

UNIVERSIDAD TECNOLÓGICA ISRAEL



CARRERA DE SISTEMAS INFORMÁTICOS

TEMA:

**INVESTIGACIÓN E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE
INTELIGENCIA DE NEGOCIOS EN SOFTWARE LIBRE**

**Trabajo de Graduación previo a la obtención del título de Ingeniería en
Sistemas Informáticos y de Computación**

AUTOR:

Mario Javier Salvador Gómez

TUTOR:

Msc. Emilio Márquez

Quito - Ecuador

2013

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del Trabajo de Graduación certifico:

Que el Trabajo de Graduación “**INVESTIGACIÓN E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE INTELIGENCIA DE NEGOCIOS EN SOFTWARE LIBRE**”, presentado por Mario Javier Salvador Gómez, estudiante de la carrera de Sistemas Informáticos y de Computación, reúne los requisitos y méritos suficientes para ser sometida a la evaluación del Tribunal de Grado, que se designe, para su correspondiente estudio y calificación.

Quito, Febrero 2013

TUTOR

Msc. Emilio Márquez

UNIVERSIDAD TECNOLÓGICA ISRAEL

AUTORÍA DE TESIS

La abajo firmante, en calidad de estudiante de la Carrera de Sistemas Informáticos y de Computación, declaro que los contenidos de este Trabajo de Graduación, requisito previo a la obtención del Grado de Ingeniera en Sistemas Informáticos y de Computación, son absolutamente originales, auténticos y de exclusiva responsabilidad legal y académica del autor.

Quito, Febrero 2013

Mario Javier Salvador Gómez

CC: 172014940-8

UNIVERSIDAD TECNOLÓGICA ISRAEL

APROBACIÓN DEL TRIBUNAL DE GRADO

Los miembros del Tribunal de Grado, aprueban la tesis de graduación de acuerdo con las disposiciones reglamentarias emitidas por la Universidad Tecnológica “ISRAEL” para títulos de pregrado.

Quito, Octubre 2013

Para constancia firman:

TRIBUNAL DE GRADO

PRESIDENTE

MIEMBRO 1

MIEMBRO 2

DEDICATORIA

A mis familiares y personas cercanas que siempre me apoyaron en la elaboración y culminación de este proyecto.

Javier Salvador

AGRADECIMIENTO

A mis padres por su apoyo incondicional y a mis profesores y tutores que todos estos años me brindaron parte de su conocimiento para culminar este proyecto.

Javier Salvador

Contenido

1. Introducción.....	1
1.1. Antecedentes.....	2
1.2. Formulación del Problema.....	3
1.3. Sistematización.....	4
1.3.1. Diagnóstico.....	4
1.3.2. Pronóstico.....	5
1.3.3. Control del Pronóstico.....	6
1.4. Objetivos.....	9
1.4.1. Objetivo general.....	9
1.4.2. Objetivos específicos.....	9
1.5. Justificación.....	10
1.5.1. Justificación Teórica.....	10
1.5.2. Justificación Práctica.....	10
1.5.3. Justificación Metodológica.....	11
1.6. Alcance y Limitaciones.....	12
1.6.1. Alcance.....	12
1.6.2. Limitaciones.....	12
1.7. Estudios de factibilidad.....	13
1.7.1. Técnica.....	13
1.7.2. Factibilidad Económica.....	18
1.7.3. Operativa.....	21
2. Marco de Referencia.....	22
2.1. Marco teórico.....	22

2.2. Marco conceptual	26
2.3. Marco Espacial	33
2.4. Marco Legal.....	33
3. METODOLOGÍA	34
3.1. Proceso de investigación.....	34
3.1.1. Unidad de Análisis	34
3.1.2. Tipo de Investigación.....	34
3.1.3. Método	34
3.1.4. Técnica.....	34
3.1.5. Instrumento	35
3.1.6. Metodología Informática	35
4. PROCESO DE DESARROLLO	37
4.1 Introducción.....	37
4.2 Descripción Proceso de Desarrollo	38
4.3. Fase de Concepción	42
4.3.1. Documento de definición del proyecto	42
4.4. Fase de Elaboración	47
4.4.1. Diagrama de Casos de Uso	47
4.4.4. Diagrama de Base de Datos	51
4.4.5. Diagrama de Procesos.....	52
4.4.6 Arquitectura Lógica de la Solución	54
4.4.7. Necesidades y Requerimientos de la Solución.....	55
4.5. Fase de Construcción	58
4.5.1. Conceptos e instrucciones sobre Pentaho sobre de la construcción de la solución	58

4.5.2. Documento de Necesidades y Requerimientos de la solución.....	88
4.5.3. Construcción del Cubo de Información	91
4.5.4. Capa de presentación	100
5. Conclusiones y Recomendaciones	114
5.1. Conclusiones	114
5.2. Recomendaciones	115
5.3. Bibliografía	116
ANEXOS	117
ANEXO 1	118
ANEXO 2	124
ANEXO 3	129
ANEXO 4	136
ANEXO 5	144
ANEXO 6	151
ANEXO 7	158
ANEXO 8	165
ANEXO 9	172
ANEXO 10	180

UNIVERSIDAD TECNOLÓGICA ISRAEL
CARRERA DE SISTEMAS INFORMÁTICOS

TEMA:

“INVESTIGACIÓN E IMPLEMENTACIÓN DE UNA HERRAMIENTA DE
INTELIGENCIA DE NEGOCIOS EN SOFTWARE LIBRE”

AUTOR

Mario Javier Salvador Gómez

TUTOR

Msc. Emilio Márquez

RESUMEN

A medida que los datos informativos crecen cada día, la necesidad de analizarlos, diferenciarlos y clasificarlos es cada vez mayor.

Por este motivo se ha buscado una herramienta en software libre especializada en Inteligencia de Negocios, para este proyecto se ha escogido la herramienta Pentaho BI Suite.

Pentaho BI Suite contiene diferentes productos en plataforma BI, que permiten el desarrollo de cubos OLAP, reportes gerenciales, representaciones gráficas de datos, diagramas, etc.

Se va a utilizar algunos productos que dispone la herramienta Pentaho BI Suite, específicamente: Pentaho Analysis Services y Pentaho Dashboard y Reporting, para implementar esta herramienta en un ejemplo práctico.

UNIVERSIDAD TECNOLÓGICA ISRAEL
COMPUTER SYSTEMS CAREER

TOPIC:

“RESEARCH AND IMPLEMENTATION OF A TOOL FOR BUSINESS
INTELLIGENCE FREE SOFTWARE”

AUTHOR

Mario Javier Salvador Gómez

TUTOR

Msc. Emilio Márquez

ABSTRACT

According as the data grow every day, the need to analyze, differentiate and classify them is increasing.

For this reason it has sought a free software tool specialized in Business Intelligence, for this project has been selected Pentaho BI Suite tool.

Pentaho BI Suite contains several BI platform products that enable the development of OLAP cubes, management reports, graphical representations of data, diagrams, etc.

It will use some products available to the Pentaho BI Suite tools, specifically:

Pentaho Pentaho Analysis Services and Reporting Dashboard, to implement this tool in a practical example.

1. Introducción

A medida que pasa el tiempo las personas han visto en varios escenarios en que la información se le ha presentado en más cantidad y por ende más difícil de acceder.

Cada vez se requiere mucha más intuición para la toma de decisiones correctas fundamentadas en información actualizada y exacta. Las herramientas para reportes, análisis y consulta de datos pueden ayudar a los usuarios de negocios a explorar a través de un sinnúmero de información para sacar solo lo necesario o valioso; por estas razones fundamentalmente se ha creado la inteligencia de negocios.

La Inteligencia de Negocios es el conjunto de productos y servicios que permiten a los usuarios finales analizar de manera rápida y sencilla, la información para la toma de decisiones de negocio a nivel operativo, táctico y estratégico. (Calaméo, 2011)

La idea es demostrar a través de un ejemplo práctico de la vida real, la utilidad que tienen las herramientas en Inteligencia de Negocios y de los beneficios que conlleva la adquisición de esta herramienta en Software Libre.

Definitivamente también se obtendrá un análisis y conceptos de lo que es Inteligencia de Negocios y su importancia, análisis exhaustivo de la herramienta de BI a utilizarse.

1.1. Antecedentes

Al alcanzar una organización una gran cantidad de información se ve en la necesidad de una mayor organización de sus datos caso contrario se puede correr el riesgo de tener información en desuso y desaprovecho de información importante.

Da como resultado que muchas organizaciones obtengan entre sus recursos equipos de computación y sistemas de información automatizados.

Pero sin embargo las personas que utilizan estos sistemas presentan numerosas inquietudes, esto con el fin de realizar tareas más complejas que le den a la organización un mayor beneficio al tomar decisiones y para lograrlo se necesita tener suficientes conocimientos que permitan determinar sus requerimientos de información y aplicar mejores soluciones.

A partir de estos antecedentes se vio la necesidad que las organizaciones recurran a Sistemas de Información más elevados, los llamados Sistemas de Soporte a Ejecutivos (ESS)

Los Sistemas de Soporte a Ejecutivos son herramientas que se enfocan en dar información y situación actual de la organización y dejan en un objetivo secundario la proyección de esta información en el futuro y de procesar inteligentemente tal cantidad de información.

Por esta razón se concibió el concepto de Inteligencia Empresarial ya que combina las utilidades de un software con la inteligencia del ser humano, por su capacidad de introspección.

Con esta herramienta logística, el especialista dispone de los elementos necesarios para planear, crear e implementar modelos de solución basados en la Inteligencia de Negocios.

1.2. Formulación del Problema

El problema surge cuando la organización crece a tal medida de tener gran cantidad de información y no poder organizarse con sus sistemas y recursos informáticos.

Cuando esto ocurre pueden suceder los siguientes aspectos:

_ Datos erróneos al elaborar reportes gerenciales sin automatizar

_ Información sin organizar esto conlleva a que exista gran número de información desvalorada, y consecuentemente se discrimine información que puede ser útil en el futuro.

_ Los usuarios gerenciales de la organización encuentran una dificultad al querer acceder a información de esta, ya que por la gran cantidad de información esta podría quedar casi inaccesible consecuencia de las formas de guardar y recurrir a la información que se pueda tener.

_ Muchas herramientas en Inteligencia de Negocios si bien son muy útiles en la organización son herramientas de software pagadas y siempre esto resulta un problema importante.

_ Inexistencia de un seguimiento sobre los movimientos profesionales de la empresa, esto lleva a un desconocimiento de la condición real de la organización llevando también a tomar decisiones al azar o por pura intuición o simplemente malas decisiones en el futuro.

1.3. Sistematización

1.3.1. Diagnóstico

A continuación se enlista los principales problemas encontrados que consecuentemente se encontraran una solución.

_ Reportes elaborados sin un estándar de presentación y manejo.

_ Información aleatoria desorganizada.

_ Gran número de datos desvalorizada.

_ Inexistencia o poco seguimiento sobre los movimientos profesionales de la empresa.

_ Sistemas de Soporte a Gerenciales costosos y poco flexibles.

_ Cuando una empresa posee una gran cantidad de información importante sin saber cómo manejarla.

1.3.2. Pronóstico

Para esta parte se busca realizar una proyección de las posibles consecuencias de los problemas encontrados en el diagnóstico, se enlista todos los problemas encontrados seguidos de sus posibles consecuencias.

_ Podría suceder que la información está en lugares de almacenamiento desconocidos para el personal a cargo de la gerencia, la gerencia estaría en desconocimiento de aquella información y por lo tanto se tomarían decisiones equivocadas.

_ Reportes elaborados sin un estándar de presentación y manejo.

Esto trae como consecuencia que cada quien en la empresa lleva la información a su manera, conlleva a malos entendidos, contratiempos y esfuerzo de trabajo innecesario.

_ Información aleatoria desorganizada.

Conlleva a tener información poco importante en prioridad alta y viceversa.

_ Gran número de datos desvalorizados.

Menosprecio de información valiosa.

_ Inexistencia o poco seguimiento sobre los movimientos profesionales de la empresa.

Esto trae como consecuencia la falta de conocimiento de los movimientos y condición real de la empresa consecuentemente se pueden tomar decisiones gerenciales equivocadas.

_ Sistemas de Soporte Gerenciales costosos y poco flexibles.

Gran pérdida de dinero en sistemas de gerenciales poco útiles y sin resultados prácticos para la organización

_ Cuando una empresa posee una gran cantidad de información importante sin saber cómo manejarla.

Es porque hay tanta cantidad de información que está llega a extraviarse, esto conlleva a que la información sea montada con supuestos y datos prácticamente inventados por ende esta información siempre será falsa.

1.3.3. Control del Pronóstico

Aquí se enuncian las posibles soluciones que se darán a los problemas encontrados en el Diagnóstico.

_ Datos prácticamente inaccesibles.

La Inteligencia de Negocios posee una técnica llamada ETL (extraer, cargar y transformar) es el proceso que se encarga de mover información y depurarla y almacenarla toda junta en un sistema de base datos organizada para toda la empresa, con esto se facilitaría el acceso a los datos.

_ Reportes elaborados sin un estándar de presentación y manejo.

Una parte importante que comprende la inteligencia de negocios es la posibilidad de llevar Reportes predefinidos o también llamados reportes a la medida, da la posibilidad a la empresa de llevar un formato de trabajo en cuanto a sus informes o reportes, disminuye la posibilidad

que se tengan diferentes formatos de reportes y que se obvие información importante en algunos.

_ Información aleatoria desorganizada.

Los Cubos OLAP es una solución utilizada en la Inteligencia de Negocios se trata de utilizar estructuras de varias dimensiones, estas contienen datos concentrados o resumidos en una estructura multidimensional que provienen de Bases de Datos de gran tamaño.

_ Gran número de datos desvalorizados.

Otra de las partes importantes que aborda la Inteligencia de Negocios es la Minería de Datos, esto quiere decir que a los datos se los analiza, se los sondea, explora y prepara para sacar la información oculta en ellos.

_ Inexistencia o poco seguimiento sobre los movimientos profesionales de la empresa.

Uno de los objetivos principales de la Inteligencia aborda es el Análisis de datos, esto permite realizar el Análisis estadísticos, pronósticos y modelado predictivo esto se utiliza cuando la información se presenta con regularidad de modo que esta características sirve para realizar pronósticos y un análisis más a fondo de la información, y por ende a tomar decisiones acertadas en el futuro.

_ Sistemas de Soporte Gerenciales costosos y poco flexibles.

Se va optar por escoger un software en Inteligencia de Negocios en Software Libre, esto permitiría disminuir los costos de adquirir esta herramienta.

_ Pérdida de información personal y valiosa para la empresa.

Este aspecto también puede encontrar una solución con las llamadas base de datos multidimensionales, porque con las estructuras de almacenamiento tradicionales almacenan datos en tablas separadas, al momento de realizar una consulta relacionada entre dos o más tablas el proceso es relativamente lento, con las bases de datos multidimensionales se obtiene información de manera rápida y disminuye la probabilidad de que se extravié.

1.4. Objetivos

1.4.1. Objetivo general

_ Mostrar por medio de una investigación y posterior implementación las ventajas que tiene para una organización la utilización de una herramienta en software libre de inteligencia de negocios.

1.4.2. Objetivos específicos

_ Investigar conceptos y procedimientos básicos que involucra la construcción de una solución en inteligencia de negocios.

_ Distinguir las diferentes características de la herramienta de software libre a utilizarse.

_ Analizar la lógica del negocio en el cuál se va a implementar la solución.

_ Implementar una solución de un problema y mostrar como se logró esta solución facilitar el trabajo gerencial de una institución.

_ Mostrar a través de reportes informativos, simples, gráficos y sencillos el resultado de la implementación.

1.5. Justificación

1.5.1. Justificación Teórica

A continuación se describen algunas ventajas que se obtiene al aplicar una solución del problema.

_ Información accesible.- Se trata de realizar una forma de que los usuarios accedan lo más sencillo posible a los datos sin importar de donde provengan estos.

_ Ayuda a la toma de Decisiones.- Tener una herramienta con acceso a datos personalizados por usuarios y estos datos sirvieran para tomar mejores decisiones que se reflejaran en el futuro.

_ Guía al usuario.- Se trata de separar los conceptos técnicos y las diferentes capacidades que pueden tener los usuarios al utilizar estas herramientas.

1.5.2. Justificación Práctica

_ Reducción de costos que implica desarrollar bajo una herramienta de software libre, por la reducción de costos que esto implica.

_ A diferencia de otras estrategias de administración de datos la Inteligencia de Negocios combina las utilidades de un software con la inteligencia del ser humano, por su capacidad de introspección.

_ Permite crear una jerarquía de indicadores de desempeño.

_ Se busca disminuir la redundancia e inconcistencia de datos.

1.5.3. Justificación Metodológica

Para un análisis de los conceptos de las herramientas posibles a utilizar y para saber cuál de estas es más conveniente utilizar, es necesario recurrir a diferentes técnicas de investigación.

Es importante saber el funcionamiento de la organización desde sus procesos más elementales, para aquello se tiene que hacer un estudio observacional desde el objetivo más elemental de la empresa, esto se puede lograr a través de entrevistas a los principales involucrados

Una vez hecho esto se elaboraran encuestas para que se despejen dudas al realizar las entrevistas antes mencionadas.

Ahora es fundamental realizar un levantamiento de requerimientos a nivel gerencial esto se puede lograr mediante entrevistas en primer lugar y elaboración de encuestas para saber qué tipo información prioriza los directivos de la empresa y que tipo de información no es tan relevante para ellos.

1.6. Alcance y Limitaciones

1.6.1. Alcance

Se llegará a investigar los diferentes conceptos que tienen que saberse para aplicar una solución en Inteligencia de Negocios, posteriormente un análisis de la herramienta en software libre de BI a utilizarse.

Se logrará hallar una estrategia en Inteligencia de Negocios en una organización donde es importante implementar una solución en este ámbito.

Se construirá un cubo de información como solución a la necesidad de organización de información sobre una base de datos ya establecida.

Terminada la construcción del cubo de información se desarrollarán reportes informativos y gráficos donde se podrá observar los beneficios de este cubo de información.

1.6.2. Limitaciones

La Solución no contemplará la construcción de la base de datos y el desarrollo de una aplicación previa, ya que estarán previamente construidas.

1.7. Estudios de factibilidad

1.7.1. Técnica

Se debe realizar un análisis de los recursos tecnológicos necesarios para el proyecto y que favorecerán su desarrollo entregando un valor agregado o carácter diferenciador, Vs los disponibles, Tomar en cuenta que los recursos tecnológicos que se requieren son los que favorecen al proyecto. Y los existentes deberían alinearse a este requerimiento.

Hardware Requerimientos Mínimos

Memoria RAM	6GB mínimo
Disco Duro	250 GB
Procesador	Dual-core AMD 64 O EMT64T

Tabla 1.- Requerimientos mínimos en hardware del proyecto solución.

Sistema Operativo

Sistema Operativo.- Los Software en Inteligencia de Negocios gratis son Multiplataforma por lo tanto es compatible su instalación bajo los siguientes Sistemas Operativos:

Versiones de Sistemas Operativos mínimos

<u>Sistema Operativo</u>	Versión Mínima
Windows	Windows XP Service Pack 2
Linux	SUSE Linux Enterprise Desktop and Server 10
Solaris	Solaris 10

Mac	Mac OS X 10.4
-----	---------------

Tabla 2.- Requerimientos mínimos en hardware del proyecto solución.

Características del Proyecto

- **Manejo de gran cantidad de datos:** La cantidad de datos que debe soportar es de grandes dimensiones ya que se va implementar en una finca donde la producción semanal es de 100000 ramos promedio.
- **Multiplataforma:** La solución debe ser desarrollada bajo un software multiplataforma, porque a medida que la información crece va aumentando su importancia y es necesario migrar a sistemas operativos más estables.
- **Disponibilidad Información:** Es necesario que la información esté disponible en todo momento.
- **Concurrencia y accesibilidad:** La concurrencia permite que todos los recursos sean utilizados simultáneamente por los usuarios, para el caso a desarrollar es necesario que la información será accesada por varios usuarios u operadores a la misma vez.

Matriz de Prioridades

Lenguaje de Programación

	Concurrencia	Multipataforma	Disponibilidad	Gran Cantidad de datos	Total	Porcentaje
Concurrencia	X	0,5	0,5	1	2	33,33
Multipataforma	0,5	X	0,5	0,5	1,5	25,00
Disponibilidad	0,5	0,5	X	1	2	33,33
Gran Cantidad de datos	0	0,5	0	X	0,5	8,33
					6	100,00

	Ponderación	Java		C#		PHP	
		Valor	Total	Valor	Total	Valor	Total
Concurrencia	33,33	8	2,6664	6	1,9998	8	2,6664
Multipataforma	25	7	1,75	7	1,75	7	1,75
Disponibilidad	33,33	7	2,3331	7	2,3331	6	1,9998
Cantidad de datos grandes	8,33	7	0,5831	6	0,4998	5	0,4165
	100		7,3326		6,5827		6,8327

Tabla 3.- Matriz de ponderación Lenguaje de Programación.

Base de Datos

	Concurrencia	Multipataforma	Disponibilidad	Gran Cantidad de datos	Total	Porcentaje
Concurrencia	X	1	0,5	1	2,5	41,67
Multipataforma	0	X	0,5	0,5	1	16,67
Disponibilidad	0,5	0,5	X	0	1	16,67
Gran Cantidad de datos	0	0,5	1	X	1,5	25,00
					6	100,00

	Ponderación	MySql		JasperSoft		SQL Server	
		Valor	Total	Valor	Total	Valor	Total
Concurrencia	41,67	6	2,5002	6	2,5002	7	2,9169
Multipataforma	16,67	8	1,3336	7	1,1669	5	0,8335
Disponibilidad	16,67	8	1,3336	7	1,1669	4	0,6668
Cantidad de datos grandes	25	7	1,75	6	1,5	9	2,25
			6,85				

Tabla 4.- Matriz de Ponderación Base de Datos.

Software BI

	Concurrencia	Multiplataforma	Disponibilidad	Gran Cantidad de datos	Total	Porcentaje
Concurrencia	X	1	0,5	1	2,5	41,67
Multiplataforma	0	X	0,5	0,5	1	16,67
Disponibilidad	0,5	0,5	X	0	1	16,67
Gran Cantidad de datos	0	0,5	1	X	1,5	25,00
					6	100,00

	Ponderación	<u>Pentaho</u>		<u>JasperSoft</u>		<u>SpagoBI</u>	
		Valor	Total	Valor	Total	Valor	Total
Concurrencia	41,67	8	3,3336	6	2,5002	7	2,9169
Multiplataforma	16,67	7	1,1669	7	1,1669	6	1,0002
Disponibilidad	16,67	7	1,1669	7	1,1669	7	1,1669
Cantidad de datos grandes	25	7	1,75	6	1,5	6	1,5
	100		7,4174		6,334		6,584

Tabla 5.- Matriz de Ponderación Software de Inteligencia de Negocios

Como conclusión conviene más desarrollar en la plataforma pensada Pentaho, porque brinda más beneficios para la solución deseada.

1.7.2. Factibilidad Económica

El desarrollo del módulo de disponibilidad propuesto para el sistema de ventas de la florícola integra un conjunto de recursos materiales, humanos y técnicos que son necesarios para su realización y posterior implantación en la empresa.

Recurso Humano

RECURSO HUMANO	DESCRIPCIÓN		
Desarrollador y Documentador	Sueldo	Sueldo x 6 meses	TOTAL
Javier Salvador	\$ 300,00	\$ 1800,00	\$ 1800,00
		SUBTOTAL	\$ 1800,00

Tabla 6.- Recurso Humano. Fuente Javier Salvador

Hardware

UNIDADES	DESCRIPCIÓN	COSTO UNITARIO	TOTAL
2	Computadores	\$650,00	\$1300,00
2	Monitores	\$130,00	\$260,00
2	Teclados	\$15,00	\$30,00
2	Mause Ópticos	\$6,50	\$13,00
1	Impresora Láser	\$70,00	\$70,00
		SUBTOTAL	\$1.673,00

Tabla 7.- Costos Hardware. Fuente Javier Salvador

Software

DESCRIPCIÓN	COSTO LICENCIA	TOTAL
Servidor Windows XP Service Pack 3	Existente	\$0,00
Servidor de BI Penaho	\$0,00	\$0,00
Servidor de Bases de Datos MySQL	\$0,00	\$0,00
	SUBTOTAL	\$0,00

Tabla 8.- Costos Software. Fuente Javier Salvador

Artículos de oficina y servicios:

UNIDADES	DESCRIPCIÓN	COSTO UNITARIO	TOTAL
3	Resmas de papel para impresora	\$6,50	\$19,50
3	Cartuchos de impresora Epson	\$7,00	\$21,00
1	Pack de DVD de 50 unidades	\$12,00	\$12,00
	Gastos Fijos (Agua, Luz,)	\$40,00	\$40,00
	Servicio Internet (1 MB)	\$19,50	\$19,50
	Varios Suministros	\$20,00	\$20,00
		SUBTOTAL	\$132,00

Tabla 9.- Costos Hardware. Fuente Javier Salvador

Costos Totales

Recurso Humano	
Hardware	\$1.800,00
Software	\$1.673,00
Artículos de oficina y servicios	\$0,00
TOTAL	\$132,00
Tabla 10.- Costos Totales. Fuente Javier Salvador	\$3,605,00

Flujo de Caja

Análisis Mensual De Flujo de Efectivo						
	<i>Julio</i>	<i>Agosto</i>	<i>Septiembre</i>	<i>Octubre</i>	<i>Noviembre</i>	<i>Diciembre</i>
Flujo Inicial de Efectivo	\$ -	(\$105)	\$263	\$631	\$999	\$1.317
Entradas						
Ingresos	2.000	800	800	800	800	800
Total de Entradas	\$2.000	\$800	\$800	\$800	\$800	\$800
Balance Disponible	\$2.000	\$695	\$1.063	\$1.431	\$1.799	\$2.117
Salidas						
Recursos Humanos	300	300	300	300	350	350
Hardware	1.673					
Software	0	0	0	0	0	0
Muebles, enseres y servicios	132	132	132	132	132	132
Comisiones						
Total Salidas	\$2.105	\$432	\$432	\$432	\$482	\$482
Balance de Flujo de Efectivo Final	(\$105)	\$263	\$631	\$999	\$1.317	\$1.635

Tabla 11.- Flujo de Caja Mensual. Fuente Javier Salvador

1.7.3. Operativa

Se debe enunciar la importancia que tendrá la aplicación y/o utilización que se dará a la información y/o producto resultante de esta investigación, si es necesario se debe incluir un análisis de impacto y resistencia al cambio por parte de los usuarios y/o la comunidad beneficiaria del producto.

Beneficios

- Mejor manejo de los datos y interactividad con el usuario final.
- Consolidar los diferentes procesos de producción de la empresa.
- Mejores maneras de visualizar los datos de la empresa.

Análisis de impacto y resistencia al cambio

- a. Resistencia por parte de los gerentes, por adoptar una nueva herramienta en BI, aferrándose a herramientas tradicionales como Excel, Access, Word, etc.
- b. La información es difícil de ordenar y clasificar, y tiene que ser trasladada por varios filtros para obtener la información deseada.
- c. Disponibilidad de los gerentes al querer enriquecer cada vez más con variedad de información que se pueda clasificar y realizar los cambios necesarios y oportunos de tal forma, que se adapte a los requerimientos de los distintos tipos de usuarios.

2. Marco de Referencia

2.1. Marco teórico

La inteligencia de negocios, se refiere al proceso mediante el cual las empresas y las compañías reúnen datos, los analizan, y lo vuelven a aplicar con el fin de aprovechar de la mejor manera posible el mejor modelo de negocio y financiero posible para su caso particular. Muchas empresas dedican todos sus recursos sólo para este proceso, con el fin de tratar de racionalizar su flujo de trabajo y hacer más dinero con menos esfuerzo.

La inteligencia empresarial podría encargarse de las ventas, beneficios, fondos perdidos, honorarios de la transacción, los importes pagados inicialmente; con datos suficientes, es posible determinar exactamente las cifras correspondientes. Un buen sistema de inteligencia de negocios básicamente gestiona el "valor corriente" de una empresa en un solo esfuerzo. Lo ideal es que el flujo de datos sea capaz de producir los mejores resultados con un mínimo de esfuerzo. Esto podría incluir desde la adición de puestos de trabajo en una zona, que convierten a un proceso de automatización, o incluso simplemente cambiando la estrategia de publicidad.

El término inteligencia empresarial se refiere al uso de datos en una empresa para facilitar la toma de decisiones. Abarca la comprensión del funcionamiento actual de la empresa, bien como la anticipación de acontecimientos futuros, con el objetivo de ofrecer conocimientos para respaldar las decisiones empresariales.

