



# Tesis de Grado

Solución para el aumento de  
visibilidad de incidencias en  
productos software

Argañaraz Francisco José

Mirizio Esteban Ricardo

---

## *Dedicatoria*

A nuestros familiares, y amigos, por el apoyo que nos brindaron a lo largo de nuestra carrera y que fue de gran ayuda para lograr la concreción de este nuestro proyecto.

## *Agradecimientos*

A nuestra tutora, la Ingeniera María Alejandra Boggio, por su apoyo y dedicación durante el desarrollo de nuestra carrera y trabajo final de grado.

A las Ingenieras Glenda del Giudice y Melisa Caffaratti, por su incondicional cooperación y predisposición a ayudarnos cada vez que lo hemos necesitado

*Solución para el aumento de la  
visibilidad de incidencias en  
productos software*

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

INSTITUTO UNIVERSITARIO AERONAUTICO – FACULTAD DE INGENIERIA

Aprobado por el Departamento Sistemas en cumplimiento de los requisitos exigidos para otorgar el título de Ingeniero en Sistemas al señor Francisco Jose Argañaraz – DNI 34164871.

### Revisado por:

.....  
Ing. Alejandra Boggio  
Tutor de Trabajo

.....  
Lic. Alicia Salamon  
Director Departamento Sistemas

.....  
Ing. Héctor Riso  
Director Dpto. Desarrollo Profesional

### Tribunal Examinador:

.....  
Ing. Patricio Maller

.....  
Ing. Alejandra Boggio

Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

INSTITUTO UNIVERSITARIO AERONAUTICO – FACULTAD DE INGENIERIA

Aprobado por el Departamento Sistemas en cumplimiento de los requisitos exigidos para otorgar el título de Ingeniero en Sistemas al señor Esteban Ricardo Mirizio – DNI 35109048.

**Revisado por:**

.....  
Ing. Alejandra Boggio  
Tutor de Trabajo

.....  
Lic. Alicia Salamon  
Director Departamento Sistemas

.....  
Ing. Héctor Riso  
Director Dpto. Desarrollo Profesional

**Tribunal Examinador:**

.....  
Ing. Patricio Maller

.....  
Ing. Alejandra Boggio

## *Declaración de derechos de autor*

Se permite la consulta y reproducción total o parcial del contenido del presente trabajo por parte de la universidad y estudiantes con fines académicos.

Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

## *Resumen/Abstract como quieras ponerle*

Rellenar, breve resumen sobre el trabajo

INDICE DE CONTENIDO

Introducción ..... 11

Situación Problemática..... 12

Problema ..... 13

Objeto de Estudio..... 13

Campo de Acción..... 13

Objetivos ..... 14

    Generales..... 14

    Específicos ..... 14

Idea a Defender / Propuesta a Justificar / Solución a Comprobar ..... 14

Delimitación del Proyecto ..... 15

Aporte Práctico..... 15

Aporte Teórico..... 15

Factibilidad ..... 16

    Factibilidad Técnica ..... 16

    Factibilidad Operativa ..... 16

    Factibilidad Económica..... 16

Métodos de Investigación ..... 17

    Métodos Lógicos..... 17

    Métodos Empíricos..... 17

Marco Teórico ..... 18

Enfoque Metodológico..... 18

    Metodología Propuesta..... 18

Planificación del Proyecto ..... 19

    Etapas y actividades ..... 19

    Diagrama Gantt ..... 20

Analisis Del Problema ..... 21

Marco Teorico ..... 22

    Introduccion ..... 22

    Desarrollo ..... 22

        Informacion vs Datos..... 22

        Indicadores, Metricas y Medidas..... 22

        Caracteristicas de un Indicador ..... 24

        Objetivos de un indicador ..... 25

        Cuadro de mando ..... 25

Planteo de la solucion ..... 29

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Esquema de la solucion .....	30
Selección de metodos para la solucion .....	31
Identificacion de metodos .....	31
Identificacion de criterios de ponderacion .....	32
Comparacion de herramientas .....	33
Bugzilla .....	49
Introduccion .....	49
Uso de la herramienta .....	49
Aspectos generales .....	49
Ciclo de vida de un bug .....	50
Home – pantalla principal .....	51
Funciones principales .....	52
Implementacion .....	60
Análisis de la fuente de datos .....	60
Medidas Bugzilla .....	60
Dimensiones .....	60
Diseño de la base .....	64
Integración con pentaho .....	66
Pentaho .....	66
Introduccion .....	67
Kettle .....	68
Trabajo y transformaciones .....	68
Start .....	71
DropFks .....	71
Fill_bugsdb .....	72
Update_NullDates .....	72
Update_nullres .....	72
FILL_severidad / estado / prioridad / etc .....	73
Addfks .....	74
éxito .....	75
TRABAJO KETTLE AUTOMATIZADO .....	76
KITCHEN .....	76
CRON .....	77
USO DE CRON .....	77
Bi server y Administration console: cREACION DE USUARIOS Y ROLES .....	80
Desarrollo .....	81
Creacion de usuarios .....	81

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Diseñador .....	83
Mondrian.....	84
Mondrian Schema Workbench.....	84
Uso.....	85
Guardado y Publicación.....	94
Creacion de Data Source en la Consola Administrativa.....	96
Saiku Analytics.....	98
Introduccion .....	98
Desarrollo .....	98
Permisos .....	103
Introduccion .....	103
Desarrollo.....	104
Permisos hacia el esquema .....	104
Permisos hacia un cubo.....	105
Permisos hacia una dimension .....	106
Permisos hacia una jerarquia .....	107
Permisos hacia un miembro .....	108
Uso del rol .....	110
Resumen estructura de permisos Mondrian.....	112
Casos de prueba .....	113
Planteo del escenario .....	113
ANEXO .....	122
Indice de ilustraciones.....	129
Bibliografia.....	132

### INTRODUCCIÓN

El manejo y transformación de los datos en información es la clave del éxito para cualquier organización. El proceso de la transformación de datos cambia el estado de conocimiento actual del sujeto o sistema y permite a los individuos o sistemas que poseen dicho nuevo estado tomar decisiones pertinentes acordes al nuevo conocimiento adquirido.

Al igual que una organización con fines de lucro cuenta con diferentes herramientas para visualizar su estado financiero, tales como gráficos de tendencia, flujos de fondo y diversos tipos de informes, un grupo de desarrollo de software también necesita de herramientas para conocer en qué estado se encuentran sus proyectos, adquirir una vista global de los mismos y poder corregir a tiempo las desviaciones que puedan surgir.

Así como existen diferentes ámbitos en una empresa como económico, financiero y administrativo por ejemplo, si tomamos a un grupo de desarrollo de software como una organización también podemos analizarlo desde diferentes enfoques. En nuestro caso nos hemos centrado en estudiar los incidentes en un producto software. Darle luz a este aspecto permite a los jefes y líderes de proyecto contar con las herramientas suficientes para poder tener una visión global de los incidentes en los diferentes productos o componentes que desarrolla un equipo y poder tomar decisiones al respecto.

Procesados los datos se logrará luego la obtención de medidas y dimensiones a través de las cuales los usuarios podrán elaborar indicadores representativos y hacer uso de ellos para mejorar la situación actual de su grupo de proyecto, producto o componente desarrollado.

Este modelo puede ser implementado a través de distintas herramientas, las cuales tienen un uso e impacto diferente en la toma de decisiones, algunas serán más flexibles pero requerirán de mayor trabajo del lado del cliente para generar el conocimiento, mientras que otras serán más rígidas, ya que necesitarán de menos esfuerzo del lado del cliente, precisarán del apoyo de otras herramientas si se quiere obtener resultados más complejos y de mayor profundidad para su análisis.

Nuestra intención es elaborar una solución que se adapte tanto a grupos de desarrollo de software que tengan vasto conocimiento sobre la gestión de incidencias, como también a otros menos experimentados.

### SITUACIÓN PROBLEMÁTICA

Desde los comienzos del desarrollo de software hasta la actualidad la etapa de pruebas en el ciclo de vida de un software ha sido tomada a la ligera por muchos grupos de desarrollo. Sin embargo esta etapa puede significar, en el mejor de los casos, hasta un 50% más del costo del proyecto si no se realiza de la manera correcta. La gestión de incidencias apunta al tratamiento de incidentes durante o después del desarrollo de un producto. Una gestión incorrecta o incompleta de “bugs” en un proyecto lleva al fracaso u obsolescencia del mismo.

A través de la ingeniería de software podemos hacer uso de un conjunto de prácticas y metodologías que ayudan al manejo de estas incidencias, sin embargo dichas herramientas no siempre se encuentran implementadas en ambientes de desarrollo de software, más específicamente científico – técnico. Esto da a lugar a diversas falencias en los productos, que inciden directamente en la calidad de los mismos y se trasladan generando insatisfacción al usuario final.

Para comprobar la actualidad y pertinencia del proyecto, se ha escogido para el estudio de la presente tesis el Grupo de Investigación y Desarrollo (en adelante I+D) del Departamento de Informática de la Facultad de Ingeniería del Instituto Universitario Aeronáutico.

Principales problemas detectados:

1. Ausencia de herramientas que permitan:
  - Controlar
  - Informar
  - Tomar decisiones más acertadas,frente a las incidencias de un producto software.
2. Ausencia de indicadores para la gestión de incidencias en proyectos.

### PROBLEMA

El presente trabajo surge como parte de una implementación del grupo de investigación del Departamento de Informática de la Facultad de Ingeniería del Instituto Universitario Aeronáutico el cual involucra tableros de control.

Este grupo de desarrollo requiere la implementación de una solución que permita recopilar y presentar los datos de incidencias generadas y almacenadas en la base de datos de una herramienta de manejo de incidencias.

### OBJETO DE ESTUDIO

El objeto de estudio es una solución que posibilite la integración de un cuadro de mando que aumente la visibilidad de las incidencias, utilizando alguna herramienta que permita su implementación en la arquitectura dispuesta por el grupo de I+D mencionado.

### CAMPO DE ACCIÓN

El proyecto trabaja sobre los datos de las incidencias de productos software obtenidos a partir de la base de datos de la herramienta Bugzilla en la arquitectura dispuesta por el grupo de I+D.

### OBJETIVOS

#### GENERALES

El objetivo del proyecto es implementar una solución basada en el concepto de cuadro de mando, tomando como partida una base de datos de una herramienta de seguimiento de bugs. Dicha solución debe posibilitar la creación de un tablero de control para presentar medidas y dimensiones para la posterior elaboración de indicadores por parte del usuario, así como también permisos que admitan resguardar dicha información, de modo que solo sea visible a quien corresponda.

#### ESPECÍFICOS

- Comprender la estructura de la base de datos del software de seguimiento de incidencias.
- Generar medidas y dimensiones útiles que permitan posteriormente elaborar indicadores representativos acordes al modelo de negocio.
- Seleccionar la herramienta que permita implementar el tablero de control y se ajuste a los requerimientos presentados en la solución.
- Elaborar una solución eficaz que presente características deseables respecto a la usabilidad para el usuario final.
- Mejorar la situación actual de incertidumbre acerca del estado de incidencias en un grupo de desarrollo.
- Elaborar escenarios de prueba que permitan planificar y simular posteriores implementaciones de la solución.

### IDEA A DEFENDER / PROPUESTA A JUSTIFICAR / SOLUCIÓN A COMPROBAR

Este trabajo pretende demostrar la eficacia de la implementación de un cuadro de mando para analizar y monitorear indicadores válidos y necesarios para el seguimiento de incidentes en el desarrollo de software.

Una opción eficiente, que sea compatible con la arquitectura dispuesta por el grupo de I+D para aumentar la visibilidad de las incidencias.

La retroalimentación obtenida posibilitara tanto guiar el desempeño actual del grupo como apuntar a un mejor desempeño futuro del mismo y a la calidad del producto.

### DELIMITACIÓN DEL PROYECTO

Debido a que actualmente el grupo de desarrollo del Instituto Universitario Aeronáutico utiliza la herramienta open source Bugzilla, este proyecto se limitará al estudio de la implementación de un cuadro de mando a partir de dicho software, quedarán fuera del alcance de esta tesis implementaciones utilizando otro software de gestión de incidencias. Más allá de esto, el concepto del presente trabajo puede ser aplicado mediante su correcta implementación a partir de otro software de gestión de incidencias que utilice para el almacenamiento de las incidencias una base de datos relacional.

Así también el estudio, uso y funcionamiento de la herramienta Bugzilla, no están contemplados en este trabajo, sólo se limitará al análisis y utilización de sus datos.

### APORTE PRÁCTICO

Nuestro estudio de tesis resulta relevante debido a que es la aplicación de un concepto derivado del entorno empresarial orientado a las finanzas, a un entorno IT.

Dicho proyecto está dirigido a cualquier grupo de desarrollo de software con o sin experiencia en ingeniería de software.

La solución aplicada brindará una poderosa herramienta para la toma de decisiones ya que permitirá tener conocimiento claro, conciso y en tiempo real sobre el estado de los incidentes en proyectos.

De esta forma, este proyecto ayudará a resolver el problema real planteado en el grupo de desarrollo de la Facultad de Ingeniería del Instituto Universitario Aeronáutico, permitiendo mejorar la calidad del software producido, aportando visibilidad de la calidad del producto y de proyectos en general.

Por otro lado permite evaluar el crecimiento de uno o más grupos de desarrollo en cuanto a su madurez sobre la gestión de incidentes, algo que actualmente no es posible.

### APORTE TEÓRICO

Nuestro proyecto aporta la investigación y exposición que efectuaremos sobre las diferentes formas de implementar el concepto de cuadro de mando a través de herramientas open source, con el fin de utilizar aquella que resulte apropiada para la solución del problema planteado.

Si bien la solución será aplicada a un caso particular tomando como fuente de datos una aplicación en particular, el estudio y análisis de la implementación permitirá que la misma pueda ser adaptada a otros productos y/o arquitecturas, heredando así sus ventajas, haciéndola compatible con distintos grupos de desarrollo de software de diversos ámbitos.

### FACTIBILIDAD

Es factible desarrollar el proyecto propuesto, tanto técnicamente, operativa y económicamente.

A continuación se detallan estas tres factibilidades:

#### FACTIBILIDAD TÉCNICA

El grupo cuenta con los conocimientos necesarios para desarrollar y concretar este proyecto.

Para los aspectos que escapen a nuestro conocimiento actual, consideramos que toda la información que precisemos existe disponible en libros, artículos, foros, blogs y sitios web, a fin de moldear el marco conceptual como una base apropiada y comprensiva.

Debido a que la solución planteada se basa en desarrollos open source existentes, la documentación de este tipo de soluciones es amplia y los conocimientos sustentados por una comunidad.

Por dichos motivos podemos asumir que el proyecto es factible en su aspecto técnico.

#### FACTIBILIDAD OPERATIVA

El grupo cuenta con todos los medios tecnológicos necesarios para concretar este proyecto.

El grupo de I+D de la Facultad de Ingeniería del Instituto Universitario Aeronáutico es quien se beneficia con este proyecto y podemos asegurar que contamos con todo su apoyo para realizar este Trabajo Final de Grado.

La arquitectura sobre la cual se montará la solución actualmente existe y se encuentra funcionando correctamente.

Por dichos motivos podemos asumir que el proyecto es factible en su aspecto operativo.

#### FACTIBILIDAD ECONÓMICA

El proyecto es económicamente factible debido a que la investigación se realizará sobre fuentes de información y publicaciones disponibles en bibliotecas y mayormente de manera online.

Por otra parte, la implementación de la posible solución será efectuada con herramientas open source únicamente, por lo que los costos serán despreciables.

La inversión principal que se realizará será de tiempo y considerando que el trabajo que realizamos es con fines académicos, se efectuará sin fines comerciales.

### MÉTODOS DE INVESTIGACIÓN

Para llevar a cabo nuestro trabajo de tesis final de grado utilizaremos métodos de investigación lógicos y empíricos.

Los métodos lógicos se basan en la utilización del pensamiento en sus funciones de deducción, análisis y síntesis, mientras que los métodos empíricos, se aproximan al conocimiento del objeto mediante su conocimiento directo y el uso de la experiencia, como la observación y la experimentación.

#### MÉTODOS LÓGICOS

Se utilizarán métodos lógicos para:

- Analizar la situación problemática y establecer claramente el problema que se pretende solucionar.
- Relacionar adecuadamente el marco teórico a desarrollar con el objeto de estudio y el campo de acción.
- Analizar las diferentes herramientas para implementar el cuadro de mando.
- Analizar y deducir cuál de esas herramientas se ajusta mejor al problema.

#### MÉTODOS EMPÍRICOS

Se utilizarán métodos empíricos para:

- Estudiar el entorno del grupo de desarrollo de software para elaborar una solución adecuada a las necesidades del desarrollo basándonos en información obtenida mediante observación directa, reuniones con el personal y búsquedas en Internet, entre otros.
- Experimentar con las diferentes herramientas disponibles, para lograr una implementación exitosa del modelo.
- Realizar escenarios de prueba para verificar si la solución se ajusta a un grupo de desarrollo genérico.
- Observar y evaluar los resultados obtenidos.

### MARCO TEÓRICO

En el desarrollo del proyecto se consideran los siguientes ejes teóricos:

- Concepto de “Tablero de Control”, el cual conforma el objetivo de este proyecto.
- Concepto de “Cuadro de Mando” extraído de “El enfoque de Kaplan y Norton”
- Definición de “Sistema de Seguimiento de Errores” (*Bug Tracking System*) para comprender el campo de acción y el objeto de estudio.
- Concepto de ETL ( Extracción, Transformación y Carga de datos)

### ENFOQUE METODOLÓGICO

#### METODOLOGÍA PROPUESTA

Debido a que no hay una metodología que se ajuste por completo este trabajo, se desarrolla una metodología propia basada en un análisis teórico y verificación empírica.

El detalle de dicha metodología se describe a continuación:

- Se constituirá un grupo de dos personas para analizar, discutir y tomar acciones pertinentes sobre el problema planteado en nuestro trabajo y las diferentes soluciones que se ajusten al dominio del mismo.
- Se contemplará tanto el trabajo en conjunto como así también la división de tareas para eliminar los tiempos muertos en el proyecto, lo cual permitirá completarlo en tiempo y forma.
- Se acordarán reuniones semanales para trabajar en conjunto, exponer lo realizado y discutir el estado y avance del proyecto tomando como guía un diagrama de Gantt.
- Se realizará un diagrama de Gantt para especificar la dedicación prevista para las diferentes tareas y elaborar el proyecto respetando los tiempos y recursos planteados en cada etapa para poder lograr el mismo con éxito.
- Se entregarán regularmente informes parciales del trabajo y avance para la revisión y feedback por parte del tutor y así supervisar el avance del mismo.

### PLANIFICACIÓN DEL PROYECTO

#### ETAPAS Y ACTIVIDADES

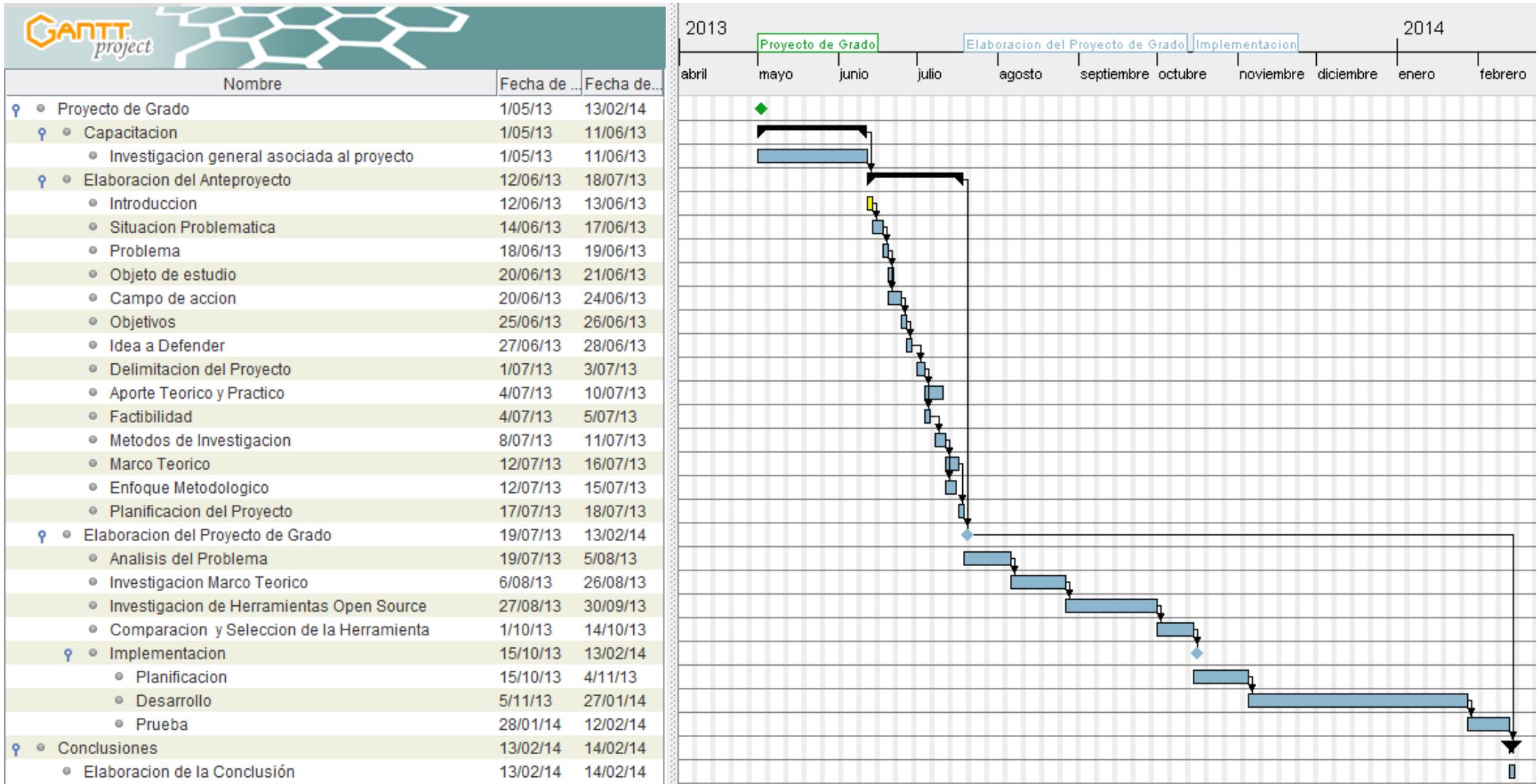
La realización del proyecto estará compuesta por cuatro grandes etapas:

- Capacitación
  - Los autores se interiorizarán con los conceptos básicos a desarrollar en el proyecto, haciendo un trabajo de investigación.
- Anteproyecto
  - Consta en la elaboración del Anteproyecto
- Proyecto de grado
  - Consta en la elaboración del Proyecto de Grado
- Conclusiones
  - Una vez finalizado el proyecto se evaluará si los objetivos fueron cumplidos y se llevará a cabo una conclusión final

# Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

## DIAGRAMA GANTT



### ANALISIS DEL PROBLEMA

Como se señaló anteriormente en el planteo del problema, en la actualidad, el control y la información que se tiene sobre los bugs o incidentes en un proyecto no permite tomar conciencia del estado en el que se encuentran diferentes componentes y/o productos. Si bien esto depende de cuan en serio se tome la etapa de testing en la realización pruebas unitarias, de integración y de sistema, en aquellos casos en que estos datos están presentes, su aprovechamiento se ve disminuido por la ausencia de herramientas que procesen estos datos y le den a un líder de proyecto un valor agregado que contribuya tanto en la concreción y entrega del software/producto al usuario final.

