encaste	9.000.000		30	Cabezas
217160	Toros	3.000.000		6	Cabezas
217200	Caballos de trabajo	720.000		6	Cabezas
313230	Paja	300.000		1.000	Fardos
313300	Stock ensilaje	2.125.000		240	ton
328430	Vaquillas lechería	7.800.000		65	Cabezas
328440	Terneros de leche	12.000.000		120	Cabezas
328445	Terneras de leche	4.720.000		59	Cabezas
500100	Crédito BCI		7.998.000		\$
500200	Crédito Banco Estado		10.000.000		\$
500300	Crédito Banco Santander		8.640.000		\$
570000	Caja	500.000			\$
	<b>Totales</b>	<b>884.935.000</b>	<b>884.935.000</b>		

### 3.11. Movimientos contables

La contabilidad general registra en sus cuentas las operaciones correspondientes a intercambios de valores. Puede representar la constitución de un activo; la incursión de un gasto (pago de un consumo, pago de impuesto, retribución por un servicio); transferencias entre activos o entre gastos, o combinación de ellos, la constitución de un pasivo, etc. La contabilidad general, atiende exclusivamente a la naturaleza de tales intercambios. El criterio del intercambio, que es de orden jurídico, económico y

financiero, constituye la norma esencial que se usa en la clasificación, agrupamiento y procesamiento de sus cuentas. [CON78]

Dentro del plan contable general se mantiene el uso del método de la partida doble. Este se basa en que todo hecho económico tiene origen en otro hecho de igual valor pero de naturaleza contraria, como por ejemplo; cuando se presta dinero, el que recibe debe y el que entrega tiene. Esta posee dos principios básicos:

- “Quien recibe debe a quien entrega”
- “No existe deudor sin acreedor, ni acreedor sin deudor”

En el sistema CONGABOV los movimientos contables se constituyen de un encabezado y líneas de detalle. El encabezado se compone de los campos explicados la tabla 25.

TABLA 25. Encabezado movimiento

<b>Campo</b>	<b>Descripción</b>
Número de factura o comprobante	Campo numérico que identifica al movimiento.
Fecha	Fecha en que se ejecutó el movimiento, el sistema verifica que esta este dentro del rango del ejercicio.
Proveedor o cliente	Entidad con la que se realizó la transacción, en caso de ser un movimiento interno, no se necesita este campo.
Número de cheque	El sistema permite la opción de llevar el registro de la chequera asociando los movimientos a un número de cheque en el caso que corresponda. Este campo no es obligatorio.

El encabezado tiene por función agrupar a las líneas de detalle, el sistema permite cualquier cantidad de estas. En la tabla 26 se muestran los campos que posee cada una.

TABLA 26. Líneas de detalle

<b>Campo</b>	<b>Descripción</b>
Glosa	Campo en el cual se puede anotar una descripción del movimiento.
Actividad	Código del rubro o actividad al cual se le carga el movimiento.
Cuenta provee	Cuenta desde donde proviene el movimiento.
Cuenta recibe	Cuenta receptora del movimiento.
Unidad	Todas las cuentas vienen con unidades predefinidas, el sistema se encarga de escoger esta.
Cantidad	Si la unidad escogida por el sistema es distinta de pesos (\$), se habilita este campo para señalar la cantidad de unidades del movimiento.
Monto	Valor neto del movimiento.
Impuestos	En la pantalla de ingreso se ofrecen varias opciones de impuestos como el IVA, ILA, etc. Los cuales son calculados internamente por el sistema. Sus valores son ingresados por el administrador del sistema.
Específico recuperable y no recuperable	Existe el caso especial de impuesto a los combustibles, estos campos solo se activan al seleccionarse cuentas del grupo combustibles y lubricantes (601500).

En la tabla 27 se ve un ejemplo de un movimiento contable.

TABLA 27. Ejemplo de movimiento contable

Encabezado del movimiento							
Número de factura	Fecha	Proveedor o cliente	Número de cheque				
425	29-06-2008	ANASAC.	46				
Líneas de detalle							
Glosa	Actividad	Provee	Recibe	Unidad	Cantidad	Monto	Impuesto
ANASAC	Gastos e ingresos generales	Caja	Regularización	\$		457.300	86.887
ANASAC	Lechería	Regularización	Sales minerales.	Kg.	800	120.000	
ANASAC	Crianza	Regularización	Sustituto lácteo.	Kg.	500	199.600	
ANASAC	Crianza vaquillas	Regularización	Sales minerales.	Kg.	850	137.700	

En este ejemplo se ve el uso de la cuenta de regularización (490.000), en la primera línea esta recibe un monto de \$457.300 de la cuenta de caja (570000), luego destina 800 Kg. por un monto de \$120.000 en sales minerales a la actividad de lechería, 500 Kg. de sustituto lácteo por un monto de \$199.600 a la actividad de crianza y finalmente 850 Kg. de sales minerales equivalentes a \$137.700 a la actividad crianza de vaquillas. **Como regla lo que recibe la cuenta de regularización tiene que ser igual a lo que provee.** El sistema alerta al usuario en caso de que estos montos sean distintos.

Las cuentas poseen unidades predefinidas en el sistema, por ejemplo el petróleo (601510) se mide en litros (lt.), heno de pradera (601320) en fardos, plantaciones forestales (211200) en hectáreas (ha), caja (570000) en pesos (\$), etc. En los datos del presupuesto al depender estos de una sola cuenta se trabaja inmediatamente con la unidad de esta, pero en los movimientos que utilizan el método de la partida doble se tienen dos cuentas, las que reciben y las que proveen. A continuación se describen los criterios usados por el software para decidir que unidad escoger en las tres situaciones posibles.

- **Unidad pesos contra unidad pesos:** en este caso las dos cuentas se miden en pesos y solo se están traspasando montos monetarios, por lo tanto se le asigna esta unidad al movimiento y se bloquea el campo de cantidad. Por ejemplo la cuenta caja (570000) provee un monto de \$364.756 a la cuenta salarios mano de obra permanente (610111) medida en \$ también.

- **Unidad pesos contra otra unidad cualquiera:** en este caso se mantiene la unidad distinta de pesos, esto debido a que se asume que se esta entregando un monto de dinero por alguna cantidad de un bien. Por ejemplo la cuenta de caja (570000) provee un monto de \$90.000 por 600 Kg. de sales minerales (601635).
- **Dos unidades iguales y distintas de la unidad pesos:** Si las dos unidades son iguales se mantiene y se habilita el campo cantidad para ingresar la cantidad. Por ejemplo la cuenta stock en ensilaje (313300) medida en toneladas provee 375 Ton a la cuenta ensilaje de pradera (601330) también medida en esta unidad por un monto de \$2.250.000.
- **Dos unidades distintas entre si y diferentes de la unidad pesos:** En este caso el sistema asume que el movimiento esta mal hecho alertando al usuario ya que no se pueden transferir unidades ni dineros entre dos cuentas de distinta naturaleza física. Por ejemplo tratar de traspasar de la cuenta potros (328110) medida en cabezas a la cuenta Lupino (601355) medida en Kg. no tiene sentido real.

Las explotaciones agropecuarias poseen tipos de movimientos comunes a las actividades que desarrollan, en los cuales se ven involucradas siempre los mismos grupos de cuentas. Debido a esto se decidió que el *software* CONGABOV tuviera estos tipos predefinidos filtrando la información que se despliegan en las listas de cuentas que proveen y reciben.

En la tabla 28 se ven los 13 tipos y sus propiedades

TABLA 28. Tipos de movimientos

Tipo de movimiento	Grupos que proveen	Grupos que reciben
Compras	Cuentas de proveedores, regularización y caja.	Cuentas del grupo gastos por naturaleza (600000)
Pago de Créditos	Cuenta regularización, cuentas corriente bancos, caja.	Deudas a largo plazo, regularización, deudas a corto plazo, cuentas por pagar, gastos financieros.
Pago de Salarios y Leyes Sociales	Cuenta corriente bancos, caja.	Gasto personal, gastos honorarios.
Depreciación de Activos del Inventario	Plantaciones permanentes, maquinaria y equipo, construcciones específicas lechería, instalaciones lechería, bovinos de leche, etc.	Cuenta de depreciaciones.
Diferencias de Inventario según Categorías de Animales	Bovinos de leche, stock animales y aves, animales en mediaría, inventario inicial y final.	Bovinos de leche, stock animales y aves, animales en mediaría, inventario inicial y final.
Ingreso a caja de créditos de corto y largo plazo	Deudas a largo plazo, deudas a corto plazo.	Caja y/o banco.
Movimientos Internos	Cuentas de stock, cuentas de ventas.	Cuentas de stock, cuentas de gastos.
Producciones de Forraje	Venta de cereales, cultivos escardados, productos forrajeros.	Stock alimentos de ganados, cereales, cultivos escardados, avance de cultivos forrajeros.
Corrección monetaria	Fondos de revalorización.	Cuentas del activo y pasivo.
Subvenciones	Cuenta bonificaciones y reembolsos.	Caja y/o banco.
Ventas Mensuales de leche	Venta de animales y subproductos.	Caja y/o banco.
Ventas con factura de distintas categorías de animales	Venta de animales y subproductos.	Caja y/o banco.
Ventas de productos de STOCK y Otros	Cuentas de stock	Cuentas de clientes y caja.

En las figuras 36, 37 y 38 se ven las pantallas de administración de los movimientos contables.

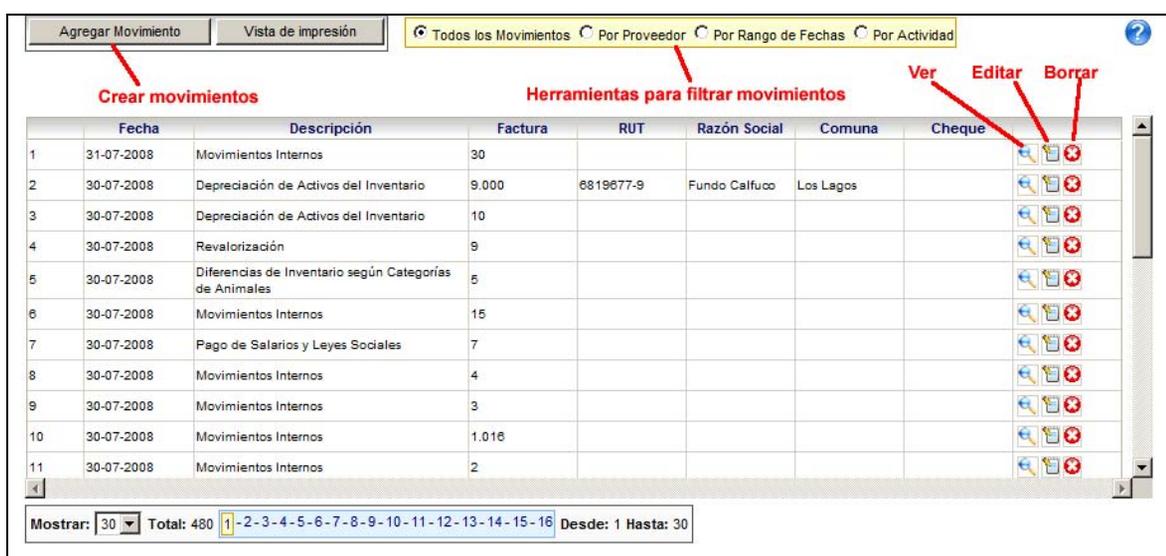


FIGURA 36. Administración movimientos contables

**Crear Movimientos**

1  Compras

2  Pago de Créditos

3  Pago de Salarios y Leyes Sociales

4  Depreciación de Activos del Inventario

5  Diferencias de Inventario según Categorías de Animales

6  Ingreso a caja de créditos de corto y largo plazo

7  Movimientos Internos

8  Producciones de Forraje.

9  Revalorización

10  Subvenciones

11  Ventas Mensuales de leche.

12  Ventas con factura de distintas categorías de animales

13  Ventas de productos de STOCK y Otros.

**Tipos de movimientos**

Siguiente >>    Cerrar Ventana

FIGURA 37. Tipos de movimientos

**Compras** — Tipo de movimiento

Crear Nuevo Movimiento · Crear Proveedor o Cliente · Crear Actividad · Crear Cuenta    **Crear cuentas, actividades, proveedores**

Nº Factura o Comprobante: 425    Fecha: 29-06-2008    Proveedor / Cliente: 91253000-0 : Anasac    Nº Cheque:

Glosa: ANASAC    Actividad: 0000: Gastos e Ingresos Generales    **Unidad**    Impuestos:  IVA (19 %)    **Impuestos**  
 ILA (8 %)  
 Impuesto Molino (7 %)  
 Impuesto Feria Ganado (8 %)

Monto: 457300

Cuenta que Provee: 570000: Caja (\$)    Unidad: It    **Editar línea**

Cuenta que Recibe: 601510: Petroleo (It)    Cantidad: 300

Editar Línea    Especifico recuperable: 2000    Especifico no recuperable: 1000    **Impuesto combustibles**    **Borrar línea**

Glosa	Actividad	Cta Provee	Cta Recibe	Un.	Ctd.	Monto	Impuesto	Esp. Rec.	Esp. No Rec.
ANASAC	0000 : Gastos e Ingresos Generales	570000 : Caja	490000 : Cuenta de Regularización	\$		457.300	86.887	0	0
ANASAC	8420 : Lecheria	490000 : Cuenta de Regularización	601365 : Sales minerales	kg	800	120.000	0	0	0
ANASAC	8441 : Crianza	490000 : Cuenta de Regularización	601302 : Sustituto lacteo	kg	500	199.600	0	0	0
ANASAC	8430 : Crianza de vaquillas	490000 : Cuenta de Regularización	601365 : Sales minerales	kg	850	137.700	0	0	0

Guardar cambios    Cerrar Ventana    **Lineas de detalle**

FIGURA 38. Pantalla para crear movimientos contables

### 3.12. Libros diarios

Los informes generados directamente de los datos ingresados en el módulo de contabilidad son el libro diario de compras y ventas, banco caja y el resumen.

### 3.12.1. Diario de compras y ventas

Este informe reagrupa las operaciones concernientes a las compras y ventas de productos y servicios, efectuadas por la explotación durante el periodo analizado. Permitiendo seleccionar por tipo de cuenta, es decir, puede mostrar solo las Compras o las Ventas. También se puede indicar un rango de fechas para un análisis más minucioso.