Las herramientas de inteligencia se basan en la utilización de un sistema de información de inteligencia que se forma con distintos datos extraídos de los datos de producción, con información relacionada con la empresa o sus ámbitos y con datos económicos.

Técnica ETL

Mediante las herramientas y técnicas ELT (extraer, cargar y transformar), o actualmente ETL (extraer, transformar y cargar) se extraen los datos de distintas fuentes, se depuran y preparan (homogeneización de los datos) para luego cargarlos en un almacén de datos.

Extraer

Se trata extraer los datos desde su sistema de origen, estos sistemas de orígenes la mayoría de las veces son más de uno.

Generalmente los sistemas de orígenes son bases de datos relacionales o no relacionales o archivos de datos. Al extraer los datos los convierte en un formato que está listo para el proceso de transformación.

Una etapa importante del proceso de extracción de los datos es la parte de analizar estos datos, comprender la naturaleza del negocio, y verificar si los datos están encaminados a lo que se espera, si no es así estos datos no son tomados en cuenta.

En esta etapa también se debe analizar la cantidad de datos que tiene el sistema de origen; si los datos que se extraen son demasiados podrían tener algún impacto sobre este sistema de origen; la solución podrían ser:

Migrar este sistema de origen a una máquina con mayores recursos o programar el uso de este sistema en horarios donde este impacto sea mínimo.

Transformar

Transformar implica la realización de manipulaciones mínimas de los datos según las reglas de negocio que conlleve; estas transformaciones pueden ser las siguientes:

La obtención de nuevos valores calculados, la unión o concatenación de dos o más columnas, identificación de columnas clave en las tablas como el nombre de la persona, etc.

Cargar

Consiste en categorizar a los datos en diferentes niveles, jerarquías o varias dimensiones de la magnitud almacenada (ejemplo totales diarios, semanales, anuales, etc)

También en esta etapa se verificarán diferentes restricciones que el sistema de base de datos de origen debería tener como: integridad referencial, campos obligatorios, valores únicos y rangos de valores.

Herramientas de Inteligencia de Negocios

Son software de aplicaciones construidos para colaborar en los procesos de Inteligencia de Negocios en las organizaciones.

Existen diversas herramientas de inteligencia de negocios, a continuación se enlistarán algunas de las herramientas de Inteligencia de Negocios que se distribuyen de forma libre:

Pentaho BI Suite, Palo, Eclipse BIRT Project, JasperReport, SpagoBI, RapidMiner, LogiReports, OpenL, entre otros.

Pentaho BI Suite

Pentaho se define a sí mismo como una plataforma de BI “orientada a la solución” y “centrada en procesos” que incluye todos los principales componentes requeridos para implementar soluciones basados en procesos y ha sido concebido desde el principio para estar basada en procesos. Las soluciones que Pentaho pretende ofrecer se componen fundamentalmente de una infraestructura de herramientas de análisis e informes integrado con un motor de workflow de procesos de negocio. La plataforma será capaz de ejecutar las reglas de negocio necesarias, expresadas en forma de procesos y actividades y de presentar y entregar la información adecuada en el momento adecuado. (Pentaho.com, 2011) (Wikipedia, 2010)

Productos de Pentaho:

Pentaho Analysis Services: cuyo nombre código Mondrian es un servidor OLAP (procesamiento analítico en línea) escrito en Java. Es compatible con el MDX (expresiones multidimensionales) y el lenguaje de consulta XML para el Análisis y especificaciones de la interfaz olap4j. (Pentaho.com, 2011)

Pentaho Data Mining: Es una envoltura alrededor del proyecto Weka. Es una suite de software que usa estrategias de aprendizaje de máquina, aprendizaje automático y minería de datos. Cuenta con series de clasificación, de regresión, de reglas de asociación, y de

algoritmos de clustering, para así apoyar las tareas de análisis predictivo. (Pentaho.com, 2011)

Pentaho Dashboard: Es una plataforma integrada para proporcionar información sobre sus datos, donde se pueden ver informes, gráficos interactivos y los cubos creados con las herramientas Pentaho Report Designer. (Pentaho.com, 2011)

Pentaho para Apache Hadoop: Es un conector de bajo nivel para facilitar el acceso a muy grandes volúmenes manejados en el proyecto Apache Hadoop, la Suite de Pentaho BI para Hadoop permite abordar los mayores desafíos que experimentan los usuarios de Hadoop -, sobre su empinada curva de aprendizaje técnico, la falta de personal técnico cualificado y la falta de disponibilidad de las aplicaciones de desarrollo y despliegue para llevar a cabo la integración de datos e inteligencia de negocios con Hadoop. (Pentaho.com, 2011)

2.2. Marco conceptual

ETL

Es el proceso que permite a las organizaciones mover datos desde múltiples fuentes, reformatearlos y limpiarlos, y cargarlos en otra base de datos, data warehouse para analizar, o en otro sistema operacional para apoyar un proceso de negocio.

Los procesos ETL también se pueden utilizar para la integración con sistemas heredados.

Almacén de datos

Es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. (Wikipedia, 2010)

Minería de datos

Consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos. (Wikipedia, 2010)

OLAP

Es una solución utilizada la Inteligencia empresarial cuyo objeto es agilizar la consulta de grandes cantidades de datos. Para ello utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP). Se usa en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares. (Bouman, 2010)

Pentaho BI Suite

Pentaho BI Suite es un aglomerado de programas gratuitos para generar inteligencia de negocios.

Pentaho Analysis Services (PAS)

Pentaho Analysis Services proporciona las capacidades OLAP de Pentaho

Plataforma. PAS permite analizar interactivamente los datos del datawarehouse estos proporcionan una interfaz de tabla de referencias cruzadas en el que diferentes dimensiones tales como tiempo, producto, cliente y pueden ser colocados. (Bouman, 2010)

Capa física

Es donde la conexión de base de datos se almacena y donde la representación física de base de datos se crea.

Capa de Negocio

Capa intermedia donde las relaciones entre tablas, fórmulas, atributos y cálculos adicionales se crean.

Vista de Negocios

Reorganiza la capa de negocio para el final usuarios y grupos de usuarios finales. (Bouman, 2010)

Cubo Multidimensional

Para un ejemplo de un cubo, podemos mostrar un cubo con tres dimensiones; éste consta de un tiempo, el producto, y el cliente dimensión, cada uno de ellos colocado en un lado del cubo. Las dimensiones son la piedra angular de cubos OLAP: un cubo es simplemente una colección de múltiples dimensiones. (Bouman, 2010)

Esquema de Análisis Tipo Estrella

Cubos MDX, dimensiones y medidas se corresponden directamente con un esquema tipo estrella, tablas de dimensiones.

En la práctica, los cubos pueden ser construidos a partir de los datos en el datawarehouse. En el caso de motores de ROLAP como Mondrian, esta asignación de las dimensiones es muy apretada, ya que los datos del cubo se construyen sobre la marcha mediante la consulta de la base de datos.

Mondrian Workbench Shema.- Esta es la herramienta visual para el diseño y prueba de los cubos estos hechos en MDX los traduce en consultas SQL.

Tablas de Dimensiones y Tablas de Hechos

Las tablas de dimensiones contienen información acerca de las entidades de negocio (clientes, productos, tiendas) y tablas de hechos acerca de los eventos de negocio (ventas, envíos, pedidos). La diferencia más notable está en las columnas medibles, tales como los ingresos, el costo y los elementos que forman parte de las tablas de hechos y todos los ángulos diferentes y atributos que son necesarios para resumir estos hechos se almacenan en las tablas de dimensiones.

Dimensiones

Las tablas de dimensiones contienen información acerca de las entidades de negocio (clientes, productos, tiendas)

Jerarquías

Una jerarquía es una estructura tipo árbol que se puede utilizar para recuperar datos desde el cubo a diferentes niveles de agregación. Por ejemplo una jerarquía de la dimensión tiempo es Año-Trimestre-Mes-Día. (Bouman, 2010)

Consulta de MDX EN FILAS y COLUMNAS

MDX puede representar la información en múltiples ejes. Por lo general, estos son los Filas y columnas. Otros ejes por defecto son páginas, secciones y capítulos, pero la mayoría de las herramientas de visualización OLAP (incluyendo JPivot) no puede trabajar con ejes más allá de los dos primeros. (Bouman, 2010)

Función filtro

La función de filtro permite limitar la selección de los miembros en el cubo. Se puede utilizar la cláusula WHERE sólo para limitar parte del cubo, pero lo que si desea para limitar los resultados a los títulos de DVD que tienen ingresos de más de \$ 10.000 en 2008, o ver sólo los títulos de películas que mostraban un aumento del 10 por ciento de los ingresos en comparación con el mes anterior? Estos son sólo dos ejemplos en los que la función de filtro puede ser útil. Filtro es una función MDX que toma dos argumentos: un conjunto y una buscar condición. (Bouman, 2010)

Miembros calculados

Un miembro calculado es un miembro de la dimensión cuyo valor se calcula en tiempo de ejecución utilizando una expresión especificada y se puede definir como miembros de

dimensiones regulares o como miembros de la dimensión. Mondrian permite agregar miembros calculados en el esquema directamente, pero esa característica utiliza la sintaxis MDX calculado sintaxis también.

Medidas

Las Medidas son parte fundamental de un Cubo de Información y son miembros medibles de lo que se necesita saber de la información almacenada en la base de datos, un ejemplo podría hacer el Total de Ventas, o el Conteo de Productos en el Inventario durante un mes. (Bouman, 2010)

Reportes Pivot

La consola de usuario del servidor de Pentaho BI ofrece la posibilidad de crear una vista de análisis, que es una tabla cruzada denominada Pivot.

Gestión del conocimiento

Es un concepto aplicado en las organizaciones. Tiene el fin de transferir el conocimiento desde el lugar dónde se genera hasta el lugar en dónde se va a emplear, e implica el desarrollo de las competencias necesarias al interior de las organizaciones para compartirlo y utilizarlo entre sus miembros, así como para valorarlo y asimilarlo si se encuentra en el exterior de estas.

Inteligencia sanitaria

Inteligencia sanitaria es "la selección y presentación de información a los agentes-profesionales clave, en forma tal que lleve a acciones que modifiquen, en el sentido deseado, la salud de los pacientes y de las poblaciones".

El objetivo de los sistemas de información en salud debe ser "brindar información oportuna y confiable a los funcionarios, personal de salud y población, que les permita tomar decisiones informadas".

La combinación de información, conocimiento y experiencia para seleccionar cursos de acción (juicio, criterio, aptitud) genera inteligencia cuando es específica para cada contexto y accionable como para aspirar a impactar sobre el comportamiento del receptor.

_ Decisión Empresarial

La Toma De Decisiones es el proceso mediante el cual se realiza una elección entre las opciones o formas para resolver diferentes situaciones de la vida en diferentes contextos: a nivel laboral, familiar, sentimental, empresarial.

La toma de decisiones consiste, básicamente, en elegir una opción entre las disponibles, a los efectos de resolver un problema actual o potencial.

La toma de decisiones a nivel individual se caracteriza por el hecho de que una persona haga uso de su razonamiento y pensamiento para elegir una solución a un problema que se le presente en la vida; es decir, si una persona tiene un problema, deberá ser capaz de resolverlo individualmente tomando decisiones con ese específico motivo.

En la toma de decisiones importa la elección de un camino a seguir, por lo que en un estado anterior deben evaluarse alternativas de acción. Si estas últimas no están presentes, no existirá decisión.

2.3. Marco Espacial

La implementación se desarrollará en un sistema de almacén de datos o base de datos relacional entre 20 y 30 tablas.

La base de datos es para un sistema de producción florícola, donde se va a contemplar el proceso de producción de flores de una finca en Pichincha - Ecuador, la producción de una finca consiste desde el tratamiento de la flor desde que se corta en los invernaderos, pasa por su agrupación y clasificación en de varios tallos, esto se llama ramos, el bodegaje de la flor y su posterior despacho.

2.4. Marco Legal

Toda la construcción del proyecto debe estar protegido en las leyes de propiedad intelectual con su respetivo registro, también con registro de marca y patentes.

3. METODOLOGÍA

3.1. Proceso de investigación

3.1.1. Unidad de Análisis

Se va a aplicar en una base de datos en el campo florícola, más específico en el campo de Pos cosecha, se necesita que se diferencia los diferentes subprocesos del área de producción y realizar las diferentes comparaciones; lo gente de la gerencia de la florícola será el personal que utilizará directamente el producto final.

3.1.2. Tipo de Investigación

Se va a utilizar la investigación aplicada ya que implicaría el uso de conocimientos establecidos, para diseñar, construir y modificar; en este caso se va a utilizar los diferentes conceptos de inteligencia de negocios y herramientas de software libre que ya existen.

3.1.3. Método

Se utilizará el método de investigación deductivo, ya que partiremos de conceptos generales para hallar una solución en particular.

3.1.4. Técnica

Se han realizado algunas entrevistas preliminares con el personal gerencial en la florícola dando a conocer los beneficios que se tendrá esta implementación y han dado la autorización necesaria para realizar la investigación en el área, se van a realizar diferentes tipos de observación tanto directa como indirectamente, también se recurrirá a unas entrevistas

preliminares para saber los procesos más básicos que se va a tratar luego se elaborará encuestas específicas para despejar dudas específicas de los que se necesita.

3.1.5. Instrumento

_ Listas de chequeo de Datos

_ Encuestas

_ Cuestionarios

_ Registros de Datos

3.1.6. Metodología Informática

Metodología AUP (Agile Unified Process)

AUP es una metodología de desarrollo muy simple, que requiere trabajo duro porque no se basa en el seguimiento de un plan, sino en la adaptación continua a las circunstancias de la evolución del proyecto. (Navegapolis, 2011)

El AUP es un desarrollo incremental y evolutivo que implica que al final de cada iteración se dispone de una parte del producto operativa que se puede inspeccionar y evaluar.

AUP es una metodología ágil, y como tal:

Es un modo de desarrollo de carácter adaptable más que predictivo.

Orientado a las personas más que a los procesos.

Emplea la estructura de desarrollo ágil: incremental basada en iteraciones y revisiones.

Intentar predecir en las fases iniciales cómo será el producto final, y sobre dicha predicción desarrollar el diseño y la arquitectura del producto no es realista, porque las circunstancias obligarán a remodelarlo muchas veces.

Para qué predecir los estados finales de la arquitectura o del diseño si van a estar cambiando.

En Agile se toma a la inestabilidad como una premisa.

El desarrollo Agile va generando el diseño y la arquitectura final de forma evolutiva durante todo el proyecto. No los considera como productos que deban realizarse en la primera “fase” del proyecto.

4. PROCESO DE DESARROLLO

4.1 Introducción

Una solución viable que se determinó fue el modelamiento y construcción de un cubo OLAP, OLAP significa, procesamiento analítico en línea (On-Line Analytical Processing, OLAP) aquí se organizan los datos contenidos en una base de datos multidimensional para que se ajusten al modo que tienen los usuarios de analizarlos: en categorías jerárquicas y en valores de resumen previamente calculados.

Un cubo OLAP proporciona sólo los resultados resumidos del gran banco de datos que se puede tener; en vez de una gran cantidad de registros individuales, para que sea posible analizar muchos datos que, de otra forma, no podrían manejar los recursos del sistema.

Cuando se crea un cubo OLAP a partir de los datos de una base de datos relacional, se llevan a cabo los pasos iniciales de la creación de informes que necesiten los usuarios.

Además de la construcción de un cubo OLAP, se busca profundizar en los conceptos del BI, desde la recolección de datos en una base de datos, la transformación de estos datos en un solo medio de almacenamiento.

4.2 Descripción Proceso de Desarrollo

El proceso que se va utilizar es AUP, es una versión simplificada del Proceso Unificado de Desarrollo RUP. Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. (Universidad Boliviana, Ingeniería de software, 2010)

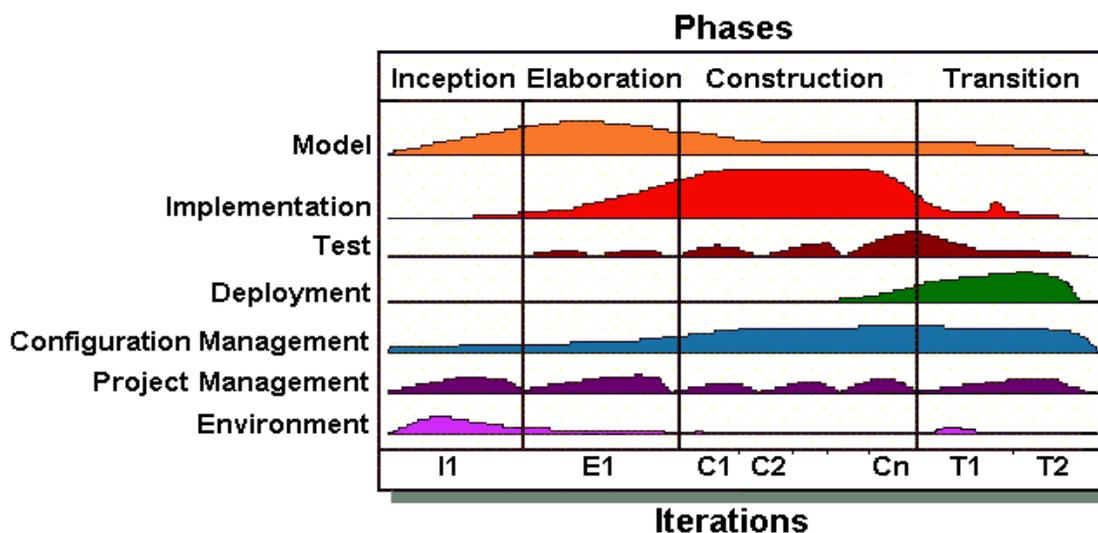


Figura 1.- Fases de la metodología AUP. Fuente <http://www.ambysoft.com>

Características AUP

- _ **Su personal sabe lo que él está haciendo:** No está enfocado en la documentación en sí, sino en el producto final y su funcionalidad.
- _ **Simplicidad:** Se describe usando poca cantidad de páginas.
- _ **Agilidad:** Está basada en los principios de los métodos ágiles en donde es desarrollado en cortos lapsos de tiempo. El software desarrollado en una unidad de tiempo es llamado una iteración, la cual debe durar de una a cuatro semanas.

_ **Poner importancia en actividades de alto valor:** Se centra en actividades reales y no en detalles.

_ **Independencia de la herramienta:** No es obligatorio utilizar un solo tipo de herramientas, el proceso de desarrollo es flexible en este aspecto. (Universidad Boliviana, Ingeniería de software, 2010)

Fases

UAP cuenta con las siguientes fases:

_ Concepción

Se Identifica el alcance inicial del proyecto, proveer una arquitectura potencial para el sistema, y obtener un financiamiento inicial del proyecto y la aceptación.

Las Tareas principales consisten en definir alcance del proyecto, estimar costos y plazos, definir riesgos, determinar factibilidad del proyecto.

_ Elaboración

Se obtiene la arquitectura del sistema.

Las Tareas principales consisten en identificar arquitectura, validar la arquitectura, desarrollar el ambiente el proyecto, Equipo del personal del proyecto.

_ Construcción

Se implementa un software sobre una base incremental la que debe estar relacionada con los objetivos de los involucrados.

Las Tareas principales consisten en el Modelado, construcción, creación de documentación de apoyo.

_ Transición

Se valida y entregar el sistema en un ambiente de producción.

Las Tareas principales consisten en el test del sistema, test de usuarios, re trabajo del sistema e instalación.

Disciplinas

Definen actividades que el equipo de desarrolladores debe realizar para construir, validar y entregar un software que satisfaga las necesidades.

_ Modelo

Entender los procesos de negocios de la organización, el dominio de problema que puede ser abordado por el software, e identificar una solución viable.

_ Implementación

Transformar los modelos en código ejecutable y aplicar pruebas básicas en unidades particulares de prueba.

_ Prueba

Realizar una evaluación objetiva para asegurar la calidad. Esto incluye encontrar defectos, validar que el sistema funcione como fue diseñado, y verificar que los requerimientos estén abordados por las funcionalidades.

_ Despliegue

Planificar la entrega del sistema y ejecutar el plan para que el sistema esté disponible para los usuarios.

_ Administración de la Configuración

Administrar el acceso a los artefactos del proyecto. Esto no solo incluye los seguimientos de las versiones de los artefactos, sino también controlar y administrar los cambios sobre ellos.

_ Administración del Proyecto

Dirigir las actividades que forman parte del proyecto. Esto incluye administración de riesgos, dirigir personas y coordinar personas con sistemas que están fuera del alcance del proyecto.

_ Ambiente

Facilitar todo el entorno que permita el normal desarrollo del proyecto. (Universidad Boliviana, Ingeniería de software, 2010)

4.3. Fase de Concepción

Esta fase tiene como objetivo identificar el alcance del proyecto, establecer plazos y entregables, estimar requisitos y recursos.

4.3.1. Documento de definición del proyecto

_ Resumen ejecutivo del proyecto

La Inteligencia de Negocios es el conjunto de productos y servicios que permiten a los usuarios finales analizar de manera rápida y sencilla, la información para la toma de decisiones de negocio a nivel operativo, táctico y estratégico.

La idea es demostrar a través de un ejemplo práctico, la utilidad que tienen las herramientas en Inteligencia de Negocios y de los beneficios que conlleva la adquisición de esta herramienta en Software Libre.

_ Visión General

Cuando se alcanza una gran cantidad de información una organización se ve en la necesidad de una mayor organización de sus datos caso contrario puede correr el riesgo de tener información en desuso y desaprovecho de información importante.

Por esta razón se concibió el concepto de Inteligencia Empresarial ya que combina las utilidades de un software con la inteligencia del ser humano, por su capacidad de introspección.

_ Objetivos del proyecto

- _ Investigar conceptos y procedimientos básicos que involucra la construcción de una solución en inteligencia de negocios.
- _ Distinguir las diferentes características de la herramienta de software libre a utilizarse.
- _ Analizar la lógica del negocio en el cuál se va a implementar la solución.
- _ Implementar una solución de un problema y mostrar como se logró esta solución facilitar el trabajo gerencial de una institución.
- _ Mostrar a través de reportes informativos, simples, gráficos y sencillos el resultado de la implementación.

_ Alcance del Proyecto

Dentro del Alcance se llegará a investigar los diferentes conceptos que tienen que saberse para aplicar una solución en Inteligencia de Negocios, posteriormente un análisis de la herramienta en software libre de BI a utilizarse.

Se logrará hallar una estrategia en Inteligencia de Negocios en una organización donde es importante implementar una solución en este ámbito.

Se construirá un cubo de información como solución a la necesidad de organización de información sobre una base de datos ya establecida.

Terminada la construcción del cubo de información se desarrollarán reportes informativos y gráficos donde se podrá observar los beneficios de este cubo de información.

Fuera del Alcance la Solución no contemplará la construcción de la base de datos.

El desarrollo de una aplicación previa, ya que estarán previamente construidas.

_ Entregables producidos

- 1) Documento de definición del proyecto
- 2) Diagrama de casos de uso
- 3) Diagrama de clases
- 4) Diagrama de base de datos
- 5) Diagrama de procesos
- 6) Arquitectura lógica de la solución
- 7) Necesidades y Requerimientos
- 8) Conceptos básicos e instrucciones sobre Pentaho antes de la construcción de la solución.
- 9) Documento de Necesidades y Requerimientos de la solución
- 10) Proceso de la construcción del cubo de información
- 11) Entrega de una capa de presentación de datos.
- 12) Documento técnica de la solución
- 13) Entrega del cubo de información

Duración estimada: 15 semanas

_ Puntos de Decisión e Hitos

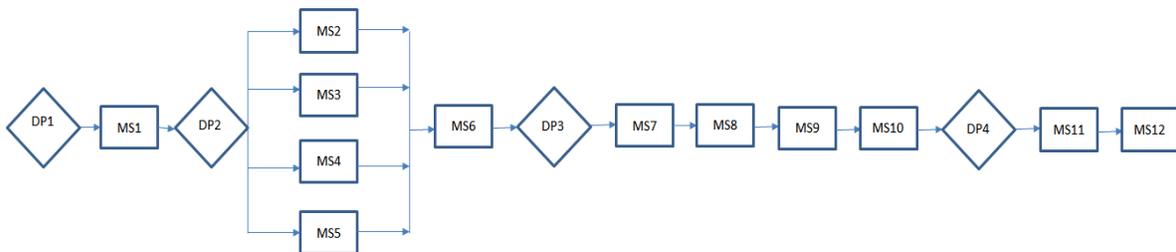
En esta sección se describe los hitos y duración del Proyecto.

Hito	Fecha de término	Entregables finalizados
MS1	8/07/2012	Documento de definición del proyecto
DP2		Aprobación de la Fase de Concepción
MS2	12/07/2012	Diagrama de casos de uso
MS3	12/07/2012	Diagrama de clases
MS4	12/07/2012	Diagrama de base de datos
MS5	12/07/2012	Diagrama de procesos
MS6	17/07/2012	Arquitectura lógica de la solución
DP3		Aprobación de la Fase de Elaboración
MS7	7/08/2012	Conceptos e instrucciones sobre Pentaho sobre la construcción de la solución.
MS8	21/08/2012	Documento de Necesidades y Requerimientos de la solución
MS9	25/09/2012	Proceso de la construcción del cubo de información.
MS10	1/10/2012	Entrega de una capa de presentación de datos.

DP4		Aprobación de la Fase de Construcción
MS11	8/10/2012	Documento técnico de la solución
MS12	11/10/2012	Entrega del cubo de información

Tabla 11.- Secuencialidad de los Entregables. Fuente Javier Salvador

La secuencialidad de los entregables y su vinculación con los hitos se describen en el siguiente gráfico.



_ Organización del proyecto

Aquí se listan los roles principales del proyecto y la gente que estará ocupando cada uno de ellos.

Una estructura organizacional apropiada es esencial para alcanzar el éxito. La siguiente lista muestra la organización propuesta para el proyecto:

Rol	Responsable
Gerente del proyecto	
Administrador del Sistema	Javier Salvador

Analista de Negocios	Javier Salvador
Analista de Sistema	Javier Salvador
Desarrollador Pentaho	Javier Salvador
Capacitador Pentaho	Javier Salvador

Tabla 12.- Cuadro de Roles. Fuente Javier Salvador.

4.4. Fase de Elaboración

Esta fase tiene como objetivo obtener la arquitectura y modelamiento del sistema.

Como la solución ya tiene un sistema de información y base de datos predeterminados, se vio necesario analizar estos recursos.

4.4.1. Diagrama de Casos de Uso

Se vio necesario realizar un diagrama de casos de uso del proceso actual de poscosecha, es de vital importancia entender y comprender a detalle del proceso actual de poscosecha, desde que la flor es cortada desde el invernadero hasta el proceso de despacho de la flor de la bodega.