Desde ya, esto se ve agravado en entornos de desarrollo de software los cuales no están familiarizados con conceptos o herramientas de la ingeniería en software que faciliten esta toma de conocimiento.

Debido a que actualmente no hay ninguna herramienta planteada para enfrentar la solución se llevara a cabo una búsqueda entre diferentes herramientas open source seleccionando aquella que se ajuste mejor brindando una solución rica y adecuada que permita tomar el control sobre las incidencias en un proyecto.

En el desenlace del marco teórico será abordado el análisis de las salidas de la herramienta Bugzilla para seleccionar aquellos datos que permitan conformar métricas representativas para un grupo de desarrollo de software genérico, tarea la cual nos sentimos preparados para llevarla a cabo aplicando nuestros conocimientos en ingeniería de software.

### MARCO TEORICO

#### INTRODUCCION

En este marco teórico se describen las teorías, tendencias, tecnologías y otros conceptos teóricos que son necesarios estudiar, analizar y dominar para plantear una solución al problema que se enfrenta en este proyecto.

Durante su desarrollo se trataran conceptos como, indicadores, métricas y cuadro de mando entre otros. La comprensión de dichos conceptos permitirá abordar la solución de una manera correcta desde el punto de vista técnico y practico.

Luego de terminar con el desarrollo conceptual se procederá a la presentación del análisis de las salidas de bugzilla. Por último se documentara la investigación realizada sobre diferentes opciones, Open Source, que se presentan para la implementación de una solución para la gestión de incidencias. Las mismas se compararan mediante una serie de criterios ponderados, que expondrán a la herramienta correcta por su propio peso.

#### DESARROLLO

---

##### INFORMACION VS DATOS

La información es un elemento fundamental para la toma acertada de decisiones, antes de continuar desarrollando el actual marco teórico es de gran importancia tener en claro la diferencia entre dos conceptos: los datos y la información.

Los datos son unidades de información que pueden incluir números, observaciones y hechos, pero que al estar desligadas de un contexto en particular, carecen de sentido informativo. Es decir, los datos por si solos no nos dicen nada.

La información, es un conjunto organizado de datos procesados, que constituyen un mensaje o hecho sobre un determinado fenómeno y proporciona significado o sentido a una situación en particular. El momento en que los datos se convierten en información es cuando aportan significado, relevancia y entendimiento, en un tiempo y lugar específico.

Cabe aclarar que no siempre la misma información es valiosa para diferentes entidades. Lo que puede ser información para una persona puede ser simplemente irrelevante para otra. El principal desafío se encuentra en separar los datos que tienen sentido y aportan información, para asegurarse de que el análisis se realice basado en la mejor información disponible.

---

##### INDICADORES, METRICAS Y MEDIDAS

Teniendo a la información como eje central para la toma de decisiones, es imprescindible que dicha información sea de calidad, que permita valorar y medir la realidad. Por esta razón, es que se incurre en la elaboración de indicadores que ayuden al análisis del comportamiento de todas las variables de aquello que se busca estudiar o comprender.

Un conjunto de indicadores debe permitir en una primera instancia extraer una foto del estado del fenómeno en un momento del tiempo, a partir del cual se elabore un diagnóstico que sirva para la formulación de planes, programas y proyectos, a las cuales se les haga permanente seguimiento y evaluación. De esta manera, un seguimiento y evaluación basado en indicadores, es una herramienta que permite valorar el nivel de cumplimiento de los objetivos propuestos de un plan, programa o proyecto, para lo cual identifica y selecciona la información que permita tomar decisiones, aplicar correctivos y sistematizar experiencias.

### MEDIDAS

Una medida *“es un hecho puntual a medir”*. La existencia de medidas numéricas facilita el conocimiento de un fenómeno. Una medida proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto.

Por ejemplo:

- Un programa tiene 10.000 líneas de código. (Medida -> Cantidad de líneas de código)
- Un programa tiene 236 Bugs. (Medida -> Cantidad de bugs)

Dado que las medidas son un hecho aislado y no sirven para comparar, es por eso que necesitamos las métricas. Podemos mapear a las medidas como datos, que obtenemos o que están a nuestra disposición a través de la medición, u/o alguna herramienta o persona que nos la brinda.

Es deseable que esas mediciones o fuentes de datos sean altamente confiables, ya que de ello dependerán los resultados que se obtengan.

En tal sentido, una adecuada medición deberá contar con las siguientes características:

- **Pertinente:** deben ser relevantes y útiles para facilitar la toma de decisiones que sobre la base de tales mediciones.
- **Precisa:** debe reflejar fielmente la magnitud que se quiere analizar, a través de una adecuada definición de las características y un instrumento de medición.
- **Oportuna:** que se encuentre disponible en el tiempo en que la información es relevante para la toma de decisiones, para corregir y/o prevenir.
- **Económica:** debe existir una proporcionalidad entre los costos en la medición y los beneficios y la relevancia de la misma.

---

### METRICAS

Son las métricas aquellas que le dan “uso” y “valor” a las medidas. Una métrica *“es la combinación entre una o más medidas y su visualización a través de una o más dimensiones”*. Llamamos dimensión a todos aquellos datos que nos permiten agrupar, filtrar o seleccionar estas medidas. Así una dimensión válida podría ser el tiempo, o provincias o algún criterio que nos permita agrupar dichas medidas.

Cada dimensión puede poseer uno o más niveles de jerarquía, dependiendo de la granularidad en que queramos observar y adentrarnos en dicha dimensión. Sin ampliar demasiado, niveles de jerarquía posibles para la dimensión tiempo serían: Año-Trimestre-Semana-Día. Todo depende del nivel de granularidad en el que estemos interesados.

Ampliando esta definición señalamos algunos ejemplos de métricas:

Ej.

- **Cantidad de bugs por mes** (Cantidad de bugs + dim. Tiempo)
- **Cantidad de bugs por componente** (Cantidad de bugs + dim. Componente)
- **Cantidad de bugs semanales por producto** (Cantidad de bugs + dim. Tiempo + dim. Producto)

### INDICADOR

Como bien señalamos anteriormente un indicador concreta “*la medición de un objetivo*”, marca una meta a cumplir y está basado en las métricas.

Ej.

**La cantidad de bugs por producto** debe ser menor 100 para que el producto sea aceptado, si llegara a ser mayor se rechazará y en caso de ser igual quedará en estado de evaluación.

**Indicador:**

- ❖ Cantidad de bugs < 100 → Producto Aceptado.
- ❖ Cantidad de bugs = 100 → Producto En Evaluación.
- ❖ Cantidad de bugs > 100 → Producto Rechazado.

Luego de este ejemplo puede causar la impresión de que un indicador bien puede confundirse con métrica, si bien son conceptos encontrados podemos decir que:

- Las métricas son el fundamento de los indicadores
- Un indicador es una métrica o combinación de métricas que proporcionan una visión, que nos brindan conocimiento acerca del estado o progreso de un producto, proyecto, etc.
- Podemos decir que las métricas por si solas funcionan como indicadores cuando son representativos para un escenario en particular. De no ser así pueden combinarse para lograr el efecto deseado.

---

### CARACTERÍSTICAS DE UN INDICADOR

Un indicador debe cumplir con tres características básicas:

- 1. Simplificación:** la realidad en la que se actúa es multidimensional, un indicador puede considerar alguna de tales dimensiones (social, cultural, política), pero no puede abarcarlas todas.
- 2. Medición:** permite comparar la situación actual de una dimensión de estudio en el tiempo o respecto a patrones.
- 3. Comunicación:** todo indicador debe transmitir información acerca de un tema en particular para la toma de decisiones.

### OBJETIVOS DE UN INDICADOR

Los indicadores son herramientas útiles para la planeación y la gestión general en cualquier ámbito.

Sus objetivos principales:

- Generar información útil para mejorar el proceso de toma de decisiones, el proceso de diseño, implementación o evaluación de un plan, programa, etc.
- Monitorear el cumplimiento de acuerdos y compromisos.
- Cuantificar los cambios en una situación que se considera problemática.
- Efectuar seguimiento a los diferentes planes, programas y proyectos que permita tomar los correctivos oportunos y mejorar la eficiencia y eficacia del proceso en general.

---

### CUADRO DE MANDO

Un cuadro de mando es un conjunto de indicadores que aportan información sumariada e inteligente al usuario.

Un cuadro de mandos ideal posee a las siguientes características:

- Sumarizado: Proporciona el resumen o "Titulares" al usuario para que este, de forma rápida y ágil pueda saber cómo se está comportando su organización.
- Inteligente: Inteligente porque responde a preguntas. Las mismas deben cumplir alguna de las 2 condiciones:
  - Responde a preguntas del tipo "por qué me das esta información". Debe ser capaz de respaldar y proporcionar toda la información que hay detrás de cada "titular" que proporciona.
  - Responde a preguntas importantes para el usuario que hacen referencia a "elecciones" que debe o quiere tomar. Una gran discusión en torno a los cuadros de mandos es acerca de qué indicadores elegir, qué se muestra en el cuadro de mandos y qué no es suficientemente relevante.

El éxito de un cuadro de mando depende de su uso. Su uso depende de:

- La relevancia de la información que aporta
- La capacidad comunicativa del mismo

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Como guía, un buen cuadro de mando debería cumplir con las características **S.M.A.R.T**:

- **Sinergia:** Debe ser ergonómico y visualmente efectivos para el usuario.
- **Motivador:** Debe mostrar KPIs (indicadores) críticos y necesarios para la toma de decisiones.
- **Acertado:** La Información que se presenta debe ser totalmente precisa. Los datos de panel deben haber sido muy probados y validados.
- **Responsivo:** Debe responder a los umbrales predefinidos por el usuario. Para atraer de inmediato la atención sobre los asuntos críticos.
- **Tiempo:** Debe mostrar la información más actualizada posible. Debe requerir el menor tiempo posible para leerlo.

El usuario es la persona o personas para las que se va a diseñar un cuadro de mando y presentar la información de forma que sea útil para él. Esto supone o pretende una relación uno a uno entre usuario y cuadro de mando, desde un punto de vista práctico es mucho más performante agrupar por tipos de usuario, normalmente en función del uso que le vaya a dar.

Debido a que existen diferentes tipos de usuarios es importante saber qué tipo va a consumir ese cuadro de mando y saber que uso le va a dar para así utilizar un diseño u otro.

Los diferentes tipos de usuarios que se pueden encontrar son:

- **Usuarios Operacionales:** Personas que necesitan gestionar el trabajo diario y necesitan acceder a datos muy concretos. No necesitan una evolución histórica sino saber que está pasando en este momento. Pueden necesitar de un cuadro de mando Operacional, que les ayude a llevar su trabajo diario.
- **Usuarios Tácticos:** Personas responsables de un departamento. Necesitan la información contextualizada y sobretodo histórico. Ej. Histórico de ventas.
- **Usuarios directivos:** Personas que tienen grandes responsabilidades y que necesitan tener la información muy rápidamente de una manera muy ágil y muy visual los cuadros de mando de estos suelen ser cuadros de mandos muy sencillos técnicamente y muy visuales.

Teniendo en cuenta la anterior clasificación, dichos usuarios pueden ser mapeados con un tipo específico de cuadro de mando según su rol. Así, cada tipo cuadro de mando es construido específicamente para adecuarse al tipo de tareas que desarrollan los usuarios que encuadran dentro de un rol.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Se puede clasificar a los diferentes cuadros de mando en:

### Operacionales

- De respuesta rápida
- Simple, fácil de utilizar, sin demasiadas opciones y muy a la vista.
- Capaz de mostrar alertas y eventos
- La información es a nivel de detalle, no quieren saber cómo va toda la organización sino porque un paquete no ha salido a tiempo o en que camión está.
- Suelen tener pocas gráficas y gran cantidad de tablas que proporcionan acceso directo a la información detallada.

Ej. Cuadros de mando para personal que maneja stock, registro de entradas y salidas.

### Tácticos

Recomendados para usos de directivos intermedios, que lo que necesitan es dar respuestas a sus preguntas. Su información no necesita tanto detalle, pero si consolidada.

- Históricos, comparativos, contextualizados en definitiva
- Con datos agregados y consolidados
- Componentes gráficos más elevados y complejos que el cuadro operacional.
- La visualización de la información se vuelve más compleja
- Su objetivo es buscar respuestas acerca de las causas, buscar explicaciones. No tanto adentrarse en los detalles concretos. Aunque deben permitir llegar hasta cierto nivel de detalle si es necesario.

### Directivos

Tiene muy poco tiempo necesita información muy gráfica y resumida. Tipos de cuadro de mando que suelen resumirse en alertas y que suelen decir que cosas van bien y que cosas que no van bien. Las respuestas al “porque” esas cosas van bien o mal deberían buscarse en el cuadro de mando táctico.

- De respuesta rápida
- Capaz de mostrar alertas y eventos
- Con datos agregados y consolidados
- Que se pueda “imprimir”
- Cuadro de mando integral como ideal.

### Adhoc

Cualquiera de los usuarios anteriores en cualquier momento puede querer acceso a cierta información. Este cuadro de mando podría considerarse de “propósito general”. Como ventaja proporcionan al usuario final gran flexibilidad e independencia. Permite jugar con la información. Como inconveniencia aquello que se gana en facilidad se pierde en flexibilidad. Es un equilibrio entre flexibilidad y potencia.

- **Tan simple como sea posible**
- **Tan flexible como sea posible:** El usuario no sabe en principio que información quiere analizar ni cómo quiere analizarla.
- **Tan escalable como sea posible:** Debe ser tan configurable y completo como demande la imaginación del usuario. Involucra opciones extra para usuarios avanzados, no vitales para el funcionamiento del cuadro.
- **Tan visual y agradable como sea posible:** Cuando se analiza información por curiosidad o interés no se busca confundir al usuario con miles de opciones.

### PLANTEO DE LA SOLUCION

Como preludeo al desarrollo de la solución es importante plantear y sentar las bases de lo que se va a llevar a cabo a lo largo de este trabajo. Dichas bases serán realmente importantes a la hora de decidir las herramientas que se van a utilizar y apoyar las decisiones de implementación tomadas en este trabajo de grado.

Luego de un análisis previo se llegó a la conclusión en ciertos aspectos, centrales para el éxito de dicha tesis como solución a un problema real.

Planteado un escenario genérico dentro de un entorno de desarrollo de software normal se espera abordar la solución apuntando a las siguientes premisas:

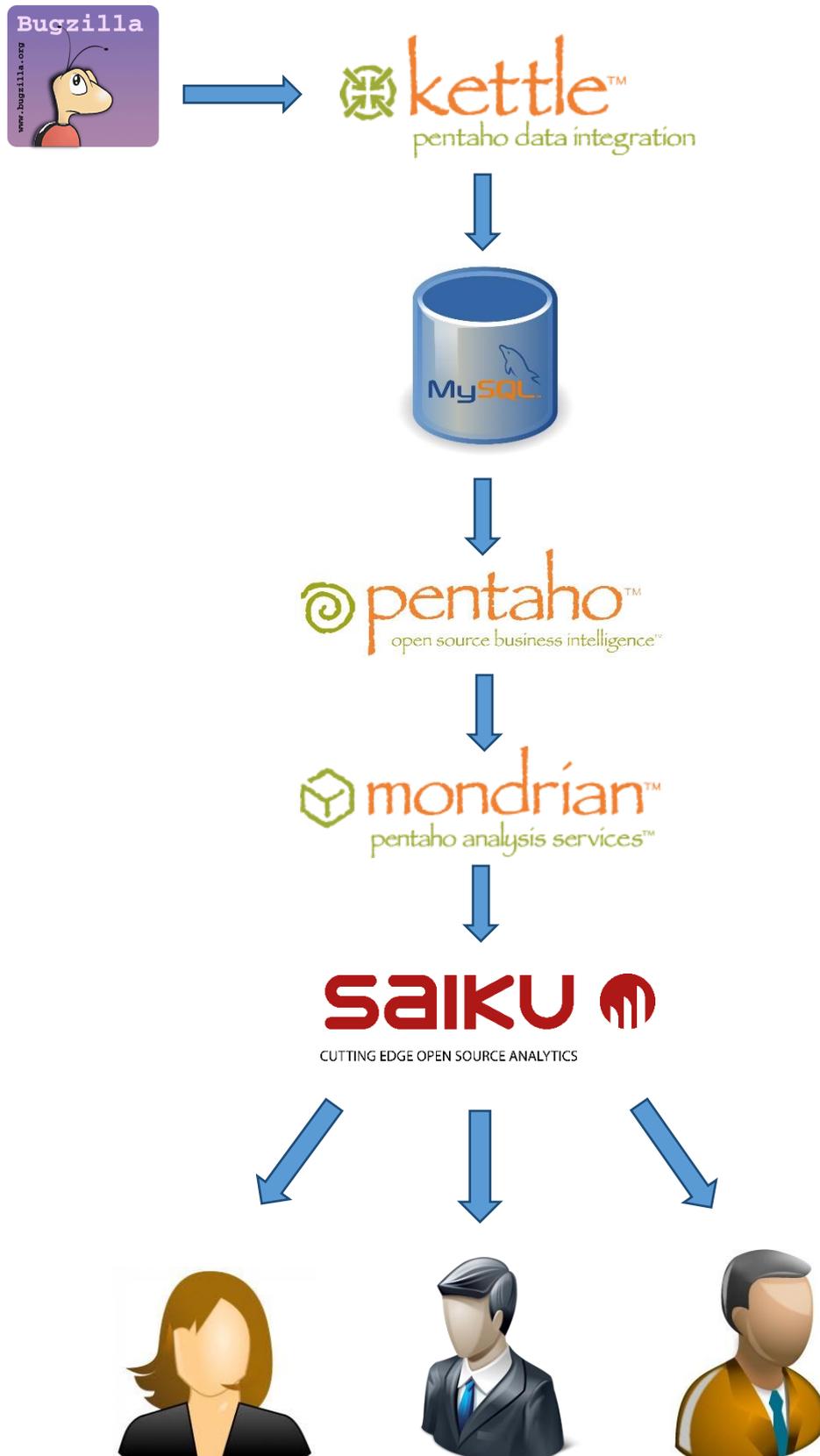
- La herramienta seleccionada como “Front-End” de la solución debe presentar un alto grado de usabilidad, facilitando el manejo de la misma a los usuarios finales, ya que de presentar gran complejidad podría ser rechazada por el entorno de desarrollo.
- Todo usuario debe tener la posibilidad de hacer un “buen” uso de la misma sin necesidad de poseer conocimientos técnicos previos sobre el manejo de métricas, medidas e indicadores.
- La solución integral debe ser de utilidad tanto para un usuario que se encuentra en el nivel “técnico/operativo” en este caso programadores, diseñadores como para aquellos que se encuentran en el nivel Ejecutivo tales como, líderes de proyecto, jefes, etc.
- En consecuencia al punto anterior, la solución debe incluir algún esquema “personalizable” de permisos jerárquico o por roles que resguarde la visualización de la información según las políticas de la organización o grupo de desarrollo. No se contemplara la implementación de una herramienta que brinde abiertamente toda la información a todos los usuarios, ya que esto no es deseable para una organización, al menos de este tipo, así también no juega a favor la exposición innecesaria de cierta información para áreas que no pueden sacar provecho de la misma.
- Es deseable que la “herramienta” brinde el mayor grado de flexibilidad posible para la “fabricación” y elaboración de métricas por parte de los usuarios finales. Permitiendo así la generación de indicadores de negocio representativos para cada grupo, área o usuario en particular. Como se dijo anteriormente no siempre la misma información es validad para diferentes entidades.
- Si bien se apunta a un grupo de desarrollo estándar es deseable lograr que la solución sea performante en todos los aspectos posibles.
- Es deseable que la herramienta permita que la información pueda ser presentada a los usuarios de diversas maneras y formatos, sobre todo gráficos simples pero representativos, para enriquecer y facilitar la toma de decisiones.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

A continuación se plantea el esquema de la solución, mostrando a gran escala, como interactúan entre si las diferentes herramientas que fueron utilizadas para realizar el trabajo final de grado.