Se despliegan la fecha, número de factura, RUT, razón social, nombre del proveedor o cliente, valor neto de la factura, IVA, impuestos específicos para combustibles y lubricantes y, por último, el monto total de la factura que considera el monto neto y los distintos impuestos. Además, en la parte inferior de la ventana, muestra el total de los movimientos seleccionados donde se detalla el valor neto, el IVA, los impuestos específicos a los combustibles y lubricantes finalizando con el monto total de los movimientos seleccionados, incluidos los respectivos impuestos. En la figura 39 se aprecia este informe.

Fecha	Nº Factura	Rut	Razón Social	Nombre	Neto	IVA	Esp Rec	Esp No Rec	Total	
31-08-2007	5913	82.392.600-6	Cooprinsem	Cooprinsem	624	119	0	0	743	
31-08-2007	5913	82.392.600-6	Cooprinsem	Cooprinsem	3.684	700	0	0	4.384	
31-08-2007	6528	08.573.181-6	Repuestos Universal	Repuestos Univ.	84.746	16.102	0	0	100.848	
31-08-2007	7439	92.347.000-K	Soprole	Soprole	2.509	0	0	0	2.509	
31-08-2007	8791	92.347.000-K	Soprole	Soprole	7.920	1.505	0	0	9.425	
31-08-2007	9753	92.347.000-K	Soprole	Dos Alamos	517	98	0	0	615	
31-08-2007	118	04.849.108-1	Loreto Walker	L. Walker	157.356	29.898	0	0	187.254	
31-08-2007	1800	92.347.000-K	Soprole	Soprole	130.150	24.729	30.981	0	185.860	
01-09-2007	7693	92.347.000-K	Soprole	Soprole	5.886	1.118	0	0	7.004	
01-09-2007	7693	92.347.000-K	Soprole	Soprole	30.600	5.814	0	0	36.414	
01-09-2007	7693	92.347.000-K	Soprole	Soprole	18.833	3.578	0	0	22.411	
04-09-2007	2507	92.347.000-K	Soprole	Soprole	32.067	6.093	0	0	38.160	
04-09-2007	2507	92.347.000-K	Soprole	Soprole	40.000	7.600	0	0	47.600	
04-09-2007	2507	92.347.000-K	Soprole	Soprole	11.070	2.103	0	0	13.173	
05-09-2007	1568	92.347.000-K	Soprole	Soprole	1.842.949	350.160	0	0	2.193.109	
05-09-2007	7867	92.347.000-K	Soprole	soprole	368.590	70.032	0	0	438.622	
05-09-2007	7868	92.347.000-K	Soprole	Soprole	1.474.359	280.128	0	0	1.754.487	
06-09-2007	7913	92.347.000-K	Soprole	Soprole	310.000	58.900	0	0	368.900	
06-09-2007	7913	92.347.000-K	Soprole	Soprole	222.000	42.180	0	0	264.180	
06-09-2007	7913	92.347.000-K	Soprole	Soprole	50.000	11.170	0	0	61.170	
<b>Totales</b>					<b>Total</b>	184.326.671	18.392.131	369.223	38.956	203.126.981

FIGURA 39. Diario compras ventas

### 3.12.2. Diario de banco caja

Este informe muestra las operaciones y movimientos contables en los que haya intervenido alguna cuenta Banco o Caja seleccionada por el usuario durante un periodo de tiempo establecido por este. Estas están dentro del rango 560000 a 580000 correspondiente a dos clases.

En las cuentas de la clase bancos se encuentran las cuentas corrientes que posee la empresa en estas entidades. El plan contable trae por defecto las cuentas 560100: Cta. Cte. Bco. Estado, 560200: Cta. Cte. Bco. Santander, 560300: Cta. Cte. Bco. Chile, etc.

Las cuentas de caja registran la disponibilidad en dinero efectivo existente en la caja de la empresa, ya sea en moneda nacional o extranjera. Además, se incluyen los fondos disponibles para la realización de gastos en funciones regulares específicas de la empresa, sujetos a rendición y reposición, para mantenerlos en niveles preestablecidos. (Caja chica, fondo fijo) [CON78]

En la tabla 29 se detallan las columnas de este informe.

TABLA 29. Diario banco caja

Columnas	Descripción
Fecha	Fecha en que se efectuó el movimiento.
Cheque	Número de cheque.
Factura	Factura del movimiento
Glosa	Glosa de la línea de detalle.
Giros	De ser un egreso de dinero a la cuenta despliega el total de la línea de detalle.
Depósitos	De ser un ingreso de dinero a la cuenta despliega el total de la línea de detalle.
Saldos	Saldo de la cuenta seleccionada al efectuarse el movimiento.

Finalmente se muestra el saldo inicial con el que comenzó la cuenta, monto obtenido del balance inicial, monto total en giros, depósitos y saldo final. A continuación, en las figuras 40 y 41, se aprecian el informe y su respectivo gráfico.

Fecha	Cheque	Factura	Glosa	Giros	Depositos	Saldo
31-07-2007			--- SALDO INICIAL ---			500.000
01-08-2007		5311		4.284	0	495.816
01-08-2007	Cuenta seleccionada	5311		743	0	494.873
01-08-2007		5311		9.818	0	485.055
01-08-2007		7137	Equipos Indus.	23.562	0	461.493
02-08-2007		2222	semilla papas	6.000.000	0	-5.538.507
02-08-2007		2222	SPT	2.023.000	0	-7.561.507
02-08-2007		2222	Muriato de K	481.950	0	-8.043.457
02-08-2007		2222	Rango	15.172	0	-8.058.629
02-08-2007		2222	Supernitro	975.800	0	-9.034.429
04-08-2007		376	Soprole	301	0	-9.034.730
04-08-2007		376	Soprole	2.988	0	-9.037.718
04-08-2007		376	Soprole	5.173	0	-9.042.891
04-08-2007		376	Soprole	4.263	0	-9.047.154
05-08-2007		9803	Schuler	799.756	0	-9.846.910
08-08-2007		502	Soprole	39.226	0	-9.886.136
09-08-2007		584	Soprole	349.860	0	-10.235.996
09-08-2007		584	Soprole	182.784	0	-10.418.780
09-08-2007		584	Soprole	583.100	0	-11.001.880

Saldo Inicial	Total Giros	Total Depósitos	Saldo Final
500.000	203.156.933	276.549.736	73.892.803

FIGURA 40. Diario banco caja

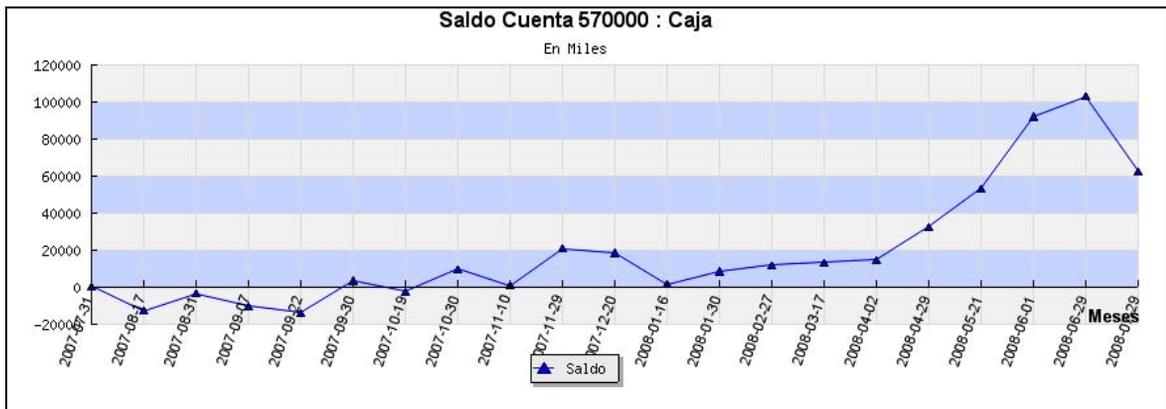


FIGURA 41. Diario banco caja

### 3.12.3. Resumen de compras y ventas

La finalidad de este informe es básicamente para información de tipo tributario, ya que globaliza los totales de compras y ventas efectuadas por la empresa. Recuerda los totales acumulados de compras y ventas hasta el proceso anterior, en sus formas de exento, afecto, IVA, específico y total; indica los montos para el período en cuestión y señala los nuevos saldos acumulados.

El I.V.A. de compras puede que no corresponda exactamente al 19% de las compras afectas. Esto se debe a que en agricultura, el impuesto específico cancelado por compras de petróleo puede recuperarse junto con el I.V.A. Por lo tanto, el I.V.A. corresponde al 19% de las compras afectas, más el impuesto específico recuperable. [Ler07] En la figura 42 se ve este informe.

Desde: 01-08-2007		Hasta: 31-07-2008		Mostrar		Vista de impresión	
<b>Resumen de Compras</b>							
	Exento	Afecto	IVA + Esp Rec	Esp No Rec	Total		
Acumulado al 01-08-2007	0	0	0	0	0		
Total periodo	87.525.979	96.800.692	18.761.354	38.956	203.126.981		
Nuevo acumulado	87.525.979	96.800.692	18.761.354	38.956	203.126.981		
<b>Resumen de Ventas</b>							
	Exento	Afecto	IVA	ILA	Total		
Acumulado al 01-08-2007	0	0	0	0	0		
Total periodo	0	232.161.994	44.110.779	0	276.272.773		
Nuevo acumulado	0	232.161.994	44.110.779	0	276.272.773		
				<b>Total IVA a pagar</b>	<b>Iva a pagar: 25.349.425</b>		

FIGURA 42. Resumen compras ventas

### 3.13. Control de gestión

En el módulo control de gestión encontramos una serie de informes que tienen como objetivo servir de apoyo a las decisiones de producción e inversión del empresario

agrícola, mediante la entrega periódica de información relacionada con el desarrollo de cada uno de los rubros que componen la empresa y con la evolución experimentada por sus capitales. Este módulo permite también controlar, en forma permanente, como se esta llevando a cabo el plan de explotación formulado a comienzos del período contable. [Ler07]

### 3.13.1. Libro Mayor

Este informe muestra los débitos y créditos realizados por una cuenta para una determinada actividad y en un rango de fechas seleccionados por el usuario. En el caso de que la fecha de inicio seleccionada por este para la generación del informe no corresponda con la fecha de inicio del ejercicio se muestra en el campo saldo anterior el monto total contabilizado por la cuenta anteriormente a este periodo.

El sistema ordena cronológicamente las operaciones efectuadas, indica cuando corresponde, las cantidades físicas y los precios unitarios (es decir, el informe permite llevar una contabilidad física), reproduciendo luego los montos monetarios bajo la columna débito si es una entrada o bajo la columna crédito si se trata de una salida. Finalmente, calcula los totales tanto físicos como monetarios, determina al Saldo de la cuenta y el precio promedio ponderado.

Este documento permite, por lo tanto, seguir la evolución, a lo largo del año, de cada uno de los elementos que componen la empresa y verificar si las operaciones asignadas a cada cuenta son rigurosamente exactas. [Ler07] En la figura 43 se ve este informe.

Fecha	Nº Factura	Glosa	Cantidad	Unidad	Precio/Unidad	Débito	Crédito
30-08-2007	128	Soprole	86.708	lt	124	0	10.751.792
30-08-2007	847	Soprole	12.400	lt	124	0	1.537.600
29-09-2007	131	Soprole	104.813	lt	123	0	12.891.999
29-09-2007	848	Soprole	12.000	lt	123	0	1.476.000
30-10-2007	132	Soprole	120.148	lt	125	0	15.018.500
31-10-2007	849	Soprole	12.400	lt	125	0	1.550.000
29-11-2007	133	Quillayes	114.376	lt	130	0	14.868.880
29-11-2007	850	Quillayes	10.500	lt	130	0	1.365.000
30-12-2007	136	Quillayes	103.188	lt	128	0	13.208.064
30-12-2007	851	Quillayes	13.000	lt	128	0	1.664.000
30-01-2008	137	Quillayes	83.180	lt	127	0	10.563.860
30-01-2008	852	Quillayes	12.400	lt	127	0	1.574.800
07-02-2008	853	Quillayes	10.278	lt	127	0	1.305.306
27-02-2008	139	Quillayes	63.255	lt	126	0	7.970.130
30-03-2008	141	Quillayes	76.218	lt	126	0	9.603.468
<b>Total Periodo</b>			1.245.527		128	0	159.675.278
<b>Saldo Anterior</b>			0			0	0
<b>Saldo Total</b>			1.245.527				159.675.278
<b>Precio Promedio</b>					128		

FIGURA 43. Libro mayor

### 3.13.2. Gestión por actividad

Mensualmente, o con la periodicidad requerida, se emite el informe de gestión por actividad. A este nivel, para cada una de las actividades o rubros de la empresa, el sistema reproduce los objetivos establecidos en el presupuesto y los confronta con los datos generados por la contabilidad hasta el momento del informe, realizando las comparaciones correspondientes. Las diferencias o desviaciones entre lo programado y lo realizado, se expresan en porcentaje. [Ler07]

Para la actividad seleccionada se revisan las cuentas de gastos (600000), ingresos (700000) y diferencias de inventario (800100 - 800200) que hayan sido presupuestadas o que tengan registrado algún movimiento durante el transcurso del ejercicio contable. El usuario puede especificar una fecha de término, por defecto se genera el informe con la fecha de fin del ejercicio. Cada una de las filas de este representa a una cuenta usada en la actividad, mostrando lo presupuestado, lo realizado en la contabilidad y la diferencia entre estos. En la tabla 30 se encuentra la descripción de los campos de este informe.

TABLA 30. Gestión por actividad.

<b>Presupuestado</b>	
<b>Columna</b>	<b>Descripción</b>
Unidad	La unidad con la que se trabaja la cuenta definida en el plan de cuentas.
Cantidad	Cantidad total presupuestada para la cuenta en la actividad seleccionada.
Valor	Monto total presupuestado.
Valor / Unidad	Monto total dividido en la cantidad presupuestada.
<b>Operaciones</b>	
<b>Columna</b>	<b>Descripción</b>
Cantidad	Cantidad de unidades contabilizadas en el ejercicio hasta la fecha indicada por el usuario.
Gastos	Si se trata de una cuenta de gasto el monto contabilizado hasta la fecha se muestra en esta columna.
Ingresos	De ser una cuenta de ingresos el monto contabilizado hasta la fecha se muestran en esta columna.
Valor / Unidad	Monto contabilizado dividido en la cantidad contabilizada.
Desviación	Muestra la diferencia en porcentaje entre lo presupuestado y lo contabilizado.