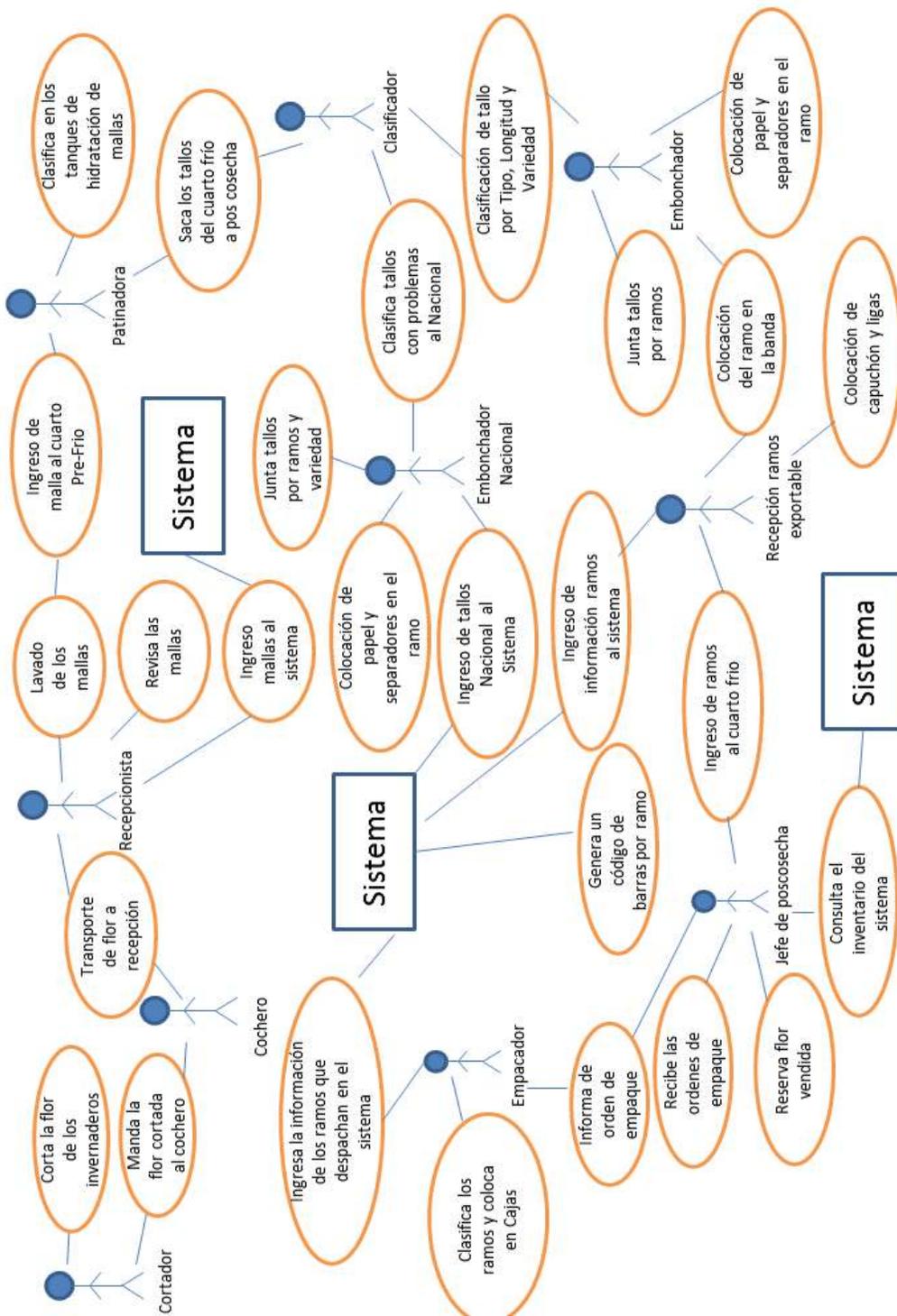
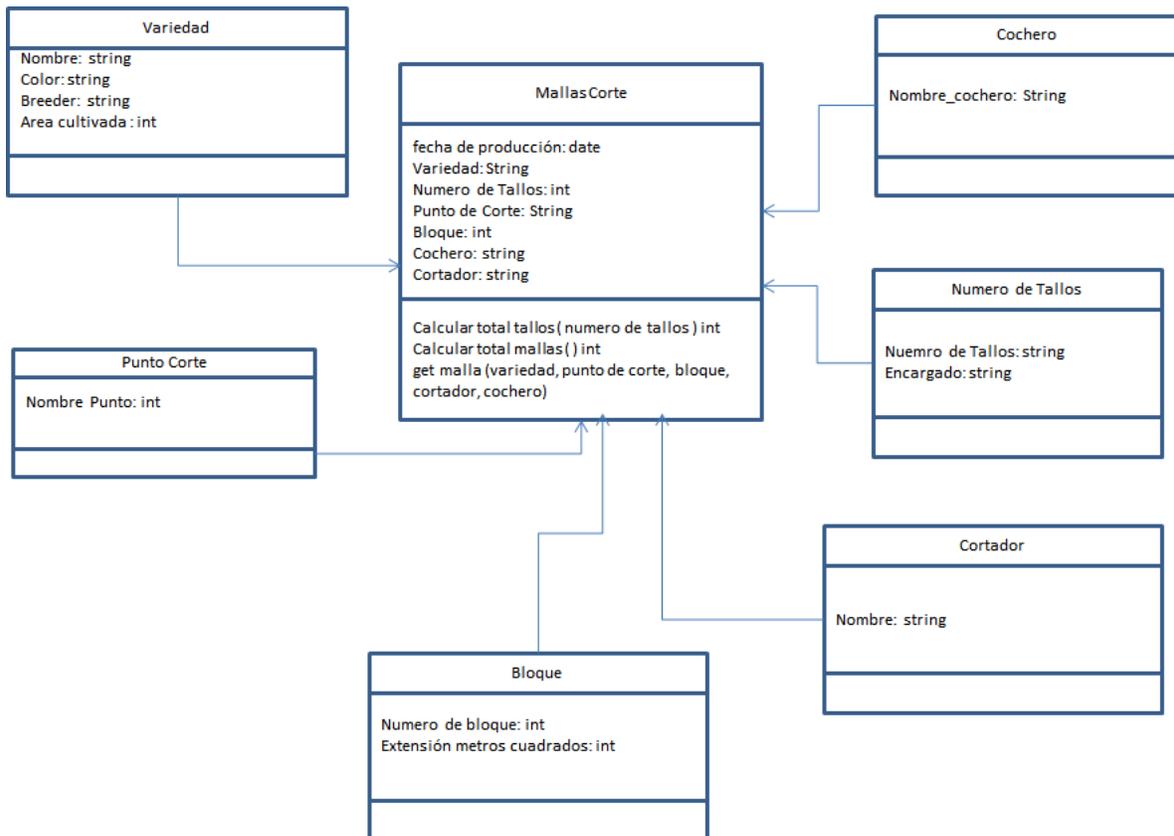


Diagrama 1.- Diagrama de Casos de Uso. Fuente Javier Salvador.

4.4.3. Diagrama de Clases

Este diagrama tiene como objetivo analizar los diferentes objetos que están involucrados en el proceso de Pos-cosecha.



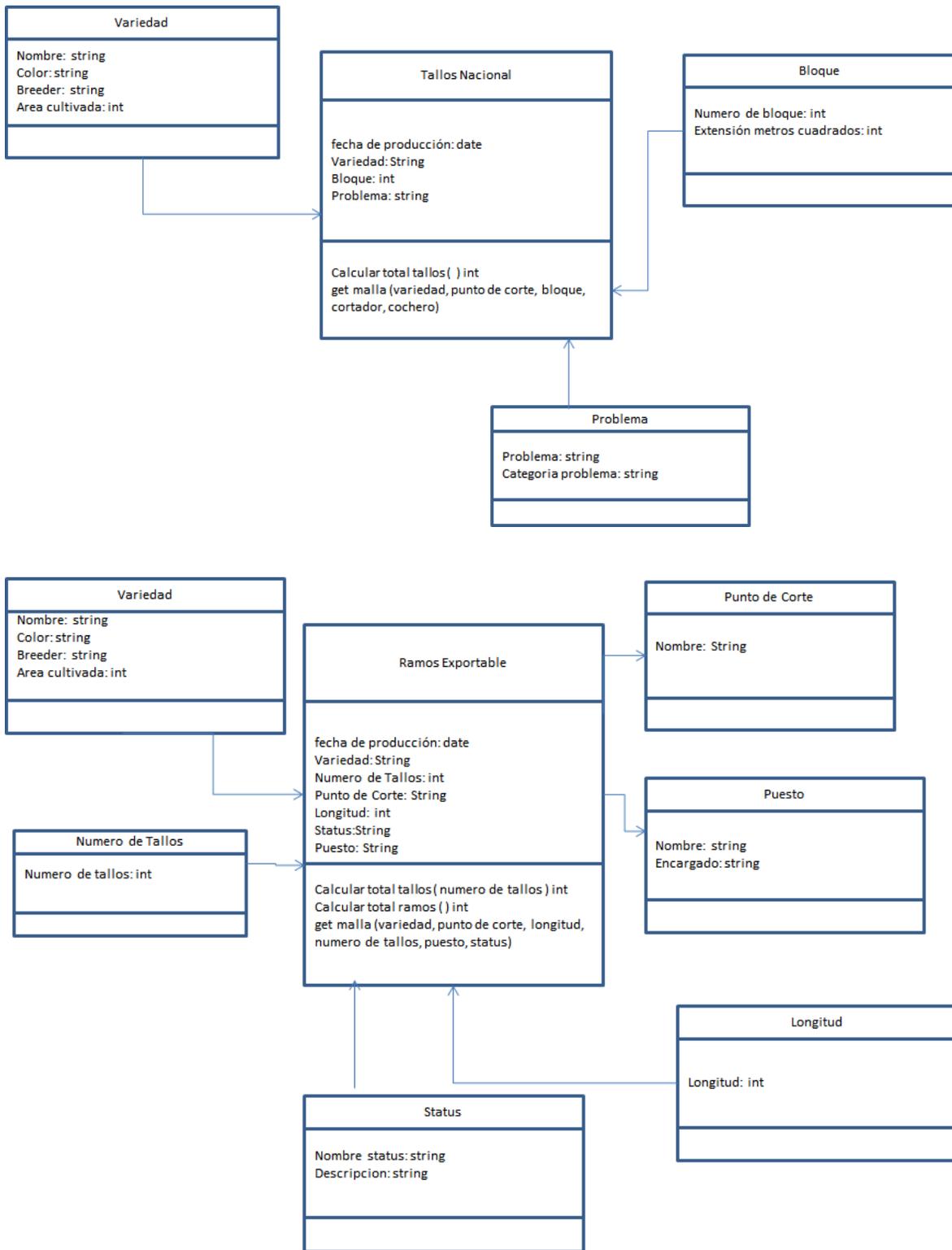


Diagrama 2.- Diagrama de Clases. Fuente Javier Salvador.

4.4.5. Diagrama de Procesos

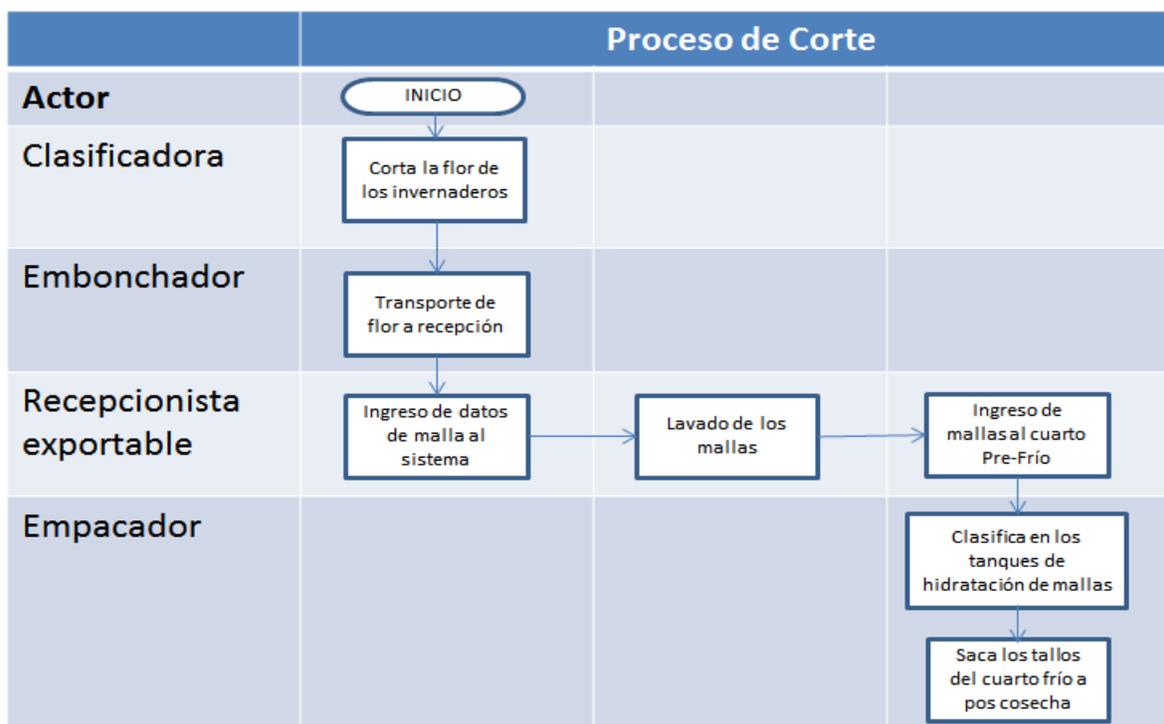
Aquí logramos mostrar los principales procesos que se identificaron:

Son 3 principales procesos que necesita a nivel gerencial saber información categorizada.

Proceso de Corte.- Es la flor que es cortada desde los invernaderos, esta información se ingresa por mallas, las mallas contienen cierto número de tallos.

Proceso Exportable.- Es la flor que está en condiciones para ser exportada, esta información es ingresada por ramos, los ramos contienen cierto número de tallos.

Proceso de Ramos Nacional.- Es la flor que no se ingresó al exportable por algún problema que tuvo o una enfermedad, esta información de ingresa por tallos.



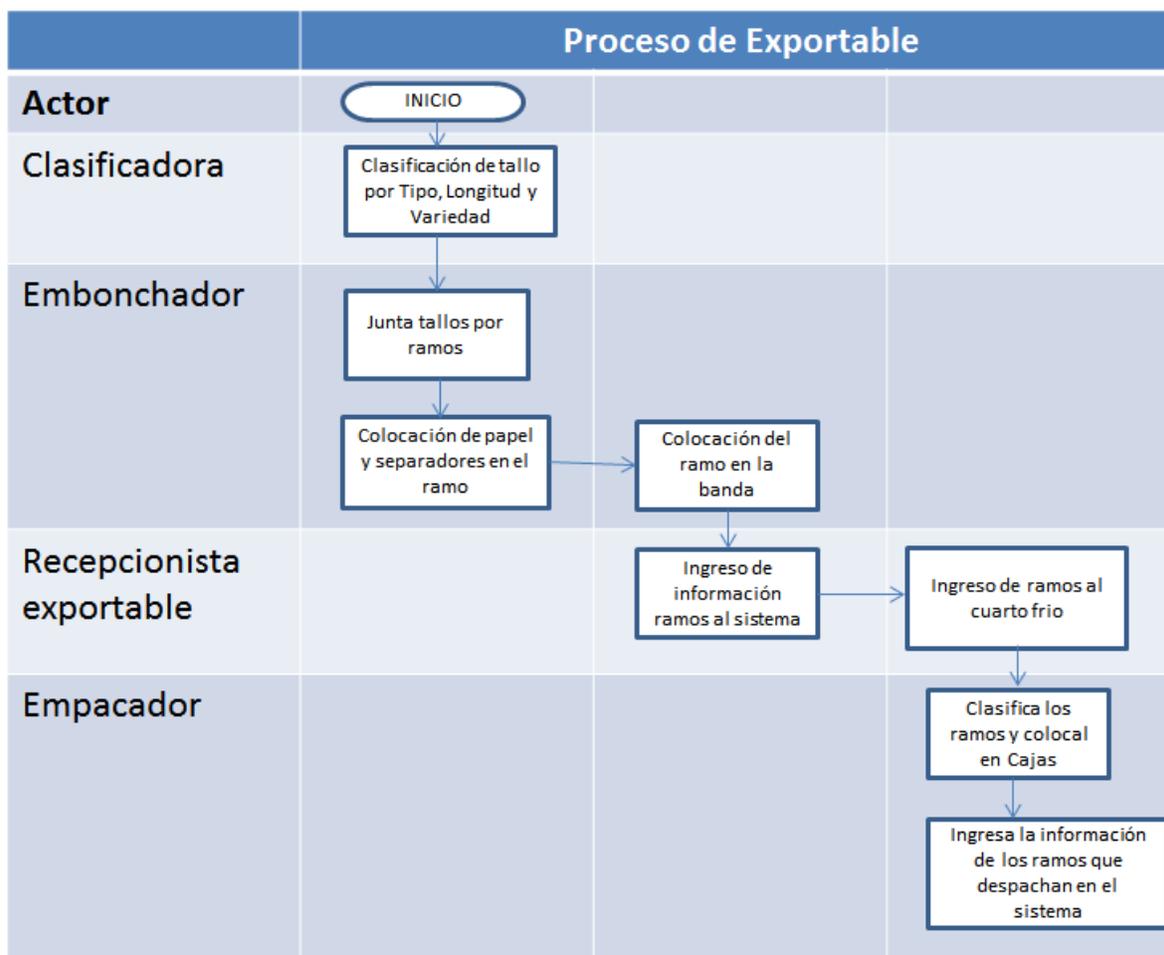
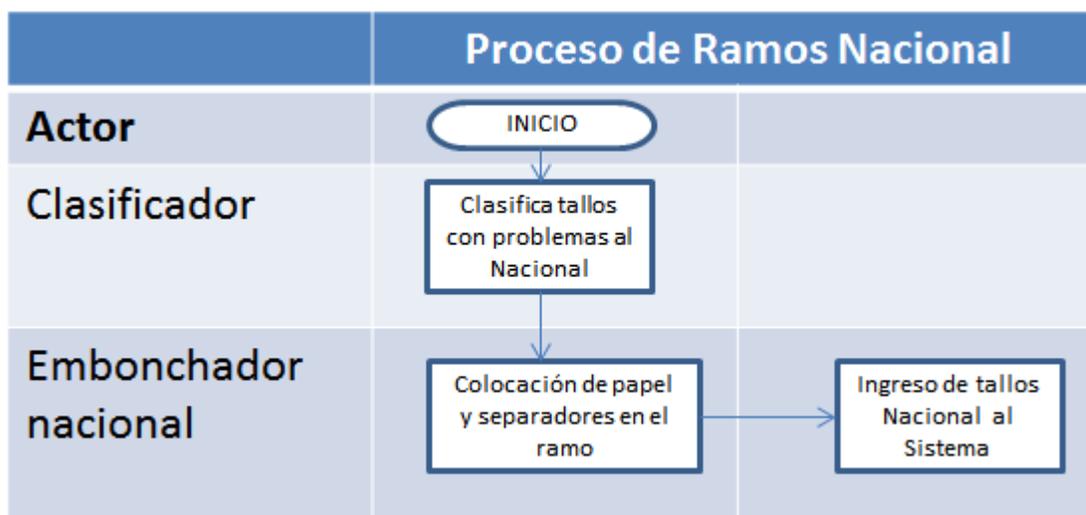


Diagrama 4.- Diagrama de Procesos. Fuente Javier Salvador.

4.4.6 Arquitectura Lógica de la Solución

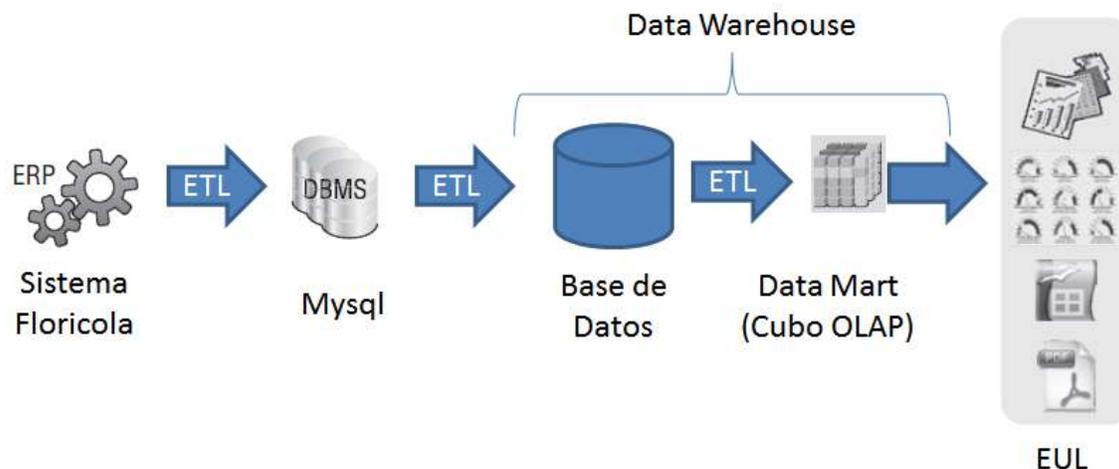


Diagrama 5.- Diagrama de Arquitectura Lógica de la Solución. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho.

ERP.- Son los sistemas de información encargados de manejar las operaciones del negocio en cual se desenvuelve, en este caso lo administra el Sistema de Información Florícola que procesa toda la información de producción de la finca.

ETL.- Significa Extract, Transform and Load, es el proceso que permite trasladar datos desde varias fuentes; en este proceso se los depura, se los reformatea y se los carga en otra fuente de almacenamiento de datos, una vez hecho esto esta información se la puede analizar o servirá para ayudar al proceso del negocio.

DBMS.- Son sistemas de gestión de bases de datos que sirven de interfaz entre el usuario y la base de datos, en este caso se utiliza MySQL.

DATA WAREHOUSE.- El almacenamiento de datos, consistente en la base de datos del almacén central.

EUL.- La capa de usuario final (EUL) con las diversas herramientas para trabajar con los datos (informes, cuadros de mando, hojas de cálculo y documentos publicados).

4.4.7. Necesidades y Requerimientos de la Solución

Entendiendo la Lógica del Negocio y la manera como se recoge la información de los ramos en producción en una finca de flores, los gerentes y administrativos de la finca buscan la siguiente información:

1. El número de Tallos Cortados desde los Invernaderos, tanto Mensual como Semanal
2. El número de Tallos Cortados desde los Invernaderos, que se diferencie su Variedad y Punto de Corte
3. El número de Tallos Cortados desde los Invernaderos, que se diferencia de que Bloque, Variedad, Años, Número de Semana en los que fueron cortados.
4. El número de Tallos Cortados desde los Invernaderos, que se agrupen por mallas.
5. Número de Plantas sembradas por Variedad.
6. Productividad del Corte = Numero de Tallos / Numero de Plantas.
7. El número de Tallos en Exportable, tanto Mensual como Semanal
8. El número de Tallos en Exportable, que se diferencie su Variedad , Punto de Corte y Longitud
9. El número de Tallos en Exportable, que se diferencie en cuantos tallos fue agrupado para conformar un ramo.
10. Productividad del Exportable = Numero de Tallos / Numero de Plantas.

11. Longitud Promedio = Longitudes sumadas del Total de Tallos / Numero de Tallos.
12. El número de Tallos ingresados al Mercado Nacional tanto mensual como semanal.
13. El número de Tallos ingresados al Mercado Nacional, que se diferencie de su Variedad y de que Bloque fue Cortado.

Una vez identificado los requerimientos de la organización definimos los siguientes cubos y medidas para su posterior construcción:

Dimensiones y Detalle de los Cubos

Cubo Exportable

Dimensiones	Jerarquía	Nivel
Fecha	Años	Mes Semana Día
	Meses	Mes Semana Día
	Semanas	Semana Día
Variedad		Nombre Color
Longitud		Número de Longitud
Número de Tallos		Número de Tallos
Punto de Corte		Nombre Punto de Corte

Cubo Corte

Dimensiones	Jerarquía	Nivel
Fecha	Años	Mes Semana Día
	Meses	Mes Semana

		Día
	Semanas	Semana Día
Variedad		Nombre Color
Bloque		Número de Bloque
Cochero		Nombre Cochero
Cortador		Nombre Cortador
Punto de Corte		Nombre Punto de Corte
Numero de Tallos		Número de Tallos

Medidas

Nombre	Aggregator	Fórmula
Tallos corte	Sum	
Plantas	Sum	
Productividad Corte		Tallos corte / Plantas
Tallos exportable	Sum	
Ramos exportable	Count	
Longitud promedio	Avg	
Productividad Corte		Tallos exportable / Plantas
Tallos Nacional	Count	

4.5. Fase de Construcción

4.5.1. Conceptos e instrucciones sobre Pentaho sobre de la construcción de la solución

El Pentaho BI Suite es un software de código abierto, usted es libre de usar y distribuir sus programas, y si lo desea, puede estudiar y modificar, incluso su código fuente.

Pentaho está programado en el lenguaje de programación Java. Antes de poder ejecutar programas Java, se necesita estar instalado Java. Para Pentaho, se necesitan al menos Java versión 1.5. (Bouman, 2010)

Inicio del servidor de Pentaho BI

Pentaho se lo puede obtener en la siguiente dirección de Internet <http://sourceforge.net/projects/pentaho>.

Directorio de instalación

El directorio descomprimido contiene dos subdirectorios: administración-console y biserver-ce.

- 1) **Administration-Console (PAC).**- Este es un servicio administrativo para la gestión y configurar el servidor de BI de Pentaho real.
- 2) **Biserver-ce (directorio principal).**- Este es el directorio verdadero de trabajo de Pentaho BI Server.

Se debe crear un directorio en un lugar de conveniencia, luego mover los directorios de administración-console y los directorios biserver-ce (incluyendo sus contenidos) de este directorio.

En el directorio biserver-ce se tiene que hacer doble clic en el script llamado start-pentaho.bat.

En primer lugar, se inicia un servidor de base de datos HSQLDB, que es utilizado por el servidor de Pentaho para almacenar los datos del sistema, así como una base de datos de muestra.

De forma predeterminada, la base de datos HSQLDB ejecuta en el puerto 9001. Usted debe asegurarse de que no se está ejecutando otro servidor en ese puerto.

En segundo lugar, se inicia un servidor Tomcat. De forma predeterminada, el servidor Tomcat es utilizado en el puerto 8080 para las peticiones web. Usted debe asegurarse de que ningún otro servidor se está ejecutando en este puerto. (Bouman, 2010)

Inicio de sesión

Después de iniciar el servidor, se puede iniciar cualquier navegador de internet. Se ingresa en el navegador a la siguiente dirección:

<http://localhost:8080>

Se le redirigirá automáticamente a la siguiente dirección:

<http://localhost:8080/pentaho/Login>

Enseguida, usted debe ver página de Bienvenida de Pentaho, allí se puede iniciar sesión en el servidor pulsando el botón grande Ingresar.

Si se pulsa el botón, aparecerá un cuadro Inicio de sesión. A partir de ahí, se puede seleccionar un nombre de usuario de la lista desplegable. Por ahora, iniciar la sesión como usuario Joe. (Bouman, 2010)

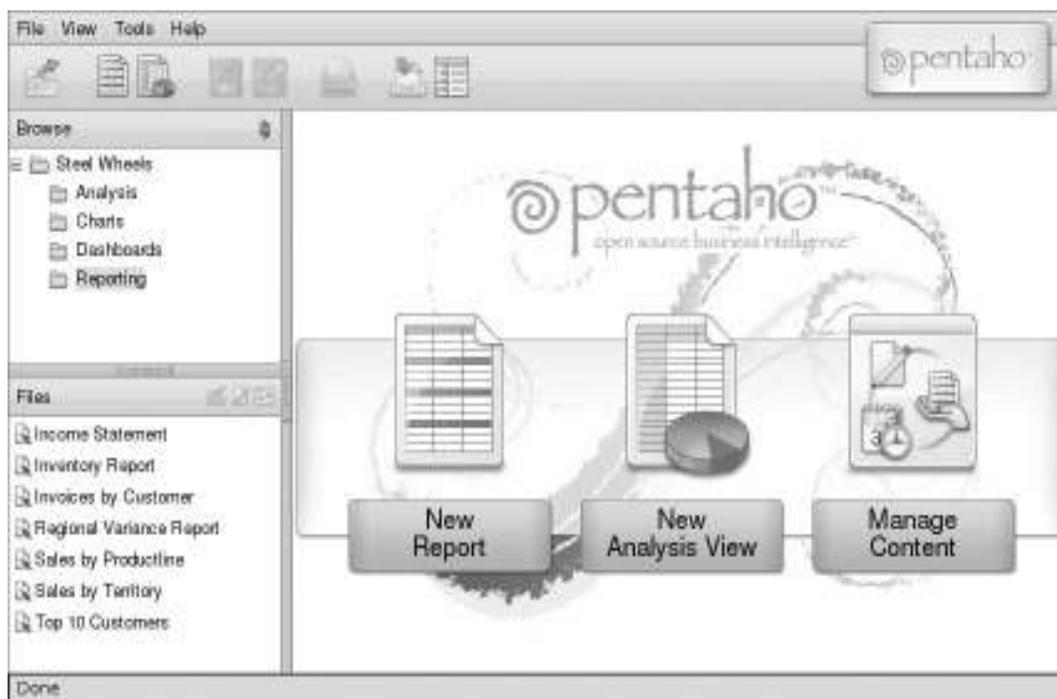


Figura 2.- Pantalla de Trabajo de Pentaho. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho.

Ya hemos probado el servidor Pentaho que esté funcionando correctamente, ahora se revisa la base de datos previamente construida; la base de datos está en un servidor mysql, puerto 3306, con el usuario root.

Para realizar la configuración y las diferentes conexiones en el directorio de administración de pentaho (administration-console).

Pentaho Administration Console (PAC)

Para iniciar el administrador de Pentaho los archivos de arranque se encuentra justo dentro de la administración-console. Los usuarios de Windows pueden iniciar PAC mediante la ejecución de startup.bat, del mismo modo, los usuarios de Windows pueden detener el PAC con el archivo stop.bat para detener el PAC.