### ESQUEMA DE LA SOLUCION



### SELECCIÓN DE METODOS PARA LA SOLUCION

Luego de establecer las premisas más importantes para la selección de un método adecuado para el manejo de datos, se procederá a la identificación y comparación entre ellos aplicando una ponderación multicriterio por scoring a fin de evaluarlos y obtener la conclusión relativa a cuál es la solución que es conveniente utilizar en el Trabajo Final de Grado.

---

#### IDENTIFICACION DE METODOS

Antes de comenzar, cabe destacar que lo que se comparan son herramientas OpenSource de soluciones para la implementación de una solución basada en la generación de un cuadro de mando. Sin embargo dichas herramientas no encuadran dentro del mismo “método” sino que forman parte de diferentes conceptos.

Durante la comparación teórica de dichos métodos se han llevado a cabo implementaciones prácticas parciales con herramientas reales para comprobar y demostrar la veracidad de los criterios puestos en escena.

A continuación se detallan los métodos posibles elegidos por nosotros para la generación de cuadros de mando:

- Dashboard
- Procesamiento Olap

Debido al tiempo acotado en el cual se desarrolla este trabajo de grado, se ha decidido seleccionar solo una herramienta representativa de cada método, que luego de análisis previo, se decidió era la más completa y que se calificaba como apta para la solución.

Dentro de las herramientas posibles para procesar y mostrar los datos se analizaron las siguientes:

- Comunity Dashboard Edition (Dashboard)
- Mondrian Schema Workbench (Procesamiento Olap) con Saiku Reporting como Front-End.

### IDENTIFICACION DE CRITERIOS DE PONDERACION

A continuación se enumeran los diferentes criterios a tener en cuenta en la comparación, con su respectiva justificación y scoring. Se contemplaran tanto criterios de uso como todo aquello referente para su futura implementación para entornos de desarrollo.

- ❖ **Complejidad de implementación:** Este criterio hace referencia a la dificultad que posee la herramienta para su implementación. La importancia reside en la posibilidad de reproducir dicha solución en un futuro para grupos de desarrollo.
- ❖ **Mantenibilidad:** Este criterio representa la facilidad con la que se pueden introducir cambios en la herramienta vs el esfuerzo que representa plasmar dichas modificaciones. Contempla modificaciones dentro de lo que la aplicación está preparada para hacer, no involucra cambios en el código fuente.
- ❖ **Usabilidad (Front-End):** Este criterio hace referencia a la dificultad que representa para el usuario el uso correcto de la aplicación para el fin deseado. Que tan “intuitiva” es la aplicación para el usuario final. Que tan difícil resulta comprender la forma de utilizar la aplicación para que el usuario obtenga la información que desea.
- ❖ **Flexibilidad:** Este criterio representa la potencialidad de la herramienta para ir más allá de lo que el usuario pueda requerir de la aplicación. En herramientas muy limitadas o poco flexibles se delimitan a funcionalidades para cubrir las necesidades básicas dentro de la solución.  
La capacidad de flexibilidad extiende estas funcionalidades básicas y posee capacidad para adaptarse a escenarios de uso de diferentes usuarios brindando más valor agregado a la aplicación.
- ❖ **Permisos:** Este criterio evalúa la capacidad de la herramienta de ofrecer un sistema o jerarquía de permisos adecuada para la solución. Esto comprende que los permisos sean personalizables, admitan una jerarquía, preferentemente con roles que agrupen tipos de usuarios, buscando como resultado final el filtro de información de manera simple y adecuada.
- ❖ **Soporte de orígenes de datos:** Este criterio evalúa la capacidad de la herramienta para aceptar diversas fuentes de datos, como archivos, bases de datos o estructuras concretas.

#### DESARROLLO TEORICO

El método del **Scoring** es una manera rápida y sencilla para identificar la alternativa preferible en un problema de decisión multicriterio.

Básicamente el método consiste en la postulación de criterios representativos para la toma de decisiones, a dichos criterios se les asigna un “peso” relativo a la importancia que poseen en la decisión. Luego se toman las diferentes alternativas de solución y se le da un puntaje a cada criterio dependiendo de cuanto cumple o no con el mismo.

Aquella alternativa que posea mayor puntuación basada en la suma y multiplicación de los pesos de los criterios por su valor es la que será elegida como la adecuada.

#### ETAPAS DEL PROCESO DE SCORING MULTICRITERIO:

---

1. Identificar la Meta General del Problema
2. Identificar las Alternativas
3. Listar los Criterios a emplear en la toma de decisión
4. Asignar una ponderación para cada uno de los Criterios
5. Establecer en cuanto satisface cada Alternativa a nivel de cada uno de los Criterios
6. Calcular el Score para cada una de las Alternativas
7. Ordenar las Alternativas en función del Score. La Alternativa con el Score más alto representa la Alternativa a recomendar.

#### Fórmula para calcular el Score

$$S_j = \sum_{i=0}^n W_i \cdot R_{ij}$$

Donde:

$R_{ij}$  = Rating de la Alternativa j en función del Criterio i

$W_i$  = Ponderación para cada Criterio i

$S_j$  = Score para la Alternativa j

---

## DESARROLLO PRÁCTICO

1. **Meta General:** Seleccionar la mejor alternativa para implementar una solución OpenSource basada en la generación de cuadros de mando.

2. **Alternativas:**

- Community Dashboard Edition
- Saiku
- Reporting

3. **Criterios:**

- Implementación
- Usabilidad
- Mantenibilidad
- Flexibilidad
- Permisos
- Escalabilidad
- Soporte Orígenes de datos

4. **Asignación de pesos**

<i>Criterio</i>	<i>Peso</i>
<i>Implementación</i>	2
<i>Usabilidad</i>	5
<i>Mantenibilidad</i>	3
<i>Flexibilidad</i>	5
<i>Permisos</i>	5
<i>Escalabilidad</i>	3
<i>Soporte orígenes de datos</i>	3

<b>Detalle de pesos</b>
1 = Muy poco importante
2 = Poco importante
3 = Medianamente importante
4 = Importante
5 = Muy importante

<b>Rating de satisfacción</b>
1 = Mu bajo
2 = Bajo
3 = Medio
4 = Alto
5 = Muy Alto

### 5. Asignación de rating de las alternativas

A continuación se presenta el análisis de cada herramienta en el contexto de los criterios definidos en los puntos anteriores.



Saiku es una herramienta liviana de análisis y reporting desarrollada por “Meteorite.” Construido sobre Mondrian (Servidor Olap) y OLAP4J (una API de java abierta para la construcción de aplicaciones olap). En esta arquitectura cumple el rol de front-end permitiendo visualizar los datos almacenados en una estructura olap, en este caso generada por el “Mondrian Schema Workbench”.

Saiku nos ofrece una solución de análisis basada en web, muy amigable para los usuarios finales. Permite a los mismos analizar datos de forma fácil y rápida además de crear y compartir informes permitiendo la generación de gráficos y tablas dinámicos, explorando datos en tiempo real.

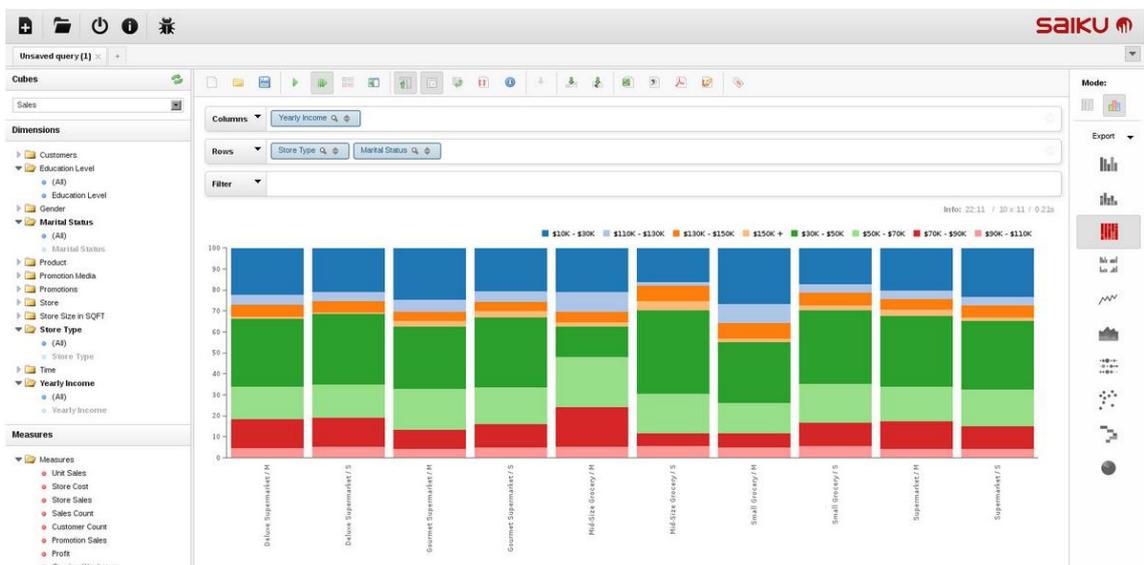
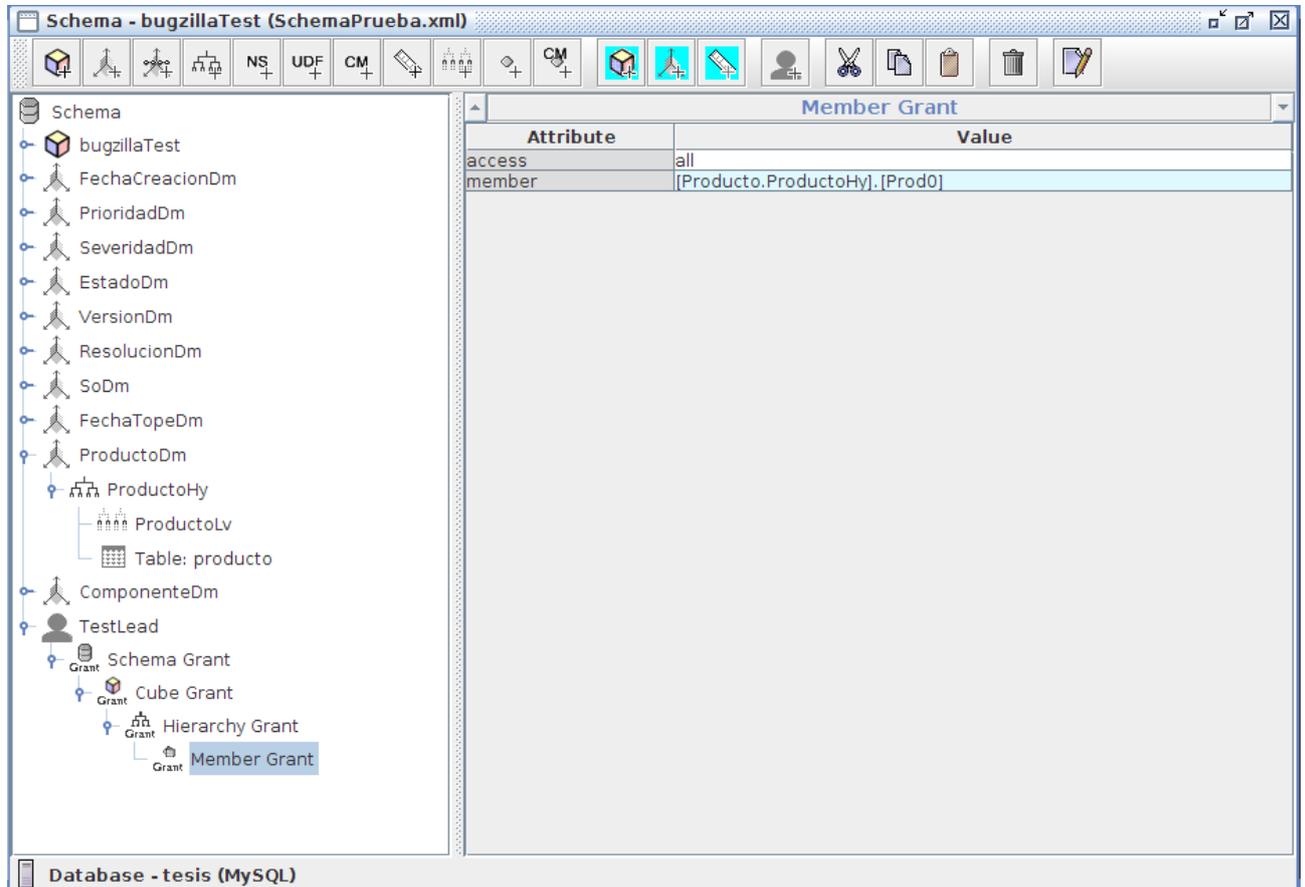


Ilustración 1: Saiku reporting (Front-End)

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Mondrian Schema Workbench (también llamado Pentaho Schema Workbench, en adelante PSW) permite a los usuarios crear y probar esquemas de cubos OLAP de manera gráfica, los cuales serán luego consumidos por saiku para permitir ser presentados gráficamente por el usuario final.



**Ilustración 2: Mondrian Schema Workbench**

### **Complejidad de implementación:**

La implementación de saiku puede ser realizada de dos maneras:

1. Como un servidor independiente
2. Como un plugin del servidor BI de la herramienta Pentaho.

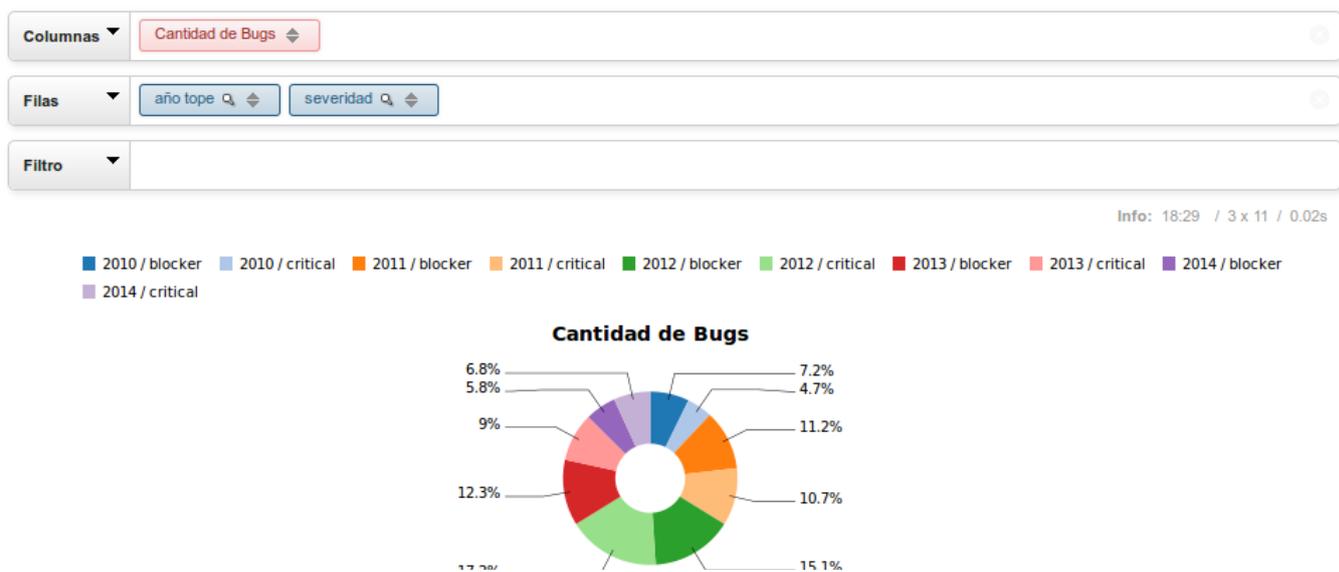
Las dos implementaciones ofrecen una interfaz casi idéntica con las mismas funcionalidades. En ambos casos la implementación de las mismas no presenta dificultad alguna en cuanto a la instalación.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Para la puesta en marcha y utilización solo basta con configurar el origen de datos y proporcionarle una estructura de datos del tipo olap. La configuración del origen de datos puede llevar un poco de tiempo debido a que debemos encontrar nuestra URL de conexión de acuerdo al driver de la fuente de datos. Toda esta información se puede encontrar ejemplos en la página oficial de saiku o en la página de los respectivos desarrolladores de la fuente de datos.

Luego la herramienta se encarga automáticamente de brindarnos las medidas y dimensiones para generar nuestros propios cuadros de mando, permitiéndonos seleccionar la forma en la que queremos visualizarlo.



**Ilustración 3: Visualización de una consulta mediante "Saiku Reporting"**

En cuanto a la implementación de "Mondrian Schema Workbench", dicha herramienta requiere conocimientos básicos de business intelligence para modelar un cubo a partir de una base de datos relacional. Esto supone una barrera alta en cuanto al conocimiento técnico requerido ya que modelar una estructura olap no es un tema trivial y los resultados posteriores obtenidos dependerán de la capacidad y calidad para modelar que posea el encargado de realizar dicha tarea.

Superada esta barrera, la interfaz que nos provee el software es relativamente sencilla de utilizar y se encuentra correctamente documentada, en la página oficial de Mondrian.

<http://mondrian.pentaho.com/documentation/schema.php>

**Score: 2 (Bajo)**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Mantenibilidad:

Una de las cosas que se puede esperar al trabajar con cuadros de mando es la necesidad de nuevas métricas, dimensiones o indicadores para la posterior generación de un cuadro de mando que permita ver una realidad en particular. En el caso de Mondrian + Saiku esto no requeriría de ninguna modificación del lado de la aplicación. Con respecto a la formulación de nuevos indicadores, los mismos pueden generarse combinando las medidas y dimensiones disponibles. Es esperable que la modelación del cubo se realice de manera correcta para solo llevarse a cabo una sola vez y en dicha tarea se extraigan todas las dimensiones y medidas posibles para poner a disposición de los usuarios, por lo cual no debería ser necesaria la remodelación del cubo. Para la aplicación seguirá siendo transparente, ya que no requiere ningún cambio, pudiéndose dejar la configuración de la fuente de datos tal como está, así como también los roles y permisos.

Un cambio en la estructura de los datos de entrada no repercute en la aplicación de ninguna manera.

El único caso en el cual solo habría que volver a modelar el cubo sería si surgiera la necesidad de cambiar la estructura de la fuente de datos, en este caso una base de datos relacional, algo que no debería, o no es deseable, que ocurra muy a menudo.

**Score: 4 (Muy Alto)**

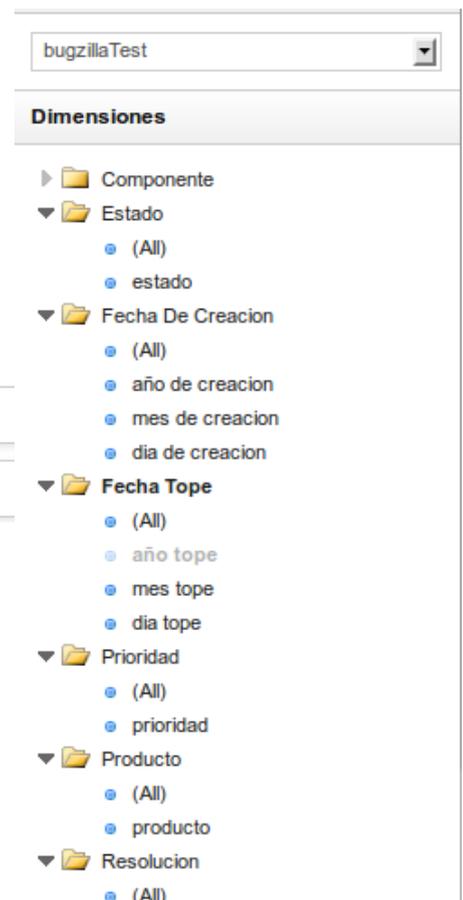
### Flexibilidad:

Las herramientas en conjunto permiten al usuario generar sus propios cuadros de mando de manera dinámica, siendo el usuario quien elabora las métricas mediante la libre combinación de las medidas y dimensiones ofrecidas. Esto presenta ventajas y desventajas. Dentro de lo favorable, notamos que el usuario no se ve limitado por las métricas posibles, puede crear las propias y visualizarlas mediante diferentes tipos de gráficos dinámicos a su elección con tan solo un click. Como desventaja señalamos que la potencia del dashboard generado puede verse disminuida en cuanto a personalización de los gráficos ya que si bien la elección del tipo de gráfico es libre y parece cubrir todas las necesidades, se ve limitada por la disponibilidad de tipos de gráficos que nos ofrece la herramienta, que podrían ser más y no permite personalización.