Finalmente se muestran el total de ingresos, total de gastos, diferencias de inventario, el margen obtenido para la actividad seleccionada y la diferencia entre lo presupuestado y lo obtenido en las operaciones para estos valores. En la figura 44 se aprecia este informe.



situación que existía a comienzos del período contable y la situación que prevalece a la fecha del balance. [Ler07]

Las cuentas del balance están constituidas por todas las clases dentro del rango 100000 – 599999. Se toman los datos del balance inicial y de la contabilidad desde el inicio del ejercicio hasta la fecha de cierre indicada por el usuario para la generación de este informe. Las cuentas del activo se ordenan de mayor a menor liquidez, y las del pasivo de mayor a menor exigibilidad, dividiéndose ambas en clases. Se muestran sus valores monetarios separándolos en activos o pasivos según corresponda indicando los valores al **inicio del ejercicio**, obtenidos del balance inicial, el total contabilizado en las **operaciones** hasta la fecha de cierre indicada por el usuario y el valor resultante al **fin del periodo**.

Al final del Balance, aparece el **resultado neto** del período que puede ser un Beneficio o una Pérdida. En la figura 46 se ve el informe.

Fecha de cierre: 31-07-2008		Mostrar		Vista de impresión		Ver Gráfico		Activos		Pasivos	
Cuenta	Descripción	Inicio	Operaciones	Fin Periodo	Fin Periodo	Operaciones	Inicio				
<b>Cuentas Financieras</b>		<b>Fecha de cierre</b>									
570.000	Caja	500.000	278.549.736	73.892.803			203.156.933				
520.000	Ctas por Pagar		21.500.000	21.500.000			0				
500.300	Credito Banco Santander			0	8.640.000		0	8.640.000			
500.200	Credito BancoEstado		10.000.000	0			0	10.000.000			
500.100	Credito BCI		7.998.000	0			0	7.998.000			
	<b>TOTAL</b>	<b>500.000</b>	<b>316.047.736</b>	<b>95.392.803</b>	<b>8.640.000</b>		<b>203.156.933</b>	<b>26.638.000</b>			
<b>Cuentas de Terceros</b>											
490.000	Cuenta de Regularización		51.691.900	0			51.691.900				
436.200	IVA a pagar		18.392.131		25.718.648		44.110.779				
	<b>TOTAL</b>		<b>70.084.031</b>	<b>0</b>	<b>25.718.648</b>		<b>95.802.679</b>				
<b>Cuentas de Stock</b>		<b>Clases de cuentas</b>									
328.445	Termeros de leche	4.720.000	7.440.000	7.440.000			4.720.000				
328.440	Termeros de leche	12.000.000	9.400.000	9.200.000			12.200.000				
328.430	Vaquillas lechería	7.800.000	10.320.000	10.320.000			7.800.000				
313.300	Stock ensilaje	2.125.000	10.357.000	302.000			12.180.000				
313.230	Paja	300.000	158.400	158.400			300.000				
313.100	Concentrados		200	200			0				
<b>Total Balance</b>		<b>884.935.000</b>	<b>533.033.930</b>	<b>1.018.585.866</b>	<b>918.852.314</b>		<b>433.300.378</b>	<b>884.935.000</b>			
<b>Resultado Beneficio Neto</b>				<b>99.733.552</b>	<b>Resultado neto</b>						

FIGURA 46. Balance general

En la figura 47 se muestran los gráficos de torta generados para este informe, en el primero se aprecian los activos y en el segundo los pasivos.

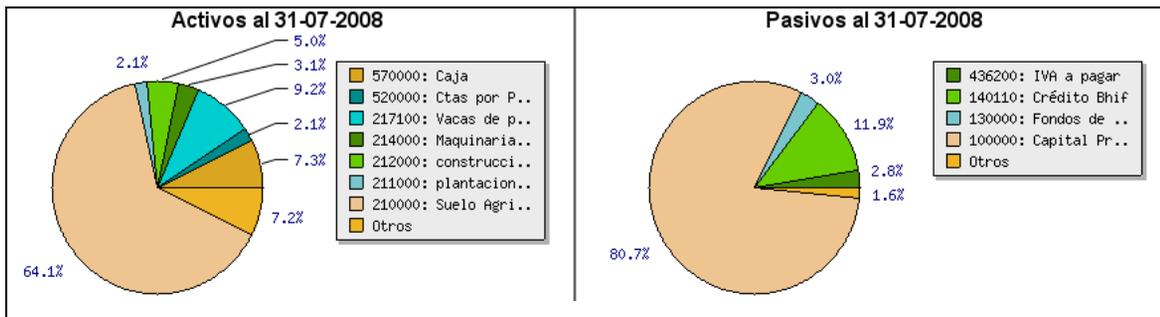


FIGURA 47. Gráfico balance general

### 3.13.4. Resumen de márgenes

Este informe es parte del control presupuestario permanente y constituye una cuenta de gestión recapitulativa por actividad. En este se reproducen los márgenes previstos a comienzos del ejercicio, se señala el total de operaciones realizadas en cada actividad (total de gastos y total de ingresos), así como los márgenes realmente obtenidos en cada una de ellas. [Ler07]

El fin es evidenciar las diferencias entre lo presupuestado y lo realizado tanto por actividad como en lo global, indicando la diferencia entre el margen total, gastos generales y beneficio neto de la explotación obtenidos.

Cada fila de este informe corresponde a una actividad previamente presupuestada en el plan de explotación, excluyéndose únicamente la actividad 0000: Gastos e ingresos generales, cuyo margen es mostrado al calcular el beneficio neto, al final del informe. El usuario puede indicar una fecha de cierre, de esta manera solo se consideran los datos de la contabilidad hasta la fecha indicada. En la tabla 31 se expone una descripción de cada columna perteneciente al resumen de márgenes.

TABLA 31. Resumen de márgenes

Presupuesto	
Columna	Descripción
Margen	Margen presupuestado para la actividad obtenido de los ingresos (700000) menos los gastos (600000)
Margen unitario	Margen total dividido en el tamaño presupuestado para la actividad.
Operaciones	
Columna	Descripción
Gastos	Gastos operacionales registrados para la actividad desde el inicio de ejercicio a la fecha indicada por el usuario.
Ingresos	Ingresos operacionales registrados para la actividad desde el inicio de ejercicio a la fecha indicada por el usuario.
Margen	Margen obtenido para la actividad hasta la fecha indicada.
Margen unitario	Margen dividido por el tamaño presupuestado para la actividad.
Desviación	Desviación entre el margen presupuestado y el obtenido.

Las desviaciones obtenidas, al igual que en los otros informes, se mide en porcentaje como se puede observar en la figura 48.

Fecha de Cierre: 31-07-2008		Mostrar		Vista de impresión		Ver Gráfico		Desviación %	
Fecha de cierre		Presupuesto		Operaciones					
Actividad	Tamaño	Margen	Margen Unitario	Gastos	Ingresos	Margen	Margen Unitario	Des.(%)	
3410 : Pradera mejorada	230 ha	0	0	16.630.997	16.630.997	0	0	0	
2210 : Papas	10 ha	5.348.500	534.850	13.999.350	19.080.000	5.080.650	508.065	-5	
8441 : Crianza	220 Cabezas	6.915.037	31.432	38.043.849	44.030.000	7.986.151	36.301	15	
4100 : Manzanos	5 ha	7.757.125	1.551.425	7.868.000	16.400.000	8.532.000	1.706.400	10	
8430 : Crianza de vaquillas	135 Cabezas	22.839.400	169.181	49.790.055	64.820.000	15.029.945	111.333	-34	
8420 : Lechería	237 Cabezas	84.246.689	355.471	176.480.249	265.695.204	89.214.955	376.434	6	
<b>Totales</b>				<b>Margen Presupuestado</b>	<b>Margen Operacional</b>	<b>Desviación</b>			
<b>MARGEN TOTAL</b>				127.106.751	125.843.701	-1			
<b>(+ -) GASTOS E INGRESOS GENERALES</b>				-30.712.522	-26.110.149	15			
<b>BENEFICIO NETO DE LA EXPLOTACION</b>				96.394.229	99.733.552	3			

FIGURA 48. Resumen de márgenes

Los gráficos generados para este informe, mostrados en la figura 49, son el de gastos e ingresos de gestión, margen presupuestado total, margen operacional total y los márgenes unitarios.

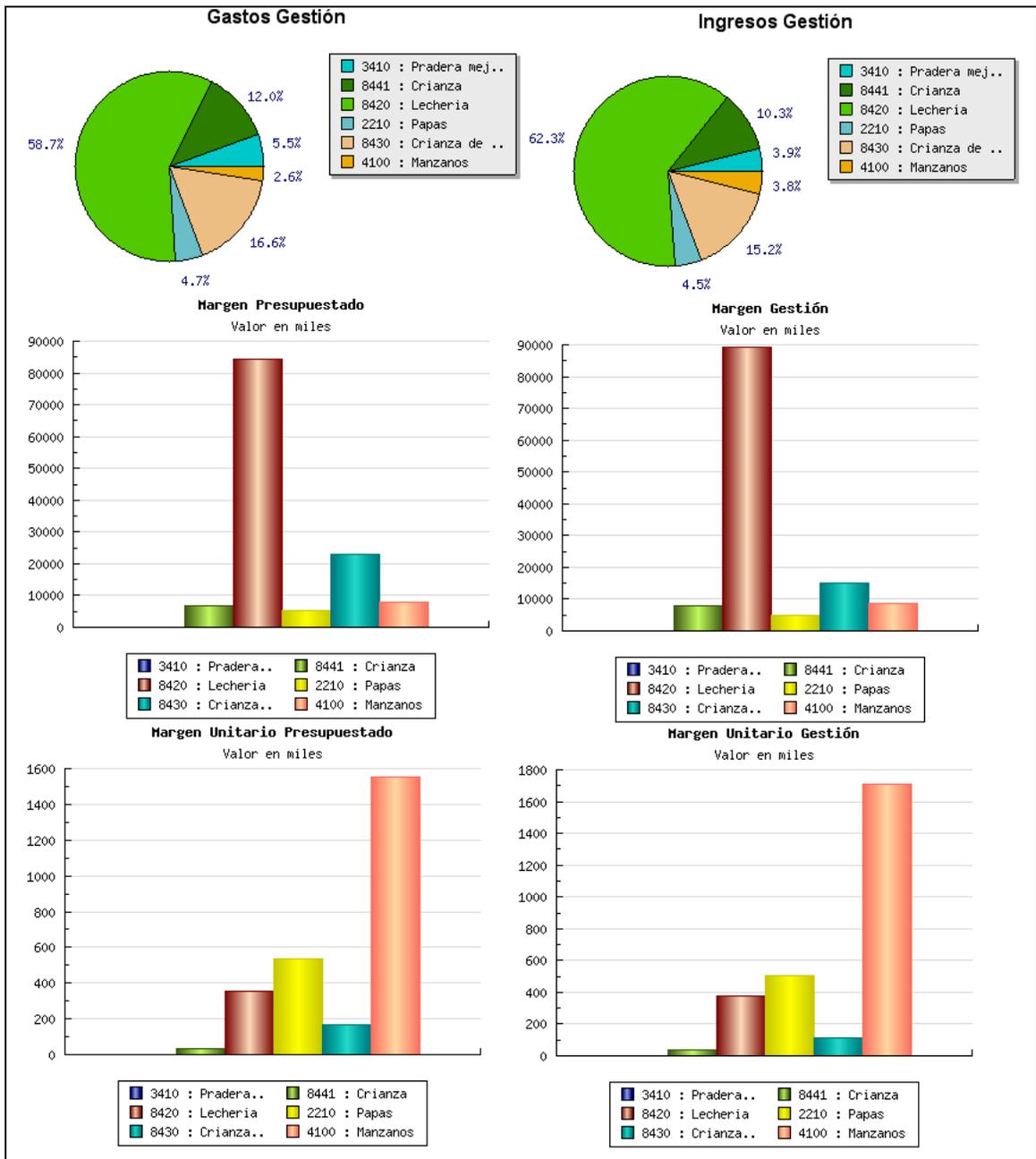


FIGURA 49. Gráficos resumen de márgenes

### 3.13.5. Indicadores financieros

En base a toda la información recopilada en el sistema, se calculan una serie de indicadores financieros que permiten comparar la evolución que ha experimentado la empresa durante el ejercicio en relación con el resultado, productividad económica, eficiencia económica, liquidez, solvencia y rentabilidad. [Ler07]

La recopilación de información necesaria para el desarrollo de este modulo fue realizada por Leandro Ahumada estudiante de Agronomía de la Universidad Austral de Chile. En

la tabla 32 se describen los indicadores financieros calculados por el sistema CONGABOV.

TABLA 32. Indicadores financieros

<b>Resultado</b>	
Ingreso bruto total	Total de ingresos de la explotación
Beneficio neto	Ingresos brutos totales – costos y gastos totales
<b>Productividad económica</b>	
Beneficio Neto / S.A.U	Beneficio neto / superficie agrícola útil
Beneficio Neto / M.O.P	Beneficio neto / mano de obra permanente.
<b>Eficiencia económica</b>	
Beneficio Neto / Ingreso Bruto Total	Divide estos dos valores obteniéndose como resultado un porcentaje.
<b>Liquidez</b>	
Capital de operación	El capital de operación, es determinado en forma estática a nivel del balance, como la diferencia entre el activo circulante (stock + realizable + disponible) y el pasivo circulante o deudas a corto plazo. <b>C.O = (stock + Act. realizable + Act. disponible) – (Pasivo Circulante + Deudas a Corto Plazo)</b>
Necesidad en capital de operación	Esta dada por el stock total, más los créditos otorgados a clientes, menos los créditos obtenidos de los proveedores. <b>N.C.O. (stock total + crédito clientes) – Crédito proveedores(que incluyen créditos operacionales a &lt; 1 año)</b>
Situación de liquidez	La diferencia entre CO y NCO proporciona la SL <b>SL = CO - NCO</b>
<b>Solvencia</b>	
Tasa de solvencia	La TS mide la relación entre capitales propios y deudas totales. <b>Tasa de Solvencia = (Capitales propios / deudas Totales) x 100</b>
Tasa de autonomía financiera	Compara el pasivo no exigible con las deudas a mediano largo plazo, proporcionando una idea sobre la capacidad de endeudamiento de la empresa. <b>Tasa Autonomía Financiera = ( capitales Propios / deudas mediano largo plazo) x 100</b>
Grado de endeudamiento	El GDE, mide la relación entre endeudamiento y pasivo total <b>Grado de Endeudamiento: (Pasivo exigible / Pasivo total )x 100</b>
<b>Rentabilidad</b>	
Capital propio (%)	Permite medir la tasa de interés que producen los fondos propios invertidos por el empresario o por los socios de la empresa, considerando como tales tanto el capital aportado al constituirse la empresa o posteriormente con motivo de una ampliación, así como beneficios obtenidos durante los años de funcionamiento y que permanezcan invertidos en la empresa bajo la forma de reservas. <b>Capital propio = ( Beneficio/ Capital Propio) x 100</b>
Capital total (%)	Esta rentabilidad mide la capacidad efectiva de la empresa para remunerar los capitales puestos a su disposición. <b>Capital total = (Beneficio/ activo Total) x 100</b>

Este informe calcula estos indicadores al comienzo del ejercicio y a la fecha de cierre ingresada por el usuario, para luego indicar la desviación porcentual entre estos valores. En la figura 50 se ve este informe.