Ahora mediante un navegador se puede ingresar al administrador.

De forma predeterminada, el PAC escucha las peticiones en el puerto 8099.

Se puede acceder con esta dirección <http://localhost:8099/>.

Al navegar en esta página, se le pide primero el nombre de usuario, el nombre de usuario y contraseñas predeterminadas son admin y password, después de acceder al sistema, se verá una página como esta.

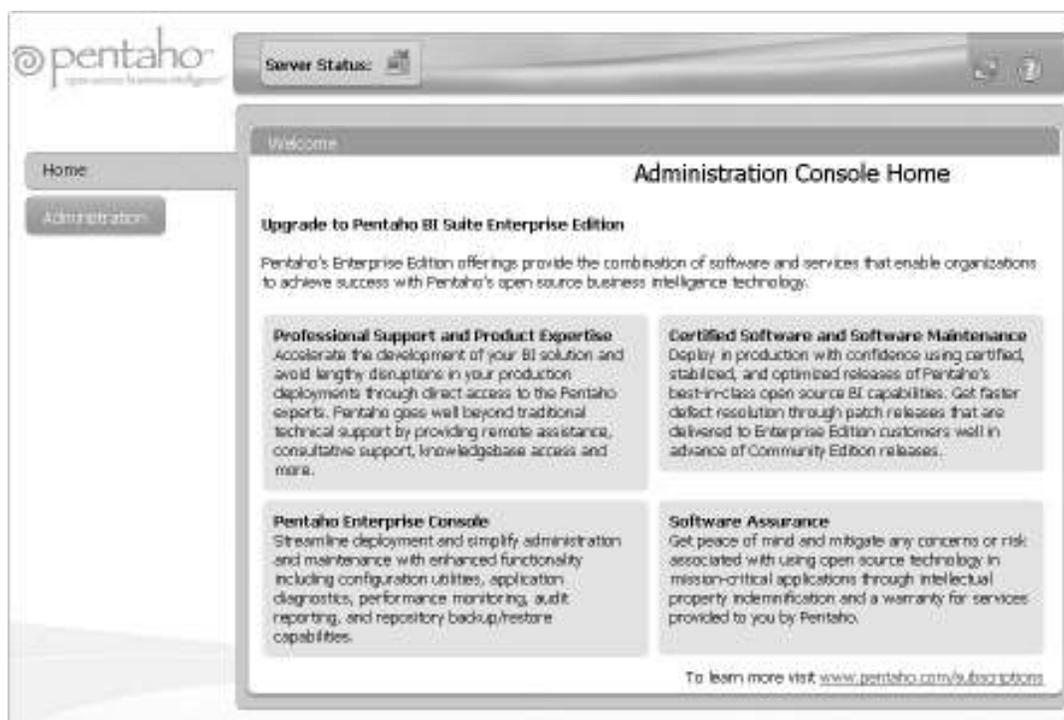


Figura 3.- Pantalla de Bienvenida de Administración de Pentaho. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Gestión de usuarios

De forma predeterminada, el servidor de Pentaho BI utiliza un sistema de autorización simple que consiste de usuarios, roles y permisos que se almacena en una base de datos relacional. PAC le permite crear roles y usuarios, y hacer que el usuario / rol se asocien. Los permisos pueden ser controlados desde la consola de usuario Pentaho (es decir, desde el lado del servidor Pentaho BI). En este caso, el usuario actual puede otorgar o revocar permisos.



Figura 4.- Pantalla de Administración de Pentaho para Usuarios y Roles. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Data Sources

PAC le permite crear y editar nombres orígenes de datos JDBC (JNDI). Puede acceder a la consola de origen de datos haciendo clic en los orígenes de datos en la parte superior de la consola de administración.

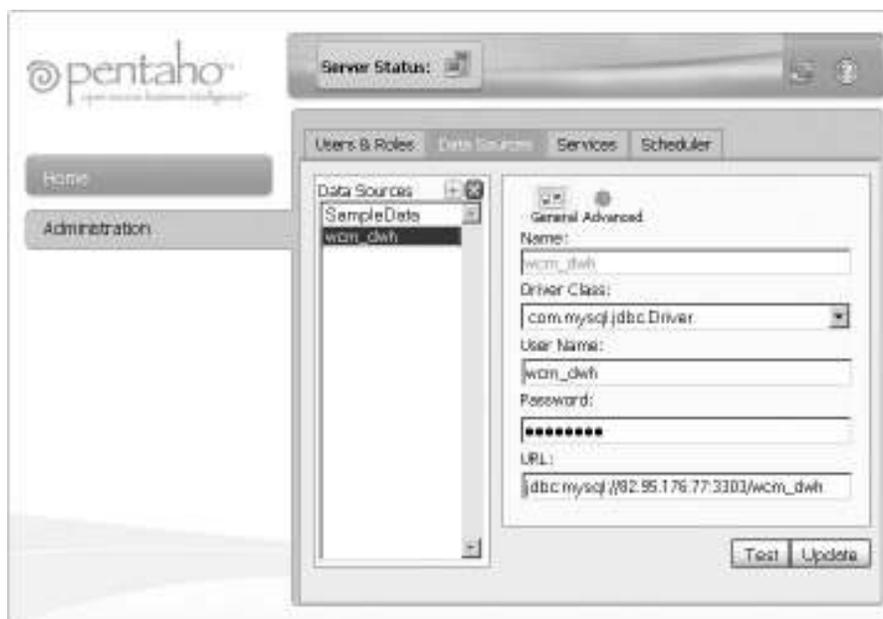


Figura 5.- Pantalla de Administración de Pentaho para Conexión con la base de datos.

Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

La lista de fuentes de datos disponibles está en el lado izquierdo de la consola. Si selecciona un elemento de la lista, puede cambiar sus propiedades en el formulario de la derecha. Datos fuentes tienen las siguientes propiedades:

Name: El nombre JNDI del origen de datos. Esto se puede utilizar en la informes y contenido BI otro para hacer referencia a este respecto.

Driver class: nombre de la clase Java que implementa el JDBC conductor. Para MySQL, puede usar `com.mysql.jdbc.Driver`.

User name: el nombre del usuario de base de datos.

Contraseña: la contraseña del usuario de base de datos.

URL: La cadena de conexión. El formato de la cadena de conexión depende sobre el conductor, y usted debe consultar la documentación del controlador para averiguar qué formato utilizar. Para MySQL, el formato es:

`jdbc:mysql://<hostname>[:<port>]/<schema_name>` (Bouman, 2010)

Capas Funcionales de Pentaho

A un nivel funcional, el servidor Pentaho se puede dividir en tres capas:

- 1) **Plataforma.-** La plataforma Pentaho BI organiza el contenido en las llamadas soluciones, estas puede ser pensado como una carpeta de sistema de archivos con todo el contenido de BI para resolver algún problema de negocio llamado action sequence.

Las action sequence se representan usando XML y se almacenan en un archivo de texto con una extensión. xaction. Las action sequence, por lo tanto también se llama xactions.

Los xactions son ejecutados por el motor de la solución de la plataforma. Cada vez que algún cliente invoca una xactions, el motor lee la definición de los xactions.

Lógicamente, las soluciones de Pentaho se almacenan y se mantienen en un repositorio de soluciones.

Físicamente, el repositorio de soluciones pueden estar almacenados en una base de datos relacional. (Bouman, 2010)

2) **Componentes BI.-** Es la base de un número de componentes que ofrecen funcionalidad de inteligencia de negocios.

Uno de los componentes principales es la MetaData.-

La MetaData consiste en tres capas:

Capa física.- Es donde la conexión de base de datos se almacena y donde la representación física de base de datos se crea.

Capa de Negocio.- capa intermedia donde las relaciones entre tablas, fórmulas, atributos y cálculos adicionales se crean.

Capa de Presentación.- Reorganiza la capa de negocio para el final usuarios y grupos de usuarios finales. (Bouman, 2010)

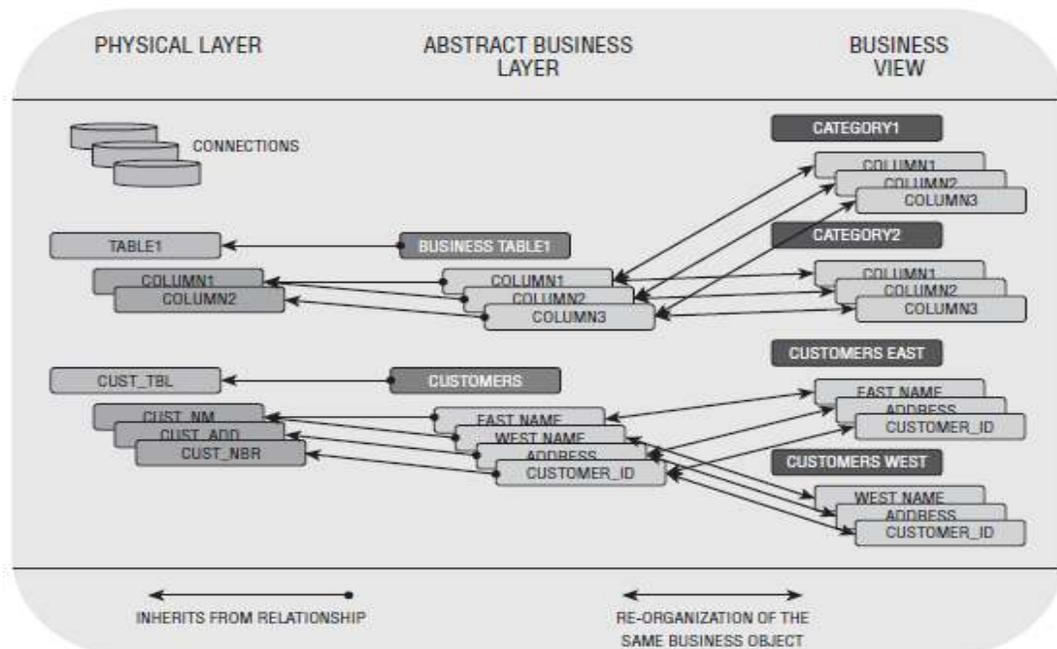


Figura 6.- Capas de la Meta Data Funcional de Pentaho. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho.

El motor OLAP.-

Mondrian es el motor de los cubos OLAP de Pentaho y es traducida en consultas en MDX a través de en un modelo multidimensional.

Esto significa que en un primer análisis se ejecuta en un modelo multidimensional.

3) Capa de presentación

Pentaho viene con una interfaz web integrada denominada consola de usuario, permite a un usuario interactuar con el servidor.

La capa de presentación puede ser utilizado para navegar y abrir el contenido existente (informes, cuadros de mando, análisis), pero en cierta medida también se puede utilizar para crear nuevo contenido de BI.

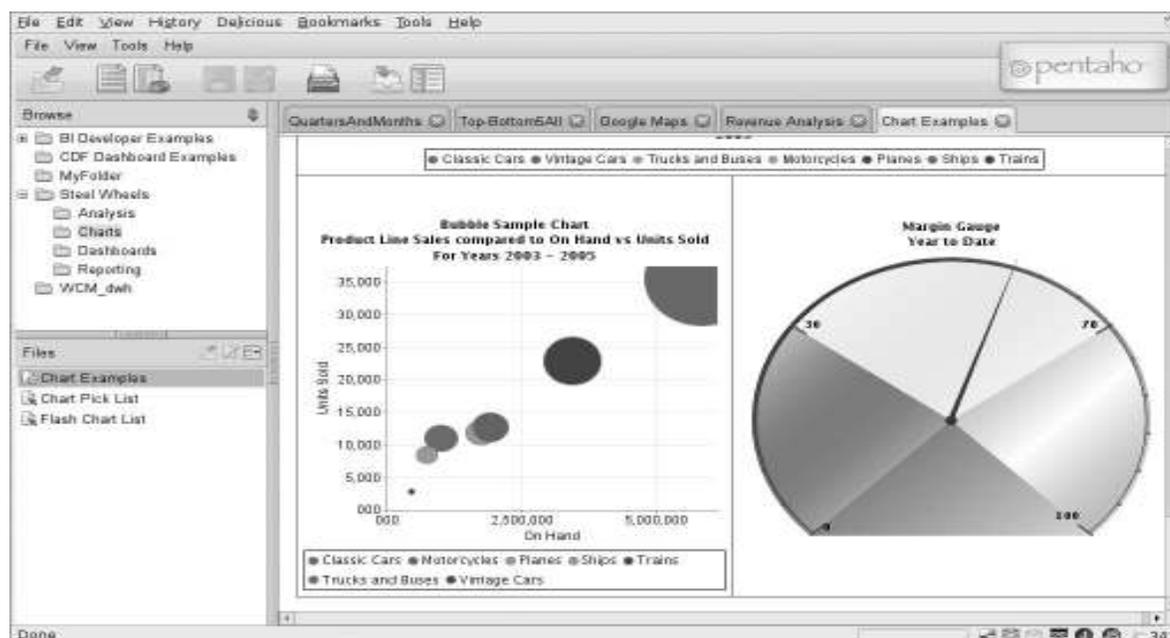


Figura 7.- Ejemplo de un Dashboard de la Presentación Final de la Solución. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho.

Este contenido en forma de informes y vistas de análisis se pueden crear en un componente de creación de informes llamado WAQR.

Pentaho proporciona una serie de programas llamado servlets que realizan una tarea específica, la capa de presentación trabaja a través de servlets que son componentes Java que funcionan independiente en el equipo local. En su lugar, se ejecutan dentro de otro programa, los servlets containers. (Bouman, 2010)

Pentaho Analysis Services (PAS)

Pentaho Analysis Services proporciona las capacidades OLAP de Pentaho

Plataforma. PAS permite analizar interactivamente los datos del datawarehouse estos proporcionan una interfaz de tabla de referencias cruzadas en el que diferentes dimensiones tales como tiempo, producto, cliente y pueden ser colocados.

Componentes de Pentaho Analysis Services

PAS consta de los siguientes cuatro componentes:

JPivot análisis.- JPivot es una herramienta de análisis basada en Java que sirve como la interfaz de usuario actual para trabajar con cubos OLAP.

Mondrian ROLAP engine.- Recibe consultas MDX de front-end de herramientas tales como JPivot.

Workbench Shema.- Esta es la herramienta visual para el diseño y prueba de los cubos hechos en Mondrian estos hechos en MDX los traduce en consultas SQL.

Aggregate Designer.- Una herramienta visual para la generación de tablas se utiliza para acelerar el rendimiento del motor de análisis.

Cubos, Dimensiones y Medidas

Cubo Multidimensional

Para un ejemplo de un cubo, podemos mostrar un cubo con tres dimensiones; éste consta de un tiempo, el producto, y el cliente dimensión, cada uno de ellos colocado en un lado del cubo. Las dimensiones son la piedra angular de cubos OLAP: un cubo es simplemente una colección de múltiples dimensiones.

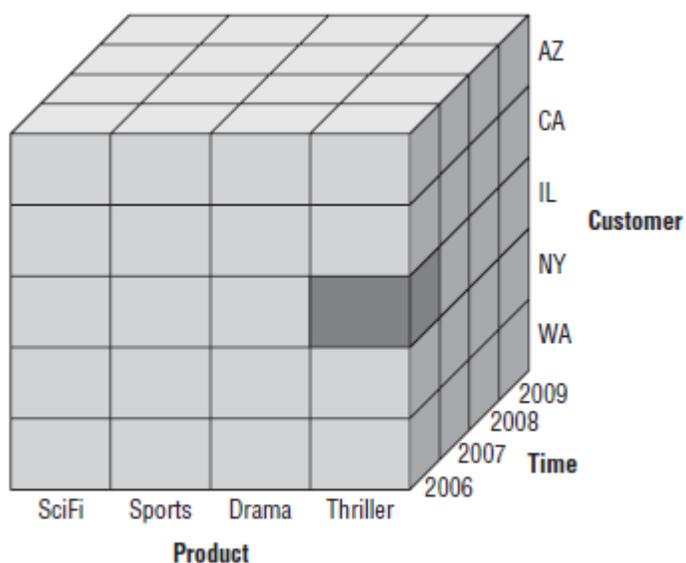


Figura 8.- Representación gráfica de un cubo multidimensional. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho.

Esquema de Análisis Tipo Estrella

Cubos MDX, dimensiones y medidas se corresponden directamente con un esquema tipo estrella, tablas de dimensiones.

En la práctica, los cubos pueden ser construidos a partir de los datos en el datawarehouse. En el caso de motores de ROLAP como Mondrian, esta asignación de las dimensiones es muy apretada, ya que los datos del cubo se construyen sobre la marcha mediante la consulta de la base de datos.

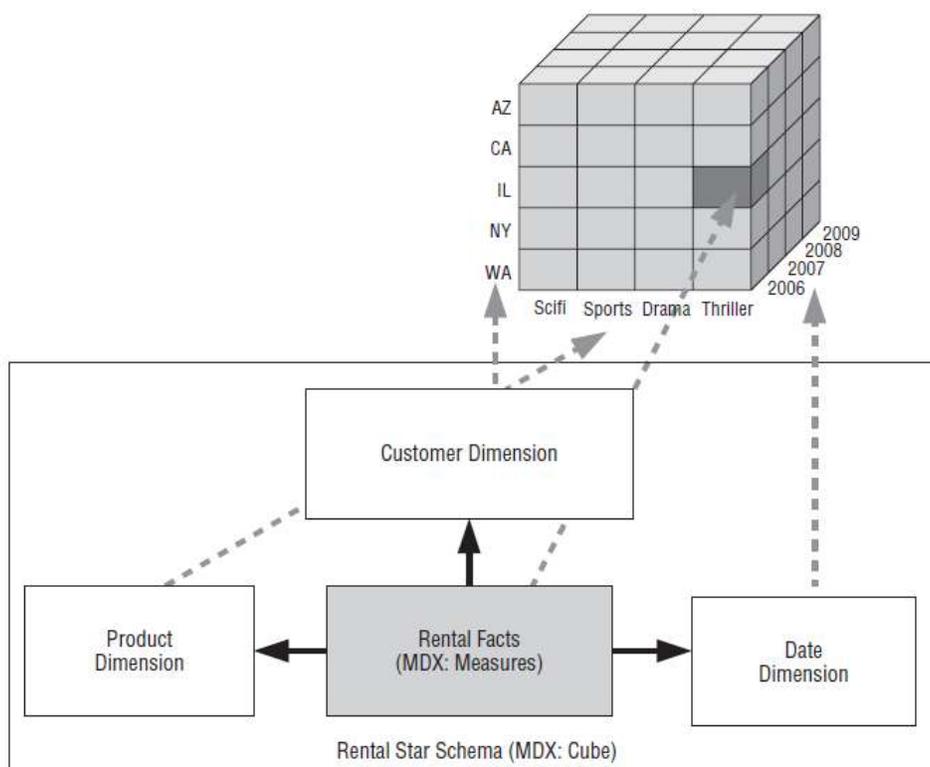


Figura 9.- Representación del Esquema Tipo Estrella. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Las jerarquías, niveles y miembros

Se necesita una forma de agregar de los datos a través de las diversas dimensiones.

Para este fin, las dimensiones se organizan en una o más jerarquías.

Jerarquías

Una jerarquía es una estructura tipo árbol que se puede utilizar para recuperar datos desde el cubo a diferentes niveles de agregación. Por ejemplo una jerarquía de la dimensión tiempo es Año-Trimestre-Mes-Día. (Bouman, 2010)

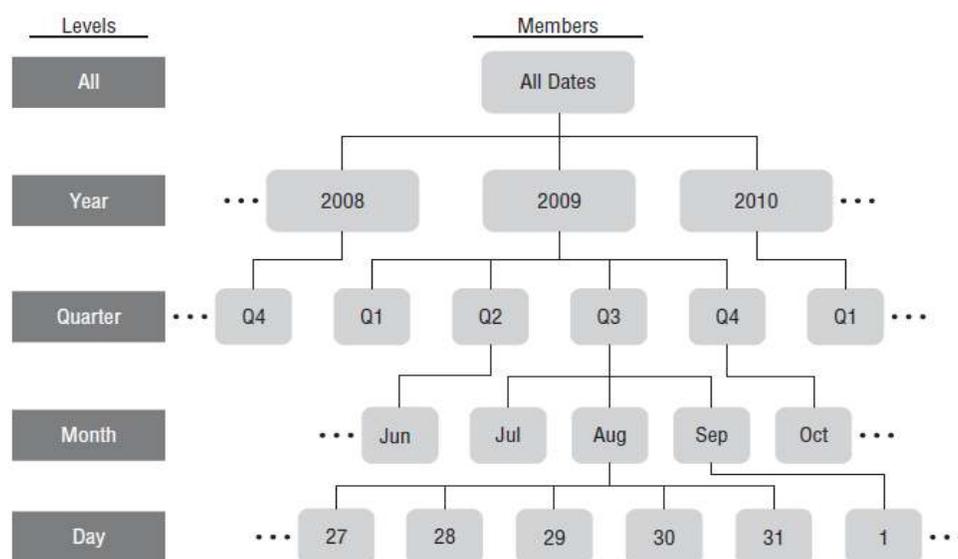


Figura 10.- Representación de Jerarquías y como se vinculan con sus niveles. Fuente

Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Uso de la sintaxis MDX sería el nombre de la dimensión, el nivel y miembro, [fecha].

[año]. [2008].

Consulta de MDX EN FILAS y COLUMNAS

MDX puede representar la información en múltiples ejes. Por lo general, estos son los Filas y columnas. Otros ejes por defecto son páginas, secciones y capítulos, pero la mayoría de las herramientas de visualización OLAP (incluyendo JPivot) no puede trabajar con ejes más allá de los dos primeros.

En la mayoría de los casos, se debe trabajar con sólo dos conjuntos de miembros a la vez, una muestra en el eje de las columnas y una en el eje de las filas. La expresión MDX por lo tanto utiliza la sintaxis genérica siguiente:

```
SELECT <member collection> ON COLUMNS,  
<member collection> ON ROWS  
FROM <cubename> (Bouman, 2010)
```

No se puede utilizar la misma dimensión en más de un eje. Por ejemplo, si utiliza la dimensión Fecha en las filas o en Columnas, ya no puede ser utilizada en el eje contrario.

Se puede restringir los datos que se puede acceder a través del cubo a un subconjunto particular (una porción), esto es denominado slicer.

Se utiliza con la palabra Where com en SQL y se la utiliza como en el siguiente ejemplo:

La siguiente consulta MDX limita el análisis al año 2008 mediante una slicer:

```
SELECT [Measures].[Revenue] ON COLUMNS,
```

```

<member collection> ON ROWS
FROM <cubeName>
WHERE [Date].[Year].[2008] (Bouman, 2010)

```

Función filtro

La función de filtro permite limitar la selección de los miembros en el cubo. Se puede utilizar la cláusula WHERE sólo para limitar parte del cubo, pero lo que si desea para limitar los resultados a los títulos de DVD que tienen ingresos de más de \$ 10.000 en 2008, o ver sólo los títulos de películas que mostraban un aumento del 10 por ciento de los ingresos en comparación con el mes anterior? Estos son sólo dos ejemplos en los que la función de filtro puede ser útil. Filtro es una función MDX que toma dos argumentos: un conjunto y una buscar condición.

```

SELECT [Country].Members ON COLUMNS,
      FILTER(
          [Title].Members,
          [Revenue]>10000
      ) ON ROWS
FROM [wcm_orders]
WHERE [Year].[2008] (Bouman, 2010)

```

Combinación de Dimensiones

MDX no se limita a mostrar una dimensión por eje, pero para realizar múltiples dimensiones en un solo eje, se debe utilizar la función CROSSJOIN. Esta función es similar a la CROSS JOIN tipo de combinación en la llanura SQL y crea una combinación de todos los valores de dos dimensiones. Por ejemplo, usando CROSSJOIN, se puede colocar [cliente]. [País] y [DVD]. [Género] en el mismo eje.

Miembros calculados

Un miembro calculado es un miembro de la dimensión cuyo valor se calcula en tiempo de ejecución utilizando una expresión especificada y se puede definir como miembros de dimensiones regulares o como miembros de la dimensión. Mondrian permite agregar miembros calculados en el esquema directamente..

WITH

MEMBER [Measures].[CoolMoviesRevenue]

AS '[Action/Adventure]+[Fantasy]+[SciFi]+[Suspense/Thriller]+[Thriller]'

SELECT NON EMPTY {[Measures].[CoolMoviesRevenue],

[Measures].[Revenue]} ON COLUMNS,

NON EMPTY {[Country].Members} ON ROWS

FROM [wcm_orders]

WHERE [Year].[2012]

Creación de los Esquemas Mondrian

Los Esquemas Mondrian se definen como documentos XML, estos esquemas pueden contruirse manualmente o a través de una interfaz gráfica o programa de edición llamado Mondrian-Schema-Workbench.

Descarga de Mondrian

Mondrian se lo puede obtener en el sitio www.sourceforge.net. Lo puedes encontrar en el sitio SourceForge buscando por la palabra Mondrian.

El sitio del proyecto Mondrian el código fuente para el motor Mondrian en sí, así como Pentaho SchemaWorkbench y el Diseñador de PentahoAggregate, descargar el SchemaWorkbench.

Instalación de Pentaho Schema Workbench

Schema Workbench se distribuyen como archivos. Zip y. Archivos tar.gz.

Los usuarios de Windows deben descargar el archivo. Zip, y los usuarios de UNIX basado en sistema deben descargar el archivo. Tar.gz. Después de la descarga, es necesario descomprimir el archivo.

Después de descomprimir el archivo el archivo es necesario colocar el archivo jdbc que es el conector de base datos, en este caso el conector Mysql correspondiente; el conector jdbc debe estar en el directorio drivers. (Bouman, 2010)

Schema-Workbench Pentaho

Para iniciar con Workbench, se debe hacer doble click en el ejecutable workbench.bat.

Aparecerá la pantalla principal del editor Workbench.

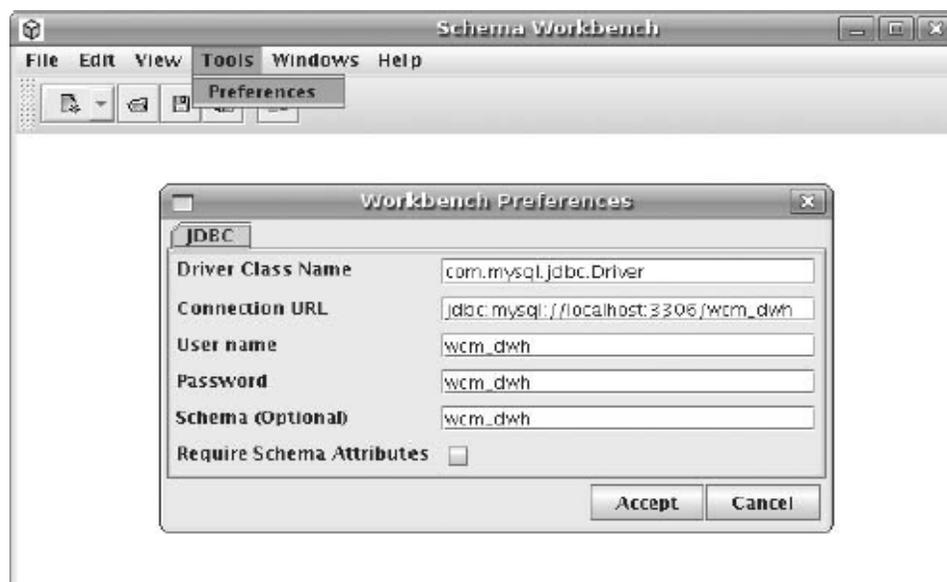


Figura 11.- Parámetros de Conexión de WorkBench. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Conexión Base de Datos

Es necesario crear una conexión de base de datos antes de iniciar con el proyecto.

Para establecer la conexión nos vamos al menú Tools / Preferences.

Driver Class Name.- Es el nombre de la clase Java del driver JDBC que va a utilizar.

Para MySQL es com.mysql.jdbc.Driver.

Connection URL.- Es el string que se utiliza para comunicarse con el servidor de base de datos. Suponiendo que usted desea conectarse a la base de datos a nivel local wcm_dwh instancia MySQL se ejecuta en el puerto por defecto 3306.

User name y password: Es el nombre y contraseña del servidor de base de datos Mysql.

Creación de un nuevo esquema

Ir al menú File / New / Schema

En un principio nuestro schema estará vacío.

En el lado derecho del Schema Workbench tiene un área de trabajo donde se puede editar elementos en el esquema.