*Ilustración 4: Combinación de medidas y dimensiones*

**Score: 4 (Alto)**



*Ilustración 5: Selección de dimensiones*

## Trabajo Final De Grado

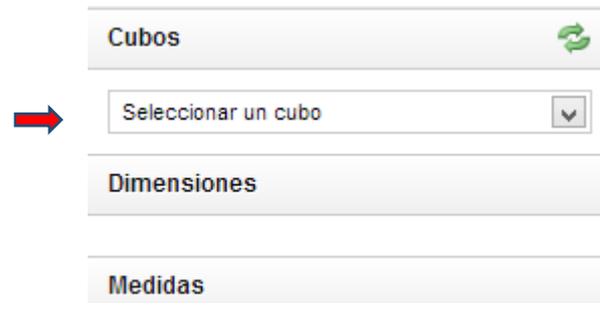
Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Usabilidad:

El programa presenta una interfaz de usuario amigable y no son muchas las acciones requeridas para la generación de un dashboard a utilizándolo.

Un caso de uso básico como generar un dashboard, consiste básicamente en:

1. Seleccionar la fuente de datos



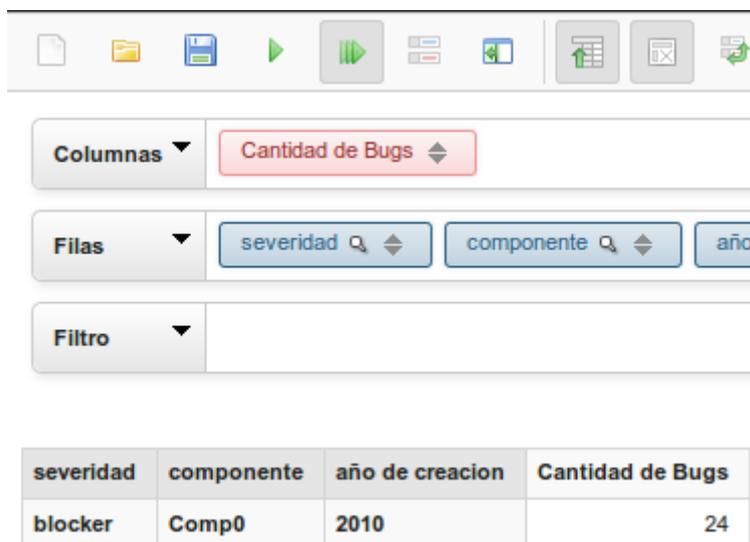
*Ilustración 6: Selección cubo*

2. Seleccionar las medida y dimensiones deseadas para la generación de la consulta

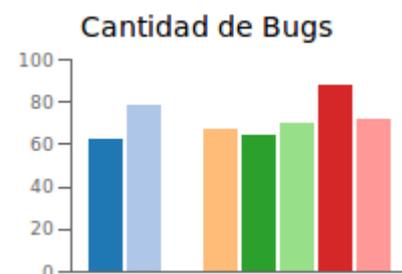


*Ilustración 7: Selección de dimensión y medidas*

3. Presionar el botón "Ejecutar Consulta". También podemos elegir el modo "chart" para representar la salida con un gráfico representativo a nuestra elección.



severidad	componente	año de creacion	Cantidad de Bugs
blocker	Comp0	2010	24



*Ilustración 8: Ejecución consulta modo "Chart"*

Score: 5 (Muy Alto)

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### **Permisos:**

En lo que respecta a los permisos, mondrian ofrece una estructura independiente del dashboard generado con saiku. Dicha estructura contempla la generación de usuarios y mapeo de roles a usuarios. Algo que cabe destacar es que los permisos no se establecen sobre el dashboard propiamente dicho sino que los mismos pueden aplicarse sobre cada medida y dimensión en particular brindando un bajo acoplamiento entre cada elemento. Esto evita la necesidad de generar un dashboard diferente para cada rol, sino que a cada rol se le asigna la medida y/o dimensión que le están permitidas utilizar para que genere su cuadro de mando libremente sin necesidad de que nadie lo haga por él.

**Score: 5 (Muy Alto)**

### **Soporte Origen de datos:**

Básicamente el front-end acepta solamente como fuente de datos bases de datos Olap. Dentro las bases aceptadas encontramos:

- Cubos Mondrian en Mysql
- XML/A (Pentaho BI Server)
- XML/A (Palo)
- Cubos Mondrian en PostgreSQL
- XML/A (Microsoft SSAS)
- DB2

Esto en cierto punto puede ser una complicación debido a la necesidad de adecuar la fuente de datos que normalmente puede ser un archivo o una base relacional a una base de datos del tipo Olap.

**Score: 3 (Medio)**



El **Comunity Dashboard Editor (CDE)** es una herramienta grafica open source que nace para facilitar la creación, edición y renderización de Dashboards.

CDE es una poderosa herramienta que combina un front-end con fuentes de datos y components personalizables para la creación de dashboards. Dicha herramienta se encuentra integrada en el servidor de Pentaho BI.

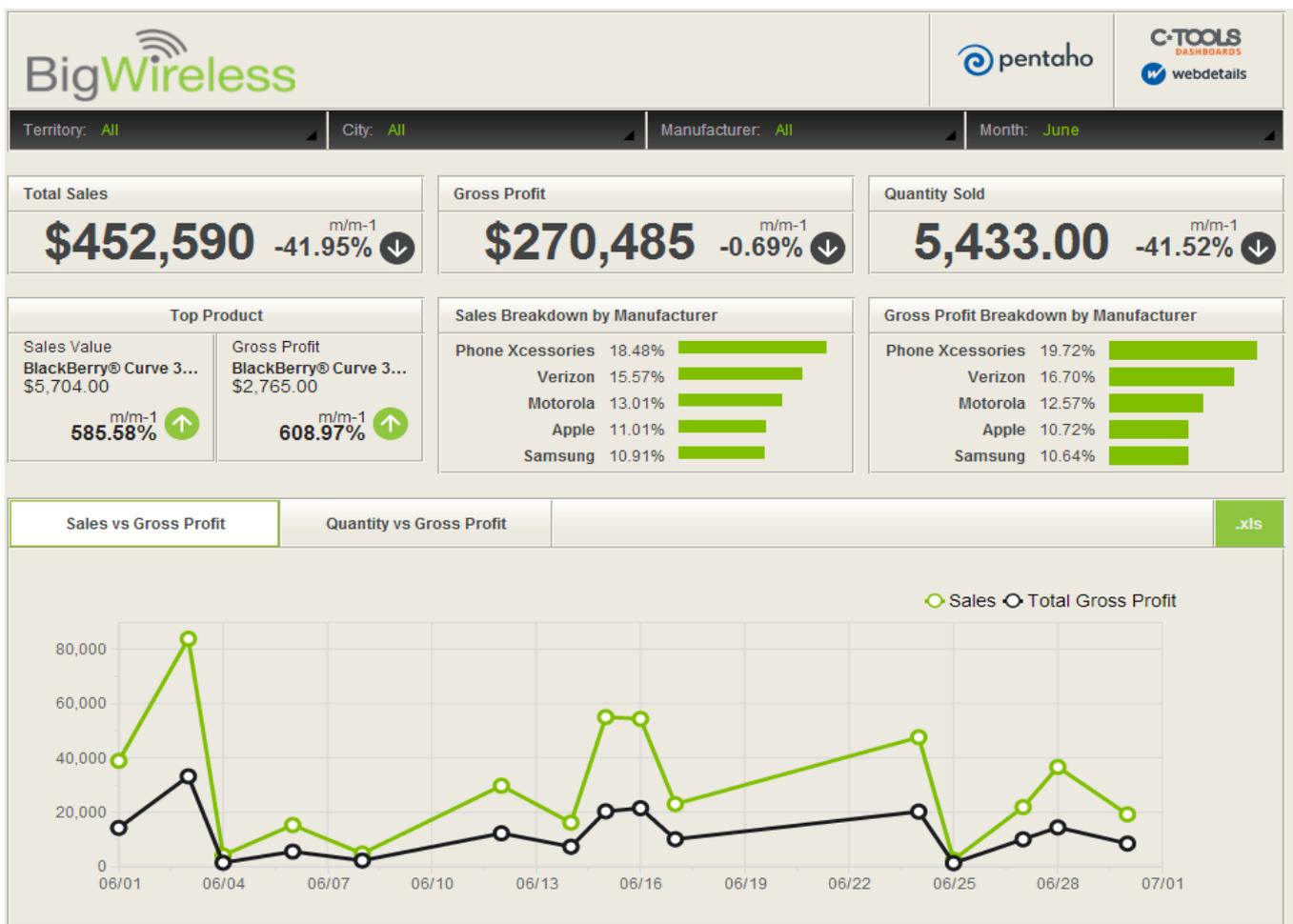


Ilustración 9: Presentación Comunity Dashboard Edition

Como puede apreciarse a simple vista en la imagen es una herramienta sumamente poderosa y requiere de mucho conocimiento para utilizarla a su máximo potencial.

## Trabajo Final De Grado

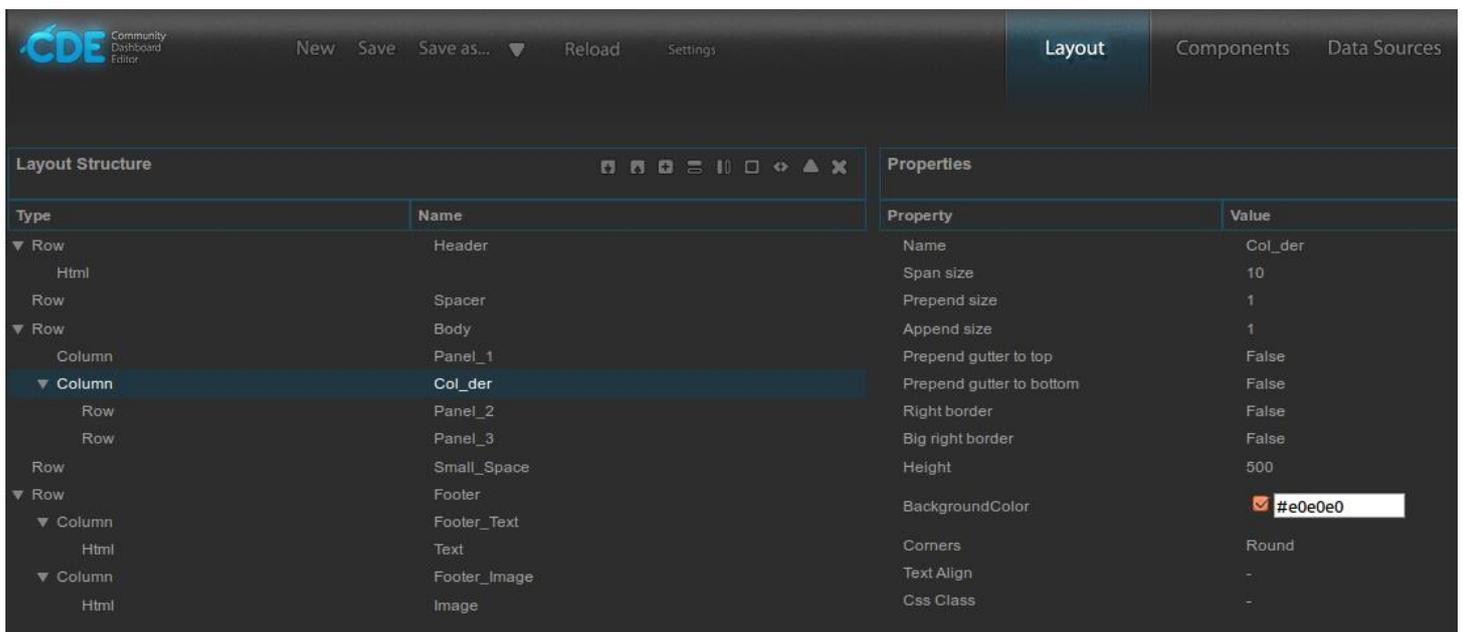
Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Complejidad de implementación:

La implementación de CDE requiere como primer punto la instalación del Servidor Pentaho BI para luego añadirle el módulo de CDE. Este paso no representa mucha dificultad pero se necesitan ciertos conocimientos y llevar a cabo configuraciones no triviales como por ejemplo la configuración de múltiples variables de entorno. Puede tomar mucho tiempo lograr poner en funcionamiento la herramienta de Pentaho con el módulo CDE debido a ciertos problemas de compatibilidad que se pueden encontrar entre las diferentes versiones del servidor BI y el módulo de CDE.

En cuanto a la puesta en marcha de la herramienta, una vez que se ha instalado el servidor de Pentaho y el módulo CDE, se procede a la configuración de alguna fuente de datos. Esto puede realizarse a través de fuentes JNDI o URL de conexión. Para ello debemos buscar las URL de conexión en la página de los desarrolladores.

Luego de configurar las fuentes de datos se debe realizar el diseño del front-end del dashboard. Dicho diseño puede ser simple, seleccionando alguna plantilla, o podemos armar nuestro propio diseño añadiendo columnas y filas al dashboard. Seguido de esto se debe elaborar la consulta sql para la extracción de los datos que queremos representar. Luego debemos programar los componentes que van a aparecer en ese Font-End, como gráficos de barra o de torta para representar los datos.



### ***Ilustración 10: Configuración Front-End CDE***

En resumen, no es una tarea para nada trivial la puesta en marcha desde cero y requiere mucho tiempo, sumado a que la documentación oficial que existe sobre CDE es paga, más allá de ser un software Open Source, y la comunidad/foro oficial de la herramienta no es de grandes dimensiones.

**Score: 1 (Muy Bajo)**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

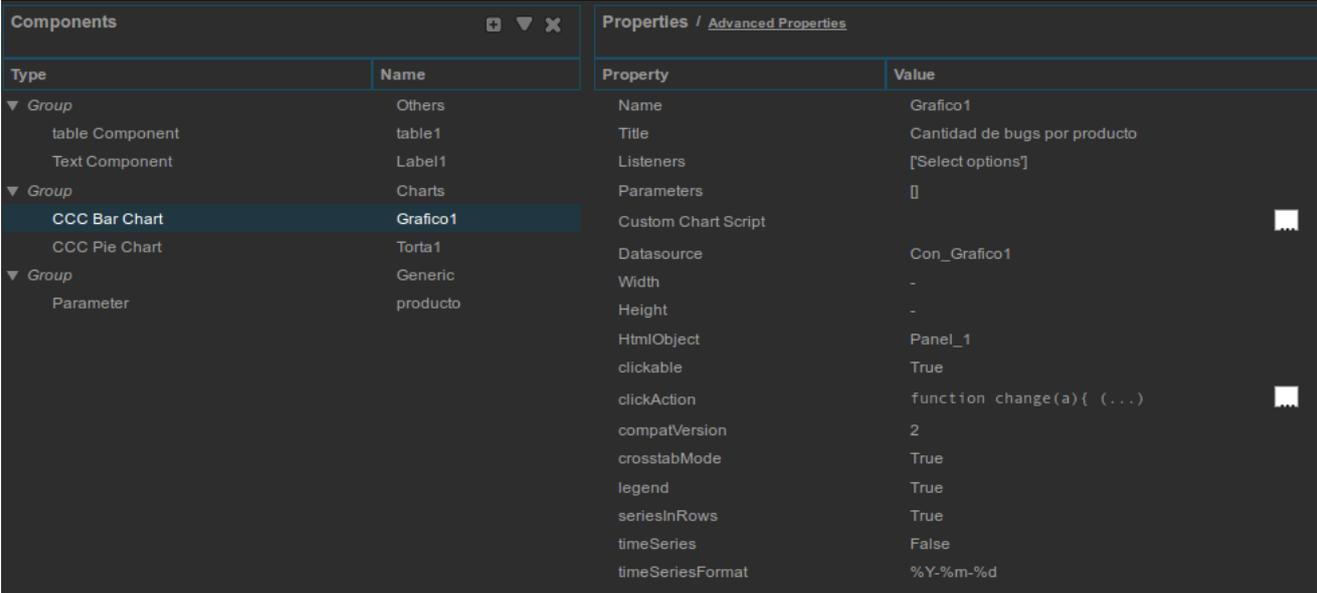
### **Mantenibilidad:**

En cuanto a la mantenibilidad de esta herramienta, podemos decir que es muy poco mantenible. Si tomamos el mismo caso de uso planteado para la herramienta anterior, podemos darnos cuenta de que requiere mucho trabajo introducir cambios en un dashboard.

En el caso en que se desee visualizar una métrica o añadir una medida o dimensión disponible ya en la base, se necesita o bien generar un dashboard totalmente nuevo o “editar” uno ya existente, en cuyo caso se debería reprogramar por completo la interacción de los gráficos (componentes) así también como las consultas a la base que extraen dichos datos.

Es decir, a menos que exista una persona dedicada y capacitada exclusivamente para esta tarea es prácticamente imposible que un usuario sin conocimientos reproduzca dichos cambios en la herramienta. En ambos casos impacta la dificultad de implementación que posee la herramienta.

En caso de que sea necesario un cambio en la estructura de la base de datos, agregando nuevas tablas o modificando su estructura, corre peligro la estabilidad del trabajo ya realizado debido a que pueden resultar erróneas u obsoletas las consultas configuradas en los diferentes dashboards contra la base de datos.



The screenshot displays a configuration interface for a dashboard component. On the left, a 'Components' panel lists various types such as 'table Component', 'Text Component', 'Charts', and 'Parameter'. The 'Charts' group is expanded, and 'CCC Bar Chart' is selected, showing its name as 'Grafico1'. On the right, the 'Properties / Advanced Properties' panel shows a list of properties and their values for the selected component.

Type	Name	Property	Value
Group	Others	Name	Grafico1
table Component	table1	Title	Cantidad de bugs por producto
Text Component	Label1	Listeners	[Select options]
Group	Charts	Parameters	[]
CCC Bar Chart	Grafico1	Custom Chart Script	
CCC Pie Chart	Torta1	Datasource	Con_Grafico1
Group	Generic	Width	-
Parameter	producto	Height	-
		HtmlObject	Panel_1
		clickable	True
		clickAction	function change(a){ (...)
		compatVersion	2
		crosstabMode	True
		legend	True
		seriesInRows	True
		timeSeries	False
		timeSeriesFormat	%Y-%m-%d

**Ilustración 11: Configuración de un componente CDE**

Como se puede apreciar en la imagen cada gráfico, tabla o componente que se presenta en el front-end se alimenta de un “Data Source” en el cual se contiene la consulta representado por dicho gráfico.

**Score: 2 (Bajo)**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

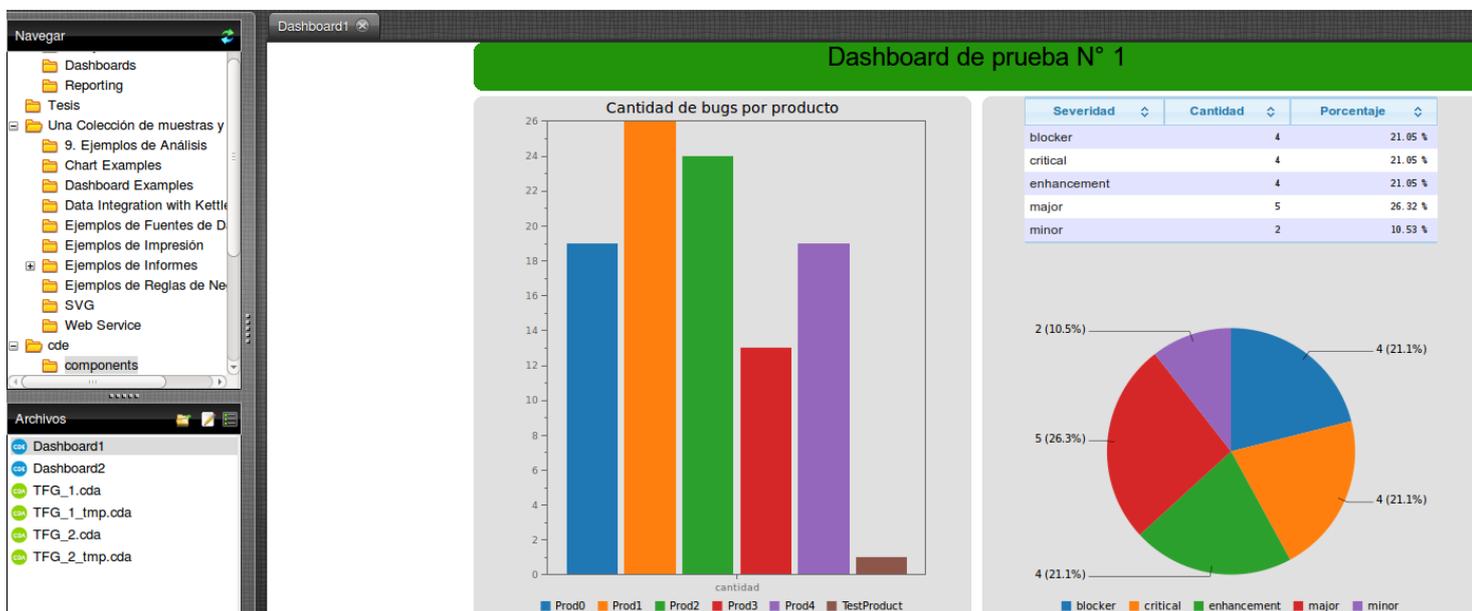
### Usabilidad:

Con respecto a la usabilidad de esta herramienta es muy simple para un usuario ya que solo se limita a la visualización de un dashboard estático el cual puede o no contener algo de interactividad dependiendo de la capacidad del encargado de desarrollar el dashboard.

Para ello el usuario solo debería ingresar a la url en donde se encuentra su dashboard, o loguearse mediante usuario y contraseña a la consola del servidor de pentaho para acceder al dashboard sobre el que tiene permisos.