Fecha Cierre: 31-07-2008			
Mostrar Indicadores			
Vista de Impresión			
Ver Gráfico			
Indicador	Comienzo Ejercicio	Fin Ejercicio	Desviacion (%)
<b>RESULTADO</b>			
Ingreso Bruto Total	302.410.195	299.934.954	-1
Beneficio Neto	96.394.229	99.733.552	3
<b>PRODUCTIVIDAD ECONÓMICA</b>			
Beneficio Neto / S.A.U	419.105	433.624	3
Beneficio Neto / M.O.P	6.885.302	7.123.825	3
<b>EFICIENCIA ECONÓMICA</b>			
Beneficio Neto / Ingreso Bruto Total (%)	32	33	3
<b>LIQUIDEZ</b>			
Capital de operación	807.000	88.454.755	9.999
Neosidad en Capital de Operación	26.945.000	23.201.952	-14
Liquidez	-26.138.000	65.252.803	350
<b>SOLVENCIA</b>			
Tasa de Solvencia	84	84	0
Tasa de Autonomía financiera	719	788	10
Grado de Endeudamiento	16	16	0
<b>RENTABILIDAD</b>			
Capital Propio(%)	13	13	0
Capital Total(%)	11	10	-9

FIGURA 50. Indicadores financieros

### 3.13.6. Movimientos de caja

Este informe tiene la finalidad de reflejar los egresos e ingresos a caja contabilizados durante las operaciones y flujos totales para cada mes del ejercicio contable. Cada fila corresponde a una cuenta usada durante el ejercicio, agrupadas por clase, primero los egresos y luego los ingresos. En la tabla 33 se describen las columnas de este informe.

TABLA 33. Movimientos de caja

<b>Cuentas</b>	
<b>Columna</b>	<b>Descripción</b>
Ítem	Cuentas utilizadas durante el ejercicio contable las cuales tengan registrado algún egreso o ingreso a caja.
Meses	Meses del ejercicio contable comenzando por el mes de inicio en los cuales se registran los flujos totales de las cuentas para cada uno de ellos.
<b>Totales</b>	
<b>Campo</b>	<b>Descripción</b>
Total ingresos	Se suman todos los saldos de las cuentas que proveyeron a caja y/o banco para el mes especificado.
Total egresos	Se suman todos los saldos de las cuentas que recibieron de caja y/o banco para el mes especificado.
Saldo mensual	Resultado del total de ingresos menos el total de egresos.
Saldo acumulado	Saldo acumulado al término del mes.

En las figuras 51 y 52 se aprecian el informe y sus respectivos gráficos de líneas para el saldo mensual y área para el saldo acumulado.

Item	Ago	Sep	Oct	Nov	Dic	Ene	Feb	Mar	Abr	May	Jun	Jul
<b>Egresos</b>												
Aplicacion de fertil			1.808.500									
Arriendo de maquinaria		1.356.600			1.875.000							
Arriendo maquinaria fumigacion		142.800		95.200								
Comisiones				92.492		85.680		10.963			21.626	
Concentrados	2.918.333	4.199.225	3.523.776	3.205.860	2.625.922	2.651.284	1.474.410	1.176.522	2.862.967	1.892.666	2.371.651	2.519.606
Concentrados												200
Credito BCI											7.998.000	
Credito BancoEstado											10.000.000	
Crédito Bhif					7.500.000							
Ctas por Pagar					21.500.000							
Cuentas de Regularización						143.518				493.850	544.187	50.698.997
Electricidad			431.538	197.588	251.650	185.536	181.130	169.596	225.670	186.354		
<b>Total Ingresos</b>												
	15.124.376	22.560.376	19.716.515	24.873.317	20.791.328	14.866.553	11.037.769	17.678.147	44.838.161	52.284.849	17.273.742	16.204.605
<b>Total Egresos</b>												
	19.197.679	16.198.044	17.287.164	8.405.197	38.085.398	9.936.609	8.343.895	13.815.683	10.570.516	9.144.886	26.076.661	25.687.022
<b>Saldo Mensual</b>												
	-4.073.303	6.362.332	2.429.351	16.268.120	-17.294.072	4.929.944	2.693.874	3.862.464	34.267.645	43.139.963	-8.802.919	-9.482.417
<b>Saldo Acumulado</b>												
	-4.073.303	2.289.029	4.718.380	20.986.500	3.692.428	8.622.372	11.316.246	15.178.710	49.446.355	92.586.318	83.783.399	74.300.982

FIGURA 51. Movimientos de caja

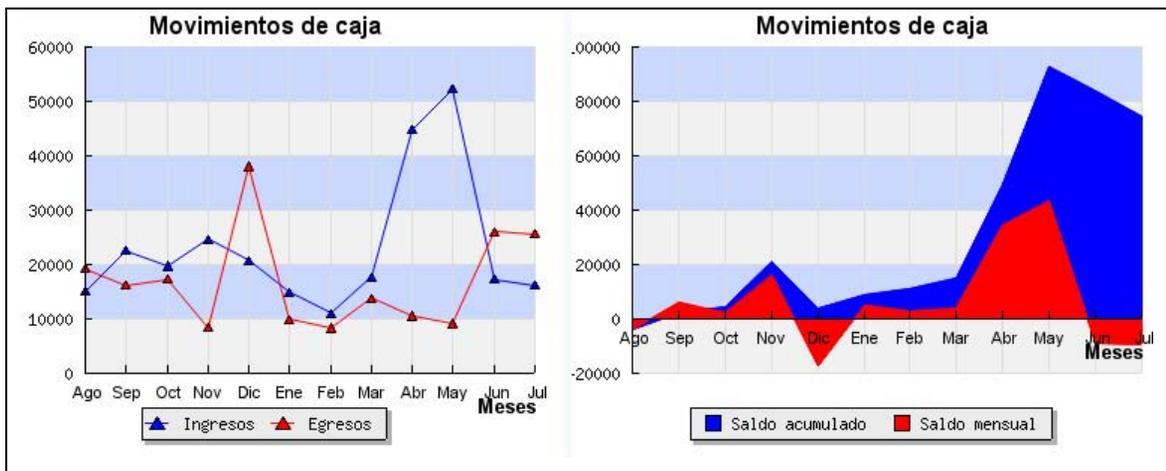


FIGURA 52. Gráfico movimientos de caja

### 3.13.7. Control de caja permanente

El fin de este informe es mostrar las diferencias entre lo presupuestado y lo contabilizado para las cuentas de egresos (600000) e ingresos (700000) en su totalidad, independiente de la actividad. De esta manera para cada una de estas cuentas se obtienen el total de montos operados en la contabilidad hasta la fecha de cierre del informe indicada por el usuario y el total presupuestado para la cuenta en todas las actividades.

Las primeras columnas muestran el código y nombre de la cuenta, nombre de la cuenta, unidad en que se mide, cantidad presupuestada, valor total presupuestado, cantidad contabilizada en las operaciones, monto total de las operaciones y finalmente la desviación entre el total presupuestado y el operacional. Esto se puede apreciar en la figura 53.

Fecha de Cierre: 31-07-2008    Mostrar    Vista de impresión    Ver Gráfico    **Desviación**

Código	Descripción	Unidad	Presupuesto		Operaciones		Desv.
			Cantidad	Valor	Cantidad	Valor	
<b>EGRESOS</b>							
601000	Aplicacion de fertil	S	0	1.500.000	0	1.350.000	-10
601010	Salitre sodico	kg	0	0	21.520	1.644.400	100
601050	Superfosfato triple	kg	9.600	1.832.000	41.900	6.240.620	282
601081	Fertilizante Mezcla 1	kg	0	0	60.000	7.565.598	100
601082	Fertilizante Mezcla 2	kg	71.400	12.138.000	0	0	-100
601083	Supernitro	kg	17.570	2.899.050	17.050	1.842.625	-36
601084	Muriato de potasio	kg	2.400	384.000	2.500	405.000	5
601100	Semillas y plantas	kg	0	0	755	572.344	100
601104	papa semilla	S	30.000	6.000.000	30.000	6.000.000	0
601200	Pesticidas	S	0	1.381.500	0	854.000	-38
601220	Herbicidas	lt	40	378.000	33	288.750	-24
601241	Raticidas	S	0	0	3	36.350	100
601302	Sustituto lacteo	kg	3.960	2.851.200	4.950	2.934.600	3
601310	Concentrados	kg	297.440	30.494.000	263.673	26.405.387	-13
601321	Heno de alfalfa	kg	76.515,9	5.356.113	54.910	3.748.916	-30
601323	Paja	Fardos	1.120	336.000	4.025	657.500	96
<b>TOTAL INGRESOS</b>				239.314.445		232.161.994	-3
<b>TOTAL EGRESOS</b>				130.350.330		127.746.253	-2
<b>SALDO</b>				108.964.115		104.415.741	-4

FIGURA 53. Control de caja permanente

Finalmente se calculan el total de ingresos (sumatoria de los totales de las cuentas 700000), total de egresos (sumatoria de los totales de las cuentas 600000) y de la resta de estos dos valores se obtiene el saldo tanto para lo presupuestado como para las operaciones calculando así la desviación para estos montos.

En la figura 54 se muestran los cuatro gráficos generados para este informe, correspondiendo a los egresos e ingresos del presupuesto y de la gestión.

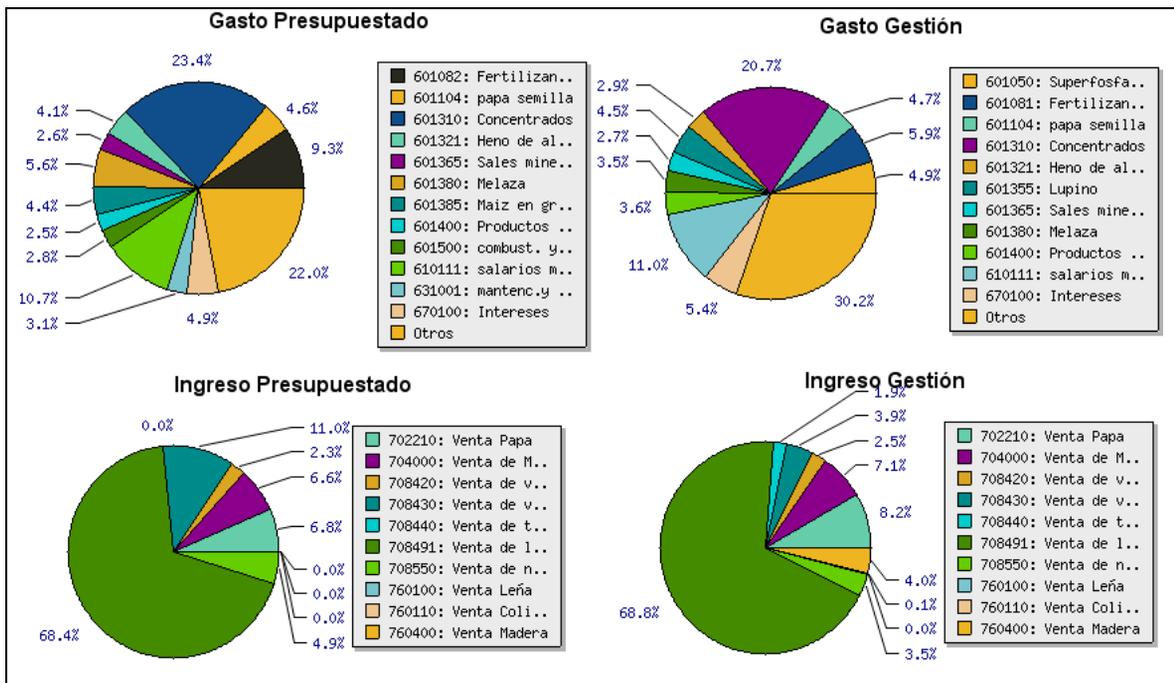


FIGURA 54. Gráfico control de caja

### 3.13.8. Costo litro de leche

El software CONGABOV esta enfocado a usuarios que tengan explotaciones agropecuarias en la zona y en las que su principal actividad sea la lechería, debido a esto uno de los módulos se centra específicamente en esta actividad permitiendo calcular el costo total de producción por litro de leche.

La recopilación de información teórica para generar las formulas a desarrollar para obtener el costo por litro de leche estuvo a cargo del estudiante de Agronomía de la Universidad Austral de Chile Jorge Vera el cual obtuvo estos datos de un documento llamado “Costo de producción del litro de leche e indicadores económicos del rubro lechería en diez predios del país” publicado por el departamento de informática agrícola de esta universidad.

A continuación, se indican los conceptos y formulas usados para el cálculo del costo del litro de leche.

- **Rubro lechería:** Vacas masa de lechería. [Ler84]
- **Costos específicos:** Costos claramente definidos para un rubro en particular. [Ler84]

La actividad lechería corresponde a la clase 8420 en el plan de actividades, el sistema agrupa todos los movimientos que tengan relación con esta y generen gastos.

- **Costos no específicos:** Costos que no pueden ser claramente definidos debiendo ser asignados a cada rubro. [Ler84].

Todos estos costos se asocian a la actividad gastos e ingresos generales (0000). Para calcular el porcentaje de costos de esta actividad que corresponde a la lechería se obtiene su total de ventas y se calcula a que porcentaje del total de ventas de la explotación corresponde obteniéndose de esta forma el costo no específico.

- **Costos directos:** Costos específicos del rubro mas costos no específicos asignados. [Ler84]
- **Costos indirectos:** Costos que se originan por concepto de intereses y depreciaciones. [Ler84] Deben distinguirse aquellos costos directos que son específicos de la lechería y los costos indirectos no específicos que deberán ser

asignados usando el criterio del porcentaje de ventas de la actividad lechería en la explotación. En la tabla 34 un ejemplo de estos.