En la parte superior del editor de esquemas existe una barra de herramientas donde contiene varios botones. La mayoría de estos botones sirven para poder añadir nuevos elementos al esquema. (Bouman, 2010)

Creando un Nuevo Cubo

Se puede agregar un cubo haciendo clic en el botón Add Cube, que es el primer botón de la barra de herramientas. El nuevo cubo se hace visible en la vista de árbol bajo el esquema como un nuevo nodo que tiene el icono de un cubo.

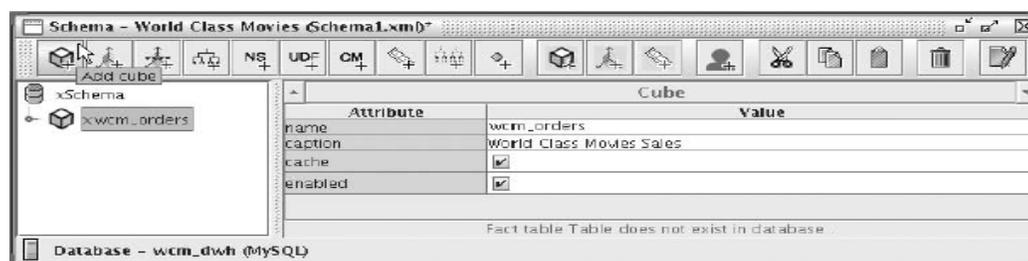


Figura 12.- Atributos principales del cubo. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Cubos de Mondrian puede tener los siguientes atributos:

Name.- Especifica el nombre que se utiliza en las consultas MDX para referirse a este cubo. Este nombre debe ser único en el esquema.

Caption.-Especifica un nombre de visualización, que será utilizado por la interfaz de usuario para presentar este cubo para el usuario final.

Cache.-Controla si los datos de la tabla deben ser almacenado en caché.

El siguiente procedimiento es elegir la tabla principal o tabla de hechos, para esto debemos crearla mediante un modelamiento llamado el Modelamiento Tipo Estrella.

Modelamiento Tipo Estrella

Se llama de Tipo Estrella porque el diagrama entidad-relación de este tipo de esquema se asemeja a una estrella. El centro de la estrella consiste en una tabla de hechos grandes y las puntas de la estrella son las dimensiones.

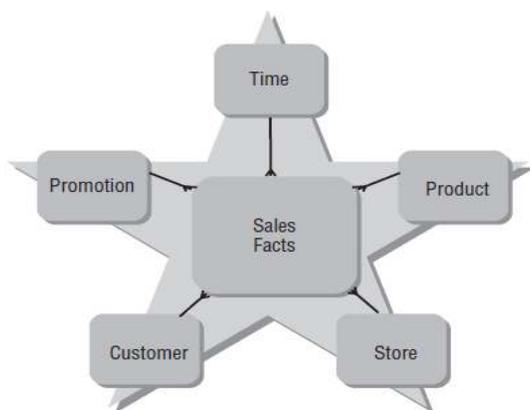


Figura 13.- Ejemplo de un Modelo Tipo Estrella donde las aristas representan las dimensiones del cubo. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Tablas de Dimensiones y Tablas de Hechos

Las tablas de dimensiones contienen información acerca de las entidades de negocio (clientes, productos, tiendas) y tablas de hechos acerca de los eventos de negocio (ventas, envíos, pedidos). La diferencia más notable está en las columnas medibles, tales como los ingresos, el costo y los elementos que forman parte de las tablas de hechos y todos los ángulos diferentes y atributos que son necesarios para resumir estos hechos se almacenan en las tablas de dimensiones.

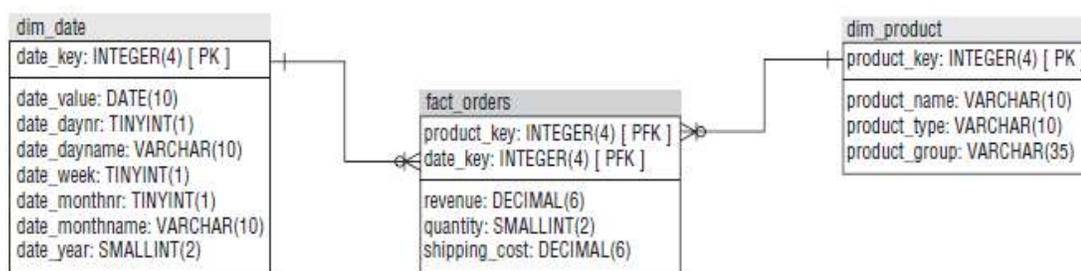


Figura 14.- Diagrama Entidad-Relación de un Esquema Tipo Estrella. . Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Realizando las Consultas para un esquema tipo Estrella

Una vez identificadas las tablas de hechos y tablas de dimensión; se debe realizar un análisis de los tipos de relaciones que tienen las tablas de dimensiones con la tabla de hechos identificada.

Si todas las tablas dimensión tienen una relación directa con la tabla de hechos se considera una relación directa caso contrario es necesario realizar algunas relaciones tipos join entre hasta encontrar el camino hacia la tabla de hechos identificada.

Esto se puede hacer a través de consultas consolidado en una Vista que contengan todas las dimensiones, esta Vista puede estar construida dentro de la estructura de base de datos y esta Vista es considerada la nueva tabla de hechos.

Escogiendo la Tabla de Hechos

Una vez identificada y construida la tabla de hechos correspondiente, debemos añadirla al cubo de información; hacemos click derecho en el cubo y agregamos un elemento tabla.

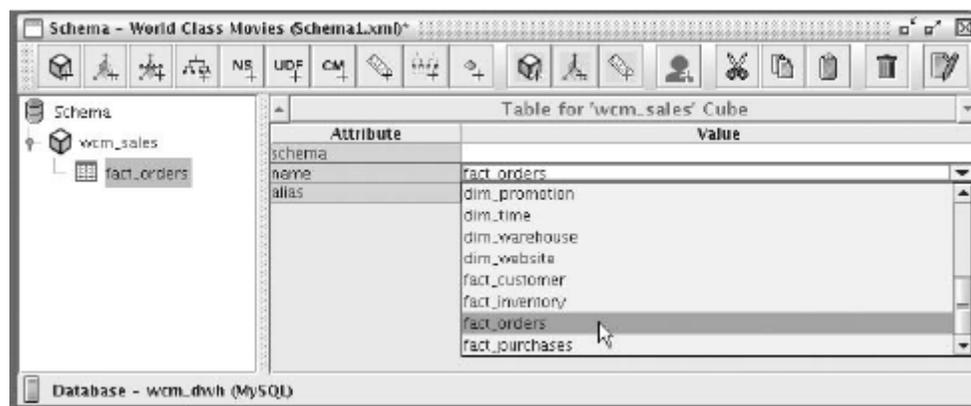


Figura 15.- Agregando un elemento tabla al nuevo Cubo. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

El nombre de la tabla es lo único que se necesita para configurar para el elemento tabla.

Sin embargo, se incluye una descripción de sus atributos:

Schema.- El identificador del esquema de base de datos que contiene la tabla de hechos. Cuando no se especifica, se asigna según lo determinado en los atributos de conexión descritos antes.

Name.- Aquí se escoge la tabla de hechos determinada en la estructura de base de datos antes construida.

Alias.- Este es el alias de la tabla de hecho.

Creando las Medidas

Las Medidas son parte fundamental de un Cubo de Información y son miembros medibles de lo que se necesita saber de la información almacenada en la base de datos, un ejemplo podría hacer el Total de Ventas, o el Conteo de Productos en el Inventario durante un mes. (Bouman, 2010)

Para agregar medidas, primero seleccione el cubo. A continuación, hacer click en el botón Agregar Medida en la barra de herramientas, el icono es una regla con un signo de más.

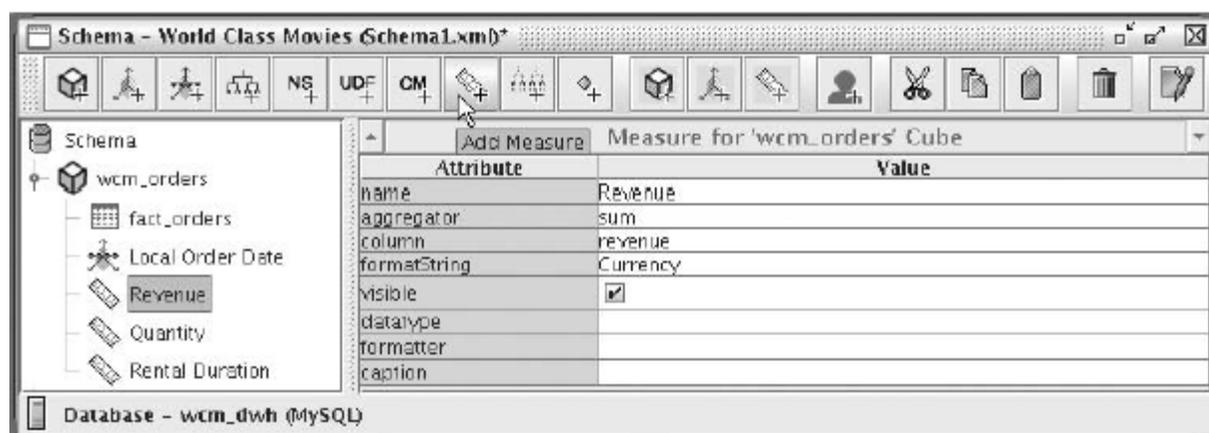


Figura 16.- Agregando un elemento Medida. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Las medidas se pueden tener los siguientes atributos:

Name.- El nombre identificador que se utiliza para referirse a esta medida en las consultas MDX. Este nombre debe ser único en el cubo.

Aggregator.- Es el nombre de la función que se utiliza para agregar la medida. Se puede elegir uno de ellos según qué es lo que se quiera medir, ya sea sumar, contar, sacar un promedio (sum, count, min, max, avg, and distinct-count). Para registrar los ingresos y cantidad, se debe elegir sum.

Column.- El nombre de la columna de la tabla de hechos del cubo. Cuando se conecta a la base de datos, el editor atributo ofrece un cuadro de lista desplegable desde el que puede elegir la columna.

formatString.- Aquí puede especificar el tipo de formato predeterminado para su visualización.

Visible.- Se especifica si la medida se visualiza para el usuario final en la interfaz de usuario.

Datatype.- Aquí se especifica el tipo de datos ya sea este tipo Cadena, Numérico, Integer, Boolean, fecha, hora.

Formatter.- Puede utilizar este atributo para especificar un tipo de formato personalizado.

Caption.- Especifica el nombre que se presenta en la interfaz de usuario. Si deja este campo en blanco, el nombre de measure se presentará en su lugar.

Creando las Dimensiones

Los esquemas Mondrian puede contener dimensiones en dos lugares:

En el interior del cubo.- Estas dimensiones se denominan dimensiones privadas debido a que sólo son conocidos por el cubo que lo contiene y no se pueden utilizar fuera del cubo.

En el interior del esquema.- Estos son dimensiones compartidas y puede ser tomada en cuenta por varios cubos o varias veces con el mismo cubo.

Para crear una dimensión compartida, primero seleccionar el esquema. Para crear una dimensión privada, seleccione el cubo que va a contener.

Luego, hacer clic en el botón Agregar Dimensión en la barra de herramientas.

Se pueden definir los siguientes atributos de dimensiones:

Name.- Para las dimensiones privadas, el nombre hace referencia a esta dimensión en las consultas MDX. Para las dimensiones comunes, el nombre se refiere cuando debe asociarla con un cubo. Para las dimensiones privadas, el nombre debe ser único entre todas las otras dimensiones utilizadas por el cubo. Para las dimensiones compartidas, el nombre debe ser único en el esquema.

ForeignKey.- Si se trata de una dimensión privada, se refiere al nombre de la columna de la tabla de hechos del cubo que corresponda a esta dimensión.

Type.- Si su dimensión es el tiempo o fecha relacionada, debe utilizar TimeDimension. Esto le permite utilizar el tiempo MDX estándar y funciones de fecha. De lo contrario, se utiliza StandardDimension.

UsagePrefix.- Esto se aplica sólo a las dimensiones particulares.

Caption.- Es el nombre para mostrar, el usuario final va a poder visualizar este nombre.

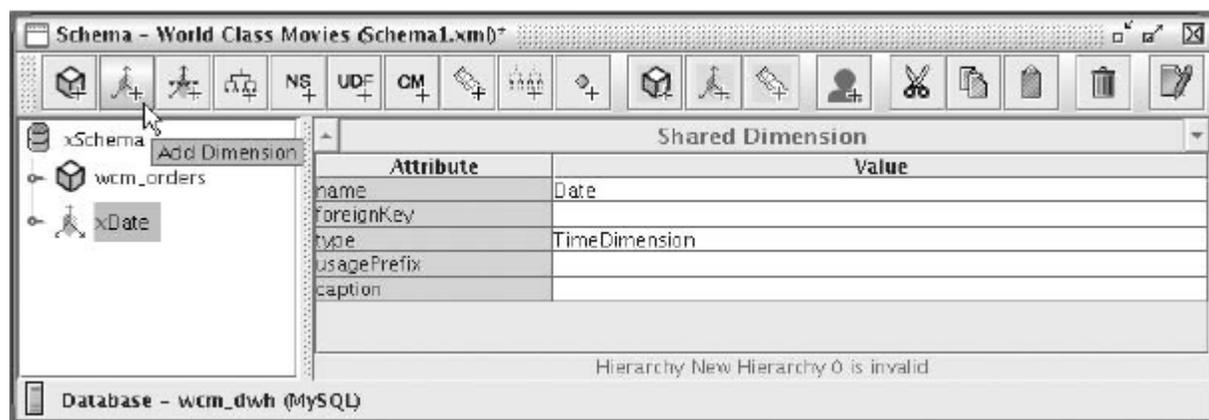


Figura 17.- Agregando un elemento Dimension. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Adición y edición de jerarquías y Elegir tablas de dimensiones

Cuando se crea una dimensión, una nueva jerarquía también se crea. Se puede ver cuando se expande el nodo dimensión.

Además, un nodo de tabla se crea automáticamente bajo el nodo de jerarquía. Antes de modificar el nodo de jerarquía, lo mejor es configurar el nodo de la tabla subyacente.

El nodo de tabla representa la tabla de dimensiones que proporcionará los valores de los niveles de la jerarquía. El procedimiento para configurar la tabla es exactamente el mismo que el procedimiento para elegir una tabla de hechos para un cubo.

Las jerarquías tienen los siguientes atributos:

Name.- El nombre utilizado para referirse a la jerarquía. Debe ser único dentro de la dimensión. Si se omite el nombre por defecto adoptará el mismo nombre que su dimensión.

Caption. - El nombre que se utiliza para presentar esta jerarquía para el usuario final

en la interfaz de usuario.

HasAll.- Una bandera que indica si la jerarquía debe tener un nivel.

AllMemberName. – Si HasAll está activado, especifica el identificador de MDX que se va a utilizar.

AllMemberCaption.- Si HasAll está habilitado, puede utilizar esta opción para especificar el nombre que se utiliza para el usuario final en la interfaz de usuario.

AllLevelName.- El nombre utilizado para hacer referencia al nivel en todas las consultas MDX. **DefaultMember.**- El nombre del miembro predeterminado.

PrimaryKeyTable.- Puede usarse para especificar el nombre de la tabla de la que está consultando.

PrimaryKey.- Normalmente, debe utilizar esta opción para especificar el nombre de la columna de clave principal de la tabla de dimensiones de esta jerarquía. Para ser exactos: es el nombre de la columna de la tabla de dimensión que hace referencia a las filas de la tabla de hechos.

Para configurar la primera jerarquía de la dimensión Date, es necesario especificar las siguientes propiedades:

Name.- Esto debe ser meses, de acuerdo con el diseño del cubo se menciona en la Tabla 15-1.

HasAll .- Esto debe estar habilitado.

PrimaryKey. - Esto se debe establecer en date_key, que es la clave principal de la tabla de dimensiones, que hemos configurado para esta jerarquía. (Bouman, 2010)

Publicar el Schema en la Solución de Mondrian

Cuando se esté satisfecho con el diseño del cubo, se puede publicar el cubo en el servidor de BI de Pentaho. Para publicar seleccionar File / Publicar en el menú principal.

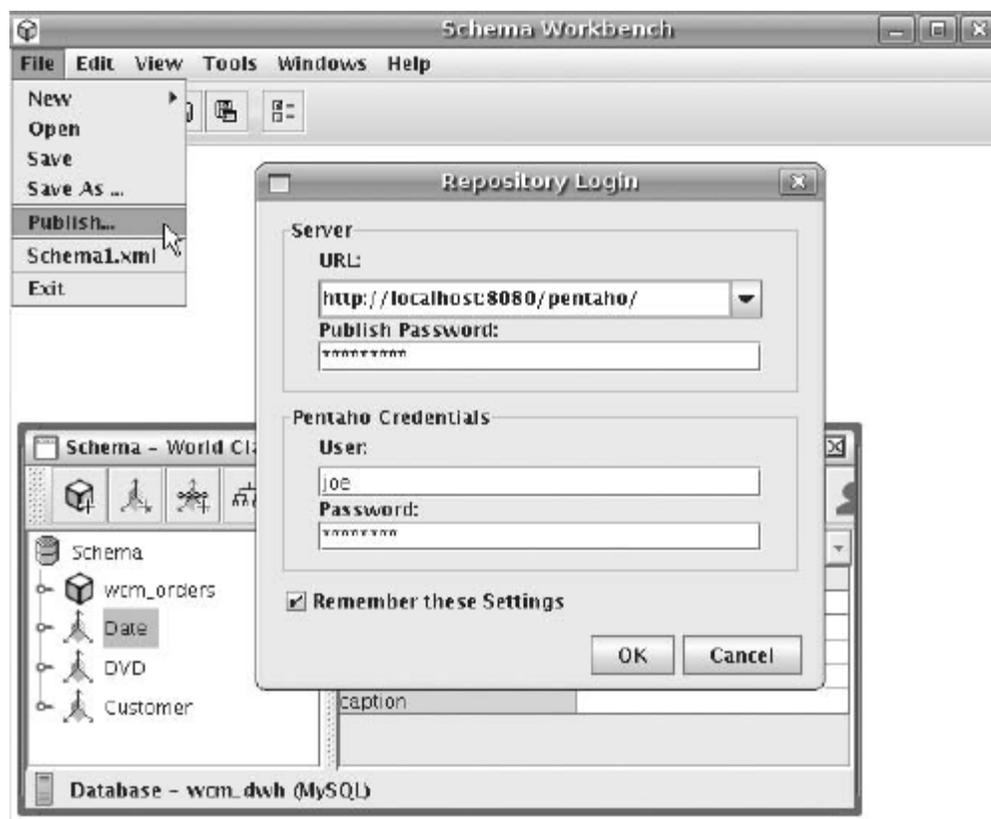


Figura 18.- Ventana de Publicación de la Solución. Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Para la dirección URL, se especifica la dirección Web del servidor de Pentaho BI a la que desea publicar el esquema. Se debe utilizar la contraseña que ha especificado en el archivo de publisher_config.xml del servidor. Para obtener el nombre de usuario y contraseña, indique las credenciales de cualquier usuario que tiene el privilegio de

publicar. Si la conexión se realiza correctamente, aparece un cuadro de diálogo que permite navegar por el repositorio de la solución.

En la parte inferior del cuadro de diálogo, se puede especificar cuál fuente JNDI proviene.

Si se activa la casilla de verificación Registrar XMLA, el esquema será registrado como un OLAP.

Vista de Análisis

La consola de usuario del servidor de Pentaho BI ofrece la posibilidad de crear una vista de análisis, que es una tabla cruzada denominada JPivot.

Ejecutar el servidor de Pentaho en el directorio de trabajo de Pentaho BI Server (ejecutar el archivo bi-server).

Luego de haber ingresado a la pantalla principal de Pentaho dar clic en el icono de Vista de análisis en la barra de herramientas o en la página de área de trabajo inicial. Pedirá que elija un esquema, y dentro del esquema, un cubo.

Después de elegir el esquema y el cubo, aparece la tabla denominada JPivot. Inicialmente, las dimensiones se muestran en el eje vertical, y la medida predeterminada se muestra en el eje horizontal.

			Measures
Date	DVD	Customer	● Revenue
+ All Date.Monthss	+ All DVDs	+ All Customers	10,921,326.03

Slicer:

Figura 19.- Ejemplo de una Vista de Análisis generada, mostrando una tabla Jpivot.

Fuente Pentaho Solutions, Business Intelligence and Data Warehousing with Pentaho

Si ejecuta la tabla JPivot significa que el cubo se publicó correctamente, sin obviar posteriores cambios en el diseño del cubo, que pueden existir.

4.5.2. Documento de Necesidades y Requerimientos de la solución

Siguiendo el concepto ETL de Inteligencia de Negocios, faltaría definir algunas medidas, dimensiones que contendrá la solución, primero es fundamental enlistar las necesidades que se obtuvo previa indagación en la organización.

_ Necesidades y Requerimientos de la Solución

Entendiendo la Lógica del Negocio y la manera como se recoge la información de los ramos en producción en una finca de flores, los gerentes y administrativos de la finca buscan la siguiente información:

1. El número de Tallos Cortados desde los Invernaderos, tanto Mensual como Semanal
2. El número de Tallos Cortados desde los Invernaderos, que se diferencie su Variedad y Punto de Corte

3. El número de Tallos Cortados desde los Invernaderos, que se diferencia de que Bloque, Variedad, Años, Número de Semana en los que fueron cortados.
4. El número de Tallos Cortados desde los Invernaderos, que se agrupan por mallas.
5. Número de Plantas sembradas por Variedad.
6. Productividad del Corte = Numero de Tallos / Numero de Plantas.
7. El número de Tallos en Exportable, tanto Mensual como Semanal.
8. El número de Tallos en Exportable, que se diferencie su Variedad , Punto de Corte y Longitud
9. El número de Tallos en Exportable, que se diferencie en cuantos tallos fue agrupado para conformar un ramo.
10. Productividad del Exportable = Numero de Tallos / Numero de Plantas.
11. Longitud Promedio = Longitudes sumadas del Total de Tallos / Numero de Tallos.
12. El número de Tallos ingresados al Mercado Nacional tanto mensual como semanal.
13. El número de Tallos ingresados al Mercado Nacional, que se diferencie de su Variedad y de que Bloque fue Cortado.

Estás son la mayoría en otras las necesidades que necesitan ser consultados por la solución BI.

_ Tabla de Dimensiones de los Cubos

Cubo Exportable

Dimensiones	Jerarquía	Nivel
Fecha	Años	Mes Semana Día
	Meses	Mes Semana Día
		Semanas
Variedad	Nombre	
Longitud	Número de Longitud	
Número de Tallos	Número de Tallos	
Punto de Corte	Nombre Punto de Corte	

Tabla 13.- Cuadro de Dimensiones Exportable. Fuente Javier Salvador

Cubo Corte

Dimensiones	Jerarquía	Nivel
Fecha	Años	Mes Semana Día
	Meses	Mes Semana Día
		Semanas
Variedad	Nombre Color	
Bloque	Número de Bloque	
Cochero	Nombre Cochero	
Cortador	Nombre Cortador	
Punto de Corte	Nombre Punto de Corte	
Numero de Tallos	Número de Tallos	

Tabla 14.- Cuadro de Dimensiones Corte. Fuente Javier Salvador

_ Tabla de Medidas

Nombre	Aggregator	Fórmula
Tallos corte	Sum	
Plantas	Sum	
Productividad Corte		Tallos corte / Plantas
Tallos exportable	Sum	
Ramos exportable	Count	
Longitud promedio	Avg	
Productividad Corte		Tallos exportable / Plantas
Tallos Nacional	Count	

Tabla 16.- Selección de Medidas / Measures. Fuente Javier Salvador

4.5.3. Construcción del Cubo de Información

Una vez identificado los diferentes cubos, dimensiones y medidas que necesitamos, se requiere identificar en donde se encuentran esa información en la base de datos.

_ Selección de Tablas de Hecho y Tablas Dimensión

Después de hacer un recorrido de la base de datos en producción el cubo requiere de las siguientes tablas:

Tablas para Cubo Mallas Corte

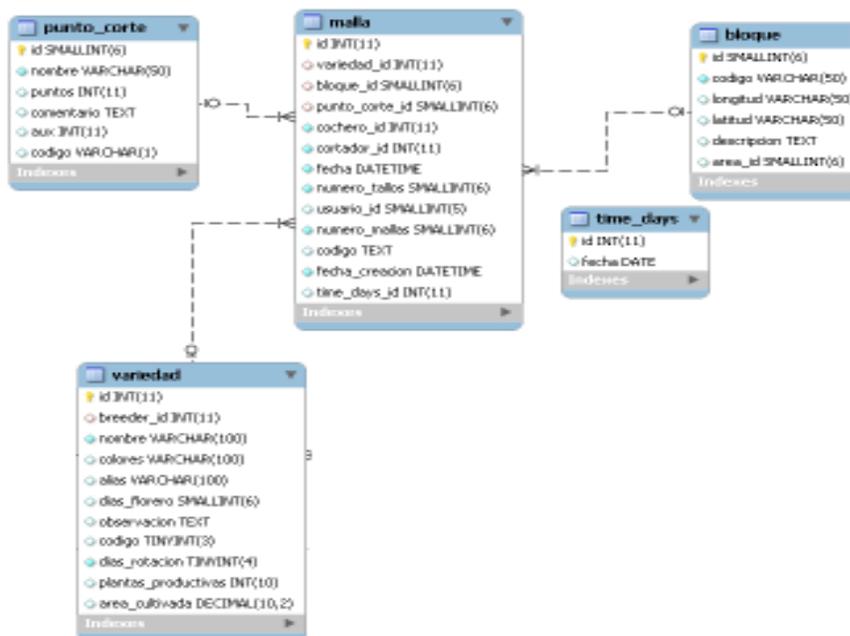


Diagrama 6.- Tablas para realizar el cubo Corte. Fuente Javier Salvador.

Esta es la parte de la base de datos que necesitamos para diseñar el Cubo de Corte

Tabla de Hecho.- Malla (Ingreso de Tallos Cortados de los invernaderos)

Tablas de Dimensión.- punto_corte, variedad, bloque, tiempo en días.

Tablas para Cubo Exportable

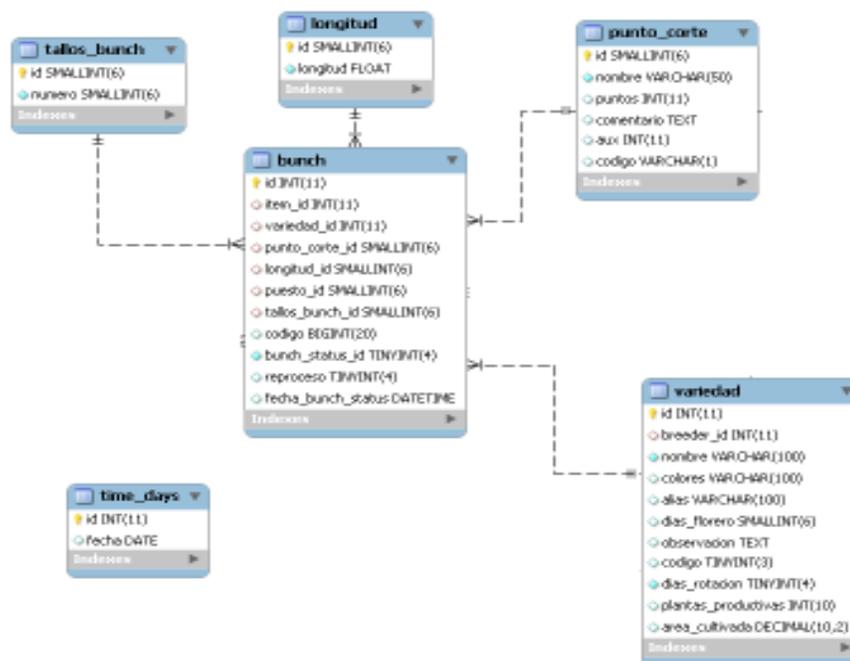


Diagrama 7.- Tablas para realizar el cubo Exportable. Fuente Javier Salvador

Esta es la parte de la base de datos que necesitamos para diseñar el Cubo de Exportable.

Tabla de Hecho.- Bunch (Ingreso de Tallos Seleccionados para el mercado de Exportación).

Tablas de Dimensión.- punto_corte, variedad, longitud, tallos_bunch, tiempo en días.

_ Conformación del Modelo Tipo Estrella

Después de identificar las tablas de hechos y las tablas para las dimensiones, se procede a realizar el modelamiento tipo estrella, para esto se necesita unir todas las tablas relacionadas en una sola; para esto se vio conveniente realizar una vista por cada cubo a realizarse y allí realizar las uniones; haciendo esto las nuevas tablas de hecho serán estas vistas.