**Ilustración 12: Logueo consola pentaho**



**Ilustración 13: Visualizacion Dashboard**

**Score: 5 (Muy Alto)**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Flexibilidad:

Dentro de la flexibilidad podemos destacar algo que extiende las capacidades básicas de un dashboard, que es la de generar dashboards con interesantes formas de interactuar con el usuario final. Esto puede contribuir a la generación de dashboards más complejos evitando la necesidad de generar muchos dashboards simples.

Si bien esto es muy costoso en cuanto a capacitación de los recursos humanos, puede dar grandes resultados. Cabe destacar que esto no puede ser realizado por un usuario no capacitado, lo que conlleva a que aquel que construye el dashboard puede no ser quien lo consume, lo cual en determinados entornos de desarrollo puede representar una desventaja, pensando en que quien sea el encargado de construir los dashboards, además de estar capacitado en la construcción de los mismos a través de CDE, debe estar empapado en el dominio del negocio para poder generar métricas representativas para los usuarios.

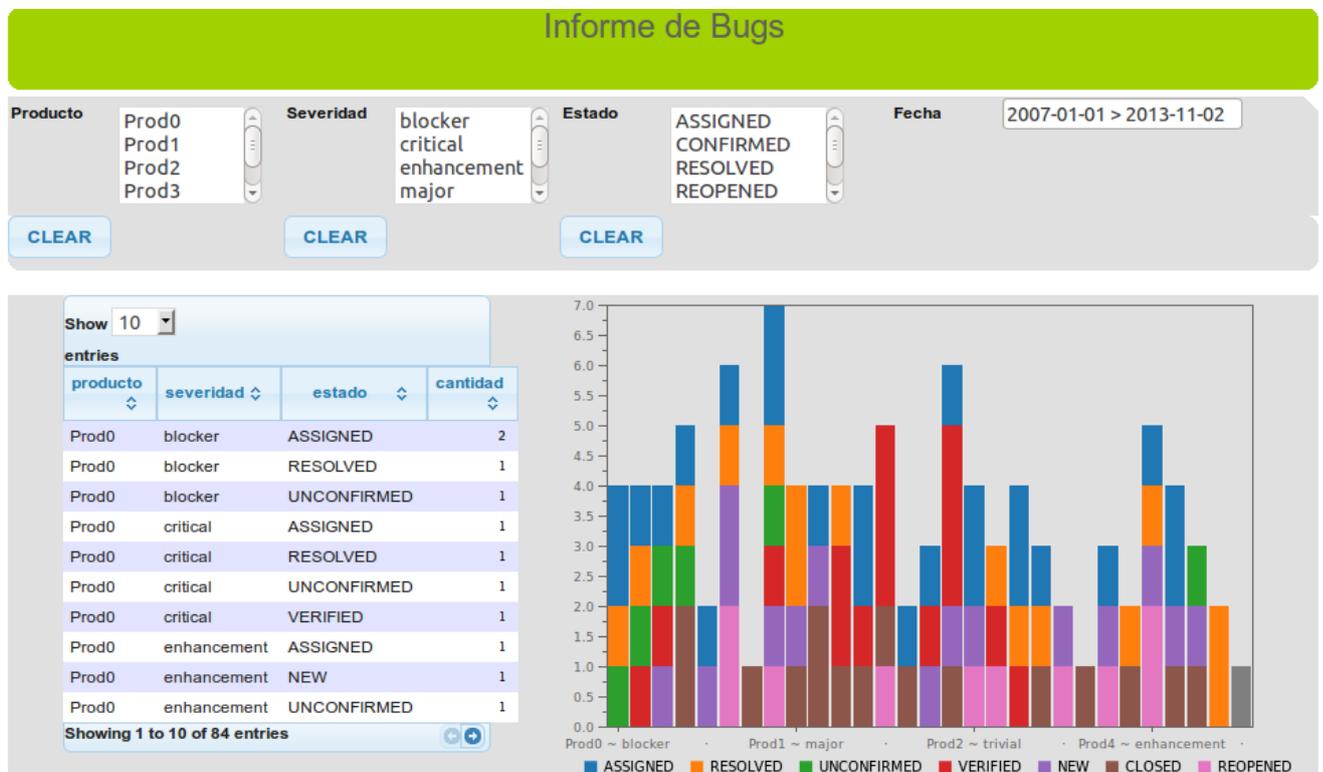


Ilustración 14: Dashboard con interacción para el usuario

Score: 5 (Muy Alto)

## Trabajo Final De Grado

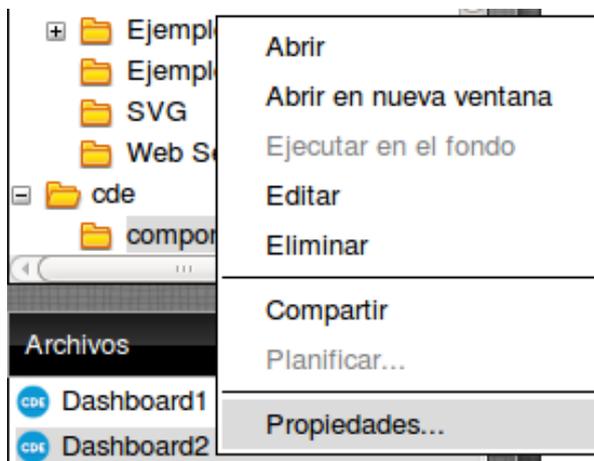
Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Permisos:

La estructura de permisos de esta aplicación se basa en la misma que trae la consola administrativa de Pentaho. Cuenta con la posibilidad de generar usuarios y mapearlos a través de la definición de roles.

La limitación que posee con respecto a los permisos es que básicamente los mismos se establecen y están ligados a un dashboard determinado. Es decir, no se puede establecer permisos sobre medidas o dimensiones, lo cual se traduce en que se deberá generar uno o tantos dashboards como sea necesario para evitar que otros roles no tengan acceso sobre determinada información.

Esto da lugar a la personalización en su grado más elevado para cada dashboard, lo nos lleva a una penalización en el tiempo que toma la elaboración de estos dashboards dado la elevada dificultad de implementación. Además de que está claro que se necesitara la ayuda externa de alguien capacitado para llevar a cabo dicha tarea.



*Ilustración 15: Propiedades del dashboard*



*Ilustración 16: Permisos sobre el dashboard*

**Score: 2 (Bajo)**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### **Soporte Origen de datos:**

El soporte de origen de datos de esta herramienta es bastante extensa. Acepta desde bases de datos hasta archivos de texto plano, xml, etc.

Dentro de las fuentes aceptadas encontramos la siguiente variedad de datasources:

- Bases de datos (Mysql, Postgresql, Access, entre otras)
- Cubos Mondrian
- Metadata de Pentaho
- Archivos XML
- Scripting - puede definir un grupo ad-hoc de origen de datos (por ejemplo, una tabla pequeña), con tan sólo escribir un fragmento de código
- Transformaciones Keetle - Keetle nos permite obtener datos de otras fuentes, como por ejemplo las hojas de Excel o servicios Web.

**Nota:** Los datasources pueden ser parametrizados lo que hace que los datos sean totalmente versátiles.

**Score: 5 (Muy Alto)**

**6. Cálculo de la ponderación para cada alternativa**

<i>Criterio</i>	<b>Ponderación (W<sub>i</sub>)</b>	<b>CDE (R<sub>i1</sub>)</b>	<b>Saiku (R<sub>i2</sub>)</b>
<i>Implementación</i>	3	1	2
<i>Mantenibilidad</i>	5	2	4
<i>Usabilidad</i>	5	5	5
<i>Flexibilidad</i>	4	5	4
<i>Permisos</i>	5	2	5
<i>Orígenes de Datos</i>	3	5	3
<b>Score (S<sub>j</sub>)</b>		<b>83</b>	<b>101</b>

*Ilustración 17: Tabla ponderación multicriterio*

$$S (1) = 3 \times 1 + 5 \times 2 + 5 \times 5 + 4 \times 5 + 5 \times 2 + 3 \times 2 = 83$$

$$S (2) = 3 \times 2 + 5 \times 4 + 5 \times 5 + 4 \times 4 + 5 \times 5 + 3 \times 3 = 101$$

- 7.** La solución “**Mondrian + Saiku**” obtuvo la ponderación más alta de **S (2)=101**, y por lo tanto representa la mejor Alternativa a recomendar.

### BUGZILLA

#### INTRODUCCION



Bugzilla es un sistema de seguimiento de errores OpenSource.

Estos tipos de sistemas se caracterizan por estar diseñados para ayudar a asegurar la calidad de software y asistir a los programadores o grupos involucrados en el desarrollo y uso de sistemas informáticos en el seguimiento de los defectos de software en sus diferentes productos.

La mayor parte de los sistemas de seguimiento de errores identifican un “ciclo de vida” al cual se le da seguimiento mediante el estado del problema desde su descubrimiento y reporte hasta su solución final.

De la misma manera, los mismos pueden ser configurados una vez creados permitiendo que distintas personas consulten o modifiquen diferentes aspectos del reporte.

Todas estas características mencionadas pueden ser encontradas en Bugzilla.

Debido que el uso de la herramienta escapa el alcance de nuestra tesis, a continuación explicaremos sin entrar en detalles como funciona este software, para luego llevar adelante el análisis de su base de datos.

#### USO DE LA HERRAMIENTA

##### ASPECTOS GENERALES

Bugzilla se centra en dos elementos básicos:

- Tareas o Bugs (incidencia, algo por resolver)
- Usuarios (persona registrada en el sistema)

Cabe destacar que un bug de Bugzilla siempre tendrá asignado un usuario, el sistema no permite otra posibilidad. Existe claro la posibilidad de reasignar dicho bug a otro usuario del sistema en caso de que sea necesario.

Dado que la administración de usuarios escapa el alcance de esta tesis no se profundizara en ello.

Cada bug cargado en el sistema siempre formara parte de un producto, nunca podrá haber una tarea que no sea relativa a un producto.

Si la complejidad del proyecto fuera mayor, este puede ser descompuesto en partes más pequeñas y organizables llamadas componentes.

Así entonces un bug siempre pertenecerá a un componente de un producto Bugzilla, y al igual que con los usuarios, se podrá reasignar a otro si fuera necesario.

Las tareas o bugs poseen muchos más atributos que serán analizados al profundizar en la base de datos y el proceso ETL implementado con Kettle.

CICLO DE VIDA DE UN BUG

Para comprender mejor las funcionalidades de este software se ha traducido al español desde la documentación oficial de Bugzilla un diagrama de flujo muy interesante que muestra los posibles estados de un bug o tarea ingresados al sistema:

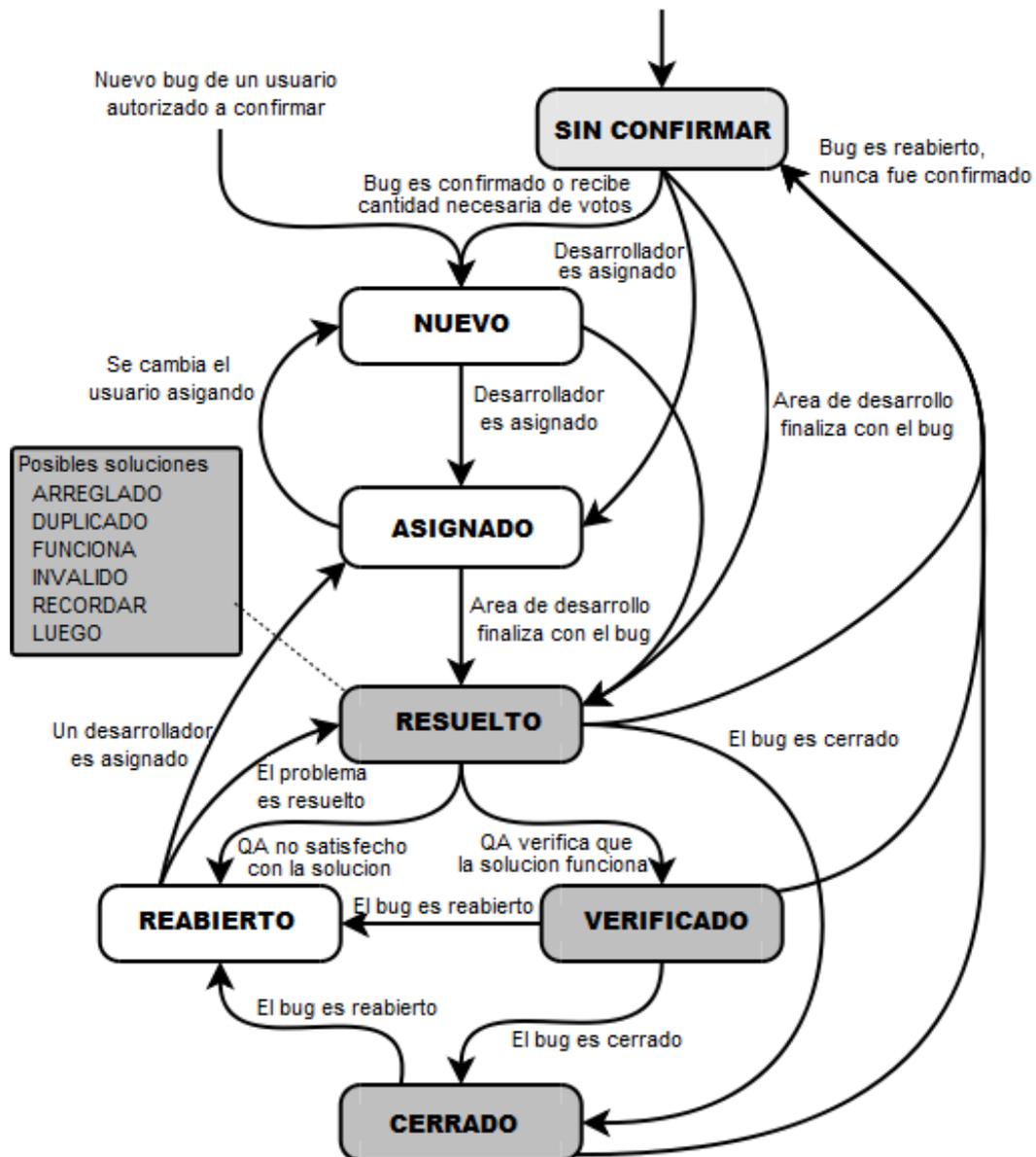


Ilustración 18: Estado de un bug

En la gráfica se puede apreciar los diferentes estados que puede adquirir un bug (en color gris) y las transiciones posibles entre estos como resultado de diferentes acciones realizadas.

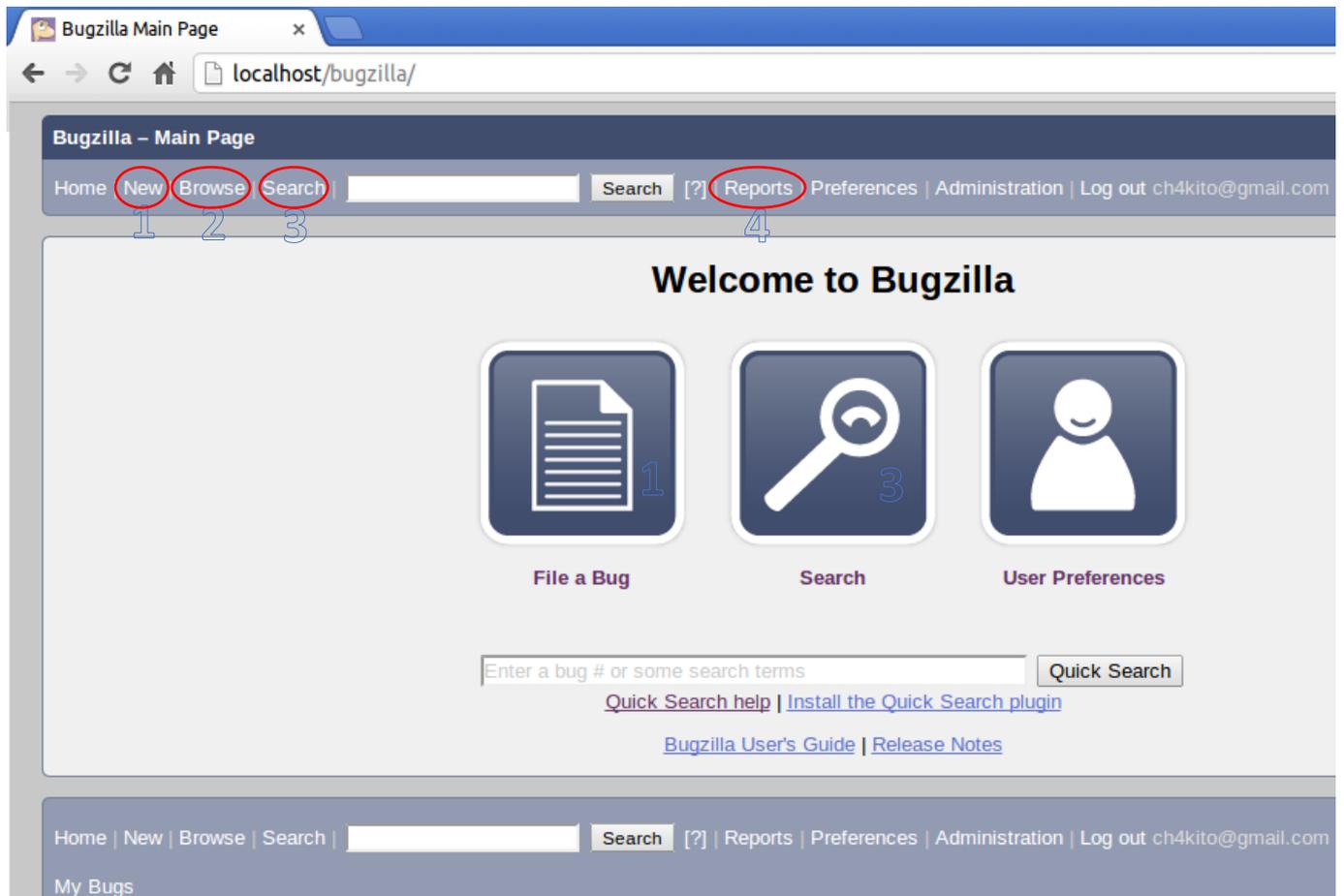
## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### HOME – PANTALLA PRINCIPAL

Bugzilla es una aplicación que corre en un servidor web, en este caso Apache y se accede a ella ingresando en cualquier navegador la siguiente URL: <http://localhost/bugzilla/>.

Asumiendo que Bugzilla fue correctamente instalado y un usuario ha sido creado, se procede a analizar la pantalla principal del software:



**Ilustración 19: Pantalla de inicio Bugzilla**

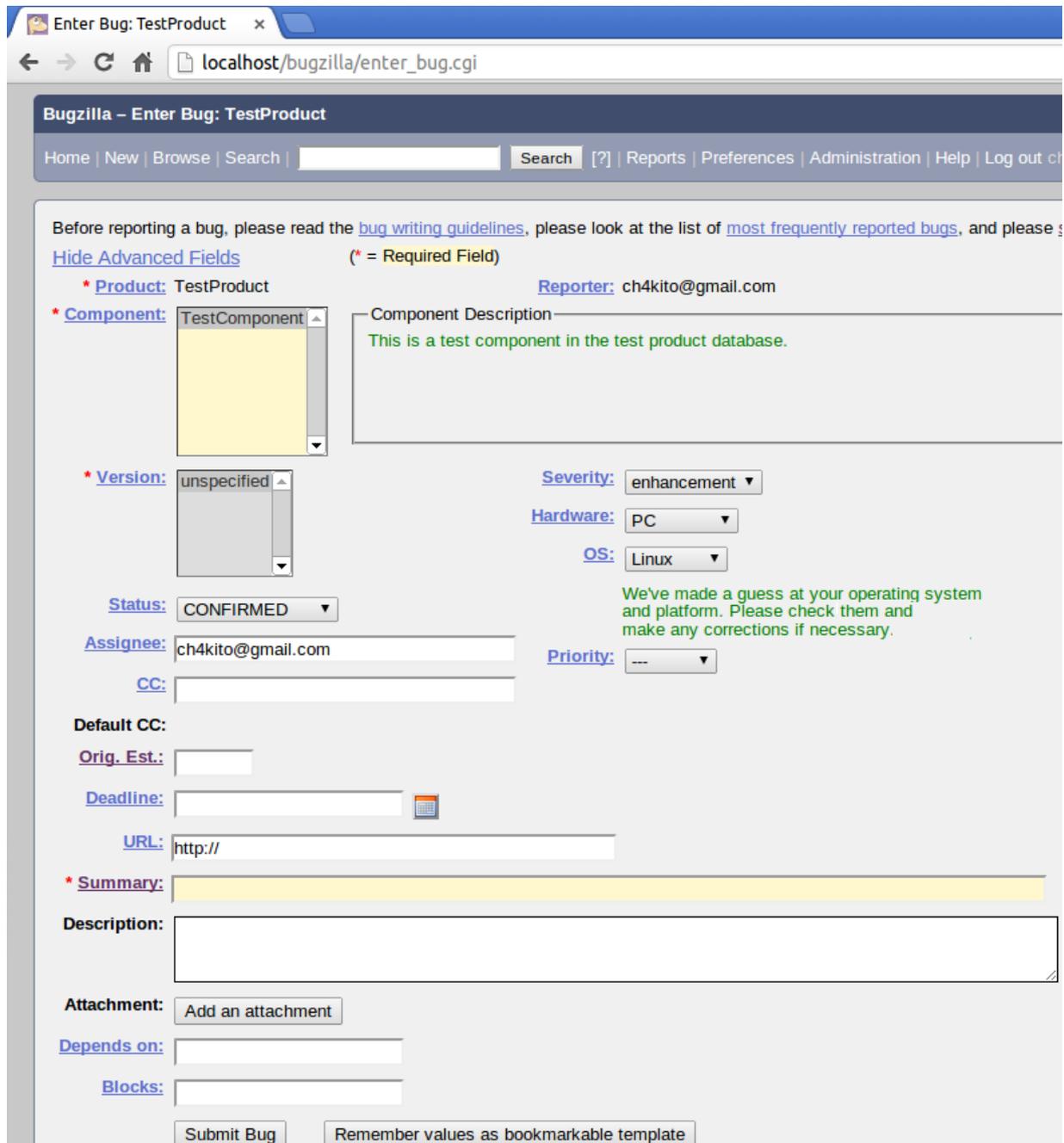
En ella se han enumerado aquellas funciones, las cuales serán explicadas brevemente ya que las son consideradas las más importantes para proveer un mejor entendimiento general del proyecto desarrollado. No se profundizara en configuraciones y administración del sistema debido a que esto escapa a la finalidad del trabajo de grado.