TABLA 34. Ejemplo Costos indirectos del rubro lechería

<b>Capitales y costos indirectos</b>	<b>Valor total</b>	<b>Depreciación</b>
Suelo	Valor activo	No se deprecia
Vacas en producción	Valor activo	No se deprecia
<b>Construcciones</b>		
Construcciones específicas para lechería	Valor activo	Valor total * 0.3
Construcciones no específicas para lechería	Valor activo * % ventas lechería	Valor total * 0.3
<b>Maquinaria y equipos</b>		
Maquinaria y equipos específicos de lechería	Valor activo	Valor total * 0.12
Maquinaria y equipos NO específicos de lechería	Valor activo * % ventas lechería	Valor total * 0.12
<b>Totales</b>	<b>Total</b>	<b>Total depreciaciones</b>
<b>Intereses</b>	<b>Total * Costo de oportunidad</b>	
<b>Total costos indirectos</b>	<b>Intereses + total depreciaciones</b>	

- **Costos de producción de leche:** (Costos directos lechería + costos indirectos – (ingresos por venta de vacas + terneros + diferencias de inventario)). [Ler84]
- **Costo litro leche:** (Costo de producción de leche) / (cantidad total de leche producida). [Ler84]

### 3.14. Herramientas para cerrar el ejercicio

Finalmente al término del ejercicio contable el sistema provee una serie de herramientas y procedimientos para poder finalizar correctamente la contabilidad y poder comenzar un nuevo periodo contable.

#### 3.14.1. Cálculo de revalorizaciones o corrección monetaria

Las revalorizaciones sirven para mantener los activos sin cambios reales frente al IPC el cual es ingresado mes a mes por el administrador del sistema. En el CONGABOV estas representan un tipo de movimiento siendo ingresadas de forma manual por el usuario o de manera automática mediante la herramienta que provee el sistema en el módulo de cierre anual. En la tabla 35 se ve un ejemplo de este tipo de movimiento.

TABLA 35. Ejemplo movimiento de revalorización

Glosa	Actividad	Provee	Recibe	Monto
Revalorización	0000 : Gastos e Ingresos Generales	130000 : Fondos de Revalorización	210000 : Suelo Agrícola	21.030.466
Revalorización	0000 : Gastos e Ingresos Generales	130000 : Fondos de Revalorización	212000 : construcciones	3.549.200
Revalorización	0000 : Gastos e Ingresos Generales	130000 : Fondos de Revalorización	213000 : Instalaciones	858.800
Revalorización	0000 : Gastos e Ingresos Generales	130000 : Fondos de Revalorización	214000 : Maquinarias y Equipo	2.248.000

Como se puede apreciar las revalorizaciones se asocian a la actividad gastos e ingresos generales (0000), siempre provee la cuenta de revalorización (130000) y recibe una cuenta del activo (200000). Este tipo movimiento tiene que realizarse antes de calcular las depreciaciones y se puede hacer de forma automática solo si el usuario no ha ingresado de forma manual alguna revalorización.

### 3.14.2. Cálculo de depreciaciones

Al igual que las revalorizaciones, en el sistema CONGABOV se pueden calcular las depreciaciones lineales de manera automática o de forma manual si el usuario desea ser mas específico. Estas deben efectuarse al término del ejercicio y después de las revalorizaciones.

Todas las clases del activo poseen propiedades que indican si la cuenta se deprecia y a cuantos años corresponde su vida útil (depreciación lineal). Con este valor, se calculan los montos de depreciación que tuvo el activo durante el periodo contable. En la tabla 36 un ejemplo de depreciaciones.

TABLA 36. Ejemplo movimiento depreciaciones

Glosa	Actividad	Provee	Recibe	Monto
Depreciación	0000 : Gastos e ingresos generales	212800 : Depreciación acumulada construcciones	680200 : Depreciaciones construcciones	2.395.710
Depreciación	0000 : Gastos e ingresos generales	213800 : Depreciación acumulada instalaciones	680300 : Depreciaciones instalaciones	579.690
Depreciación	0000 : Gastos e ingresos generales	214800 : Depreciación acumulada maquinaria y equipo	680400 : Depreciaciones maquinaria y equipos	3.034.800

Estos movimientos se asocian a la actividad gastos e ingresos generales (0000), proveen cuentas del activo especiales para este efecto, que registran la depreciación acumulada

de la clase (212800: Depreciación acumulada construcciones) y reciben cuentas de gastos, también confeccionadas para este efecto (680200: Depreciaciones construcciones). Este tipo de movimiento solo se puede realizar de manera automática si no se han ingresado depreciaciones de manera manual.

### 3.14.3. Cierre del ejercicio

Finalmente, después de haber efectuado las revalorizaciones y depreciaciones, corresponde cerrar el ejercicio y comenzar uno nuevo. Esta etapa se divide en tres procesos explicados a continuación.

- **Bloqueo del ejercicio en curso:** Esto consiste en bloquear la aplicación para que no se puedan seguir agregando datos al ejercicio actual, así se bloquean todos los componentes de la interfaz para este efecto.
- **Creación de nuevo ejercicio:** Se habilita un nuevo ejercicio para la explotación y se migran los datos como comienzo y fin del ejercicio, plan de cuentas y actividades. De esta manera el nuevo ejercicio posee los mismos parámetros que el anterior pudiéndose modificar estos si el usuario lo requiere.
- **Creación de balance inicial:** Se calcula automáticamente el balance inicial para el nuevo ejercicio. La fórmula de cálculo de este (de forma muy resumida) es la siguiente;  
**Balance inicial n+1 = Balance inicial n + contabilidad en el periodo n.**
- **Creación de presupuestos:** Para facilitar la creación del plan de explotación al usuario se copian todos los presupuestos creados por este del ejercicio recién cerrado al nuevo.

Todos los datos e informes del ejercicio recién cerrado se pueden seguir revisando, el único cambio es que el sistema no permite modificaciones a estos.

### 3.14.4. Reabrir ejercicio

Como se acaba de comentar al cerrarse un ejercicio a este no se le pueden agregar datos ya que según la fórmula para la creación del balance inicial la contabilidad del ejercicio anterior influye directamente sobre este.

Puede ocurrir el caso de que un usuario necesite agregar movimientos al ejercicio cerrado, debido a esto se habilitó una funcionalidad en el sistema, que permite reabrir este, borrando todos los datos del ejercicio actual en curso (recién creado). Esto se hace para que los datos del balance no queden con errores debido a la modificación de la contabilidad del ejercicio anterior.

Claramente esta operación hay que realizarla en casos en que sea necesario y ojala antes de ingresar movimientos al nuevo ejercicio, para que de esta manera se borre solo la información creada automáticamente con la herramienta de cierre.

## CAPÍTULO IV

### 4. ANÁLISIS DE LAS TECNOLOGÍAS USADAS PARA LA CREACIÓN DE APLICACIONES AJAX

Luego de haber estudiado las tecnologías usadas en la creación de RIA`s con AJAX y usado un conjunto de estas para desarrollar un *software* de gestión contable para el rubro agropecuario, se procederá a hacer un análisis indicando ventajas, desventajas y recomendaciones para obtener el máximo provecho de estas.

#### 4.1. Ventajas del uso de AJAX

AJAX es un enfoque impresionante del uso de distintas tecnologías, que nos está conduciendo a una nueva generación de aplicaciones *Web*, las cuales poseen muchas ventajas comparativas con las antiguas páginas que los internautas estaban acostumbrados a ver.

##### 4.1.1. Usabilidad

Según la definición de Jakob Nielsen<sup>45</sup>, la usabilidad es el atributo que mide lo fáciles que son de usar las interfaces *Web*. Esta va desde la forma en que se presenta una aplicación al usuario, haciéndola mas agradable y fácil de entender para el público objetivo, hasta los componentes gráficos a nivel de interface que posee para facilitar su uso.

Debido a la gran cantidad de datos que deberá manejar el *software* CONGABOV, este tiene que contar con algunos componentes en la interfaz de usuario, para facilitar su manejo. Estos elementos poco usuales en aplicaciones de Internet no vienen incorporados en el paquete de *widjets* estándar de GWT, siendo desarrollados a partir de la mezcla de estos.

---

<sup>45</sup> Nacido en 1957, en Copenhague, Dinamarca, es una de las personas más respetadas en el ámbito mundial sobre usabilidad en la web.

- **Cabeceras siempre visibles en grillas:** En la mayoría de las grillas, la gran cantidad de datos, hace necesario el tener que usar la barra de desplazamiento (*scroll*), siendo incomodo el perder el nombre de las columnas. Debido a esto se desarrollaron cabeceras que siempre están visibles en las grillas, como se puede apreciar en la figura 55.

	Grupo Cuentas	Descripción	Total	/Ha	%
4.-	601500	Gastos en Combustibles y Lubricantes	2.600.000	7.182	4,41
5.-	608000	Gastos en Reposición de Animales	25.265.000	69.793	42,84
6.-	610000	Gastos de Personal	5.632.000	15.558	9,55
7.-	620000	Gastos en Tasas e Impuestos	2.700.000	7.459	4,58
8.-	630000	Gastos en Arriendos	210.000	580	0,36
9.-	631000	Gastos en Mantenciones y Reparaciones	600.000	1.657	1,02
10.-	634000	Gastos en Servicios Pagados por la Empresa	2.580.000	7.127	4,38

FIGURA 55. Cabeceras siempre visibles en grillas.

- **Paginación de la información:** Cuando los informes son demasiado extensos (mas de 100 filas por ejemplo), se incrementa en gran medida el tiempo de conversión del formato XML, haciendo tedioso el uso del sistema. Para evitar esto, se hizo indispensable dividir la información en páginas, poniendo un tope a los datos obtenidos y reduciendo los tiempos de espera como se ve en la figura 56.

Mostrar: 30 Total: 480 -2-3-4-5-6-7-8-9-10-11-12-13-14-15-16 Desde: 271 Hasta: 300

Cantidad a mostrar Total de datos Página actual Rango mostrado

FIGURA 56. Paginación.

- **Ordenamiento por columnas:** Para mejorar la búsqueda y comprensión de la información, se habilitaron ordenamientos por columnas en las grillas de datos e informes, con excepción de los que por su naturaleza no admiten el reordenamiento por perder el sentido de lo que muestran (algunas columnas llevan totales acumulados dependiendo de la fecha de ingreso de la información). Esto se ve en la figura 57.

Unidad	Total
kg	5.400.000
kg	5.500.000
\$	7.525.750
ton	9.570.000
kg	11.824.100
ton	15.735.000
qqm	18.000.000
\$	65.100.000
lt	163.755.345

Unidad	Total
lt	163.755.345
\$	65.100.000
qqm	18.000.000
ton	15.735.000
kg	11.824.100
ton	9.570.000
\$	7.525.750
kg	5.500.000
kg	5.400.000

FIGURA 57. Ordenamiento de grillas

- **Suggest list:** Algunas de las listas de la aplicación, contienen demasiados datos, haciendo difícil encontrar lo buscado (listas que despliegan las cuentas y actividades). Para esto se construyo un componente especial, mostrado en la figura 58, en el cual el usuario puede escribir una parte de la información y este *widget* le sugiere datos que tengan alguna semejanza con lo ingresado.

FIGURA 58. Suggest list

- **Controles de ayuda:** Todas las pantallas tienen un control de ayuda para el usuario, siendo todas específicas según de donde se soliciten. Esta información es ingresada por el administrador del sistema. En la figura 59, un ejemplo de este componente.

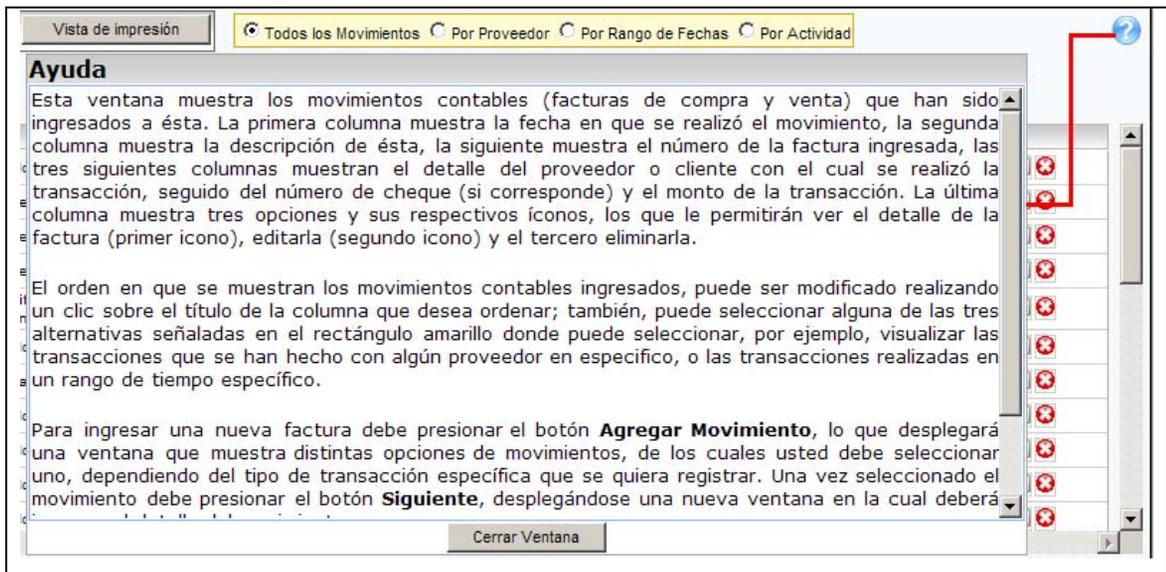


FIGURA 59. Control de ayuda

#### 4.1.2. Basado en estándares

AJAX se compone básicamente de JavaScript, HTML, XML y CSS, los cuales son todos estándares propiciados por la W3C, motivo por el cual los navegadores implementan estos lenguajes según las normas indicadas por este organismo. Sin embargo el objeto XMLHttpRequest, con el cual se logran hacer las peticiones asíncronas no es todavía un estándar oficial de la W3C, habiendo distintas implementaciones de este para cada navegador. Esto obliga a los desarrolladores a implementar las solicitudes al servidor según el navegador que este usando el usuario o bien usar *frameworks* o librerías que traten este problema.

La W3C ya ha terminado el 15 de abril de 2008 la primera revisión de una versión oficial para la implementación del objeto XMLHttpRequest, así que es cosa de tiempo para que todas las tecnologías usadas en la técnica AJAX sean estándares aceptados por todos los navegadores modernos.

Por el momento es recomendable usar *frameworks* que provean funcionalidades para solucionar el tema de la compatibilidad, disminuyendo así el tiempo de desarrollo. En el caso del software CONGABOV se uso GWT, el cual, al momento de compilar los fuentes JAVA a JavaScript, se preocupa internamente de que las solicitudes asíncronas funcionen en la mayoría de los navegadores.