Vista para el Cubo Corte

```
DROP VIEW IF EXISTS `florprueba`.`view_malla`;  
  
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`%` SQL SECURITY  
INVOKER VIEW `florprueba`.`view_malla` AS  
  
select `malla`.`id` AS `id`,  
  
`malla`.`variedad_id` AS `variedad_id`,  
  
`malla`.`bloque_id` AS `bloque_id`,  
  
`malla`.`punto_corte_id` AS `punto_corte_id`,  
  
`malla`.`cochero_id` AS `cochero_id`,  
  
`malla`.`cortador_id` AS `cortador_id`,  
  
`malla`.`fecha` AS `fecha`,
```

```

`malla`.`numero_tallos` AS `numero_tallos`,

`malla`.`usuario_id` AS `usuario_id`,

`malla`.`numero_mallas` AS `numero_mallas`,

(`malla`.`numero_tallos` * `malla`.`numero_mallas`) AS `numero`,

`malla`.`codigo` AS `codigo`,

`malla`.`problema_id` AS `problema_id`,

`malla`.`fecha_creacion` AS `fecha_creacion`,time_format(`malla`.`fecha`, '%H:%i')
AS `hora`,

concat(trim(concat(`cochero`.`apellido1`,`cochero`.`apellido2`)),' -
`,`cochero`.`nombres`) AS `cochero`,

`bloque`.`codigo` AS `bloque`,

concat(trim(concat(`cortador`.`apellido1`,`cortador`.`apellido2`)),' -
`,`cortador`.`nombres`) AS `cortador`,

`variedad`.`nombre` AS `variedad`,`punto_corte`.`nombre` AS
`punto_corte`,`usuario`.`usuario` AS `usuario`,

`malla`.`time_days_id` AS `time_days_id`

from (((((`malla` join `persona` `cochero` on((`malla`.`cochero_id` = `cochero`.`id`)))
join `bloque` on((`malla`.`bloque_id` = `bloque`.`id`))) join `persona` `cortador`
on((`malla`.`cortador_id` = `cortador`.`id`))) join `variedad` on((`malla`.`variedad_id`

```

```
= `variedad`.`id`))) join `punto_corte` on((`malla`.`punto_corte_id` =
`punto_corte`.`id`))) join `usuario` on((`malla`.`usuario_id` = `usuario`.`id`));
```

Como se puede apreciar en esta vista se generan las uniones con todos los elementos que se necesita para el Cubo Corte.

Vista para el Cubo Exportable

```
`base_flor`.`view_bunch_exportable` AS
```

```
Select
```

```
`bunch`.`id` AS `id`,
```

```
`bunch`.`variedad_id` AS `variedad_id`,
```

```
`bunch`.`punto_corte_id` AS `punto_corte_id`,
```

```
`bunch`.`longitud_id` AS `longitud_id`,`bunch`.`
```

```
`bunch`.`puesto_id` AS `puesto_id`,
```

```
`bunch`.`tallos_bunch_id` AS `tallos_bunch_id`,
```

```
time_format(`Bunch`.`fecha`,`%H:%i`) AS `hora`,
```

```
`variedad`.`nombre` AS `variedad`,
```

```
`punto_corte`.`nombre` AS `punto_corte`,
```

```
`longitud`.`longitud` AS `longitud`,
```

```
`tallos_bunch`.`numero` AS `numero`,
```

```
`bunch`.`fecha` AS `fecha`,  
  
`bunch`.`time_days_id` AS `time_days_id`  
  
from (((`bunch` join `variedad` on((`bunch`.`variedad_id` = `variedad`.`id`)))  
  
join `punto_corte` on((`bunch`.`punto_corte_id` = `punto_corte`.`id`)))  
  
join `longitud` on((`bunch`.`longitud_id` = `longitud`.`id`)))  
  
join `tallos_bunch` on((`bunch`.`tallos_bunch_id` = `tallos_bunch`.`id`)));
```

Como se puede apreciar en esta vista se generan las uniones con todos los elementos que se necesita para el Cubo Exportable.

_ Construcción y Publicación del Cubo de Información

Debemos tener claro que hay dos maneras de construir el Cubo de Información, una es generando de manera manual el archivo xml que produce el cubo; otra manera es a través del programa WorkBench.

Recordemos que antes de construir el cubo debemos dejar establecida por lo menos una conexión jndi en la consola de administración de Pentaho; para esto debemos ejecutar la consola desde el directorio de instalación de Pentaho `biserver/administration-console/ start-pac.bat`.

Luego de esto ingresamos a la consola de administración a través del Url en un navegador: `localhost:8099`.

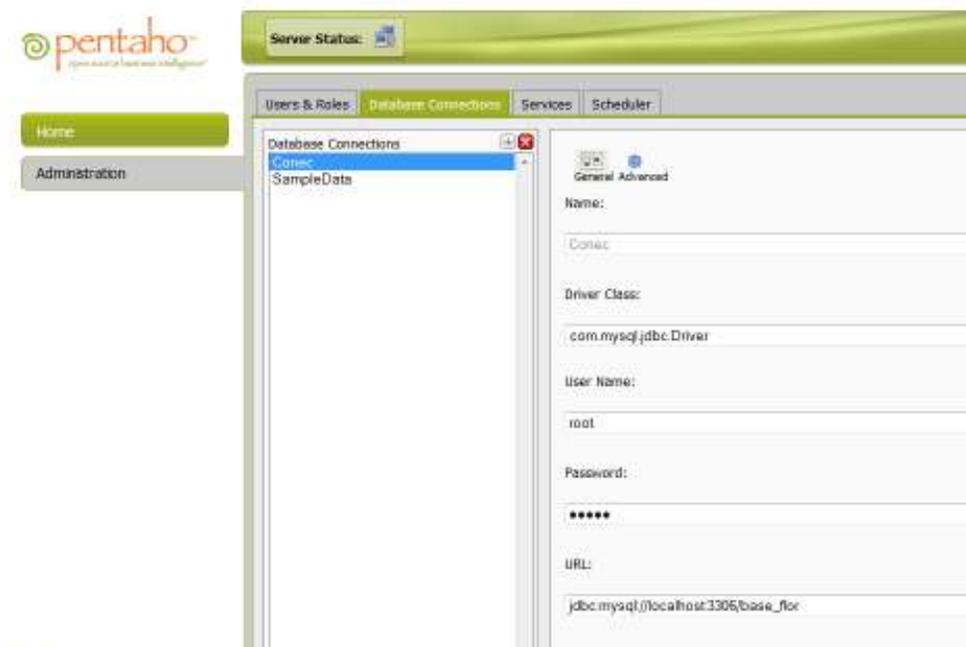


Figura 20.- Consola de Administración, estableciendo los parámetros de conexión

Fuente Javier Salvador, Pentaho PAC

Para este caso establecimos los siguientes parámetros de conexión (JNDI):

Name: jndi_florprueba

Driver Class: com.mysql.jdbc.Driver (es el driver jdbc de base de datos mysql)

User Name: root (Nombre de usuario de la base datos)

Password: **** (Es la contraseña de la base de datos)

URL: jdbc:mysql://localhost:3306/florprueba

El siguiente proceso es ejecutar el programa Workbench, para esto vamos a la carpeta de instalación schema workbench y entramos al programa en workbench.bat

Una vez en el programa vamos a Options / Connections

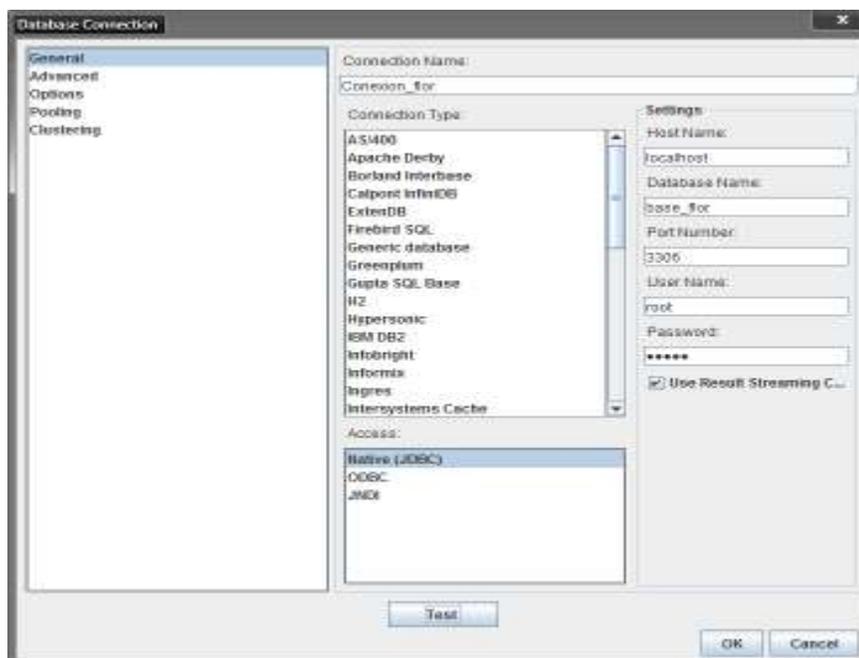


Figura 21.- Schema Workbench, estableciendo los parámetros de conexión Fuente

Javier Salvador, Pentaho Workbench

Ahora podemos construir el Cubo de información tal cual se indica en el capítulo 4.2.3.1. (Conceptos e instrucciones sobre Pentaho sobre de la construcción de la solución).

Una vez construido y publicado el cubo de información nos va dar como resultado un archivo xml, donde va estar situado en un directorio que convenga.

En nuestro caso se situa en el directorio C:\Pentaho 4.5\biserver\biserver-ce\pentaho-solutions\Cubo_Prueba.

Y el archivo se lo nombró de la siguiente manera: CuboPruebaAB.xml

El detalle del código xml del cubo de información se encuentra más abajo en el capítulo Anexos.

4.5.4. Capa de presentación

Para que se pueda ver de una manera interactiva y brindar reportes gráficos para el usuario final se pensó diseñar una Capa de presentación mediante una herramienta de Pentaho llamada el CDE o constructor de Dashboards.

Para poder construir los Reportes Dashboards, nos vamos a ejecutar la consola de trabajo de Pentaho, en este caso en Pentaho/biserver/biserver-ce/start-pentaho.bat

Una vez en la consola vamos a File / New / CDE DashBoards.



Figura 22.- Creando un Nuevo Proyecto CDE Dashboard. Fuente Javier Salvador, Pentaho CDE.

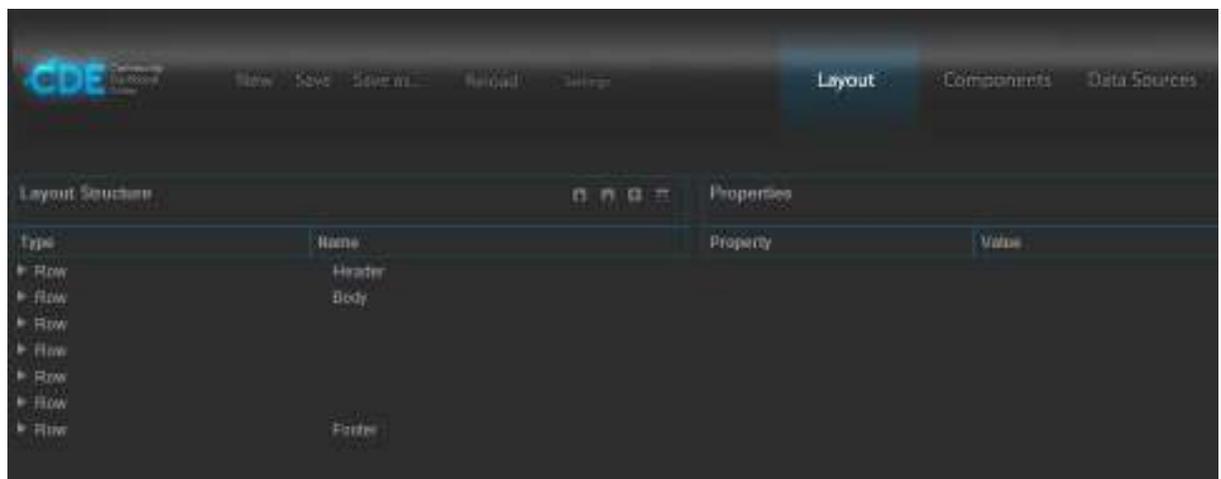


Figura 23.- Pantalla del editor de CDE Dashboard. Fuente Javier Salvador, Pentaho CDE.

Este Editor tiene tres partes importantes

- a) Layout.- Es donde se crea los paneles donde se va a diseñar el reporte
- b) Components.- Aquí se van añadir los componentes gráficos a insertarse en el dashboard, tales como cajas de texto, gráficos de líneas, tablas, cajas de selección, etc.

c) Data Sources.- Aquí van a estar las diferentes consultas en lenguaje Mdx que se necesita del Cubo de Información para ejecutar el reporte, en todo caso este sería la parte más importante del diseño de Dashboards.

Para empezar vamos a crear un gráfico donde se muestre una curva con el total de tallos cortados por semanas, filtrado por Año, Variedad, Punto de Corte y Bloques.

En primer lugar vamos a construir los paneles donde van a estar los componentes gráficos del reporte, para ello vamos a necesitar 5 paneles:

Uno para el título, dos para los filtros y dos para los gráficos que vamos a añadir.

Una vez establecidos los paneles vamos a agregar los Componentes,

Para este reporte vamos a necesitar 6 cajas de Selección: para filtrar el año, puntos de corte, bloques, variedades y para las semanas; cabe recalcar que por cada caja de Selección se genera un elemento tipo parámetro que utilizaremos más adelante.

También vamos a tener que añadir dos componentes gráficos llamados Chart: uno para mostrar la curva en semanas y otro para mostrar la curva en meses.

Ya añadidos estos componentes gráficos estableceremos las consultas Mdx para cada uno de estos componentes, a continuación veremos estas consultas:

a) Caja de Selección para Año:

```
with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName'
```

```
select TopCount( filter( {[Anio].[2012]}, not isempty( ([Anio].CurrentMember) ) ) , 50)
on ROWS,
```

```
{[Measures].[Name]} on Columns
```

```
from [Corte]
```

b) Caja de Selección para Punto de Corte

```
with member [Measures].[Name] as '[Punto de corte].CurrentMember.UniqueName'
```

```
select TopCount( filter( {Descendants([Punto de corte].[Total puntos]}), not
isempty( ([Punto de corte].CurrentMember) ) ) , 50) on ROWS,
```

```
{[Measures].[Name]} on Columns
```

```
from [Corte]
```

```
where ( ${S1_anoParameter} , ${S2_variedadParameter} )
```

c) Caja de Selección para el Número de Bloques

```
with member [Measures].[Name] as '[Bloque].CurrentMember.UniqueName' select
```

```
filter( {Descendants([Bloque].[Total bloques]}), not
isempty( ([Bloque].CurrentMember) ) ) on ROWS,
```

```
{[Measures].[Name]} on Columns
```

```
from [Corte]
```

```
where ( ${S1_anoParameter} , ${S3_puntoParameter} )
```

d) Caja de Selección para la Variedad

```

with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName' select
filter({Descendants([Variedad].[Total variedades])}, not
isempty(([Variedad].CurrentMember)) ) on ROWS,

{[Measures].[Name]} on Columns

from [Corte]

where ({S1_anoParameter} , {S3_puntoParameter} , {SbloquesParameter})

```

e) Componente Grafico en Semanas

```

select NON EMPTY({Ss1Parameter}:Ss2Parameter}) on ROWS,

NON EMPTY({[Measures].[Tallos corte]}) on Columns

from [Corte]

where ({S1_anoParameter} , {S3_puntoParameter} , {SbloquesParameter} ,
SVariedad1Parameter})

```

f) Componente Gráfico en Meses

```

select NON EMPTY({Mm1Parameter}:Mm2Parameter}) on ROWS,

NON EMPTY({[Measures].[Tallos corte]}) on Columns

from [Corte]

```

where ($\{S1_anioParameter\}$, $\{S3_puntoParameter\}$, $\{SbloquesParameter\}$, $\{SVariedad1Parameter\}$)

Teniendo todos estos elementos consolidados vamos a tener un resultado como el siguiente:

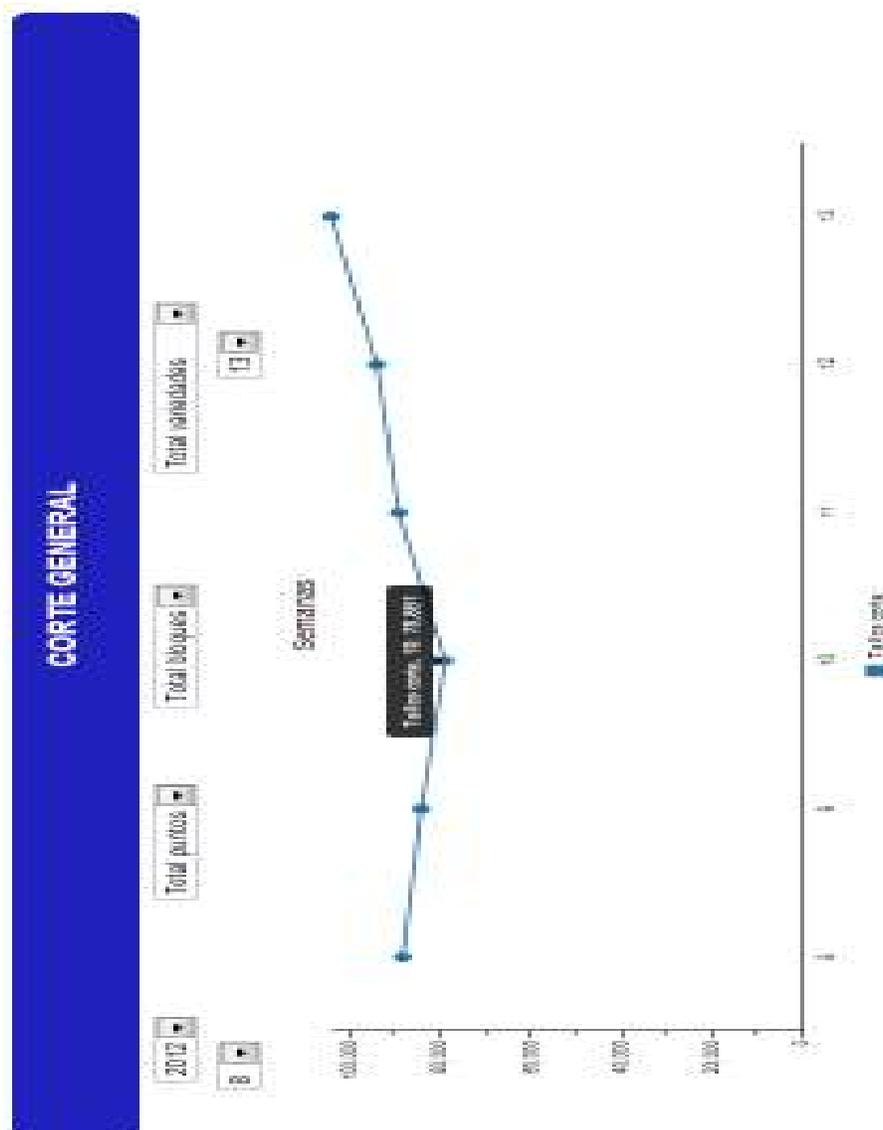


Figura 24.- Dashboard resultante donde se puede observar la información diseñada en el cubo de información. Fuente Javier Salvador

Por último este Reporte DashBoard genera un archivo de extensión XACTION.

Este archivo se genera en el directorio biserver/biserver-
ce/pentahosolution/DashDef/ Corte_GeneralDef.xaction

Este archivo está constituido en lenguaje xml y mdx y es el resultante del reporte
final DashBoard que va a manipular el usuario final.

```
<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
<DataSources>
<Connection id="FiltroPuntoQuery" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="FiltroVariedadQuery" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="FiltroFecha1Query" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="FiltroFecha2Query" type="mondrian.jndi">
```

```

<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="GraficoSemanas1Query" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="GraficoSemanasQuery" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
  <Jndi>jndi_florprueba</Jndi>
</Connection>
<Connection id="FiltroMes1Query" type="mondrian.jndi">
<Catalog>/CuboPruebaConexion/CuboPruebaAB.xml</Catalog>
<Jndi>jndi_florprueba</Jndi>
</Connection>
</DataSources>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="FiltroPuntoQuery"
  id="FiltroPuntoQuery"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>

```

```

<Parameters/>

    <Query>with member [Measures].[Name] as '[Punto de
Corte].CurrentMember.UniqueName' select filter({Descendants([Punto de
Corte].[Total puntos]}), not isempty(([Punto de Corte].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>

<DataAccess access="public" cache="true" cacheDuration="3600"
connection="FiltroVariedadQuery"
id="FiltroVariedadQuery"
type="mdx">

<BandedMode>compact</BandedMode>

<Columns/>

<Output indexes="1,0" mode="include"/>

<Parameters>

<Parameter default="[Punto de Corte].[Total puntos]" name="FiltroPuntoParameter"
type="String"/>

</Parameters>

    <Query>with member [Measures].[Name] as
'[Variedad].CurrentMember.UniqueName' select TopCount(
filter({Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) , 350) on ROWS,
{[Measures].[Name]} on Columns

```

```

from [Corte]

where ({FiltroPuntoParameter})</Query>

</DataAccess>

<DataAccess access="public" cache="true" cacheDuration="3600"

connection="FiltroFecha1Query"

id="FiltroFecha1Query"

type="mdx">

<BandedMode>compact</BandedMode>

<Columns/>

<Output indexes="1,0" mode="include"/>

<Parameters/>

    <Query>with member [Measures].[Name] as '[Fecha].CurrentMember.UniqueName'

select filter({Descendants([Fecha].[Total fecha] ,[Fecha].[Dia_Fecha])}, not

isempty(([Fecha].CurrentMember)) ) on ROWS,

{[Measures].[Name]} on Columns

from [Corte]</Query>

</DataAccess>

<DataAccess access="public" cache="true" cacheDuration="3600"

connection="FiltroFecha2Query"

id="FiltroFecha2Query"

type="mdx">

<BandedMode>compact</BandedMode>

<Columns/>

```

```

<Output indexes="1,0" mode="include"/>
<Parameters/>
  <Query>with member [Measures].[Name] as '[Fecha].CurrentMember.UniqueName'
select filter({Descendants([Fecha].[Total fecha] ,[Fecha].[Dia_Fecha])}, not
isempty(([Fecha].CurrentMember)) ) on ROWS,
  {[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
  connection="GraficoSemanas1Query"
  id="GraficoSemanas1Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
<Parameter default="[Punto de Corte].[Total puntos]" name="FiltroPuntoParameter"
  type="String"/>
<Parameter default="[Variedad].[Total variedades]" name="FiltroVariedadParameter"
  type="String"/>
  <Parameter default="[Fecha].[Total fecha]" name="FiltroFecha1Parameter"
  type="String"/>
<Parameter default="[Fecha].[Total fecha]" name="FiltroFecha2Parameter"
  type="String"/>

```

```
</Parameters>
```

```
    <Query>select NON EMPTY {Descendants([Bloque].[Total Bloques],
[Bloque].[Bloque])} ON COLUMNS,
NON EMPTY {{FiltroFecha1Parameter}:$ {FiltroFecha2Parameter}} ON ROWS
from [Corte]
where ($ {FiltroPuntoParameter} , $ {FiltroVariedadParameter})</Query>
```

```
</DataAccess>
```

```
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="GraficoSemanasQuery"
id="GraficoSemanasQuery"
type="mdx">
```

```
<BandedMode>compact</BandedMode>
```

```
<Columns/>
```

```
<Parameters>
```

```
  <Parameter default="[Punto de Corte].[Total puntos]" name="FiltroPuntoParameter"
type="String"/>
```

```
  <Parameter default="[Variedad].[Total variedades]" name="FiltroVariedadParameter"
type="String"/>
```

```
</Parameters>
```

```
    <Query>select NON EMPTY({Descendants([Semana].[Total semana]
,[Semana].[Semana])) on ROWS,
NON EMPTY({[Measures].[Tallos Corte]}) on Columns
from [Corte]
```

```

where ({FiltroPuntoParameter} , {FiltroVariedadParameter})</Query>
</DataAccess>

    <DataAccess access="public" cache="true" cacheDuration="3600"
connection="FiltroMes1Query"
    id="FiltroMes1Query"
    type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters/>
    <Query>with member [Measures].[Name] as
'[Mes].CurrentMember.UniqueName' select filter({Descendants([Mes].[Total meses]
,[Mes].[Mes])}, not isempty(([Mes].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
</CDADescriptor>

```

Adicionalmente se construirán 8 Reportes más con el siguiente contenido:

_ Un gráfico donde se muestre curvas comparativas entre Bloques con el total de tallos cortados por semanas, filtrado por Año, Variedad y Punto de Corte.

_ Un gráfico donde se muestre una curva comparando la producción del corte entre dos variedades, filtrando bloques, puntos y años.

- _ Un gráfico donde se muestre una curva comparativa del corte hecho por Años.
- _ Un gráfico donde se muestre una curva comparativa entre años del exportable producido.
- _ Un gráfico donde se muestre una curva comparativa entre las diferentes longitudes de los ramos del exportable, filtrado por Año, Variedad, Punto de Corte y Longitud.
- _ Un gráfico donde se muestre una curva con el total de tallos para exportar por semanas, filtrado por Año, Variedad, Punto de Corte y Longitud.
- _ Un gráfico donde se muestre una curva comparando la producción del exportable entre dos variedades, filtrando bloques, puntos de corte y años.
- _ Un gráfico donde se muestre varias curvas comparativas entre el exportable y el corte.

El los códigos XML de los dos cubos realizados se encuentran en los Anexos 1 y 2

Los reportes y los códigos XML y MDX de los 8 reportes realizados se encuentran detallados en los Anexos 3, 4, 5, 6, 7, 8, 9 y 10.

5. Conclusiones y Recomendaciones

5.1. Conclusiones

_ Como conclusión de la investigación respectiva, se puede decir que la Inteligencia de Negocios depende mucho de nuestra capacidad de clasificar y organizar la información, desde poder distinguir en donde pertenece un datos, hasta poder visualizar una gran cantidad de datos de una manera gráfica.

_ Podemos concluir que la herramienta de Inteligencia de Negocios Pentaho brinda muchas facilidades, ya sea en su instalación como en su programación; la característica más importante que se pudo destacar es el versionamiento periódico que tiene esta herramienta.

_ El Negocio de la Producción de Flores en el cuál se implementó la solución, brinda un sinnúmero de información y de formas de presentación de los datos, por eso es uno de los mejores escenarios para aplicar un BI, en una finca de flores la información es ingresada cada 5 minutos aproximadamente en 8 horas laborables de trabajo diarios.

_ Se pudo apreciar que la solución gráfica y el filtraje de datos interactivo es la mejor manera para que el personal gerencial muestre la información correspondiente a su empresa y de esta manera puedan tomar las mejores decisiones.

5.2. Recomendaciones

_ Se recomienda la implementación de Software conjuntamente con una solución en Inteligencia de Negocios, ya que la información recolectada mediante un BI es el producto de toda la implementación de un Sistema Informático.

_ Se recomienda hacer un análisis muy detallado de cómo está constituida la Lógica del Negocio donde se implemente, solo así podemos tener como resultado una Solución de BI confiable.

_ Se recomienda transformar la información almacenada de la mayoría de bases de datos tradicionales a una solución de inteligencia de Negocios, para poder sacar mayor provecho a estos datos.

5.3. Bibliografía

Universidad Boliviana, Ingeniería de software. (2010). Obtenido de

http://ingenieriadesoftware.mex.tl/63758_AUP.html

Wikipedia. (2010). Obtenido de <http://es.wikipedia.org/wiki/Pentaho>

Calaméo. (2011). Obtenido de <http://es.calameo.com/read/000366046e572e0d49e11>

Navegapolis. (2011). Obtenido de http://www.navegapolis.net/files/s/NST-010_01.pdf

Pentaho.com. (2011). Obtenido de <http://www.pentaho.com/>

Bouman, R. (2010). *Pentaho Solutions*. Indianapolis: Wiley Publishing, Inc.