### FUNCIONES PRINCIPALES

#### 1. NEW – CARGA DE UN BUG

Al seleccionar esta opción se presenta la siguiente pantalla.

A continuación se pueden ver los campos que posee un bug, los cuales son requeridos para su carga:



The screenshot shows the Bugzilla 'Enter Bug' form for 'TestProduct'. The browser address bar shows 'localhost/bugzilla/enter\_bug.cgi'. The form includes a navigation bar with links like 'Home', 'New', 'Browse', 'Search', 'Reports', 'Preferences', 'Administration', 'Help', and 'Log out'. Below the navigation bar, there is a message: 'Before reporting a bug, please read the [bug writing guidelines](#), please look at the list of [most frequently reported bugs](#), and please [Hide Advanced Fields](#)'. A note indicates that fields with an asterisk (\*) are required. The form fields are: Product (TestProduct), Reporter (ch4kito@gmail.com), Component (TestComponent), Version (unspecified), Severity (enhancement), Hardware (PC), OS (Linux), Status (CONFIRMED), Assignee (ch4kito@gmail.com), Priority (---), CC (empty), Default CC (empty), Orig. Est. (empty), Deadline (empty), URL (http://), Summary (empty), Description (empty), Attachment (Add an attachment), Depends on (empty), and Blocks (empty). At the bottom, there are buttons for 'Submit Bug' and 'Remember values as bookmarkable template'.

Ilustración 20: Carga de un bug

Como se puede apreciar, la interfaz permite seleccionar algunos parámetros desde listas (Producto, Componente, Versión, Estado, etc...) e ingresar otros manualmente (URL, CC, Descripción, etc...).

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

En la siguiente sección será analizada la estructura de la base de datos y se ampliara sobre estos campos.

Una vez completados todos los campos debe hacerse click en el boton "Submit" para cargar el bug en la base.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### 2. BROWSE – EXPLORAR BUGS

Al seleccionar esta opción se presentara una pantalla con un listado de productos y componentes como la siguiente:

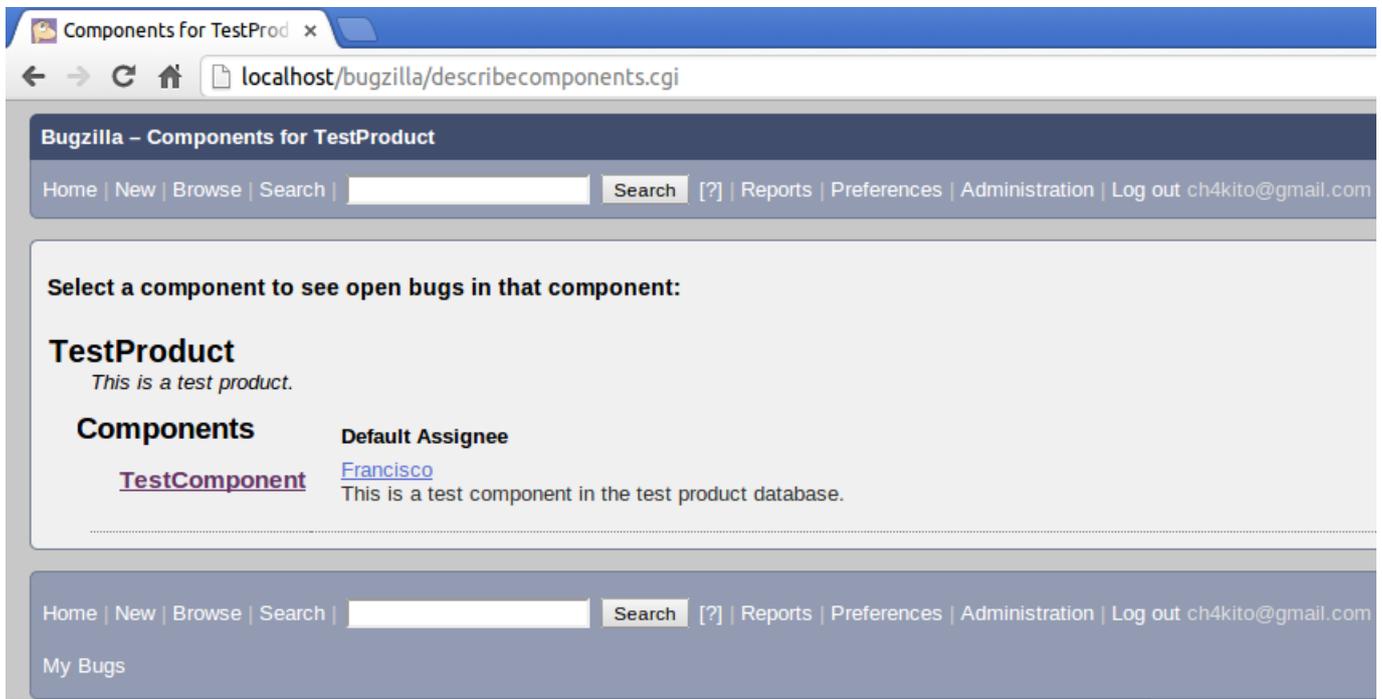


Ilustración 21: Explorador de bugs

Una vez seleccionado un producto y componente se pueden visualizar todos los bugs que corresponden a este:

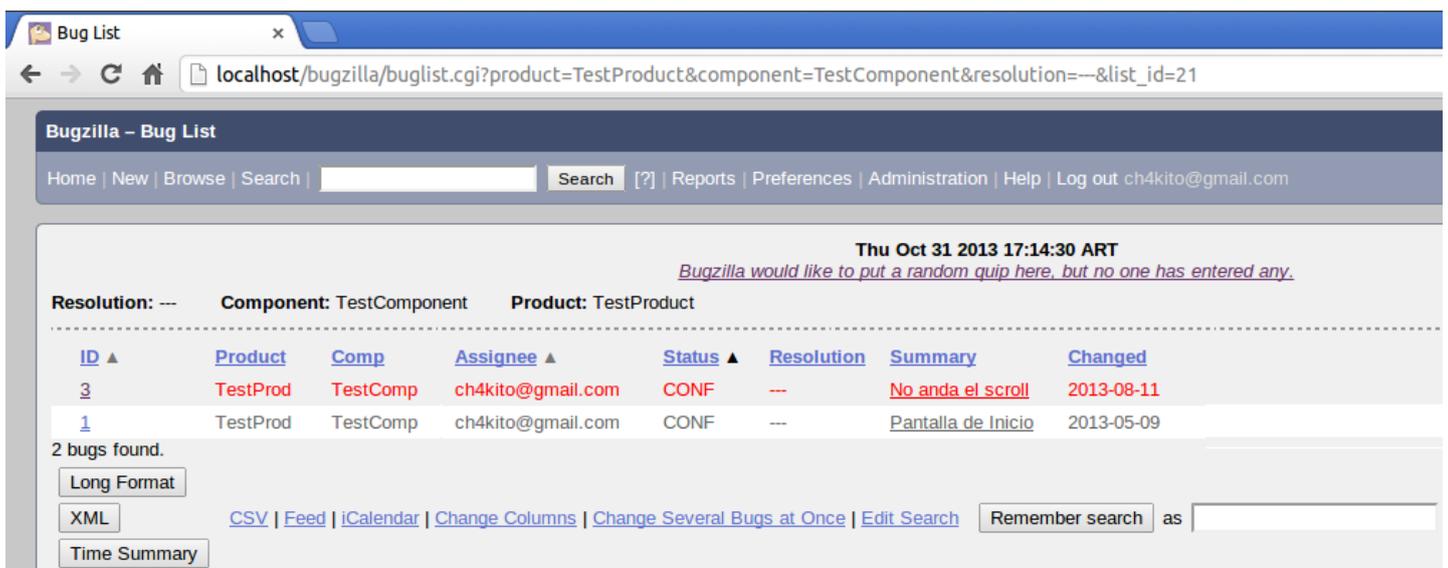


Ilustración 22: Visualización de bugs en el explorador

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Al seleccionar algún bug se ingresa a la pantalla de modificación la cual será analizada más adelante.

Cabe aclarar que esta forma de llegar hasta un bug para su modificación puede ser válida en grupos de desarrollo pequeños con un número reducido de productos y componentes como el ejemplo brindado, pero la búsqueda manual se volvería muy engorrosa en grupos de desarrollo de gran volumen con múltiples proyectos, es por esto que existe la herramienta de búsqueda que se presenta a continuación.

### 3. SEARCH – BUSQUEDA DE BUGS

Al seleccionar esta opción se presentan dos posibilidades:

- Búsqueda Simple
- Búsqueda Avanzada

#### BUSQUEDA SIMPLE



Ilustración 23: Bugzilla, búsqueda simple

La búsqueda simple cuenta con solo 3 filtros: Estado, Producto y Palabras Clave.

Esta opción básica puede ser útil a la hora de querer ver por ejemplo “todos los bugs cerrados de tal producto”, pero si este tuviera múltiples componentes que también deseen ser filtrados se debe pasar a una búsqueda avanzada.

#### BUSQUEDA AVANZADA

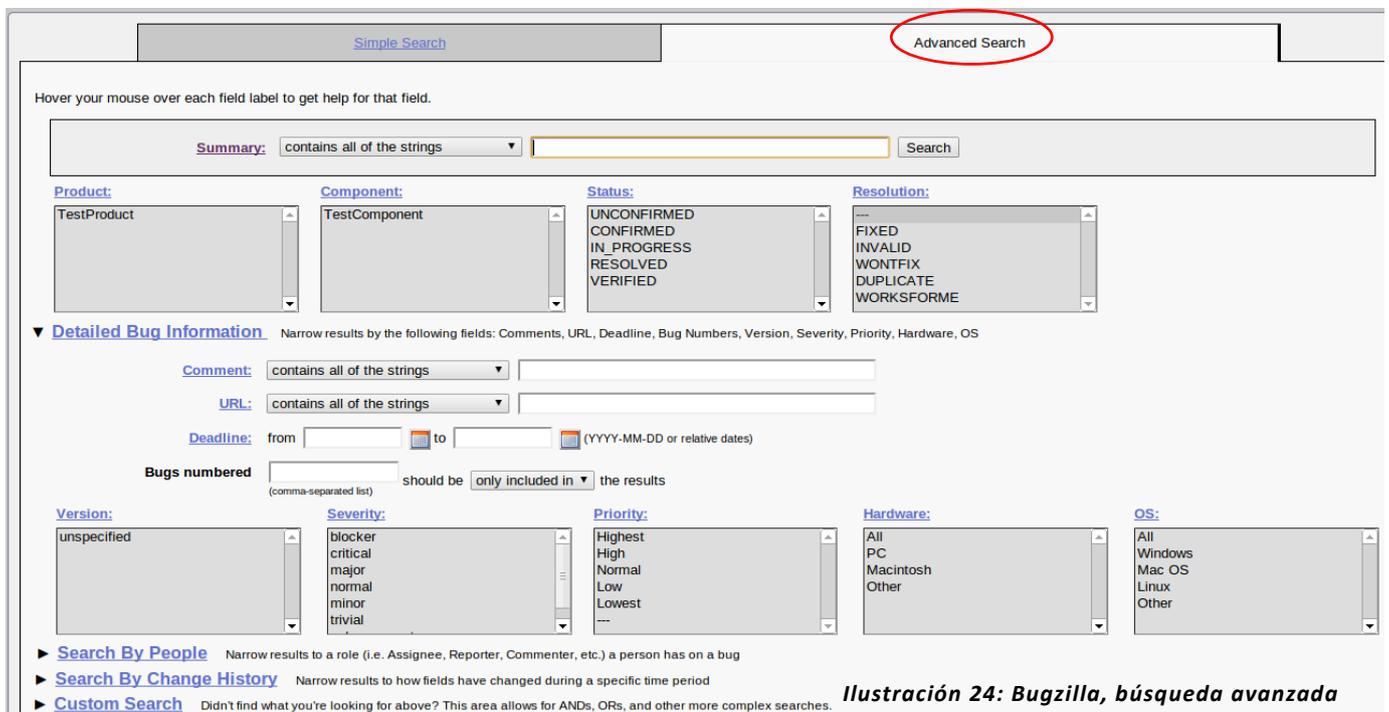


Ilustración 24: Bugzilla, búsqueda avanzada

Como se aprecia en la imagen la búsqueda avanzada posee filtros para cada campo existente, sin embargo en cada lista se presentan todas las posibilidades, no solo aquellas existentes, por lo que es muy posible terminar realizando búsquedas sin resultados lo cual es un punto que se desea mejorar con la implementación de este trabajo de grado.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### 4. MODIFICACION DE UN BUG

Al seleccionar un bug, ya sea desde la pantalla de “Explorar” o de alguno de los tipos de búsquedas, se presentara una pantalla similar a la siguiente:

**Bugzilla – Bug 1** Pantalla de Inicio

Home | New | Browse | Search |  Search [?] | Reports | Preferences | Administration | Help | Log out ch4

**Bug List:** (2 of 2) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#)

**Bug 1 - Pantalla de Inicio (edit)**

**Status:** CONFIRMED ([edit](#)) **Reported:** 2013-05-09 15:22 ART by [Francisco](#)

**Product:** TestProduct **Modified:** 2013-05-09 15:22 ART ([History](#))

**Component:** TestComponent **CC List:**  Add me to CC list  
0 users ([edit](#))

**Version:** unspecified **See Also:** ([add](#))

**Platform:** PC  Linux

**Importance:** ---  enhancement

**Assigned To:** [Francisco](#) ([edit](#))

**URL:**

**Depends on:**

**Blocks:**

[Show dependency tree / graph](#)

Orig. Est.	Current Est.	Hours Worked	Hours Left	%Complete	Gain	Deadline
0.0	0.0	0.0 + <input type="text"/>	0.0	0	0.0	<input type="text"/>

[Summarize time \(including time for bugs blocking this bug\)](#)

**Attachments**

[Add an attachment](#) (proposed patch, testcase, etc.)

**Additional Comments:**

**Status:** CONFIRMED

[Mark as Duplicate](#)

[Francisco](#) 2013-05-09 15:22:54 ART [Description](#) [reply](#) [-] [Collapse All Comments](#)  
[Expand All Comments](#)

La pantalla de inicio pose un boton muy chico para aceptar.

[Add Comment](#)

**Bug List:** (2 of 2) [First](#) [Last](#) [Prev](#) [Next](#) [Show last search results](#) [Format For Printing](#) - [XML](#) - [Clone This Bug](#) - [Top of page](#)

Ilustración 25: Modificación de un bug

Como puede apreciarse es posible modificar prácticamente todos los campos asociados al bug en cuestión, desde el nombre del bug, el desarrollador asignado a solucionarlo, fecha límite, horas trabajadas, descripción, etcétera.

Además de modificar sus campos se puede encontrar en esta pantalla la opción “Clone this bug” o clonar este bug, la cual lleva a la creación de un nuevo bug, con todos los campos ya rellenos con los datos del actual facilitando así el trabajo a la hora de agregar tareas de características similares.

También se puede imprimir el resultado actual, y navegar a través de los resultados con las opciones al final de la página.

5. REPORTES

Bugzilla permite a sus usuarios generar reportes estáticos de distintos tipos:

### Current State

- [Search](#) - list sets of bugs.
- [Tabular reports](#) - tables of bug counts in 1, 2 or 3 dimensions, as HTML or CSV.
- [Graphical reports](#) - line graphs, bar and pie charts.
- [Duplicates](#) - list of most frequently reported bugs.

### Change Over Time

- [Old Charts](#) - plot the status and/or resolution of bugs against time, for each product in your database.
- [New Charts](#) - plot any arbitrary search against time. Far more powerful.

Ilustración 26: Tipos de reportes en bugzilla

Ya sea que se quiera generar un reporte tabular o reportes gráficos como de línea, barra o torta, se nos presentara una pantalla donde primero deben seleccionarse los campos que van a ser utilizados como ejes:

Vertical Axis: OS

Plot Data Sets:  
 Individually  
 Stacked

Multiple Images: Component

Format:  
 Line Graph  
 Bar Chart  
 Pie Chart

Horizontal Axis: <none>

Vertical labels:

Ilustración 27: Selección de ejes en reportes

Por ejemplo si para el eje X elegimos “Componente” y para el eje Y “Sistema Operativo” con los dos bugs que hemos cargado a modo de prueba obtenemos los siguientes gráficos:

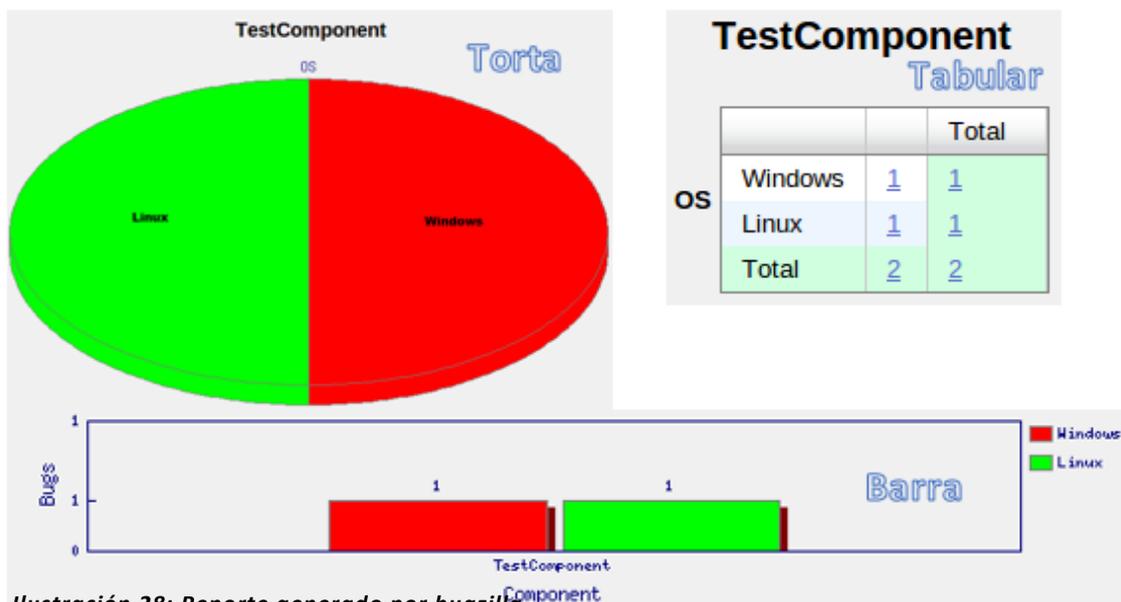


Ilustración 28: Reporte generado por bugzilla

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

También es posible realizar un filtro más avanzado con las mismas opciones que la búsqueda avanzada, la cual es bastante compleja y carece de la posibilidad de realizar consultas en tiempo real sin actualizar los campos haciendo el proceso bastante engorroso, generando muchas veces gráficos en blanco.

### IMPLEMENTACION

En esta sección se llevara a cabo el desarrollo de la implementación del trabajo final de grado. Todo aquello referente a instalación y configuración de a las diferentes herramientas utilizadas será tratado en el apartado “Anexo” de este documento.

### ANALISIS DE LA FUENTE DE DATOS

Como primer paso, luego de comprender la estructura de Bugzilla, se seleccionaron aquellos campos que permitirán posteriormente al usuario la conformación de indicadores del negocio. Esto se traduce como la selección de las dimensiones y medidas que se ofrecerán a los usuarios.

A continuación se detallan las medidas y dimensiones seleccionadas, a partir de los datos que nos brinda la herramienta, acompañadas de una breve justificación de su importancia para con la solución.

---

#### MEDIDAS BUGZILLA

- **Cantidad de Bugs**
  - Esta medida permitirá cuantificar y sumarizar la cantidad de bugs existentes según los diferentes criterios de agrupación que se apliquen.
  
- **Tiempo estimado de resolución**
  - Esta medida permitirá cuantificar y sumarizar la cantidad de horas estimadas de resolución de bugs agrupados por el criterio deseado.
  - Obtenido de
    - **Tabla:** Bugs
    - **Campo:** Estimated\_time

---

#### DIMENSIONES

- **Fecha de creación y fecha de cierre**
  - Se utilizara dicha dimensión, para visualizar la evolución de los bugs en un proyecto. Permitirá la granularidad en unidades de día, mes y año.  
  