### **4.1.3. No requiere plugins adicionales**

AJAX es solo un enfoque en la manera de usar las tecnologías que se venían viendo en aplicaciones *Web* desde años atrás, estas están soportadas por la mayoría de los navegadores modernos. Debido a esto los usuarios no tienen que descargar nada adicional para poder ver los sitios que utilicen esta técnica, a diferencia de las aplicaciones desarrolladas en Flex, las cuales necesitan el *plugin* de Flash instalado en el navegador para poder acceder a estas, o aquellas que usan Applets de JAVA, para las cuales hay que tener instalada la JVM en el computador del usuario.

### **4.1.4. Reducción del tamaño de la información intercambiada**

Al momento en que el usuario se conecta a una aplicación AJAX, lo primero que hace es descargar los fuentes en JavaScript, en los cuales reside la lógica de la interfaz (controles, validaciones, etc.) y CSS, los cuales dan el estilo gráfico, si bien estos archivos por lo general pueden llegar a pesar bastante, una vez descargados se guardan en la carpeta temporales de internet (en caso de que el usuario tenga esta opción activada), logrando con esto que solo se vuelvan a descargar en caso de que hayan sufrido modificaciones.

Una vez descargados los fuentes JavaScript en el computador, solo se envían solicitudes HTTP por la información que desea visualizarse, la cual viene en algún formato predefinido como XML, en el caso del CONGABOV, el cual al no tener toda la lógica de la interfaz ni estilo gráfico, tiene menor tamaño que si hubieran sido construidos completamente (lógica, diseño e información) en el lado del servidor.

## **4.2.Desventajas del uso de AJAX**

A pesar de lo espectaculares que pueden llegar a ser las aplicaciones desarrolladas con este enfoque, existen varias consideraciones que hay que tomar en cuenta antes de emprender un desarrollo en AJAX.

### 4.2.1. Incompatibilidades entre navegadores

Como ya se ha venido nombrando en apartados anteriores, el mayor problema de compatibilidad radica en el objeto XMLHttpRequest, debido a que todavía no existe una versión oficial de su implementación emitida por la W3C. Esto es un problema si se pretende desarrollar una *Web* AJAX desde cero, pero se pueden usar *frameworks* como Prototype, Dojo, Rico, GWT, etc. Los cuales abstraen al programador de este inconveniente, manejando internamente el tema de la compatibilidad.

Debido a lo reciente de estas técnicas y adopción de las tecnologías usadas para crear RIA`s, se tiene que no todas las versiones de los navegadores soportan estas funcionalidades, obligando de esta manera a los usuarios a actualizarse para poder acceder a aplicaciones AJAX.

Otro punto importante es que cuando el HTML no estaba bien normalizado por la W3C, existían etiquetas creadas por los mismos programadores de navegadores para sus productos, de estos tiempos vienen los famosos *LAYER* de Netscape e *IFRAME* de Explorer, con los cuales se lograron los primeros efectos similares a lo que se conoce hoy en día por AJAX. Esto se ha superado en sus nuevas versiones, debido al elemento regulador de la W3C, pero todavía quedan aspectos gráficos a considerar como por ejemplo; como norma se establece que la etiqueta *TABLE* debe soportar el atributo *CELLSPACING* para que se pueda definir el espacio entre celdas. Pero los estándares no definen que valor lleva ese atributo por defecto, por lo tanto, si no se especifica uno, los navegadores son libres de poner cualquiera. En la figura 60 se aprecian este tipo de diferencias entre Explorer y Firefox en la aplicación CONGABOV.

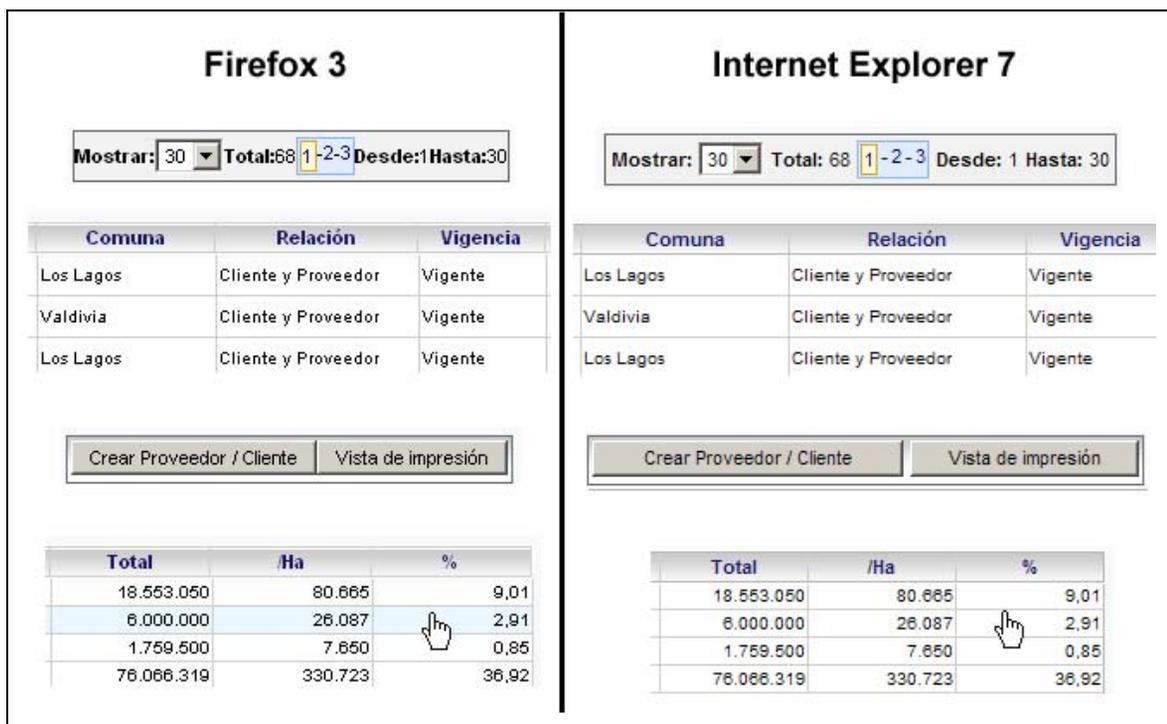


FIGURA 60. Ejemplo de diferencias entre navegadores

La mayoría de estas son de índole gráfico, en el software CONGABOV estas no se trataron, ya que no era requerimiento de la aplicación ser estéticamente atractiva (claro que debía cumplir con un mínimo aceptable), pero existen productos en los que este aspecto hay que considerarlo a fondo, teniéndose que invertir mas tiempo en el desarrollo del CSS.

El ultimo ejemplo de la imagen, es de aspecto funcional, en algunas grillas se implemento que al pasar el mouse por encima de una fila, esta cambiara su color (estilo CSS), para indicar al usuario en donde se encuentra y facilitar la lectura de la información, sin embargo al momento de la compilación solo funciono en Firefox, no observándose cambio alguno en Internet Explorer.

#### 4.2.2. Pérdida de funcionalidades comunes de los navegadores

Las funciones JavaScript no generan cambios en la URL, motivo por el cual el navegador no puede recordar las distintas páginas en que se encuentra una aplicación AJAX, perdiéndose de esta manera las funcionalidades ATRÁS, ADELANTE y ACTUALIZAR, comúnmente usadas por los usuarios. Esto quiere decir que los botones no funcionan dentro de la aplicación, así al querer retroceder se volverá al sitio *Web*

previamente cargado por el usuario y al actualizar, se recargará nuevamente la *Web* AJAX en su punto de partida, pero no recordando en que pantalla de esta se encontraba.

De la misma forma, lo antes comentado afecta a la función FAVORITOS de los navegadores, pudiéndose almacenar en esta solo el punto de entrada a la aplicación, ya que no hay una URL específica para un determinado estado de esta.

Otro ejemplo es que en la aplicación CONGABOV, los datos se despliegan en una grilla que posee una *Scrollbar*, de esta manera las pantallas tienen una altura fija, además de esto se tienen *widgets* para la manipulación y paginación de la información, como se puede apreciar en la figura 61.

	Fecha	Descripción	Factura	RUT	Razón Social	Comuna	Cheque	
1	31-07-2008	Movimientos Internos	30					🔍 📄 🗑️
2	30-07-2008	Depreciación de Activos del Inventario	9.000	6819677-9	Fundo Calfuco	Los Lagos		🔍 📄 🗑️
3	30-07-2008	Depreciación de Activos del Inventario	10					🔍 📄 🗑️
4	30-07-2008	Revalorización	9					🔍 📄 🗑️
5	30-07-2008	Diferencias de Inventario según Categorías de Animales	5					🔍 📄 🗑️
6	30-07-2008	Movimientos Internos	15					🔍 📄 🗑️
7	30-07-2008	Pago de Salarios y Leyes Sociales	7					🔍 📄 🗑️
8	30-07-2008	Movimientos Internos	4					🔍 📄 🗑️
9	30-07-2008	Movimientos Internos	3					🔍 📄 🗑️
10	30-07-2008	Movimientos Internos	1.016					🔍 📄 🗑️
11	30-07-2008	Movimientos Internos	2					🔍 📄 🗑️

Mostrar: 30 Total: 480 1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16 Desde: 1 Hasta: 30

FIGURA 61. Widgets en una pantalla

El problema con esto es que si el usuario desea imprimir los datos usando el control del navegador, aparecen todos los componentes gráficos y no se mostraría toda la información por la *Scrollbar*. Debido al contexto financiero de esta aplicación, se da a entender que es un requisito de esta, la impresión de todos los informes e información ingresada al sistema, motivo por el cual se tuvo que desarrollar un módulo que transformara todas las pantallas a un formato imprimible. Esto se aprecia en la figura 62.

Imprimir Cerrar Ventana

(Explotación: Predio Respaldo | Ejercicio:1)

**Balance Inicial**

Información a imprimir

Controles

Datos de la explotación

Cuenta	Descripción	Tipo	Monto	Cantidad	Unidad
100000	Capital Propio	Crédito	741.297.000	0	\$
140110	Crédito Bhif	Crédito	117.000.000		\$
210000	Suelo Agrícola	Débito	631.800.000	351	ha
211000	plantaciones permanentes	Débito	22.320.000	5	ha
212000	construcciones	Débito	44.365.000		\$
213000	Instalaciones	Débito	10.735.000		\$
214000	Maquinarias y Equipo	Débito	28.100.000		\$
214200	Equipo de riego	Débito	4.350.000	5	ha
217100	Vacas de produccion	Débito	89.600.000	224	Cabezas
217140	Vaquillas preñadas	Débito	13.500.000	27	Cabezas
217145	vaquillas encaste	Débito	9.000.000	30	Cabezas
217160	Toros	Débito	3.000.000	6	Cabezas
217200	Caballos de trabajo	Débito	720.000	6	Cabezas
313230	Paja	Débito	300.000	1.000	Fardos
313300	Stock ensilaje	Débito	2.125.000	240	ton
328430	Vaquillas lechería	Débito	7.800.000	65	Cabezas
328440	Terneros de leche	Débito	12.000.000	120	Cabezas
328445	Terneras de leche	Débito	4.720.000	59	Cabezas
500100	Credito BCI	Crédito	7.998.000		\$
500200	Credito BancoEstado	Crédito	10.000.000		\$

FIGURA 62. Versión imprimible

En esta versión se elimina todo el formato CSS y controles adicionales, dejando solamente el texto básico, para así no tener problemas en la impresión. Cabe destacar que el hecho de haber separado las lógicas del cliente y del negocio facilito bastante la creación de este módulo, debido a que solo se cambio la interfaz gráfica de los informes, desarrollando un grupo de funciones que reciben el resultado de una consulta SQL y generan una tabla en base a esta.

### 4.2.3. Limitado por seguridad del navegador

Esta es una desventaja comparativa con las aplicaciones de escritorio y es un problema general a todas las soluciones para crear RIA's que hoy existen, ya que al ejecutarse estas dentro de un navegador, se encuentran encapsuladas en este entorno, no pudiendo acceder ni modificar archivos que residan en la maquina cliente.

Es sin duda una muy buena práctica de seguridad, ya que disminuye en gran medida los ataques a los equipos de los internautas, pero limita las posibilidades de desarrollo para aplicaciones en la *Web*. Por ejemplo, no se pueden guardar los estados en que se encuentra un usuario en la aplicación una vez cerrado el navegador, o se desconecte por problemas en su conexión a *Internet*.

Se ha querido comentar este punto, debido a que se han desarrollado soluciones para poder guardar los estados en que se encuentra un usuario, e incluso que permiten trabajar de modo *offline*.

Google ha liberado una API para los navegadores llamada Gears, la cual dentro de muchas otras funcionalidades, posee las opciones de permitir almacenar en caché, proporcionar recursos de aplicaciones (HTML, JavaScript, imágenes, etc.) y almacenar datos en una base de datos relacional, en la que se pueden realizar todo tipo de búsquedas, todo esto de manera local en el computador del cliente, permitiendo trabajar en modo *offline*, una vez que se vuelve a conectar el usuario, sus datos son enviados al servidor. [URL29]

Para lograr resultados similares, Adobe ha desarrollado, para su plataforma FLEX, la solución ADOBE AIR, la cual no se ejecuta en un navegador, lo hace en el *plugin* de Flash, de manera local en la maquina, con lo cual sus aplicaciones ya no están restringidas a las políticas de seguridad de los navegadores *Web*. [URL30]

#### **4.2.4. Aumento excesivo en el uso de JavaScript**

EL primer problema con el JavaScript es que todos los navegadores proveen la opción de deshabilitar su uso como medida de seguridad. Esto no se tratara en mayor detalle, debido a que esta decisión corresponde a los usuarios.

En los anteriores modelos de creación de aplicaciones *Web*, desarrolladas completamente en lenguajes ejecutados en el lado del servidor, la mayor carga de procesamiento claramente se encuentra en este. Ahora con AJAX y el uso de JavaScript, el cliente es quien ejecuta el mayor número de funciones y se lleva gran parte de esta carga, lo que puede aumentar el tiempo de espera del usuario. Ósea irónicamente el tiempo ahorrado en la disminución del tráfico de red (por disminución del tamaño de la información intercambiada), se puede perder en la ejecución del JavaScript.

Debido a esto, es necesario asegurarse que las funciones a ejecutarse en el lado del cliente, no se demoren un tiempo excesivo. No olvidar que todas las personas no poseen equipos de última generación, así que es una buena práctica el tratar de averiguar las

capacidades promedio de los computadores del público objetivo al que apunta la aplicación a desarrollar, en el caso de que esto sea factible.