ANEXOS

ANEXO 1

Código Xml Cubo Corte

```

<Cube name="Corte" cache="true" enabled="true">
  <Table name="vista_malla">
  </Table>
  <Dimension type="StandardDimension" foreignKey="variedad_id" highCardinality="false"
name="Variedad">
  <Hierarchy hasAll="true" allMemberName="Total variedades" primaryKey="id">
  <Table name="variedad">
  </Table>
  <Level name="Variedad" table="variedad" column="nombre" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
  </Level>
  </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" foreignKey="bloque_id" highCardinality="false"
name="Bloque">
  <Hierarchy hasAll="true" allMemberName="Total Bloques" primaryKey="id">
  <Table name="bloque">
  </Table>
  <Level name="Bloque" column="codigo" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  </Level>
  </Hierarchy>
  </Dimension>

```

```

    <Dimension type="StandardDimension" foreignKey="punto_corte_id"
highCardinality="false" name="Punto de Corte">
    <Hierarchy hasAll="true" allMemberName="Total puntos" primaryKey="id">
    <Table name="punto_corte">
    </Table>
    <Level name="Punto de Corte" column="nombre" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
    </Level>
    </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Fecha">
    <Hierarchy hasAll="true" allMemberName="Total fecha" primaryKey="id">
    <Table name="tiempo_fecha">
    </Table>
    <Level name="Dia_Fecha" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
    </Level>
    </Hierarchy>
    </Dimension>
    <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Anio">
    <Hierarchy hasAll="true" allMemberName="Total anio" primaryKey="id">

```

```

<Table name="tiempo_fecha">
</Table>

<Level name="Anio" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  <KeyExpression>
    <SQL dialect="generic">
      <![CDATA[YEAR(tiempo_fecha.fecha)]]>
    </SQL>
  </KeyExpression>
</Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Mes">
  <Hierarchy hasAll="true" allMemberName="Total meses" primaryKey="id">
    <Table name="tiempo_fecha">
</Table>

    <Level name="Mes" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
      <KeyExpression>
        <SQL dialect="generic">
          <![CDATA[MONTH(tiempo_fecha.fecha)]]>
        </SQL>

```

```

    </KeyExpression>
  </Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Semana">
  <Hierarchy hasAll="true" allMemberName="Total semana" primaryKey="id">
    <Table name="tiempo_fecha">
      </Table>
      <Level name="Semana" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
        <KeyExpression>
          <SQL dialect="generic">
            <![CDATA[WEEK(tiempo_fecha.fecha)]]>
          </SQL>
        </KeyExpression>
      </Level>
    </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Dia">
    <Hierarchy hasAll="true" allMemberName="Total dias" primaryKey="id">
      <Table name="tiempo_fecha">

```

```
</Table>

<Level name="Dia Numero" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  <KeyExpression>
    <SQL dialect="generic">
      <![CDATA[DAY(tiempo_fecha.fecha)]]>
    </SQL>
  </KeyExpression>
</Level>
</Hierarchy>
</Dimension>
<Measure name="Tallos Corte" column="numero_tallos" aggregator="sum"
visible="true">
  </Measure>
<Measure name="Mallas corte" column="id" aggregator="count" visible="true">
  </Measure>
</Cube>
```

ANEXO 2
Código Xml Cubo
Exportable

```

<Cube name="Exportable" cache="true" enabled="true">
  <Table name="vista_bunch">
  </Table>
  <Dimension type="StandardDimension" foreignKey="variedad_id" highCardinality="false"
name="Variedad">
  <Hierarchy hasAll="true" allMemberName="Total variedades" primaryKey="id">
  <Table name="variedad">
  </Table>
  <Level name="Variedad" table="variedad" column="nombre" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
  </Level>
  </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" foreignKey="longitud_id" highCardinality="false"
name="Longitud">
  <Hierarchy hasAll="true" primaryKey="id">
  <Table name="longitud">
  </Table>
  <Level name="Longitud" column="longitud" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  </Level>
  </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" foreignKey="punto_corte_id"
highCardinality="false" name="Punto de Corte">
  <Hierarchy hasAll="true" allMemberName="Total puntos" primaryKey="id">
  <Table name="punto_corte">
  </Table>
  <Level name="Punto de Corte" column="nombre" type="String"
uniqueMembers="false" levelType="Regular" hideMemberIf="Never">
  </Level>

```

```

    </Hierarchy>
  </Dimension>
  <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Fecha">
    <Hierarchy hasAll="true" allMemberName="Total fecha" primaryKey="id">
      <Table name="tiempo_fecha">
        </Table>
        <Level name="Fecha_Dia" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
          </Level>
        </Hierarchy>
      </Dimension>
      <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Anio">
        <Hierarchy hasAll="true" allMemberName="Total Anios" primaryKey="id">
          <Table name="tiempo_fecha">
            </Table>
            <Level name="Anio" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
              <KeyExpression>
                <SQL dialect="generic">
                  <![CDATA[YEAR(tiempo_fecha.fecha)]]>
                </SQL>
              </KeyExpression>
            </Level>
          </Hierarchy>
        </Dimension>
        <Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Mes">
          <Hierarchy hasAll="true" allMemberName="Total Meses" primaryKey="id">
            <Table name="tiempo_fecha">

```

```

</Table>
<Level name="Mes" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  <KeyExpression>
    <SQL dialect="generic">
      <![CDATA[MONTH(tiempo_fecha.fecha)]]>
    </SQL>
  </KeyExpression>
</Level>
</Hierarchy>
</Dimension>
<Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Semana">
  <Hierarchy hasAll="true" allMemberName="Total Semanas" primaryKey="id">
    <Table name="tiempo_fecha">
      </Table>
    <Level name="Semana" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
      <KeyExpression>
        <SQL dialect="generic">
          <![CDATA[WEEK(tiempo_fecha.fecha)]]>
        </SQL>
      </KeyExpression>
    </Level>
  </Hierarchy>
</Dimension>
<Dimension type="StandardDimension" foreignKey="time_days_id"
highCardinality="false" name="Dia">
  <Hierarchy hasAll="true" allMemberName="Total Dia" primaryKey="id">
    <Table name="tiempo_fecha">
      </Table>

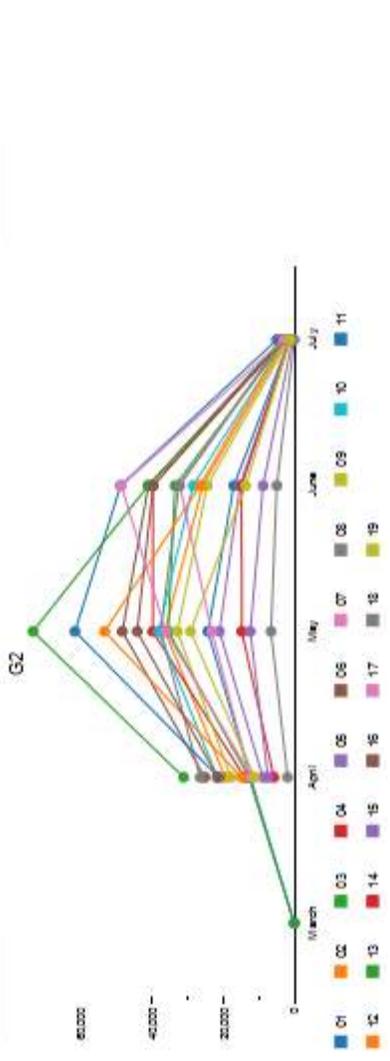
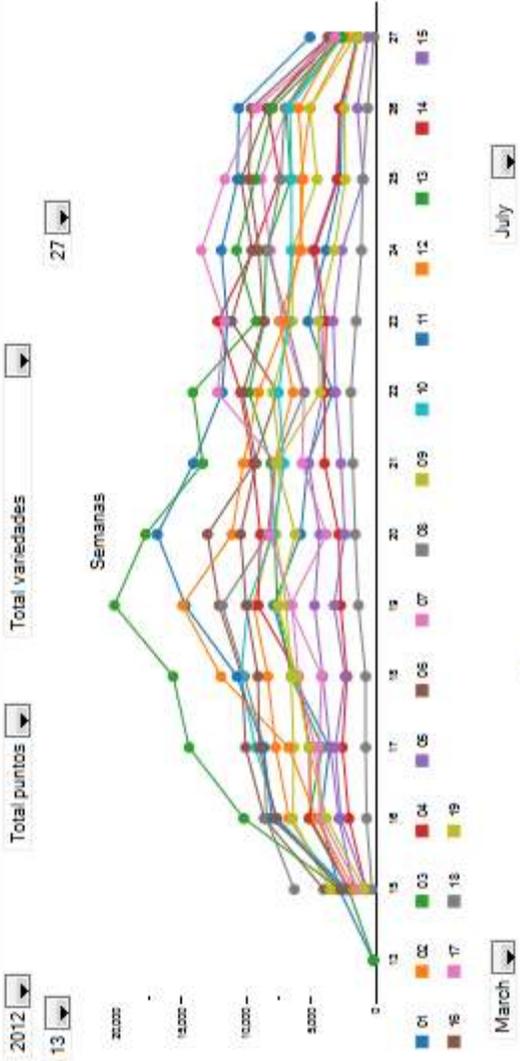
```

```
<Level name="Dia" column="fecha" type="String" uniqueMembers="false"
levelType="Regular" hideMemberIf="Never">
  <KeyExpression>
    <SQL dialect="generic">
      <![CDATA[DAY(tiempo_fecha.fecha)]]>
    </SQL>
  </KeyExpression>
</Level>
</Hierarchy>
</Dimension>
<Measure name="Tallos Exportable " column="numero_tallos" aggregator="sum"
visible="true">
  </Measure>
  <Measure name="Ramos Exportable" column="id" aggregator="count" visible="true">
  </Measure>
</Cube>
```

ANEXO 3

Reporte 1 Corte

BLOQUES COMPARACION



Código XML/MDX del Reporte 1

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S2_variedadQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
  id="S1_anioQuery"
  type="mdx">
    <BandedMode>compact</BandedMode>

```

```

    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {[Anio].[2012],[Anio].[2011],[Anio].[2010],[Anio].[2009]}, not
isempty(([Anio].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S2_variedadQuery"
    id="S2_variedadQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select TopCount( filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
    id="S3_puntoQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
    type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]}), not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} , ${S2_variedadParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss1Query"

```

```

        id="Ss1Query"
        type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
  <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss2Query"
  id="Ss2Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
  <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm1Query"
  id="Mm1Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
  <Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter( {Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]

```

```

where ({S1_anioParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm2Query"
    id="Mm2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as [Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anioParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G1Query"
    id="G1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
            type="String"/>
        <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
            type="String"/>
        <Parameter default="[TiempoS].[Total tiempo]" name="Ss1Parameter" type="String"/>
        <Parameter default="[TiempoS].[Total tiempo]" name="Ss2Parameter" type="String"/>
    </Parameters>
    <Query>select NON EMPTY {Descendants([Bloque].[Total bloques],
[Bloque].[Bloque])} ON COLUMNS,
NON EMPTY {{Ss1Parameter}:Ss2Parameter}} ON ROWS
from [Corte]
where ({S1_anioParameter} , {S2_variedadParameter} , {S3_puntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G2Query"
    id="G2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>

```

```

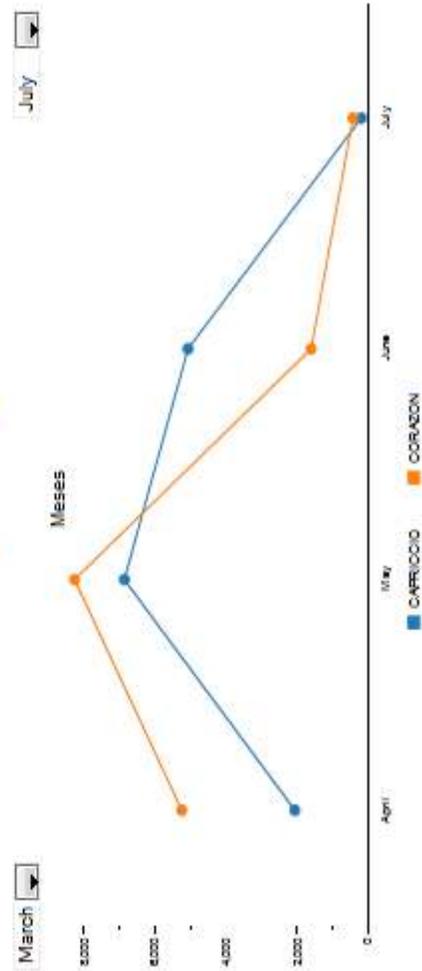
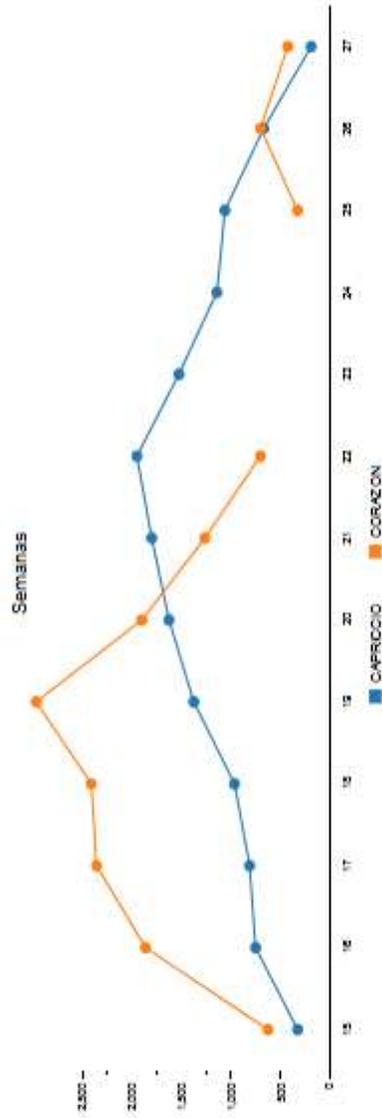
<Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
  type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
  type="String"/>
<Parameter default="[Tiempo].[Total tiempo]" name="Mm1Parameter" type="String"/>
<Parameter default="[Tiempo].[Total tiempo]" name="Mm2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY {Descendants([Bloque].[Total bloques],
[Bloque].[Bloque])} ON COLUMNS,
NON EMPTY({{Mm1Parameter}:Mm2Parameter}) on ROWS
from [Corte]
where ({S1_anoParameter} , {S2_variedadParameter} , {S3_puntoParameter})</Query>
</DataAccess>
</CDADescriptor>

```

ANEXO 4

Reporte 2 Corte

VARIETADES COMPARACION



Código XML/MDX Reporte 2 Corte

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SbloquesQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariedad1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariedad2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G_02Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
  
```

```

    <Jndi>SF</Jndi>
  </Connection>
</DataSources>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
  id="S1_anioQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters/>
  <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {[Anio].[2012],[Anio].[2011],[Anio].[2010],[Anio].[2009]}, not
isempty(([Anio].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
  id="S3_puntoQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
      type="String"/>
  </Parameters>
  <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]} , not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} , ${S2_variedadParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SbloquesQuery"
  id="SbloquesQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"

```

```

        type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Bloque].CurrentMember.UniqueName'
select filter({Descendants([Bloque].[Total bloques]}), not
isempty(([Bloque].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anioParameter} , {S3_puntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariedad1Query"
id="SVariedad1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select filter({Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anioParameter} , {S3_puntoParameter} , {SbloquesParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariedad2Query"
id="SVariedad2Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
</Parameters>

```

```

    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anoParameter} , {S3_puntoParameter} , {SbloquesParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss1Query"
id="Ss1Query"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss2Query"
id="Ss2Query"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm1Query"
id="Mm1Query"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>

```

```

<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($S1_anoParameter)</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm2Query"
  id="Mm2Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($S1_anoParameter)</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G1Query"
  id="G1Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
  <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
    type="String"/>
  <Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
  <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
    type="String"/>
  <Parameter default="[TiempoS].[Total tiempo]" name="Ss1Parameter" type="String"/>
  <Parameter default="[Variedad].[Total variedades]" name="SVariedad2Parameter"
    type="String"/>
  <Parameter default="[TiempoS].[Total tiempo]" name="Ss2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY({$Ss1Parameter}:{$Ss2Parameter}) on ROWS,

```

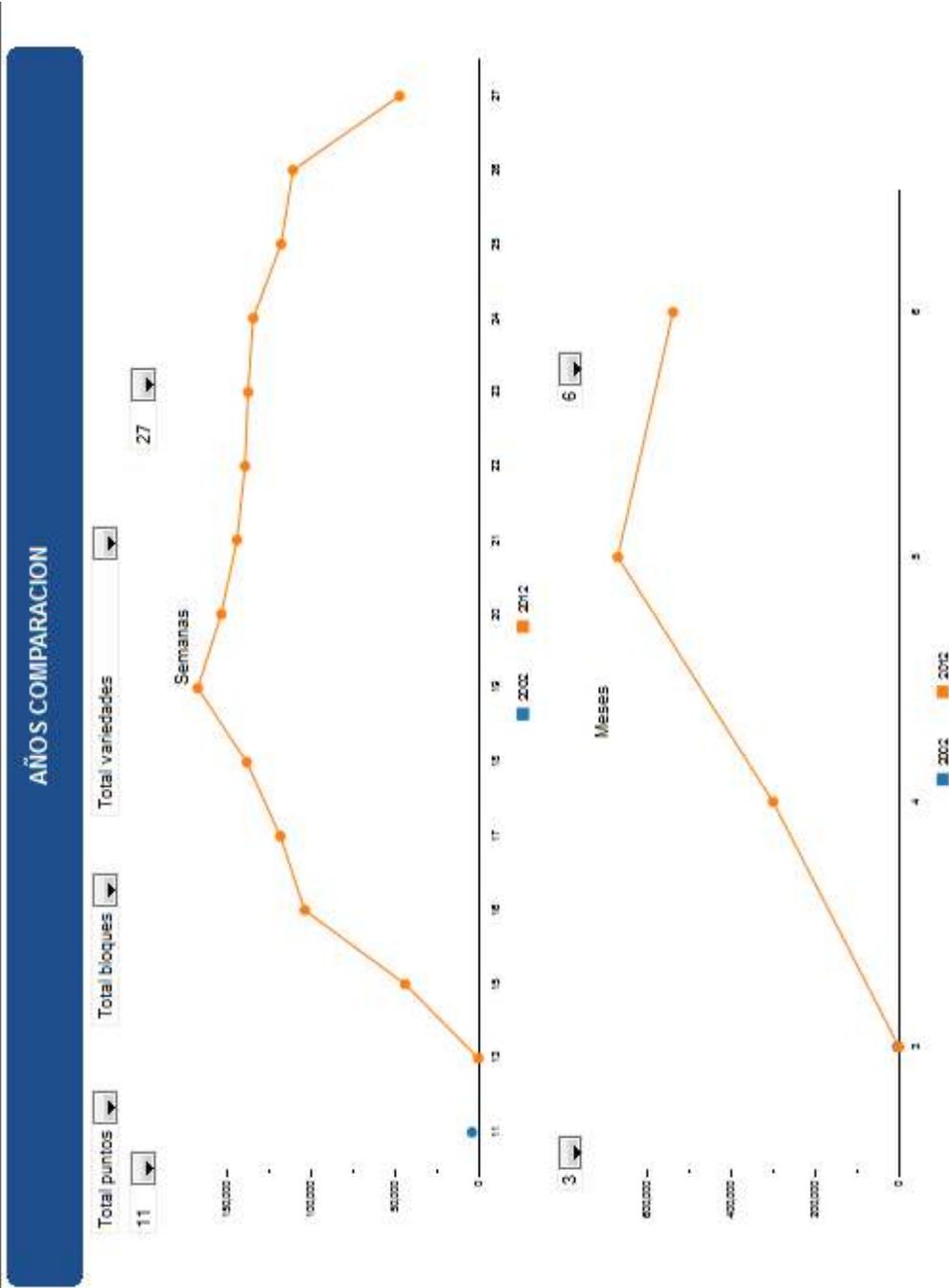
```

NON EMPTY({$SVariedad1Parameter},{$SVariedad2Parameter}) on Columns
from [Corte]
where ($S1_anoParameter} , {$S3_puntoParameter} , {$SbloquesParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="G_02Query"
      id="G_02Query"
      type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
      type="String"/>
    <Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
      type="String"/>
    <Parameter default="[Tiempo].[Total tiempo]" name="Mm1Parameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedad2Parameter"
      type="String"/>
    <Parameter default="[Tiempo].[Total tiempo]" name="Mm2Parameter" type="String"/>
  </Parameters>
  <Query>select NON EMPTY({$Mm1Parameter}:{$Mm2Parameter}) on ROWS,
NON EMPTY({$SVariedad1Parameter},{$SVariedad2Parameter}) on Columns
from [Corte]
where ($S1_anoParameter} , {$S3_puntoParameter} , {$SbloquesParameter})</Query>
</DataAccess>
</CDADescriptor>

```

ANEXO 5

Reporte 3 Corte



Código XML/MDX Reporte 3 Corte

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="SPuntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SBloqueQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariadadQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SSemana1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SSemana2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SMes1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SMes2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>

```

```

<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SPuntoQuery"
    id="SPuntoQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select filter( {Descendants([Punto de corte].[Total
puntos]}), not isempty((([Punto de corte].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SBloqueQuery"
    id="SBloqueQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Bloque].CurrentMember.UniqueName'
select filter( {Descendants([Bloque].[Total bloques]}), not
isempty((([Bloque].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {SPuntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariedadQuery"
    id="SVariedadQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
    <Parameter default="[Bloque].[Total bloques]" name="SBloqueParameter"
type="String"/>
    </Parameters>

```

```

    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select NON EMPTY( filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember, [Measures].[Tallos corte])) ) ) on ROWS,
NON EMPTY( {[Measures].[Name],[Measures].[Tallos corte]} ) on Columns
from [Corte]
where ( ${SPuntoParameter} , ${SBloqueParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SSemana1Query"
    id="SSemana1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Semana].CurrentMember.UniqueName'
select filter( {Descendants([Semana].[Total semanas] ,[Semana].[Semana])}, not
isempty(([Semana].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SSemana2Query"
    id="SSemana2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Semana].CurrentMember.UniqueName'
select filter( {Descendants([Semana].[Total semanas] ,[Semana].[Semana])}, not
isempty(([Semana].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SMes1Query"
    id="SMes1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Mes].CurrentMember.UniqueName' select
filter( {Descendants([Mes].[Total meses] ,[Mes].[Mes])}, not
isempty(([Mes].CurrentMember)) ) on ROWS,

```

```

{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SMes2Query"
id="SMes2Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters/>
<Query>with member [Measures].[Name] as '[Mes].CurrentMember.UniqueName' select
filter({Descendants([Mes].[Total meses] ,[Mes].[Mes])}, not
isempty(([Mes].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g1Query"
id="g1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
<Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SBloqueParameter"
type="String"/>
<Parameter default="[Variedad].[Total variedades]" name="SVariedadParameter"
type="String"/>
<Parameter default="[Semana].[Total semanas]" name="SSemana1Parameter"
type="String"/>
<Parameter default="[Semana].[Total semanas]" name="SSemana2Parameter"
type="String"/>
</Parameters>
<Query>select NON EMPTY {${SSemana1Parameter}:${SSemana2Parameter}} ON
ROWS,
NON EMPTY {Descendants([Anio].[Total anios], [Anio].[Anio])} on Columns
from [Corte]
where (${SPuntoParameter} , ${SBloqueParameter} , ${SVariedadParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g2Query"
id="g2Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>

```

```

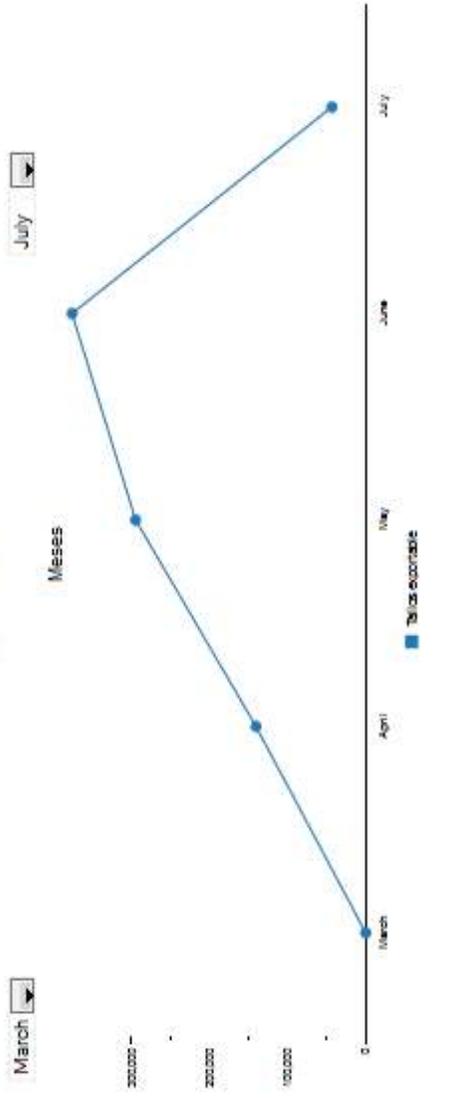
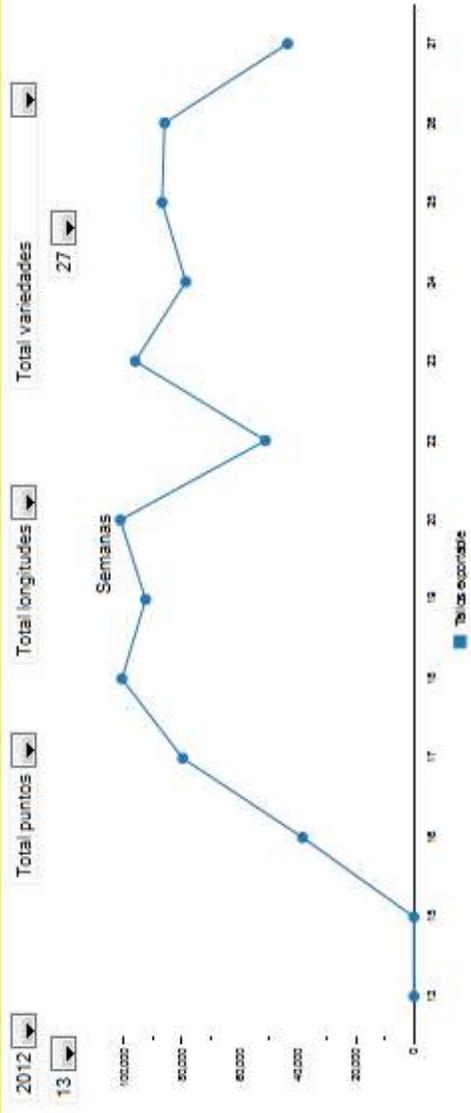
    <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
    <Parameter default="[Bloque].[Total bloques]" name="SBloqueParameter"
type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedadParameter"
type="String"/>
    <Parameter default="[Mes].[Total meses]" name="SMes1Parameter" type="String"/>
    <Parameter default="[Mes].[Total meses]" name="SMes2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY {{SMes1Parameter}}:{{SMes2Parameter}} ON ROWS,
NON EMPTY {Descendants([Anio].[Total anios], [Anio].[Anio])} on Columns
from [Corte]
where ({{SPuntoParameter}} , {{SBloqueParameter}} , {{SVariedadParameter}})</Query>
</DataAccess>
</CDADescriptor>

```

ANEXO 6

Reporte 4 Exportable

EXPORTABLE GENERAL



Código XML/MDX Reporte 4 Exportable

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="SPuntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariiedadQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SLongitudQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SSemana1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SSemana2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SMes1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SMes2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="SPuntoQuery"
  id="SPuntoQuery"
  type="mdx">
    <BandedMode>compact</BandedMode>

```

```

    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select filter({Descendants([Punto de corte].[Total
puntos])}, not isempty(([Punto de corte].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVarietadQuery"
    id="SVarietadQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Varietad].CurrentMember.UniqueName'
select filter({Descendants([Varietad].[Total variedades])}, not
isempty(([Varietad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Exportable]
where ($ {SPuntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SLongitudQuery"
    id="SLongitudQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
    <Parameter default="[Varietad].[Total variedades]" name="SVarietadParameter"
type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Longitud].CurrentMember.UniqueName'
select filter({Descendants([Longitud].[Total longitudes])}, not
isempty(([Longitud].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Exportable]
where ($ {SPuntoParameter} , $ {SVarietadParameter})</Query>

```

```

</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SSemana1Query"
    id="SSemana1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Semana].CurrentMember.UniqueName'
select filter( {Descendants([Semana].[Total semanas] ,[Semana].[Semana])}, not
isempty(([Semana].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SSemana2Query"
    id="SSemana2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Semana].CurrentMember.UniqueName'
select filter( {Descendants([Semana].[Total semanas] ,[Semana].[Semana])}, not
isempty(([Semana].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SMes1Query"
    id="SMes1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Mes].CurrentMember.UniqueName' select
filter( {Descendants([Mes].[Total meses] ,[Mes].[Mes])}, not
isempty(([Mes].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SMes2Query"
    id="SMes2Query"