Ej. Cantidad de Bugs generados en el mes de enero del 2013  
Ej. Cantidad de Bugs críticos cerrados en el 2012
  - Obtenido de
    - **Tabla:** Bugs
    - **Campos:**Creation\_ts , Deadline

- **Estado**

- Se propone dicha dimensión, debido a que para llevar un mejor control de la calidad de los productos, es necesario disponer del estado relacionado a un bug.

**Estados posibles:**

- Sin Confirmar (Unconfirmed)
- Confirmados (Confirmed)
- En Progreso (In Progress)
- Resueltos (Resolved)
  - Arreglados (Fixed)
  - Invalidos (Invalid)
  - No serán Arreglados (WontFix)
  - Duplicados (Duplicate)
  - Irreproducible (Works for me)
- Verificados (Verified)

Ej. Cantidad de Bugs que no serán Arreglados.

Ej. Tiempo estimado de resolución estimado de bugs confirmados en el periodo 2012

- Obtenido de
  - **Tabla:** Bugs
  - **Campo:**Status

- **Severidad**

- Se propone dicha dimensión, ya que hace a la calidad del software conocer el grado de incidencia de los bugs en los diferentes productos.

**Grados de severidad**

- Bloqueador (Blocker)
- Critico (Critical)
- Mayor (Major)
- Normal
- Menor (Minor)
- Trivial
- Mejora (Enhancement)

Ej. Cantidad de bugs críticos reportados.

Ej. Cantidad de bugs bloqueadores en progreso de resolución.

- Obtenido de
  - **Tabla:** Bugs
  - **Campo:**Bug\_Severity

- **Prioridad**

- Se propone dicha dimensión, debido a que es de gran importancia poder conocer la prioridad de resolución relacionado a un bug. Esto hace a la calidad permitiendo tener un mayor conocimiento sobre el nivel de atención que requieren las incidencias de un producto software.

- **Grados de prioridad**

- Muy Alta (Highest)
    - Alta (High)
    - Normal
    - Baja (Low)
    - Muy Baja (Lowest)

- **Ej.** Cantidad de bugs de prioridad muy alta de resolución cuya severidad es normal.

- **Ej.** Tiempo estimado de resolución de bugs de prioridad baja en progreso de resolución.

- Obtenido de
  - **Tabla:** Bugs
  - **Campo:** Priority

- **Producto**

- Esta dimensión permite realizar un seguimiento a bugs de un producto específico.

- **Ej.** Cantidad de bugs que posee producto “Simulador Aereo” cuya prioridad sea normal.

- Obtenido de
  - **Tabla:** Product
  - **Campo:** Name

- **Componente**

- Esta dimensión permite realizar un seguimiento a bugs de un componente específico.

- **Ej.** Cantidad de bugs que posee el componente “Modulo de creación” del producto “Simulador Aereo” cuyo estado sea verificado.

- Obtenido de
  - **Tabla:** Component
  - **Campo:** Name

- **Sistema Operativo**

- Dicha dimensión permite clasificar los bugs por la plataforma en la cual se presentan, posibilitando de esta manera el control de las incidencias sobre las diferentes plataformas.

- **Tipos de S.O.**

- Windows
    - Linux
    - Otro

- **Ej.** Cantidad de bugs reportados en las plataformas Linux.

- Obtenido de
      - **Tabla:** Bugs
      - **Campo:**Op\_sys

- **Versión**

- Dicha dimensión posibilita el control de incidencias sobre las diferentes versiones de un componente. Dicha dimensión aporta mucha visibilidad para diferenciar el estado de las incidencias en un componente/producto según su estadio de desarrollo.

- **Ej.** Cantidad de bugs de severidad Mayor en la versión 1.3 del producto “Simulador de vuelos”.

- Obtenido de
      - **Tabla:** Bugs
      - **Campo:**Version

Con el planteo de dichas medidas y dimensiones, podrán responderse a muchas preguntas que hacen referencia al control de la calidad del producto software.

## DISEÑO DE LA BASE

Luego de seleccionar los campos que se extraerán de la base de datos bugzilla se procede al diseño de una estructura de datos para el almacenamiento de los mismos y posterior transformación. Para ello se optó por almacenar dichos registros en una base de datos relacional utilizando el motor MYSQL.

La justificación del diseño e implementación de esta base de datos se centra en dos motivos

- 1. Evitar la realización de consultas complejas directamente a la base de bugzilla**  
Esto comprometería la performance de la base en un entorno de producción, provocando que la aplicación bugzilla presente inconvenientes como, retrasos al cargar un bug o traer información desde la base a la aplicación. Todo esto se verá reflejado en una disminución de la calidad del servicio ofrecida a los usuarios que hacen uso de bugzilla.
- 2. Necesidad de realizar transformaciones antes de mostrar los datos**  
Debido a que en la base de bugzilla los datos presentan cierta heterogeneidad o inconsistencias, ya que no han sido controladas por la aplicación, es necesario homogeneizar dichos registros para que estos no se traduzcan en lecturas erróneas posteriormente.  
Esto genera la necesidad de creación de una estructura de datos donde los mismos presenten consistencia y coherencia. Además, esto permite que, en un futuro, otras aplicaciones puedan nutrirse de una fuente de datos confiable.

A continuación se presenta la estructura de la base para el almacenamiento de los campos seleccionados.

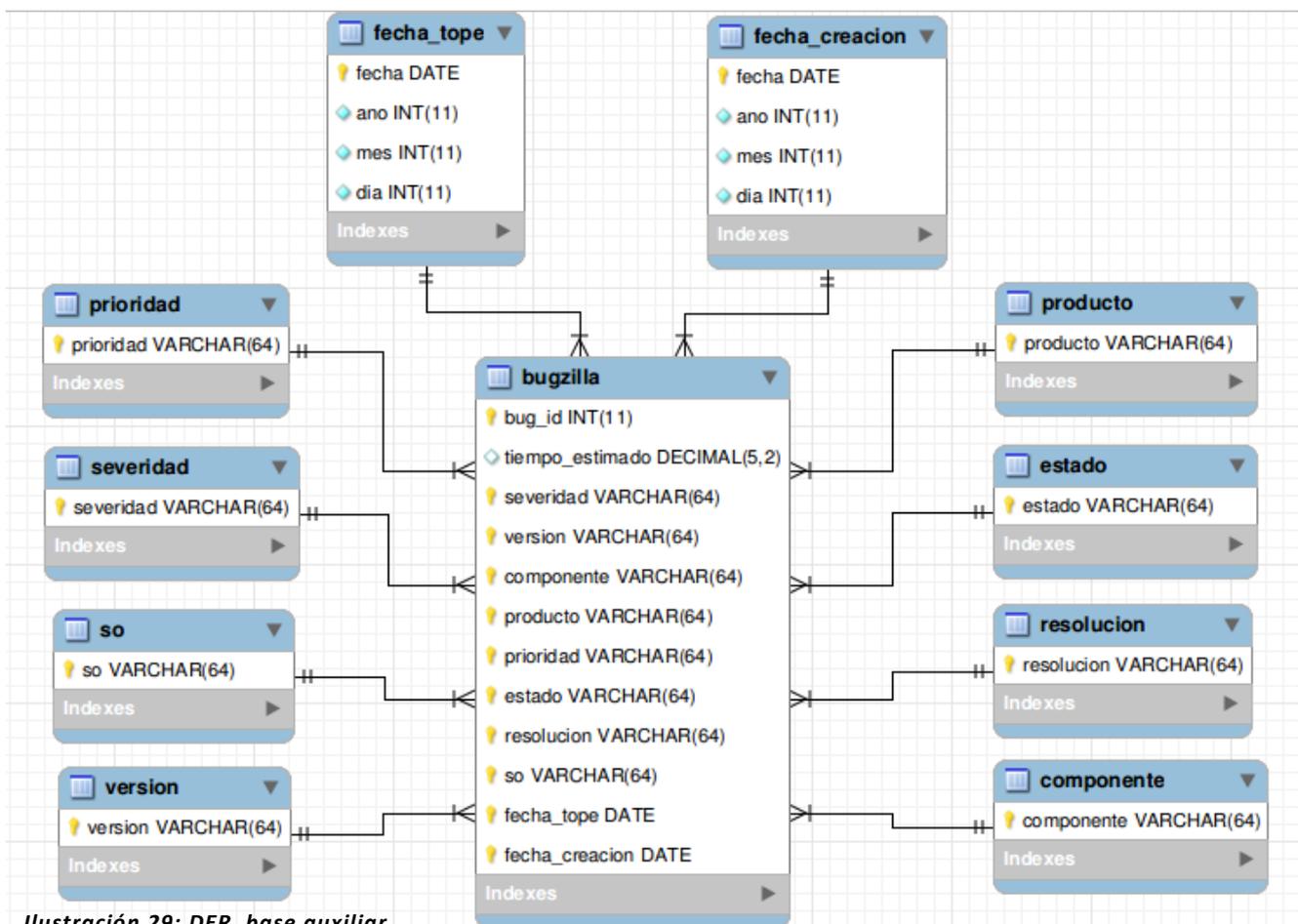


Ilustración 29: DER, base auxiliar

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Este diseño responde simplemente a la agrupación de los datos extraídos de la herramienta de seguimiento de errores en una tabla llamada “Bugzilla”. Los campos de la tabla central “Bugzilla” se corresponden con los valores extraídos de la herramienta.

La estructura que presenta es necesaria para ser aceptada por la herramienta olap que se utiliza luego para la visualización de la información.

### INTEGRACIÓN CON PENTAHO

En este apartado se desarrolla la integración de pentaho con el resto de las herramientas y el rol que cumple dentro de esta arquitectura.

#### PENTAHO



Pentaho es una compañía que ofrece soluciones Open Source en productos de Inteligencia de Negocios. Incluye los principales componentes requeridos para implementar soluciones basadas en procesos y ha sido concebido desde el principio para estar basada en procesos. Este producto ofrece diversas herramientas como:

- Reporting
- Análisis
- Dashboards
- Integración de datos
- Data Mining
- Plataforma Business Intelligence

El modelo de negocios open source de Pentaho, elimina los costos de licencia de software, provee soporte, servicios, y una optimización del producto vía una suscripción anual. En los años desde el inicio de Pentaho como pionero en las herramientas BI open source, sus productos han sido descargados más de tres millones de veces con implementación a producción en compañías en el rango desde pequeñas a las globales.

Gracias a su diseño modular y arquitectura basada en plugins, permite embeber aplicaciones de terceros en la plataforma. Esta última característica también resulta útil permitiendo la integración de Saiku, en la plataforma de pentaho.

De esta manera todo se encuentra centralizado en una sola aplicación brindando más comodidad al usuario.

### INTRODUCCION

La herramienta pentaho proveerá , entre otras cosas, de un front-end que permitirá, gráficamente y de manera muy sencilla, el manejo de usuarios y permisos.

Además sobre dicha plataforma se integran:

- El modulo “Keetle” encargado de ejecutar los trabajos y transformaciones necesarias sobre la fuente donde se extraerán los datos (Bugzilla).
- La consola de administración de usuarios y roles
- Servidor Mondrian para el mapeo de consultas Olap contra una fuente relacional.
- El software Saiku para la generación de consultas al estilo de cuadros de mando.

Para llevar a cabo la concreción de dichas tareas, como primera instancia es necesaria la instalación de la plataforma de Business Intelligence. Esta plataforma provee la arquitectura e infraestructura requerida para construir soluciones a problemas de inteligencia de negocios. Como se señaló anteriormente el framework proveerá de servicios como autenticación a través de credenciales de usuarios, una funcionalidad de la cual se hará uso para elaborar la jerarquía de permisos que requiere la solución.

A continuación se abordara la herramienta keetle, explicando detalladamente el trabajo realizado y el rol que cumple el mismo dentro de esta arquitectura.

### KETTLE

Kettle es una herramienta que permite realizar operaciones ETL (Extract, transform and load - extracción, transformación y carga), facilitando el manejo de los datos y la creación de una estructura de tablas tipo OLAP.

Con Kettle se pueden guardar los trabajos y transformaciones que se diseñan de dos maneras:

- **Repositorio**  
disponemos de una base de datos, con una estructura especial, donde son guardadas las transformaciones y trabajos construidos. Puede ser útil para el trabajo en equipo y para disponer de un lugar centralizado donde se va registrando todo lo realizado.
- **Ficheros**  
Las transformaciones y trabajos son guardados a nivel del sistema de ficheros, en archivos xml (con extensión .ktr para las transformaciones y .kjb para los jobs). Cada transformación y trabajo tiene un fichero asociado, que incluye en formato Xml el metadata que define su comportamiento.

### TRABAJO Y TRANSFORMACIONES

Antes de continuar se considera necesario detenerse a analizar los siguientes conceptos para un mejor entendimiento de la sección:

- **Trabajo o Job.**  
Su concepto es similar al de un proceso, es decir un conjunto sencillo o complejo de tareas (transformaciones) con el objetivo de realizar una acción determinada. Dichas tareas están asociadas mediante pasos o saltos.
- **Transformación**  
Es el elemento básico de diseño de los procesos ETL en PDI.

Una transformación se compone de pasos o steps, que están enlazados entre sí a través de los saltos o hops, los cuales constituyen el elemento a través del cual fluye la información entre los diferentes pasos (siempre es la salida de un paso y la entrada de otro).

En este trabajo de grado, la función que cumple Kettle es la de tomar los datos provenientes de Bugzilla, más precisamente de su base de datos relacional y transformarlos.

Este paso se justifica partiendo de la premisa de que los datos almacenados en Bugzilla no son en su totalidad homogéneos ni consistentes.

Con “no consistentes” se hace referencia a:

- Registros en blanco
- Registros Incompletos
- Registros con contenidos erróneos o que no corresponden
- Registros que por su naturaleza no permiten ser representados

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Todos estos datos, en primera instancia no están preparados para ser representados por alguna herramienta para dar informes o resultados, ni mucho menos para proporcionar algún tipo de información coherente al usuario final. Es por eso que necesitan ser tratados previamente por alguna herramienta que permita homogeneizar y ordenar estos datos.

A continuación se puede apreciar el trabajo “Kettle” completo, el cual consta de un camino de transformaciones conectadas entre si

6

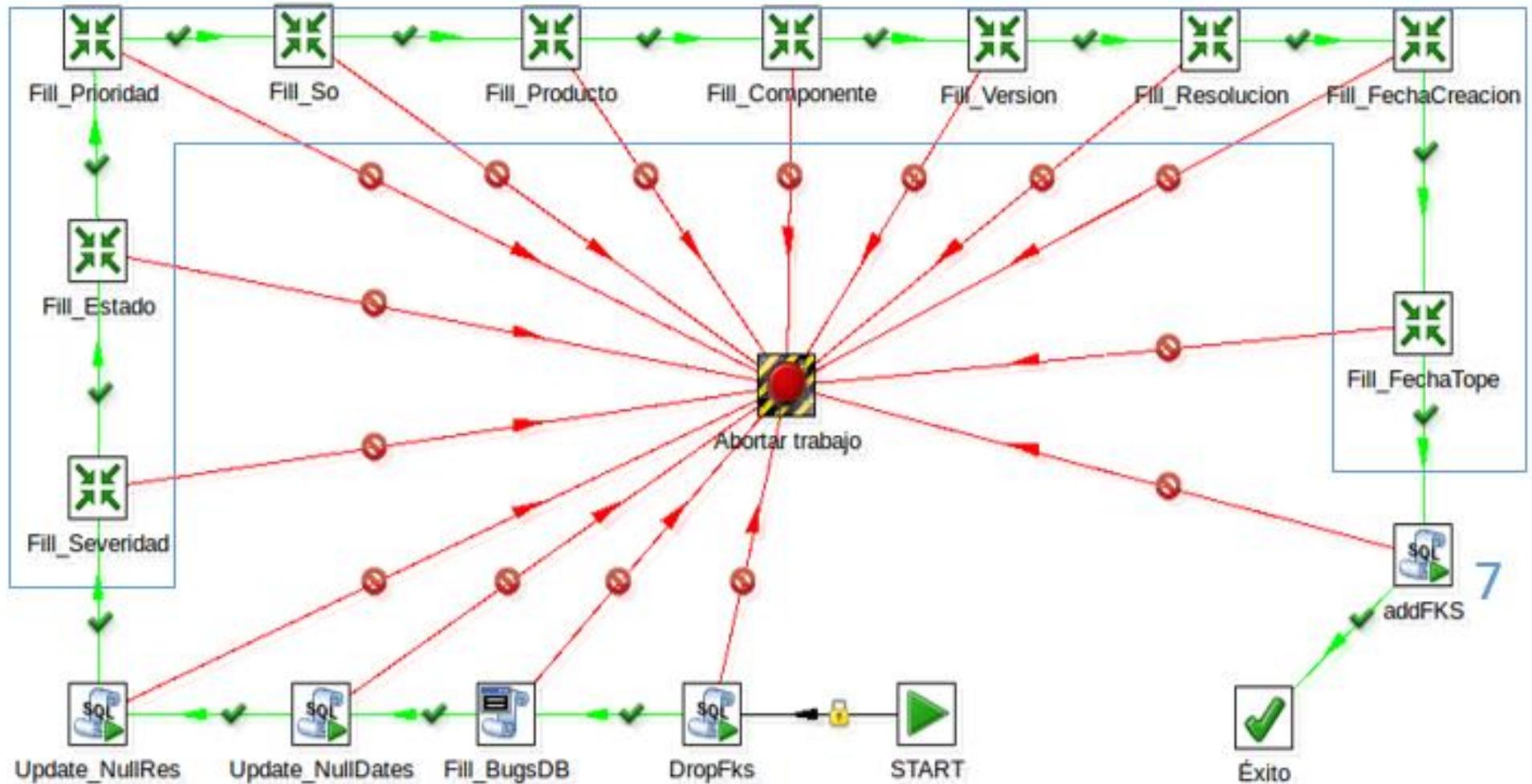


Ilustración 30: Trabajo Kettle completo



START

Indica el comienzo del trabajo.

Este elemento puede ser editado con el fin de automatizar la ejecución del trabajo.

A continuación se pueden apreciar las diferentes opciones de programación para su ejecución:

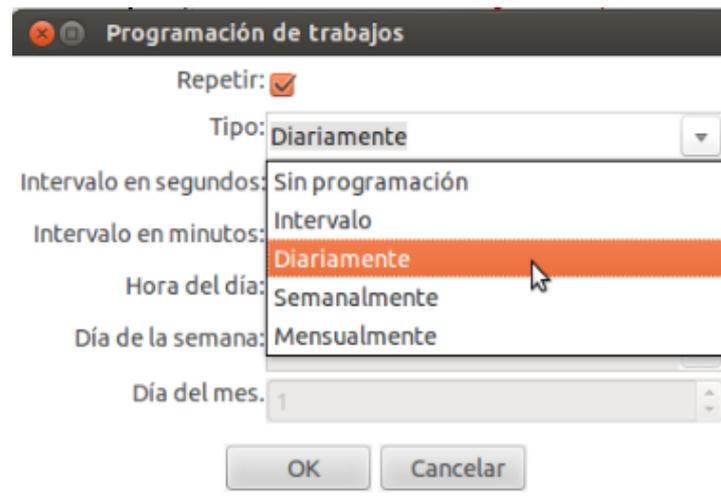


Ilustración 31: Programación de trabajos Kettle



START



DropFks

Para poder actualizar las tablas de la base de datos creada es necesario eliminar las claves foráneas, las cuales serán creadas nuevamente al finalizar el trabajo.

Esta transformación se realiza mediante un script SQL que lleva a cabo dicha tarea:

### Script SQL:

```
ALTER TABLE `tesis`.`bugzilla`
DROP FOREIGN KEY `fk_fechaCreacion` ,
DROP FOREIGN KEY `fk_fechaTope` ,
DROP FOREIGN KEY `fk_so` ,
DROP FOREIGN KEY `fk_resolucion` ,
DROP FOREIGN KEY `fk_prioridad` ,
DROP FOREIGN KEY `fk_producto` ,
DROP FOREIGN KEY `fk_componente` ,
DROP FOREIGN KEY `fk_version` ,
DROP FOREIGN KEY `fk_severidad` ,
DROP FOREIGN KEY `fk_estado` ;
```

Ilustración 32: Script "Dropfks"



### .L\_BUGSDB

Fill\_BugsDB

Todos los bugs cargados en la base de datos de “Bugzilla” deben ser tomados en cuenta a la hora de tomar decisiones, incluso los más recientes.

Esta transformación consta de un script en bash que se conecta a dicha base, toma los campos necesarios y los guarda la tabla “bugzilla” de la base de datos creada para los posteriores análisis.

A continuación se puede apreciar dicho script:

General Script

```
#!/bin/bash

mysql bugs --user=root --password=root -e
"DELETE FROM tesis.bugzilla;
INSERT INTO tesis.bugzilla(severidad, estado, prioridad, so, version, producto, componente, resolucion, fecha_creacion, fecha_tope, tiempo_estimado)
SELECT bug_severity, bug_status, priority, op_sys, version, p.name as product, c.name as component, resolution, creation_ts, deadline, estimated_time
FROM bugs.bugs b, bugs.components c, bugs.products p
WHERE b.component_id=c.id and b.product_id=p.id"
```

**Ilustración 33: Script “Fill\_BugsDB”**



### DATE\_NULLDATES

Update\_NullDates

Si al ingresar un nuevo bug mediante la interfaz de Bugzilla el usuario no selecciona una fecha tope, dicho campo obtendría un valor “null”. Al pasar los datos a nuestra tabla bugzilla se decide transformar dicho valor en 0000-00-00, el cual es inválido, para denotar que la fecha es inexistente.