En la aplicación CONGABOV, se escogió el XML como lenguaje de intercambio de datos entre el servidor y el cliente. Este al ser recibido, tiene que ser convertido a un lenguaje entendible por JavaScript, proceso que al aumentar el tamaño del árbol XML, aumenta en tiempo. Por la naturaleza contable de la aplicación, la cantidad de nodos obtenidos, en ocasiones es del orden de los cientos, provocando esto que los informes demoren mucho en generarse y obligando así a traer estos por partes mediante paginación de la data.

#### 4.2.5. Aumento de solicitudes HTTP al servidor

Al desarrollar aplicaciones con el enfoque AJAX, se deja de pensar en esta como una página, ya no se cargan pantallas completas, generándose todos sus componentes gráficos en el lado del servidor, se tiende a identificar fuentes de datos que mostrar al usuario y solicitudes HTTP al servidor para obtenerlas. En la figura 63 se aprecia esto.

The screenshot shows a web application interface for 'Compras' (Purchases). It features several input fields and a table. Red arrows point to specific elements, labeled 1 through 7, representing different data sources:

- 1.- Fecha: Points to the date field 'Fecha: 30-05-2008'.
- 2.- Proveedores: Points to the dropdown menu 'Proveedor / Cliente: 91253000-0 : Anasac'.
- 3.- Tipos de IVA: Points to the 'Impuestos' section with checkboxes for IVA (19%), ILA (8%), Impuesto Molino (7%), and Impuesto Feria Ganado (8%).
- 4.- Unidades: Points to the 'Unidad: kg' field.
- 5.- Actividades: Points to the 'Actividad: 0000: Gastos e Ingresos Generales' dropdown.
- 6.- Lista de cuentas: Points to the 'Cuenta que Recibe: 601084: Muriato de potasio (kg)' dropdown.
- 7.- Líneas de detalle: Points to the table below.

Glosa	Actividad	Cta Provee	Cta Recibe	Un.	Ctd.	Monto	Impuesto	Esp. Rec.	Esp. No Rec.
Anasac	0000 : Gastos e Ingresos Generales	570000 : Caja	490000 : Cuenta de Regularización	\$		415.000	78.850	0	0
Anasac	8441 : Crianza	490000 : Cuenta de Regularización	601302 : Sustituto lacteo	kg	500	310.000	0	0	0
Anasac	8420 : Lecheria	490000 : Cuenta de Regularización	601365 : Sales minerales	kg	700	105.000	0	0	0

FIGURA 63. Ejemplo solicitudes HTTP

A simple vista se identifican 7 fuentes de datos distintas, para las cuales un enfoque podría ser generar una solicitud por cada una de ellas. El problema de esto es que podría

aumentar considerablemente la latencia<sup>46</sup>, en redes que experimenten problemas de lentitud por el poco tamaño de los *buffers* dentro de los equipos de conectividad o desperfectos en el cableado de red por ejemplo. Si bien esto no es factible arreglarlo directamente en el desarrollo, si se puede considerar el disminuir las solicitudes HTTP, al traer por ejemplo toda la información de un formulario en una sola petición.

### **4.3.Recomendaciones de uso**

En base al análisis hecho anteriormente, se entregan una serie de recomendaciones para minimizar las desventajas nombradas y se puedan explotar al máximo las cualidades de AJAX.

#### **4.3.1. Utilizar Frameworks y librerías de componentes**

Como se ha venido recalando, es extremadamente recomendable seleccionar un Framework para comenzar el desarrollo de una aplicación en AJAX, estos solucionan varias de las desventajas nombradas anteriormente, como es la incompatibilidad entre navegadores, funciones que permiten recordar los estados de la aplicación, para así no perder el uso de los botones ATRÁS, ADELANTE, ACTUALIZAR y FAVORITOS, además proveen *widgets* para dar mayor funcionalidad a esta.

Para la creación del CONGABOV, se uso el *framework* GWT, el cual sin duda alguna facilito mucho su desarrollo, al permitir crear la aplicación en JAVA, con todas sus potencialidades como son la orientación a objetos, depuración de código, *widgets*, etc.

La mantención y reutilización del código en JAVA se hace sencilla, en comparación a lo complejo que puede llegar a ser esta tarea en JavaScript. GWT provee funciones para mantener el historial URL y permite incorporar librerías directamente en JavaScript, mediante la interfaz JSN.

No se han estudiado todos los *frameworks* disponibles para la creación de RIA en AJAX, como para poder recomendar uno por sobre otro, pero si se puede asegurar que la tarea de desarrollar una aplicación con estas características desde cero es ardua.

---

<sup>46</sup> Es el tiempo o lapso necesario para que un paquete de información se transfiera de un lugar a otro. La latencia, junto con el ancho de banda, son determinantes para la velocidad de una red.

### 4.3.2. Disminuir el número de solicitudes al servidor

Para disminuir el tiempo de respuesta de una aplicación AJAX, debido a la latencia de las redes, se recomienda pedir o enviar la mayor cantidad posible de datos de una sola vez. Por ejemplo en la figura 64 se muestra la creación de un movimiento contable en la aplicación CONGABOV, esta acción esta vinculada a varias tablas y procedimientos almacenados en el servidor.

**Compras**

Crear Nuevo Movimiento · Crear Proveedor o Cliente · Crear Actividad · Crear Cuenta

**T\_CMovimientos**

Nº Factura o Comprobante: 2222 Fecha: 02-08-2007 Proveedor / Cliente: 82392600-6 : Cooprinsem Nº Cheque:

Glosa: Actividad: 0000: Gastos e Ingresos Generales Monto: Impuestos:  
 IVA (19 %)  
 ILA (8 %)  
 Impuesto Molino (7 %)  
 Impuesto Feria Ganado (8 %)

Cuenta que Provee: Unidad: Unidad  
Cuenta que Recibe: Cantidad:

Agregar Linea

Glosa	Actividad	Cta Provee	Cta Recibe	Un.	Ctd.	Monto	Impuesto	Esp. Rec.	Esp. No Rec.
semilla papas	2210 : Papas	570000 : Caja	601104 : papa semilla	kg	30.000	6.000.000	0	0	0
SPT	2210 : Papas	570000 : Caja	601050 : Superfosfato triple	kg	4.500	1.700.000	323.000	0	0
Muriato de K	2210 : Papas	570000 : Caja	601084 : Muriato de potasio	\$	2.500	405.000	76.950	0	0
Rango	2210 : Papas	570000 : Caja	601220 : Herbicida	lt	3	12.750	2.432	0	0
Supernitro	2210 : Papas	570000 : Caja	601083 : Supernitro	kg	5.000	820.000	155.800	0	0

**T\_CImpuestos**

**T\_CDetalleMovimiento**

Guardar cambios · Cerrar Ventana

FIGURA 64. Ejemplo formulario

Suponiendo el peor de los casos, se podría enviar primero una solicitud para crear el movimiento, luego otra por cada línea de detalle que se le quiera agregar a este y otra por cada impuesto. Así para un movimiento que posea 4 líneas de detalle y a cada una de estas se le aplique el IVA, se necesitarían 9 solicitudes para completar la operación.

Gracias al uso de JavaScript, se puede almacenar en memoria toda la información del movimiento, en arreglos u otras estructuras de datos, de manera local hasta que el usuario termine de ingresar toda la información, enviándose esta en una sola solicitud a un servicio *Web* encargado de crear los registros.

### 4.3.3. Pronosticar la cantidad de datos a obtener por petición.

JavaScript necesita transformar a un formato entendible el XML que recibe desde el servidor, acción que consume recursos del sistema y en algunos casos puede hacer que el equipo se quede congelado esperando a que esta operación se termine.

Para evitar que esto ocurra, es necesario pronosticar la cantidad de datos promedio que el usuario solicitará por árbol XML, en cada módulo del sistema, y en base a esta información, decidir que acción tomar para evitar el consumo excesivo de recursos en las maquinas de los clientes, situación que hace muy tedioso el uso de las aplicaciones.

En el caso del CONGABOV, debido a que la cantidad de datos en los informes puede llegar a ser considerable, dependiendo del número de movimientos ingresados al sistema, se opto por usar la paginación, limitando de esta manera la cantidad de nodos en los arboles XML a obtener por solicitud y asegurando el correcto funcionamiento del *software*.

Para dar una idea de este proceso se preparo una prueba, en la cual se miden los tiempos de procesamiento de un documento XML simple, que consiste en entidades que representan personas. En la figura 65 se muestra el documento XML y el formato que se le desea dar.

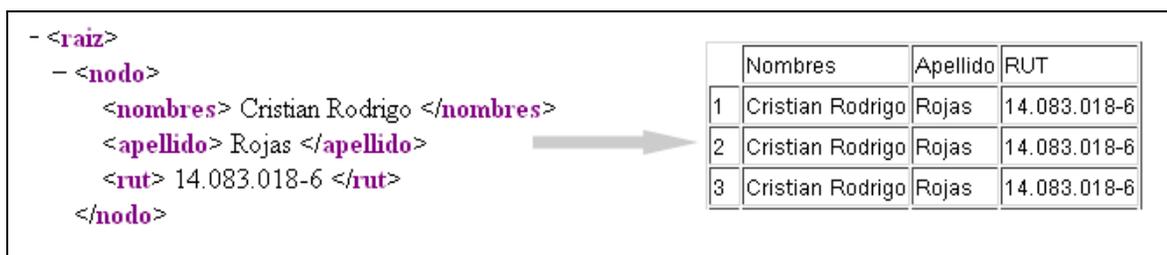


FIGURA 65. Archivo XML y su representación gráfica

En esta prueba, se ha simplificado al máximo la forma del documento XML y su representación gráfica. También se opto por realizar esta de manera local, para descartar la latencia generada en las solicitudes entre equipos a través de *Internet*.

Las características de la máquina utilizada son las siguientes; CPU Pentium 4 3.0 GHZ, 768 MB de RAM, SO Windows XP SP3.

Se realizó reiteradamente la prueba para obtener un promedio, eliminando los valores máximos y mínimos obtenidos, se aumentó la cantidad de entidades desde 10 hasta 1200. En el gráfico 66 se muestran los resultados obtenidos.

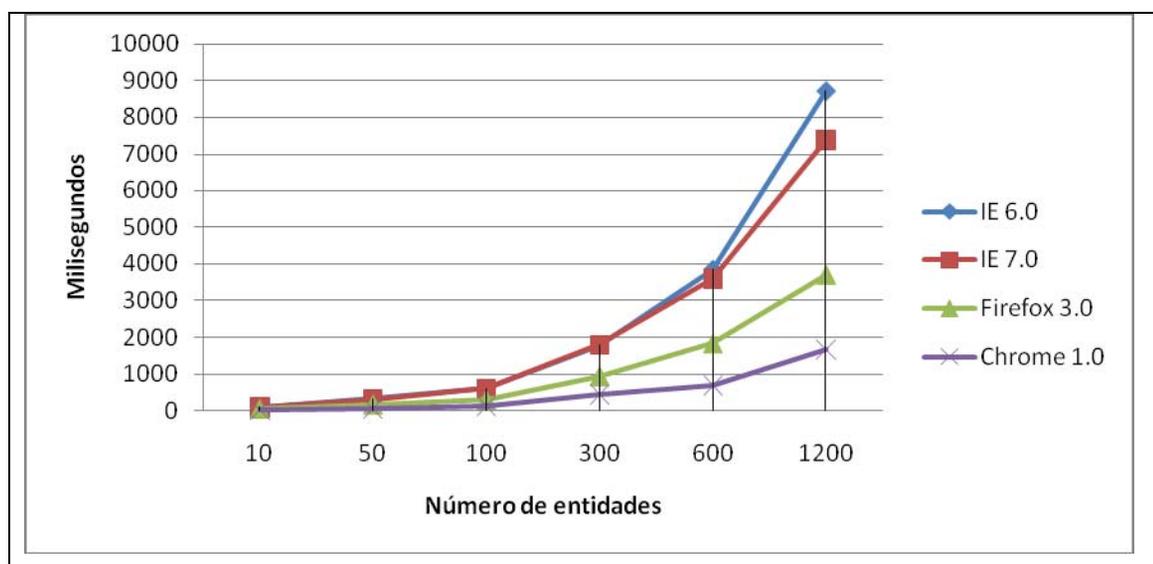


FIGURA 66. Gráfico tiempos de procesamiento

Se probó con distintos navegadores, pero **en ningún caso esta prueba pretende medir las diferencias entre estos y recomendar el uso de uno por sobre otro**, su único objetivo es justificar la decisión de paginar la información en el *software* CONGABOV.

Las primeras mediciones son rapidísimas, pero al ir aumentando el número de entidades, se aprecia un deterioro en el tiempo de procesamiento. Si bien no es común que una aplicación necesite mostrar más de mil entidades, esto sí se da en el CONGABOV, llegando a demorar este proceso casi 10 segundos, en los cuales el usuario no puede realizar otra acción.

Si consideramos que la simplicidad del documento XML usado en esta prueba, se aleja de la complejidad de una aplicación real y que además se eliminó la latencia al trabajar con los archivos de manera local y esto a su vez disminuye el tiempo de descarga, los 10 segundos pueden llegar a convertirse en un tiempo mucho mayor como se pudo comprobar en la aplicación CONGABOV.

En cuanto a la técnica de paginación usada, no se mostrará en este documento, ya que depende de la forma en que se quiera programar esta, pudiendo haber opciones más óptimas que la usada para esta aplicación. Solo se recomienda comenzar la paginación desde la base de datos, así las solicitudes desde la interfaz del cliente, llevan los

parámetros (desde, hasta y cantidad), para que la consulta SQL obtenga solo los datos solicitados, disminuyendo los tiempos de obtención en todas las capas del software.

#### 4.3.4. Indicar al usuario sobre el estado de la aplicación

Para usuarios que están acostumbrados a la *Web* tradicional, se hace difícil el identificar cuando una aplicación AJAX esta solicitando datos y en que momento estos han terminado de transmitirse.

Por esto es necesario implementar alguna función que se encargue de esto, o usar las que usualmente vienen con los *frameworks*. Para el caso del CONGABOV, se utilizaron los siguientes avisos, uno para la pantalla principal y otro para los formularios como se puede ver en la figura 67.



FIGURA 67. Indicadores de actividad

#### 4.3.5. Separar las lógicas de usuario y del negocio

Esta práctica no es exclusiva de aplicaciones AJAX pero tiende a darse de manera natural en aplicaciones *Web*, debido a la arquitectura de estas. La idea principal es dejar todo lo que tiene que ver con la lógica del usuario en lado del cliente, generada en JavaScript en el navegador *Web* y todo lo referente a la funcionalidad de la aplicación en el servidor, ya sea en algún lenguaje de programación que funcione de este lado (PHP, ASP, etc.), o en el lenguaje SQL que provea el motor de base de datos.