```

```

        type="mdx">
        <BandedMode>compact</BandedMode>
        <Columns/>
        <Output indexes="1,0" mode="include"/>
        <Parameters/>
        <Query>with member [Measures].[Name] as '[Mes].CurrentMember.UniqueName' select
filter({Descendants([Mes].[Total meses] ,[Mes].[Mes])}, not
isempty(([Mes].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G1Query"
        id="G1Query"
        type="mdx">
        <BandedMode>compact</BandedMode>
        <Columns/>
        <Parameters>
        <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="SVariedadParameter"
type="String"/>
        <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>
        <Parameter default="[Semana].[Total semanas]" name="SSemana1Parameter"
type="String"/>
        <Parameter default="[Semana].[Total semanas]" name="SSemana2Parameter"
type="String"/>
        </Parameters>
        <Query>select NON EMPTY {{SSemana1Parameter}}:{{SSemana2Parameter}} ON
ROWS,
NON EMPTY {Descendants([Anio].[Total anios], [Anio].[Anio])} on Columns
from [Exportable]
where ({{SPuntoParameter}} , {{SVariedadParameter}} , {{SLongitudParameter}})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G2Query"
        id="G2Query"
        type="mdx">
        <BandedMode>compact</BandedMode>
        <Columns/>
        <Parameters>
        <Parameter default="[Punto de corte].[Total puntos]" name="SPuntoParameter"
type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="SVariedadParameter"
type="String"/>
        <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>

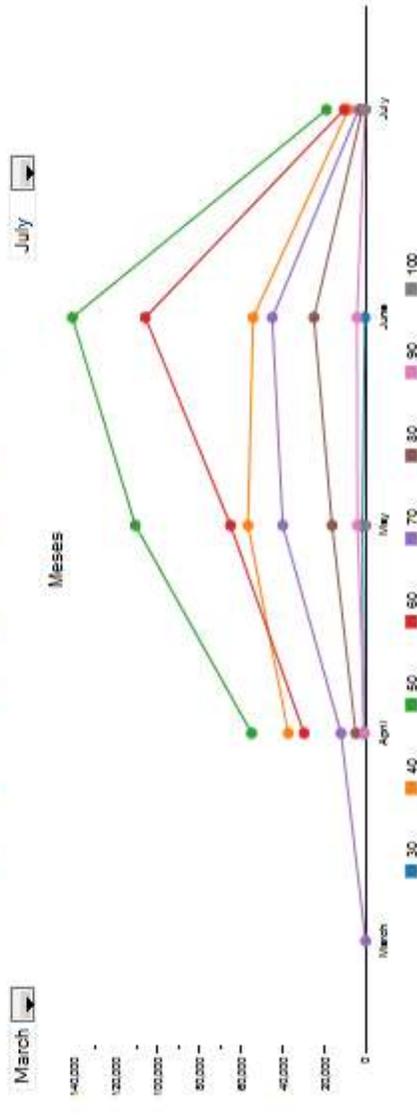
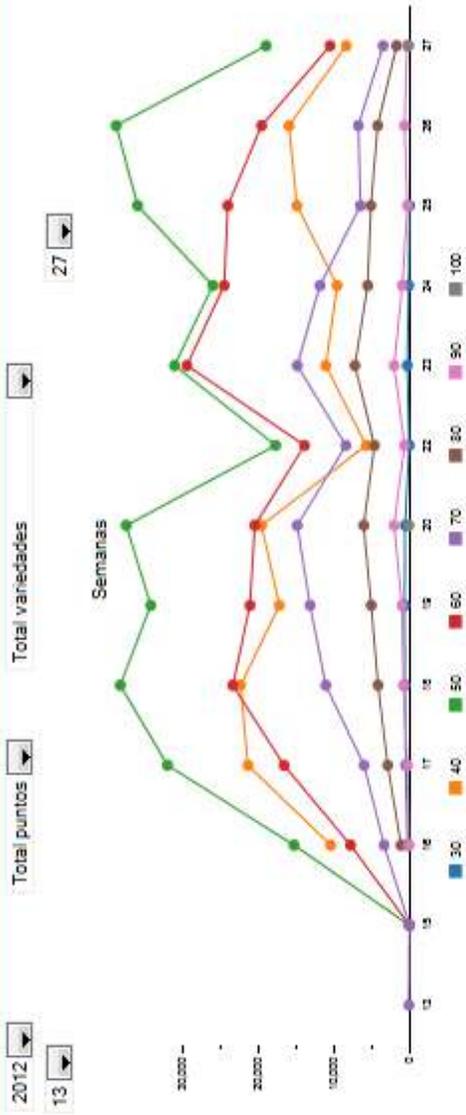
```

```
<Parameter default="[Mes].[Total meses]" name="SMes1Parameter" type="String"/>
<Parameter default="[Mes].[Total meses]" name="SMes2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY {${SMes1Parameter}:${SMes2Parameter}} ON ROWS,
NON EMPTY {Descendants([Anio].[Total anios], [Anio].[Anio])} on Columns
from [Exportable]
where (${SPuntoParameter} , ${SVariedadParameter} , ${SLongitudParameter})</Query>
</DataAccess>
</CDADescriptor>
```

ANEXO 7

Reporte 5 Exportable

LONGITUDES COMPARACION



Código XML/MDX Reporte 5 Exportable

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S2_variedadQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
  id="S1_anioQuery"
  type="mdx">
    <BandedMode>compact</BandedMode>

```

```

    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {[Anio].[2012],[Anio].[2011],[Anio].[2010],[Anio].[2009]}, not
isempty(([Anio].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S2_variedadQuery"
id="S2_variedadQuery"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select TopCount( filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
id="S3_puntoQuery"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]}), not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} , ${S2_variedadParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss1Query"

```

```

        id="Ss1Query"
        type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
  <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
</DataAccess>
  <DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss2Query"
    id="Ss2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
      <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
  </DataAccess>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm1Query"
    id="Mm1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
      <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter( {Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]

```

```

where ({S1_anioParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm2Query"
    id="Mm2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ({S1_anioParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g1Query"
    id="g1Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
            type="String"/>
        <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
            type="String"/>
        <Parameter default="[TiempoS].[Total tiempo]" name="Ss1Parameter" type="String"/>
        <Parameter default="[TiempoS].[Total tiempo]" name="Ss2Parameter" type="String"/>
    </Parameters>
    <Query>select NON EMPTY({Ss1Parameter}:Ss2Parameter}) on ROWS,
NON EMPTY {Descendants([Longitud].[Total longitudes], [Longitud].[Longitud])} on
Columns
from [Exportable]
where ({S1_anioParameter} , S2_variedadParameter} , S3_puntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g2Query"
    id="g2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>

```

```

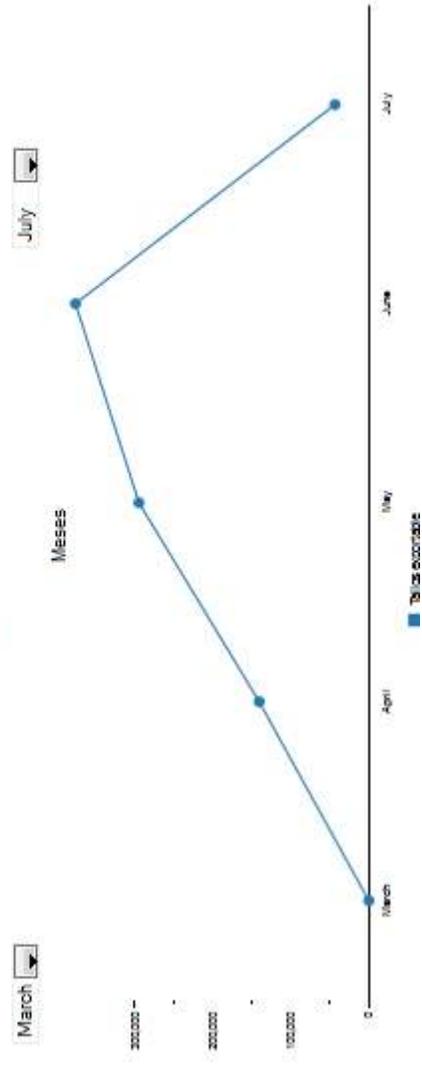
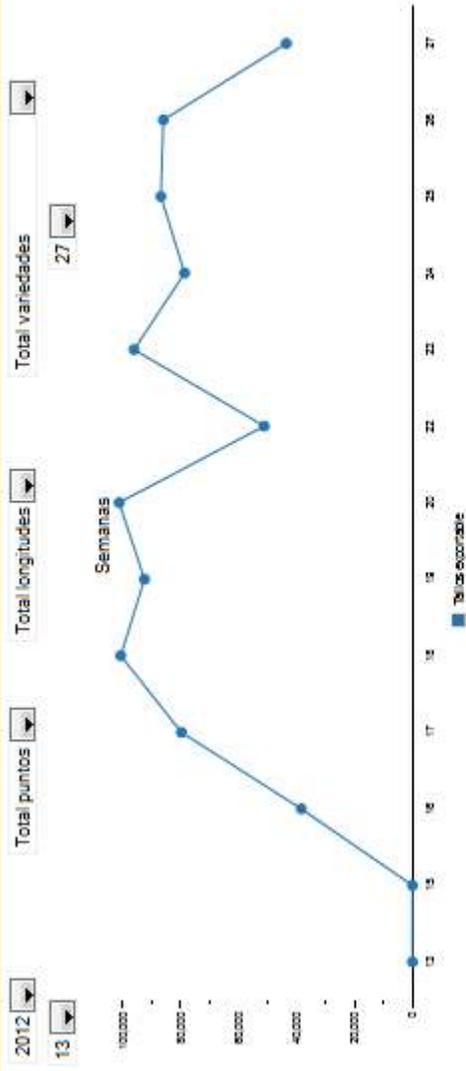
<Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
  type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
  type="String"/>
<Parameter default="[Tiempo].[Total tiempo]" name="Mm1Parameter" type="String"/>
<Parameter default="[Tiempo].[Total tiempo]" name="Mm2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY({${Mm1Parameter} : ${Mm2Parameter}}) on ROWS,
NON EMPTY {Descendants([Longitud].[Total longitudes], [Longitud].[Longitud])} on
Columns
from [Exportable]
where (${S1_anoParameter} , ${S2_variedadParameter} , ${S3_puntoParameter})</Query>
</DataAccess>
</CDADescriptor>

```

ANEXO 8

Reporte 6 Exportable

EXPORTABLE GENERAL



Código XML/MDX Reporte 6 Exportable

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SLongitudQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariadad1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g01Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="G02Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>

```

```

<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
    id="S1_anioQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {[Anio].[2012],[Anio].[2011],[Anio].[2010],[Anio].[2009]}, not
isempty(([Anio].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
    id="S3_puntoQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
        type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]} , not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} , ${S2_variedadParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SLongitudQuery"
    id="SLongitudQuery"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
        type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
        type="String"/>

```

```

    </Parameters>
    <Query>with member [Measures].[Name] as '[Longitud].CurrentMember.UniqueName'
select TopCount( filter( {Descendants([Longitud].[Total longitudes]}), not
isempty(([Longitud].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [Exportable]
where ( ${S1_anioParameter} , ${S3_puntoParameter} , ${SVariabilidadParameter} ) </Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariabilidadQuery"
id="SVariabilidadQuery"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Variabilidad].CurrentMember.UniqueName'
select filter( {Descendants([Variabilidad].[Total variabilidades]}), not
isempty(([Variabilidad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anioParameter} , ${S3_puntoParameter} , ${SbloquesParameter} ) </Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss1Query"
id="Ss1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] , [TiempoS].[Semana]}), not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anioParameter} ) </Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss2Query"

```

```

        id="Ss2Query"
        type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
  <Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter({Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana])}, not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
</DataAccess>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm1Query"
  id="Mm1Query"
  type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
      <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($ {S1_anoParameter})</Query>
  </DataAccess>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm2Query"
  id="Mm2Query"
  type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
      <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns

```

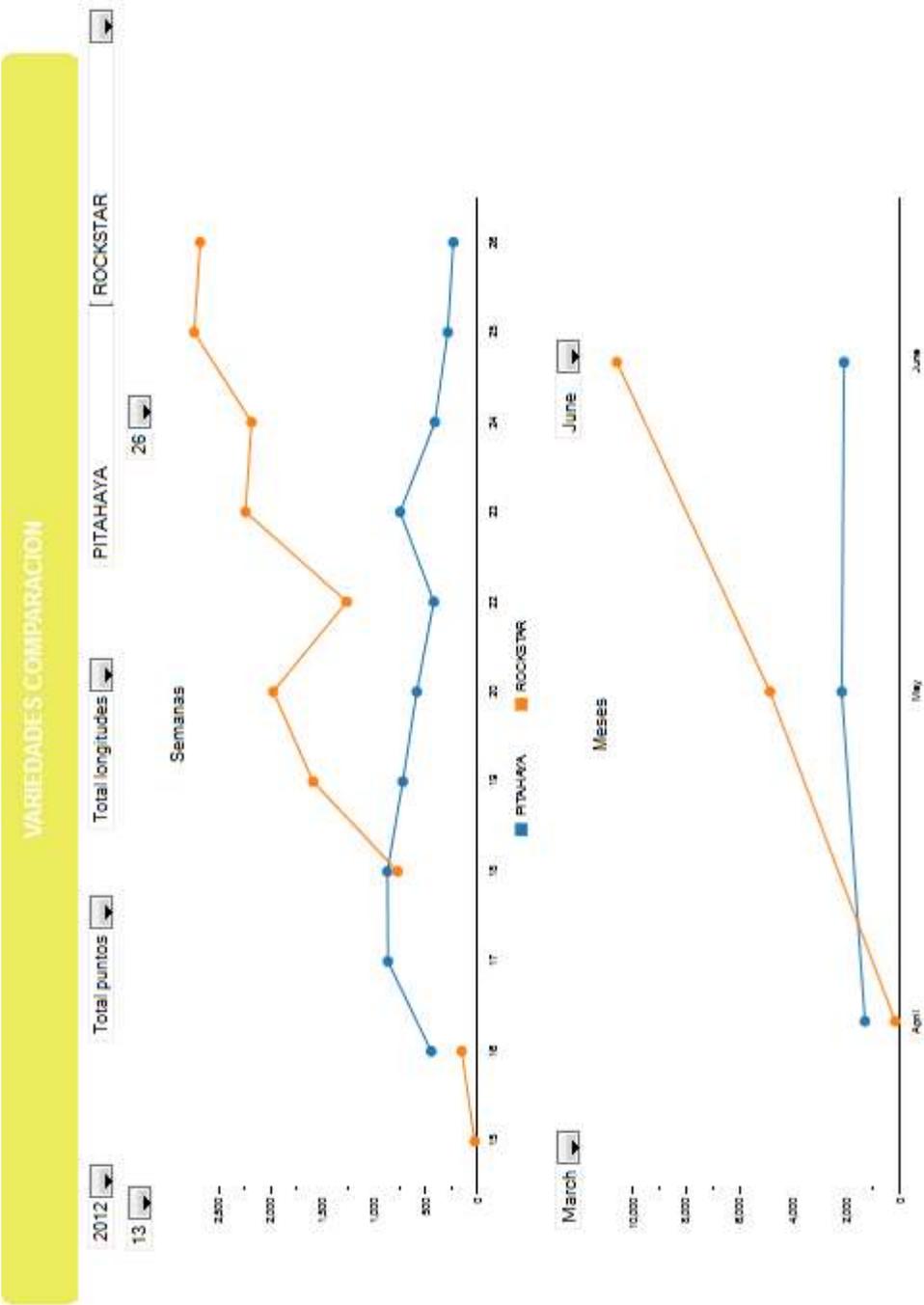
```

from [Corte]
where ( ${S1_anoParameter} ) </Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g01Query"
  id="g01Query"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
      type="String"/>
    <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
      type="String"/>
    <Parameter default="[TiempoS].[Total tiempo]" name="Ss1Parameter" type="String"/>
    <Parameter default="[TiempoS].[Total tiempo]" name="Ss2Parameter" type="String"/>
  </Parameters>
  <Query>select NON EMPTY( { ${Ss1Parameter} : ${Ss2Parameter} } ) on ROWS,
NON EMPTY( { [Measures].[Tallos exportable] } ) on Columns
from [Exportable]
where ( ${S1_anoParameter} , ${S3_puntoParameter} , ${SLongitudParameter} ,
${SVariedad1Parameter} ) </Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="G02Query"
  id="G02Query"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
      type="String"/>
    <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
      type="String"/>
    <Parameter default="[Tiempo].[Total tiempo]" name="Mm1Parameter" type="String"/>
    <Parameter default="[Tiempo].[Total tiempo]" name="Mm2Parameter" type="String"/>
  </Parameters>
  <Query>select NON EMPTY( { ${Mm1Parameter} : ${Mm2Parameter} } ) on ROWS,
NON EMPTY( { [Measures].[Tallos exportable] } ) on Columns
from [Exportable]
where ( ${S1_anoParameter} , ${S3_puntoParameter} , ${SLongitudParameter} ,
${SVariedad1Parameter} ) </Query> </DataAccess></CDADescriptor>

```

ANEXO 9

Reporte 7 Exportable



Código XML/MDX Reporte 7 Exportable

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SLongitudQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariadad1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="SVariadad2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Ss2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Mm2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g1Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="g2Query" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
  
```

```

    <Jndi>SF</Jndi>
  </Connection>
</DataSources>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
  id="S1_anioQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters/>
  <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {[Anio].[2012],[Anio].[2011],[Anio].[2010],[Anio].[2009]}, not
isempty(([Anio].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
  id="S3_puntoQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
      type="String"/>
  </Parameters>
  <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]} , not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ( ${S1_anioParameter} , ${S2_variedadParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SLongitudQuery"
  id="SLongitudQuery"
  type="mdx">
  <BandedMode>compact</BandedMode>
  <Columns/>
  <Output indexes="1,0" mode="include"/>
  <Parameters>
    <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"

```

```

        type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Longitud].CurrentMember.UniqueName'
select TopCount( filter( {Descendants([Longitud].[Total longitudes]}), not
isempty(([Longitud].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [Exportable]
where ( ${S1_anioParameter} , ${S3_puntoParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariedad1Query"
id="SVariedad1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anioParameter} , ${S3_puntoParameter} , ${SbloquesParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="SVariedad2Query"
id="SVariedad2Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
<Parameter default="[Bloque].[Total bloques]" name="SbloquesParameter"
type="String"/>
</Parameters>

```

```

    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select filter( {Descendants([Variedad].[Total variedades]} , not
isempty(([Variedad].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anoParameter} , ${S3_puntoParameter} , ${SbloquesParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss1Query"
id="Ss1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana]}), not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anoParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="Ss2Query"
id="Ss2Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[TiempoS].CurrentMember.UniqueName'
select filter( {Descendants([TiempoS].[Total tiempo] ,[TiempoS].[Semana]}), not
isempty(([TiempoS].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ( ${S1_anoParameter} )</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm1Query"
id="Mm1Query"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>

```

```

<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($S1_anoParameter)</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Mm2Query"
  id="Mm2Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Output indexes="1,0" mode="include"/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
</Parameters>
<Query>with member [Measures].[Name] as '[Tiempo].CurrentMember.UniqueName'
select filter({Descendants([Tiempo].[Total tiempo] ,[Tiempo].[Mes])}, not
isempty(([Tiempo].CurrentMember)) ) on ROWS,
{[Measures].[Name]} on Columns
from [Corte]
where ($S1_anoParameter)</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g1Query"
  id="g1Query"
  type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
  <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
  <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
    type="String"/>
  <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>
  <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
    type="String"/>
  <Parameter default="[TiempoS].[Total tiempo]" name="Ss1Parameter" type="String"/>
  <Parameter default="[Variedad].[Total variedades]" name="SVariedad2Parameter"
    type="String"/>
  <Parameter default="[TiempoS].[Total tiempo]" name="Ss2Parameter" type="String"/>
</Parameters>
<Query>select NON EMPTY({$Ss1Parameter}:{$Ss2Parameter}) on ROWS,

```

```

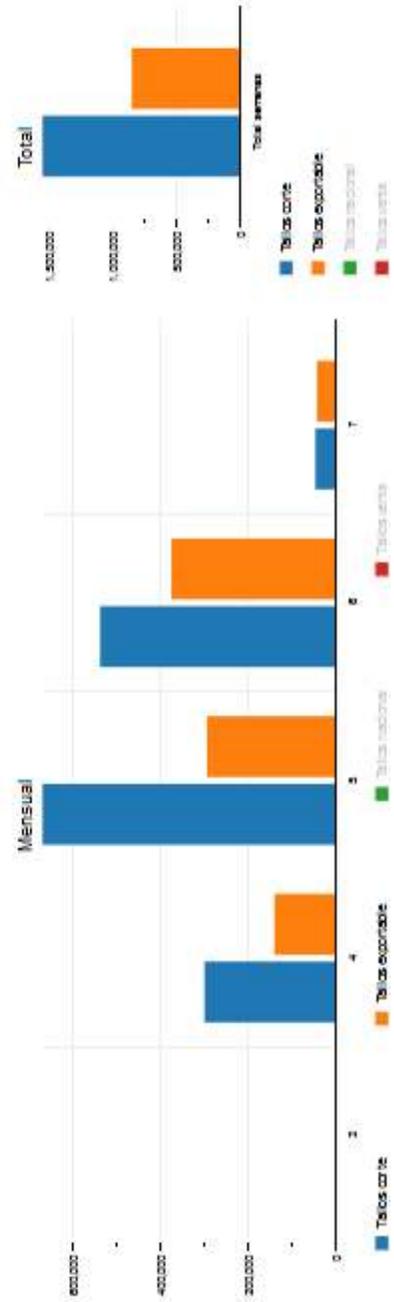
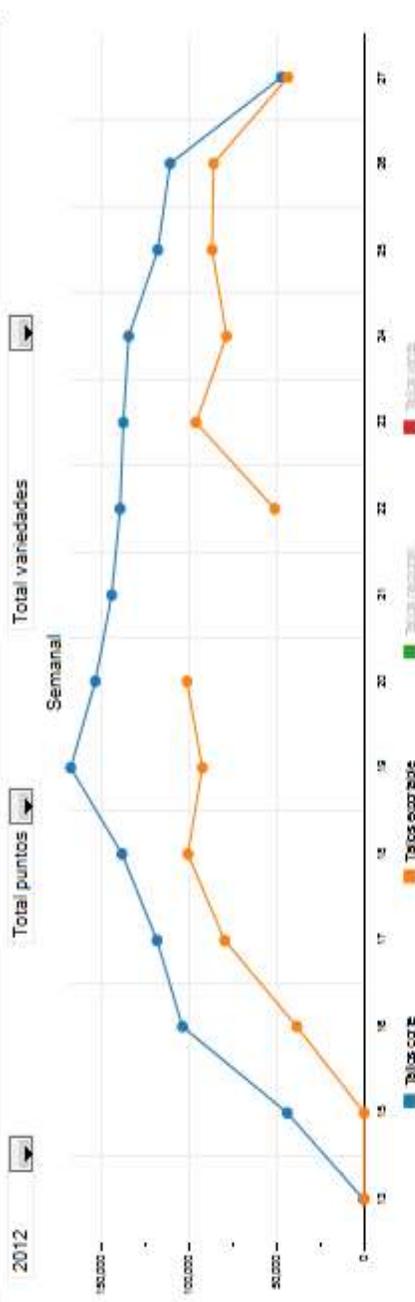
NON EMPTY({$SVariedad1Parameter},{$SVariedad2Parameter}) on Columns
from [Exportable]
where ($S1_anoParameter} , {$S3_puntoParameter} , {$SLongitudParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600" connection="g2Query"
    id="g2Query"
    type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
        <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
            type="String"/>
        <Parameter default="[Longitud].[Total longitudes]" name="SLongitudParameter"
type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="SVariedad1Parameter"
            type="String"/>
        <Parameter default="[Tiempo].[Total tiempo]" name="Mm1Parameter" type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="SVariedad2Parameter"
            type="String"/>
        <Parameter default="[Tiempo].[Total tiempo]" name="Mm2Parameter" type="String"/>
    </Parameters>
    <Query>select NON EMPTY({$Mm1Parameter}:{$Mm2Parameter}) on ROWS,
NON EMPTY({$SVariedad1Parameter},{$SVariedad2Parameter}) on Columns
from [Exportable]
where ($S1_anoParameter} , {$S3_puntoParameter} , {$SLongitudParameter})</Query>
    </DataAccess>
</CDADescriptor>

```

ANEXO 10

Reporte 8 Exportable/Corte

REPORTE GENERAL



Código XML/MDX Reporte 8 Exportable/Corte

```

<?xml version="1.0" encoding="UTF-8"?>
<CDADescriptor>
  <DataSources>
    <Connection id="S1_anioQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S2_variedadQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="S3_puntoQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Grafico_semanaQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="Grafico_mesQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
    <Connection id="TotalSMQuery" type="mondrian.jndi">
      <Catalog>/Alli/SF_Schema.xml</Catalog>
      <Jndi>SF</Jndi>
    </Connection>
  </DataSources>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="S1_anioQuery"
  id="S1_anioQuery"
  type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters/>
    <Query>with member [Measures].[Name] as '[Anio].CurrentMember.UniqueName' select
TopCount( filter( {Descendants([Anio].[Total anios]}, not isempty(([Anio].CurrentMember))
) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]</Query>
  </DataAccess>
  <DataAccess access="public" cache="true" cacheDuration="3600"
connection="S2_variedadQuery"
  id="S2_variedadQuery"

```

```

        type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Variedad].CurrentMember.UniqueName'
select TopCount( filter( {Descendants([Variedad].[Total variedades]}), not
isempty(([Variedad].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ({S1_anioParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="S3_puntoQuery"
id="S3_puntoQuery"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Output indexes="1,0" mode="include"/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
type="String"/>
    </Parameters>
    <Query>with member [Measures].[Name] as '[Punto de
corte].CurrentMember.UniqueName' select TopCount( filter( {Descendants([Punto de
corte].[Total puntos]}), not isempty(([Punto de corte].CurrentMember)) ) , 50) on ROWS,
{[Measures].[Name]} on Columns
from [General]
where ({S1_anioParameter} , {S2_variedadParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Grafico_semanaQuery"
id="Grafico_semanaQuery"
type="mdx">
    <BandedMode>compact</BandedMode>
    <Columns/>
    <Parameters>
        <Parameter default="[Anio].[Total anios]" name="S1_anioParameter" type="String"/>
        <Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
type="String"/>
        <Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
    </Parameters>

```

```

    <Query>select NON EMPTY({Descendants([Semana].[Total semanas]
,[Semana].[Semana])) on ROWS,
NON EMPTY({[Measures].[Tallos corte],[Measures].[Tallos exportable],[Measures].[Tallos
nacional],[Measures].[Tallos venta]}) on Columns
from [General]
where ({S1_anoParameter} , {S2_variedadParameter} , {S3_puntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="Grafico_mesQuery"
id="Grafico_mesQuery"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
<Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
</Parameters>
<Query>select NON EMPTY({Descendants([Mes].[Total meses] ,[Mes].[Mes])) on
ROWS,
NON EMPTY({[Measures].[Tallos corte],[Measures].[Tallos exportable],[Measures].[Tallos
nacional],[Measures].[Tallos venta]}) on Columns
from [General]
where ({S1_anoParameter} , {S2_variedadParameter} , {S3_puntoParameter})</Query>
</DataAccess>
<DataAccess access="public" cache="true" cacheDuration="3600"
connection="TotalSMQuery"
id="TotalSMQuery"
type="mdx">
<BandedMode>compact</BandedMode>
<Columns/>
<Parameters>
<Parameter default="[Anio].[Total anios]" name="S1_anoParameter" type="String"/>
<Parameter default="[Variedad].[Total variedades]" name="S2_variedadParameter"
type="String"/>
<Parameter default="[Punto de corte].[Total puntos]" name="S3_puntoParameter"
type="String"/>
</Parameters>
<Query>select {[Semana].[Total semanas]} on ROWS,
NON EMPTY({[Measures].[Tallos corte],[Measures].[Tallos exportable],[Measures].[Tallos
nacional],[Measures].[Tallos venta]}) on Columns
from [General]
where ({S1_anoParameter} , {S2_variedadParameter} , {S3_puntoParameter})</Query>
</DataAccess>

```