Para la aplicación CONGABOV se opto por dejar la lógica del negocio en el motor de base de datos, debido a que esto produce un bajo acoplamiento<sup>47</sup>, alto rendimiento en las consultas, posee un acceso universal y mejora el nivel de seguridad. [URL31]

En la figura 68, se muestra un extracto de un informe, generado en un procedimiento almacenado en el servidor.

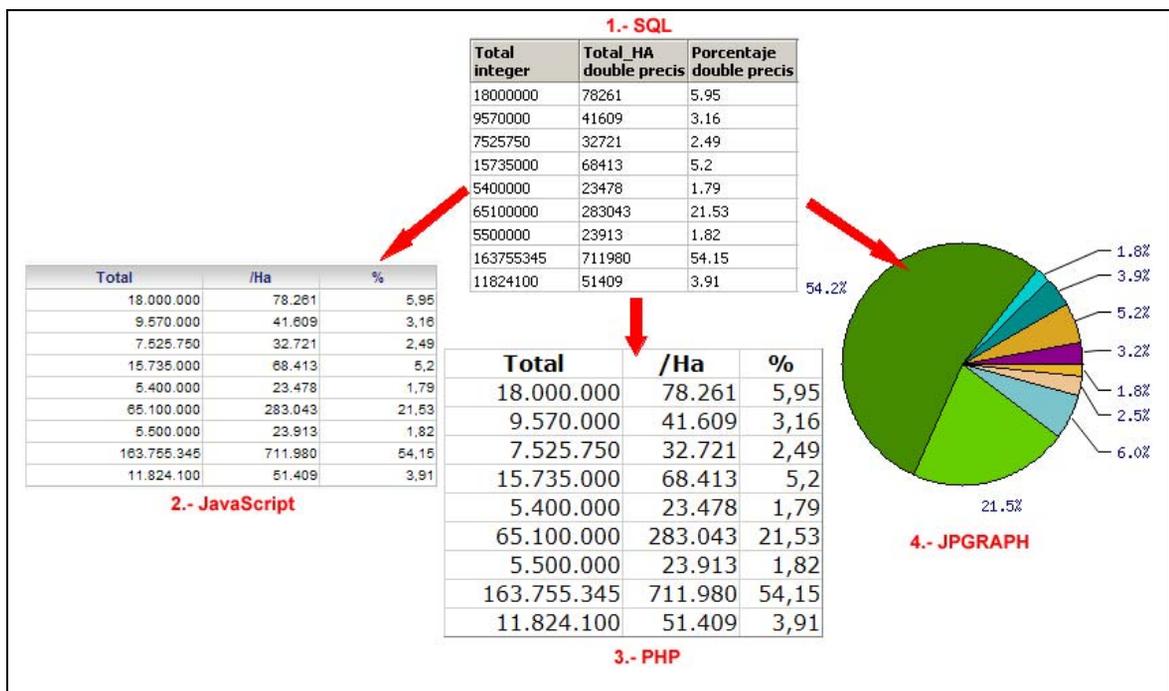


FIGURA 68. Ejemplo separación de lógicas

Lo primero que se ve es el procedimiento almacenado, el cual es el encargado de generar todos los valores, luego tenemos la vista construida en el lado del cliente en JavaScript, después otra visión de la misma información, pero esta vez generada en PHP para el módulo de impresión y finalmente la librería JpGraph, que genera un gráfico de torta. Si por ejemplo el procedimiento almacenado no calculara la columna porcentaje, las siguientes tres vistas mostradas tendrían que incluir esta lógica en sus códigos fuentes.

Al generarse los informes de contabilidad completamente en la base de datos, permite generar funciones genéricas en el resto de lenguajes, ahorrando bastantes líneas de código.

<sup>47</sup> Este indica el nivel de dependencia entre las unidades de software de un sistema informático, es decir, el grado en que una unidad puede funcionar sin recurrir a otras; dos funciones son absolutamente independientes entre sí (el nivel más bajo de acoplamiento) cuando una puede hacer su trabajo completamente sin recurrir a la otra.

Esto también facilita la migración de la aplicación a otras plataformas, por ejemplo si se quisiera cambiar AJAX por FLEX, solo se tendría que recrear la lógica de usuario y funciones que sean capaces de desplegar los informes contables, pero no se tendría que volver a programar la lógica de estos.

#### **4.4. Análisis del *Framework* GWT.**

Si bien en el capítulo dos se dio una descripción completa de este *framework*, ahora se procederá a nombrar sus ventajas y desventajas exclusivamente desde el punto de vista del desarrollo de la aplicación CONGABOV, para luego dar recomendaciones para su uso.

##### **4.4.1. Ventajas de GWT.**

Sin duda GWT fue de gran ayuda en el desarrollo de este software, las ventajas consideradas más importantes a la hora de desarrollar esta aplicación, fueron las siguientes.

- **Programación en JAVA:** Al poder usar este avanzado lenguaje de programación, se pudieron incorporar herramientas como depuradores de código y control de versiones, haciendo mucho más fácil la creación y mantención de la aplicación.
- **Compatibilidad:** El compilador de GWT se encarga de generar código JAVASCRIPT para la mayoría de los navegadores modernos, librando de esta ardua tarea a los desarrolladores, sin duda una de las mejores ventajas de este *framework*.
- **Permite trabajar con JAVASCRIPT:** Esta característica permitió lograr todas las funcionalidades que no se pudieron hacer directamente con GWT, un ejemplo de esto fue la impresión de los documentos, la cual tuvo que realizarse directamente con librerías en JAVASCRIPT.

#### 4.4.2. Desventajas de GWT.

A pesar de todos los beneficios que tiene este *framework*, se observaron ciertos aspectos de este que se podrían mejorar en las próximas versiones.

- **Difícil incorporación sobre proyectos realizados en otras plataformas:** Como se menciona anteriormente la aplicación CONGABOV tuvo que ser incorporada como módulo en una herramienta llamada EGROUWARE, desarrollada en PHP. Este proceso tuvo varios problemas técnicos debido a las restricciones de seguridad y estructura que tienen las páginas en esta herramienta, motivo por el cual no se pudo conservar el historial (botones atrás, adelante y actualizar) del navegador en el CONGABOV.
- **Poca variedad de Widgets:** GWT no cuenta con componentes avanzados, siendo necesario desarrollar e incorporar librerías externas para poder contar en el CONGABOV con todas las funcionalidades requeridas y comentadas anteriormente en este capítulo.

#### 4.4.3. Recomendaciones para el uso de GWT.

Este *framework* actúa como una capa intermedia entre el desarrollo y el resultado AJAX, teniendo como principal objetivo el simplificar la tarea de tratar con todas las peculiaridades que tienen estas tecnologías. En base a la experiencia obtenida en el desarrollo del *software* CONGABOV, se recomienda contar con un IDE y depuradores de código adecuados para JAVA y obtener así el máximo provecho de este lenguaje. También se necesitan buscar librerías externas de *Widgets* avanzados, para suplir la falta de estos en GWT, ejemplos de estas son GWT Widget Library y GWT-EXT entre otras.

Al ser las aplicaciones de GOOGLE creadas con este *framework*, se pueden ver la calidad de sus resultados, dando confianza al momento de escoger GWT. Sin duda es un *framework* altamente recomendado a la hora de crear aplicaciones AJAX.

## CAPÍTULO V

### 5. CONCLUSIONES

Después de un arduo trabajo analizando y estudiando la teoría contable que posee el sistema CONGABOV, se logró crear una versión en *Internet* para satisfacer las necesidades de sus usuarios. Esto significó un gran reto, desde el punto de vista teórico y tecnológico, el cual se pudo vencer debido al previo estudio de tecnologías que implica AJAX y ver cual de todas las plataformas es la que mas se ajustaba a las necesidades requeridas por el contexto contable de este *software*.

El desarrollo de una aplicación de esta naturaleza en *Internet* implica el contar con funcionalidades necesarias para manipular la gran cantidad de datos asociados a la contabilidad, extensos informes contables, generación de gráficos, tiempos de respuesta aceptables y muchas consideraciones más para asegurar su viabilidad.

Al tener todo este conjunto de requisitos, inmediatamente se considero AJAX para su creación, específicamente con el *framework* GWT, escogido después del estudio y análisis tecnológico. El conocimiento adquirido de AJAX y GWT sirvió para poder entregar al lector un análisis de estos, junto con algunas recomendaciones para su uso, y obteniendo así el máximo provecho de estas tecnologías.

Las aplicaciones en *Internet* cada vez poseen mas funcionalidades y mejor rendimiento, estrechando la brecha existente entre las aplicaciones de escritorio y la Web. Sin embargo, todavía falta mucho camino por recorrer, lo más importante ahora es lograr que todas las tecnologías involucradas sean estándares aceptados por todos los navegadores modernos.

Es difícil pronosticar que tecnologías serán las que logren mejores resultados en el futuro, ya sea AJAX, FLEX, JAVAFX, Silverlight o alguna otra solución, pero lo que si esta claro, es que la creación de aplicaciones en *Internet* cada vez va a abarcar mas campo y se verán capacidades que antes solo eran imaginables en software de escritorio.

Las ventajas de tener este *software* en línea para el usuario, van desde un respaldo continuo y asegurado de todos sus datos contables, actualizaciones inmediatas de la

aplicación, sin necesidad de descargas ni instalaciones, disponibilidad desde cualquier localización con *Internet*. Todos estos aspectos le dan valor agregado al producto desarrollado en comparación con su versión de escritorio.

Con la creación y puesta en marcha del Software CONGABOV, se han logrado cumplir todos los objetivos propuestos en esta tesis y se ha podido entregar a los usuarios del proyecto FONDEF D03I1151, una aplicación capaz de cumplir con sus exigencias en el ámbito técnico - económico y financiero.

## REFERENCIAS

### LIBROS Y PUBLICACIONES

[Ler07] Lerdón F, (2007), "CONGA, UN SISTEMA PARA LA PLANIFICACIÓN Y EL CONTROL DE GESTIÓN DE EMPRESAS AGRICOLAS", Universidad Austral de Chile.

[Sep95] Sepúlveda L, (1995), "DICCIONARIO DE TÉRMINOS ECONÓMICOS", Editorial universitaria.

[CON78] CONARA (Comisión nacional de la reforma administrativa), (1978), "PLAN CONTABLE GENERAL, SECTOR EMPRESARIAL".

[Ler84] Juan Lerdón, Gonzalo Jara, Nelson Parra, (1984), "Costo de producción litro de leche e indicadores económicos del rubro lechería en diez predios del país", Departamento de informática agrícola.

### MATERIAL EN INTERNET

[URL01] Angermeier M., The huge cloud lens bubble map web2.0, Disponible en <http://kosmar.de/archives/2005/11/11/the-huge-cloud-lens-bubble-map-web20/>.

[URL02] Garret J., Ajax: A New Approach to Web Applications, Disponible en <http://adaptivepath.com/ideas/essays/archives/000385.php>.

[URL03] Lerdón J., Tecnologías de infocomunicación apoyando las decisiones de producción bovina, Disponible en [http://www.produccionbovina.cl/wwwfondof/docs/revista\\_agrarias.pdf](http://www.produccionbovina.cl/wwwfondof/docs/revista_agrarias.pdf).

[URL04] Berners-Lee T., HTML A Representation of Textual Information and MetaInformation for Retrieval and Interchange, Disponible en <http://www.w3.org/MarkUp/draft-ietf-iiir-html-01.txt>.

[URL05] O'Reilly, An Introduction to Cascading Style Sheets, Disponible en <http://nwalsh.com/docs/articles/css/>.

- [URL06] W3C, Extensible Markup Language (XML), Disponible en <http://www.w3.org/XML/>.
- [URL07] Bray T., Extensible Markup Language (XML) 1.0 (Fourth Edition 2006), Disponible en <http://www.w3.org/TR/2006/REC-xml-20060816/>.
- [URL08] Wikipedia, XML, Disponible en <http://es.wikipedia.org/wiki/XML>.
- [URL09] JavaScript, Disponible en <https://developer.mozilla.org/es/JavaScript>.
- [URL10] Wikipedia, JavaScript, Disponible en <http://es.wikipedia.org/wiki/Javascript>.
- [URL11] W3C, The XMLHttpRequest Object, Disponible en <http://www.w3.org/TR/XMLHttpRequest/>.
- [URL12] W3C, Document Object Model (DOM) Level 2 Events Specification, Disponible en <http://www.w3.org/TR/DOM-Level-2-Events/>.
- [URL13] W3C, Document Object Model (DOM) Level 3 Core Specification, Disponible en <http://www.w3.org/TR/DOM-Level-3-Core/>.
- [URL14] W3C, A vocabulary and associated APIs for HTML and XHTML, Disponible en <http://www.w3.org/html/wg/html5/>.
- [URL15] IETF, Hypertext Transfer Protocol -- HTTP/1.1, Disponible en <http://www.ietf.org/rfc/rfc2616>.
- [URL16] W3C, Web Services Architecture, Disponible en <http://www.w3.org/TR/ws-arch/>.
- [URL17] Google, Google Web Toolkit, Disponible en <http://code.google.com/intl/es-ES/webtoolkit/overview.html>.
- [URL18] Eclipse, Eclipse, Disponible en <http://www.eclipse.org/>.

- [URL19] Instantiations, GWT Designer, Disponible en <http://www.instantiations.com/gwt designer/>.
- [URL20] Lacomella F., Concurrent versions system, Disponible en <http://francoiacomella.org/libros/ CVS Manejo descentralizado de proyectos.pdf>.
- [URL21] PostgreSQL, PostgreSQL, Disponible en <http://www.postgresql.org/about/>.
- [URL22] pgadmin, pgadmin, Disponible en <http://www.pgadmin.org/>.
- [URL23] PHP, Manual de PHP, Disponible en <http://www.php.net/manual/es/>.
- [URL24] PHPEclipse, PHPEclipse, Disponible en <http://www.phpeclipse.com/>.
- [URL25] Robert Hanson's, GWT Widget Library, Disponible en <http://gwt-widget.sourceforge.net/>.
- [URL26] Wikipedia, Explotación agraria, Disponible en <http://es.wikipedia.org/wiki/ Explotación agraria>.
- [URL27] Wikipedia, Débito, Disponible en <http://es.wikipedia.org/wiki/ Débito>.
- [URL28] Ministerio de economía, Disponible en <http://www.economia.cl/>.
- [URL29] Google, Gears, Disponible en <http://code.google.com/intl/es/apis/gears/>.
- [URL30] Adobe, Adobe AIR, Disponible en <http://www.adobe.com/products/air/>.
- [URL31] Eduardo Ostertag, Rich Internet Applications, Disponible en <http://www.obcom.cl/obdocum/charlaria.pdf>.