Esta transformación es un script SQL que actualiza aquellos bugs cuya fecha tope era nula a un valor único que hemos seleccionado (el primer valor valido para un campo de tipo DATE en MySQL).

#### Script SQL:

```
UPDATE tesis.bugzilla
SET
fecha_tope = '1000-01-01'
where fecha_tope='0000-00-00' AND bug_id>0;
```

**Ilustración 34: Script “Update\_NullDates”**



### DATE\_NULLRES

Update\_NullRes

Si al ingresar un nuevo bug mediante la interfaz de Bugzilla el usuario no indica un valor para el campo resolución, este adquiere un valor “null”. Al normalizar nuestra tabla “bugzilla” dichos valores nulos se convierten en un problema, ya que claves primarias no pueden ser nulas.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Para solucionar esto esta transformación es un script SQL que asigna la cadena “Sin Valor” a los casos antes mencionados:

### Script SQL:

```
UPDATE tesis.bugzilla
SET
resolucion = 'Sin Valor'
where LENGTH(resolucion)=0 AND bug_id>0;
```

Ilustración 34: Script “Update\_Null\_Res



L\_ SEVERIDAD / ESTADO / PRIORIDAD / ETC

Fill\_XXX

Una vez poblada la tabla “bugzilla” procedemos a normalizar la base creada.

Todas las transformaciones agrupadas en este punto poseen la misma lógica, es por esto que solo se explicara a modo de ejemplo “Fill\_Severidad”.

Compuesta por dos pasos, esta transformación toma todas las filas de la columna Severidad de la tabla “Bugzilla” y sin incluir duplicados mediante el modificador

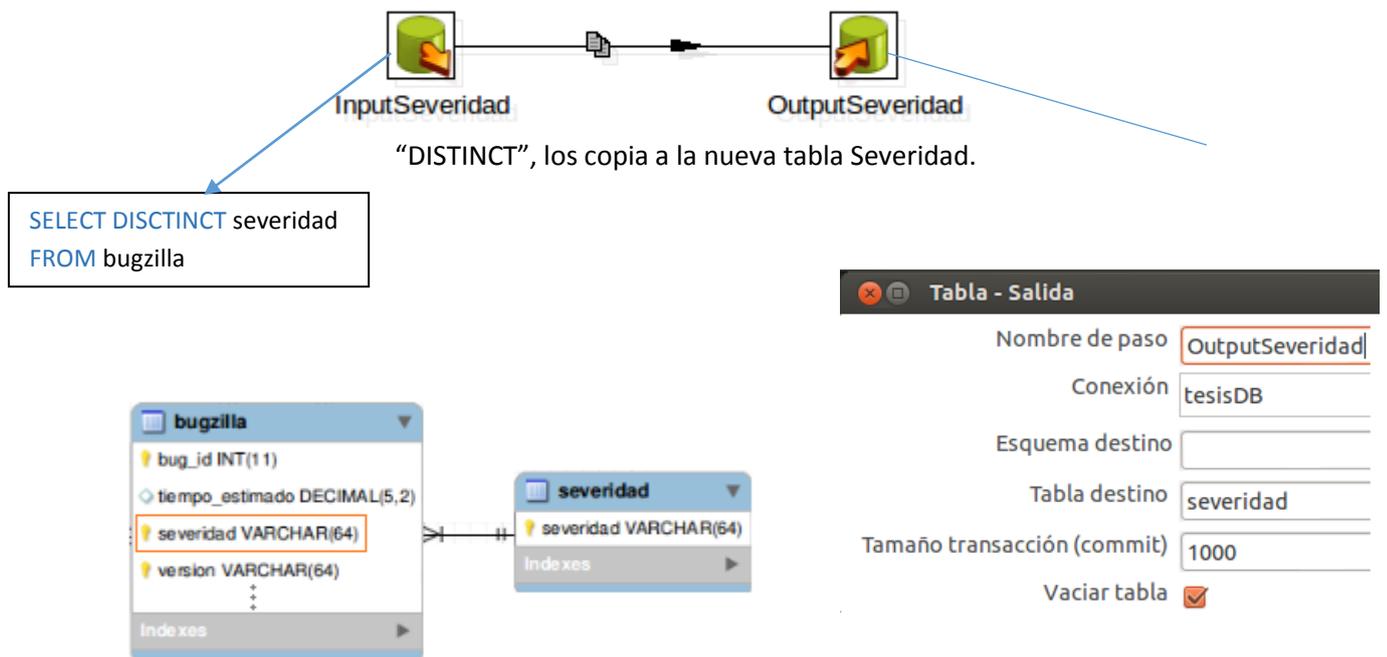


Ilustración 35: “Fill” tablas externas

Debido al pequeño tamaño de los resultset de estas consultas hemos optado por vaciar las tablas antes de poblarlas utilizando una opción de Kettle que ya trae integrada ( ) en vez de actualizarlas.

De este modo las siguientes tablas, además de severidad, son también pobladas:

- Estado
- Prioridad
- SO (Sistema Operativo)
- Producto
- Componente
- Versión
- Resolución
- FechaCreacion
- FechaTope



DFKS

Esta transformación es un script SQL que vuelve a crear las claves foráneas dropeadas en el paso 1.

### Script SQL:

```
ALTER TABLE `tesis`.`bugzilla`
ADD CONSTRAINT `fk_estado`
  FOREIGN KEY (`estado` )
  REFERENCES `tesis`.`estado` (`estado` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_severidad`
  FOREIGN KEY (`severidad` )
  REFERENCES `tesis`.`severidad` (`severidad` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_version`
  FOREIGN KEY (`version` )
  REFERENCES `tesis`.`version` (`version` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_componente`
  FOREIGN KEY (`componente` )
  REFERENCES `tesis`.`componente` (`componente` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_producto`
  FOREIGN KEY (`producto` )
  REFERENCES `tesis`.`producto` (`producto` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_prioridad`
  FOREIGN KEY (`prioridad` )
  REFERENCES `tesis`.`prioridad` (`prioridad` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_resolucion`
  FOREIGN KEY (`resolucion` )
  REFERENCES `tesis`.`resolucion` (`resolucion` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_so`
  FOREIGN KEY (`so` )
  REFERENCES `tesis`.`so` (`so` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_fechaCreacion`
  FOREIGN KEY (`fecha_creacion` )
  REFERENCES `tesis`.`fechaCreacion` (`fechaC` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
ADD CONSTRAINT `fk_fechaTope`
  FOREIGN KEY (`fecha_tope` )
  REFERENCES `tesis`.`fechaTope` (`fechaT` )
  ON DELETE NO ACTION
  ON UPDATE NO ACTION;
```

**Ilustración 36: Transformación “AddFks”**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

---



Éxito

ÉXITO

Si se llega a este paso significa que el trabajo se ha ejecutado con éxito y todas las tablas se han actualizado.

### TRABAJO KETTLE AUTOMATIZADO

Para asegurar que los datos representados en los reportes de Saiku estén actualizados y los cambios en la base de datos se vean reflejados en tiempo real, la ejecución del trabajo Kettle debe ser automatizada.

Para lograr dicha automatización se utiliza el script de Kettle para ejecutar trabajos por línea de comandos, Kitchen, junto a una herramienta nativa de Unix para programar la ejecución del mismo, Cron.

#### KITCHEN

Kitchen es uno de los módulos de Kettle, y se utiliza para ejecutar trabajos diseñados con Spoon y almacenados como XML (extensión .kjb) ó en el repositorio de base de datos.

Al ejecutar un trabajo por línea de comandos utilizando kitchen existen ciertos parámetros que pueden ser especificados para personalizar su ejecución. Los más importantes son:

- **-file=** nombre\_del\_archivo  
Esta opción ejecuta el archivo .kjb definido.
- **-log=** nombre\_del\_archivo  
Esta opción guarda un registro o log en el archivo definido.
- **-level=** nivel\_de\_registro  
Esta opción define el nivel de registro o log que se desea, estos pueden ser:
  - **Error**  
Mostrar sólo errores.
  - **Nothing**  
No mostrar salida.
  - **Minimal**  
Información mínima.
  - **Basic**  
Información básica (opción por defecto si no se especifica).
  - **Detailed**  
Información detallada.
  - **Debug**  
Salida muy detallada, utilizada comúnmente para corrección de errores.
  - **Rowlevel**  
Información a nivel de registro, genera una gran cantidad de datos.
- **-rep=** nombre\_del\_repositorio  
Esta opción define la conexión al repositorio ingresado si se utiliza uno. Si se utiliza también será necesario especificar las opciones -user, -pass, -dir y -job.
  - **-user=** nombre\_de\_usuario  
Esta opción define el usuario con el cual se conectará al repositorio si se utiliza uno.
  - **-pass=** contraseña  
Esta opción define la contraseña del usuario con el cual se conectará al repositorio si se utiliza uno.
  - **-dir=** /.../.../nombre\_directorio  
Esta opción define el directorio en el cual se encuentra el repositorio si se utiliza uno.
  - **-job=** nombre\_del\_trabajo  
Esta opción define el trabajo almacenado en el repositorio que se desea ejecutar.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

A continuación se presenta un ejemplo de la ejecución del trabajo Kettle realizado por los tesisistas desde la línea de comando utilizando Kitchen con algunos de los parámetros disponibles:

```
usuario@tfg:~/pentaho/data-integration$ ./kitchen.sh -file=/home/usuario/TrabajoKettle/TesisJob.kjb
-log=/home/usuario/logs/log -level=Minimal
INFO 23-01 21:10:50,509 - Kitchen - Nivel de registro: Registro mínimo
INFO 23-01 21:10:50,511 - Kitchen - Comienzo de la ejecución.
INFO 23-01 21:10:50,669 - TesisJob - Ejecución del trabajo comenzada
INFO 23-01 21:10:56,920 - TesisJob - Ejecución del trabajo finalizada
INFO 23-01 21:10:56,923 - Kitchen - ¡Finalizado!
INFO 23-01 21:10:56,924 - Kitchen - Comienzo=2014/01/23 21:10:50.512, Final=2014/01/23 21:10:56.924
INFO 23-01 21:10:56,925 - Kitchen - El procesamiento terminó luego de 6 segundos.
usuario@tfg:~/pentaho/data-integration$
```

Ejecución con logeo activado y nivel de registro mínimo

```
usuario@tfg:~/logs$ cat log
2014/01/23 21:09:51 - Kitchen - Nivel de registro: Registro mínimo
2014/01/23 21:09:51 - Kitchen - Comienzo de la ejecución.
2014/01/23 21:09:52 - TesisJob - Ejecución del trabajo comenzada
2014/01/23 21:10:04 - TesisJob - Ejecución del trabajo finalizada
```

*Ilustración 37: Ejecución trabajo kettle por línea de*

## CRON

Cron es un administrador de procesos en segundo plano (demonio) que ejecuta procesos o scripts a intervalos regulares (por ejemplo, cada minuto, día, semana o mes).

Cron es un demonio (servicio), lo que significa que solo requiere ser iniciado una vez, generalmente con el mismo arranque del sistema.

Para controlar el estado del mismo solo es necesario escribir el comando “service cron status” en la consola:

```
usuario@tfg:/etc$ service cron status
cron start/running, process 845
```

*Ilustración 38: Servicio “Cron”*

## USO DE CRON

### CRONTAB

Que procesos y a la hora que estos deben ser ejecutados se especifican en el fichero crontab el cual se encuentra ubicado bajo el directorio /etc:

```
usuario@tfg:/etc$ cat crontab
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
usuario@tfg:/etc$
```

*Ilustración 39: Visualización del crontab*

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Por defecto se puede observar que crontab trae definida la ejecución automática de los archivos que se encuentran dentro de las carpetas cron.hourly/ daily/ weekly/ monthly. Los atributos que se aprecian en la imagen y deben ser definidos para cada proceso o script a ejecutar son:

- m → Minuto (0-59).
- h → Hora (0-23).
- dom → Día del Mes (1-31)
- mon → Mes (1-12), o por el nombre del mes en inglés (primeras tres letras).
- dow → Día de la semana (0-7) o por el nombre del día en inglés, (primeras tres letras).
- user → Usuario que ejecuta el comando.
- command → Comando, script o programa que se desea ejecutar.

Un asterisco (\*) en cualquiera de los atributos significa todos los valores, por ejemplo un \* en el campo dom indicaría todos los días del mes.

Esta opción de uso de cron permite una mayor flexibilidad y personalización a la hora de ejecutar comandos, pero para los fines del presente proyecto se considera más apropiado una ejecución cada una hora del trabajo Kettle para actualizar posibles cambios.

---

### FICHEROS CRON.DAILY/HOURLY/MONTHLY/WEEKLY

Linux trae por defecto ya definidos como se vio en crontab la ejecución de todos los scripts que se encuentren dentro de las carpetas:

```
usuario@tfg:/etc$ locate cron.*ly
/etc/cron.daily
/etc/cron.hourly
/etc/cron.monthly
/etc/cron.weekly
```

*Ilustración 40: Linux "Cron" scripts*

Si se coloca un archivo tipo script en cualquiera de estos directorios, entonces el script se ejecutará cada hora, cada día, cada semana o cada mes, dependiendo del directorio.

A continuación se analizan los parámetros de crontab con los que se ejecutarán las líneas mencionadas definidas por defecto:

```
# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
```

*Ilustración 41: Parámetro Hourly*

**Hourly:** Se ejecuta el comando todos los días del mes a los 17 minutos de cada hora

```
# m h dom mon dow user  command
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
```

*Ilustración 42: Parámetro Daily*

**Daily:** Se ejecuta el comando todos los días del mes a las 6:25

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

```
# m h dom mon dow user  command
47 6   * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
```

**Ilustración 43: Parámetro Weekly**

**Weekly:** Se ejecuta el comando todos los días 7 de la semana (Domingo) a las 6:47

```
# m h dom mon dow user  command
52 6   1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
```

**Ilustración 44: Parámetro Monthly**

**Monthly:** Se ejecuta el comando todos los días 1ro de cada mes a las 6:52

Se logra entonces la automatización de la ejecución del trabajo Kettle cada una hora agregando el siguiente script al directorio /etc/cron.hourly/

```
usuario@tfg:/etc/cron.hourly$ cat correrTrabajoKettle.sh
#!/bin/bash
touch /home/usuario/logs/log_$(date +%m) # Se crea el archivo de log
#representado por la hora a la que fue creado
cd /home/usuario/pentaho/data-integration # Se accede al directorio de Kettle
./kitchen.sh -file=/home/usuario/TrabajoKettle/TesisJob.kjb -log=/home/usuario/logs/log_$(date +%m) -level=basic #Se ejecuta kitchen
```

**Ilustración 45: Seteo script en "Cron Hourly"**

## BI SERVER Y ADMINISTRATION CONSOLE: CREACION DE USUARIOS Y ROLES

La plataforma de pentaho posee dos componentes centrales:

### 1. Servidor BI

El servidor es el lugar en donde se alojan los diferentes módulos de BI integrados en el servidor. Al abrirlo nos encontramos con el “Design Studio”, una colección de editores, visores y módulos integrados en una misma aplicación que proveen un entorno grafico para la construcción y testing de reportes.

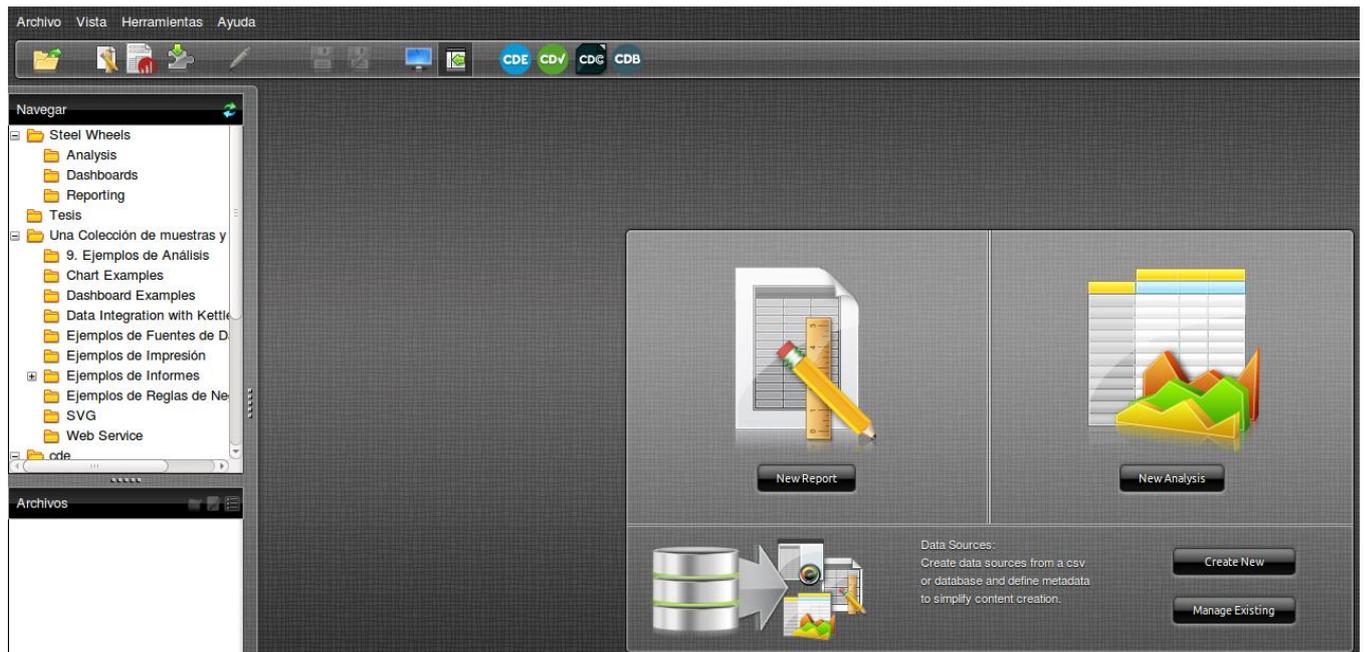


Ilustración 44: Servidor BI

### 2. Consola Administrativa

Provee un lugar central desde el cual se pueden administrar los despliegues en pentaho. La consola agrega y simplifica muchas tareas administrativas como el manejo de usuarios, roles, programación de trabajos y manejo de servicios.

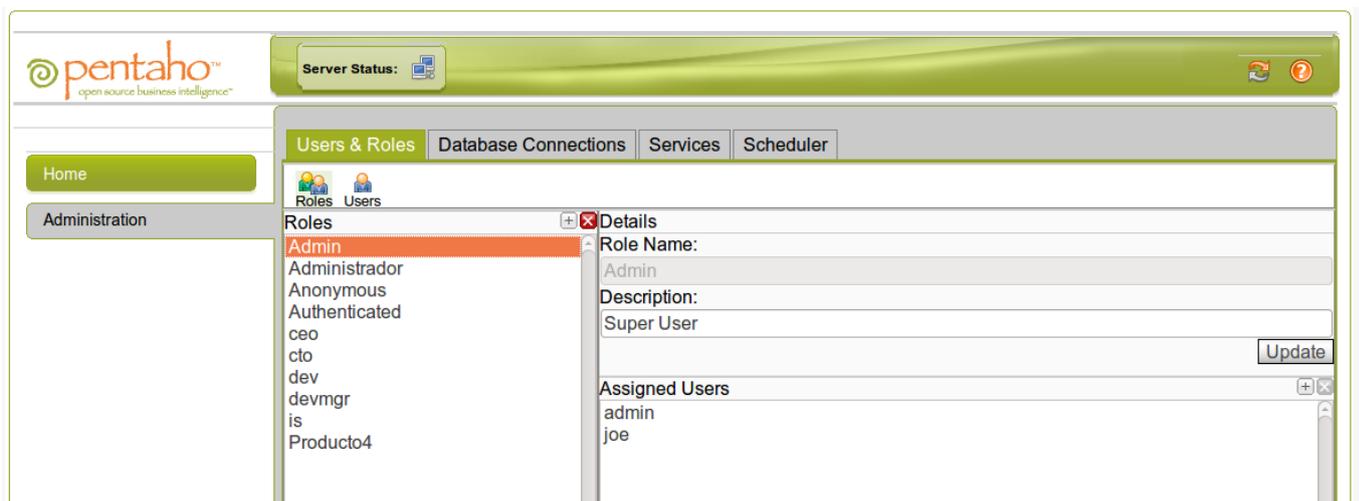


Ilustración 45: Consola Administrativa

**Nota:** La instalación y puesta en marcha de esta herramienta, esta detallada paso a paso en el anexo del actual documento.

### DESARROLLO

Una vez terminado el trabajo con mondrian, el primer paso a seguir es crear un usuario para poder acceder al diseñador de pentaho.

Esto se realiza accediendo a la consola administrativa. Para lanzar dicha consola se debe ejecutar el siguiente script situado en el path “*\$PentahoHome/administration-console/*”:

```
./start-pac.sh
```

Dicho script inicia la consola administrativa de pentaho en el puerto 8099, accedemos a ella ingresando la siguiente url en el navegador:

<http://localhost:8099>

Para poder dar usuarios de alta debemos ingresar con un usuario que tenga permisos de administrador. Al ingresar la aplicación solicitara un usuario y contraseña, que por defecto es **admin : admin**. Por cuestiones de seguridad el mismo debe ser cambiado luego de la instalación.

---

### CREACION DE USUARIOS

La jerarquía de permisos de pentaho nos permite la creación de usuarios y roles. Estos roles pueden representar a uno o más usuarios, y pueden ser mapeados siguiendo algún criterio particular.

Pentaho posee 3 roles por defecto con permisos especiales asociados. Estos no pueden eliminarse:

- **ADMIN:** Rol que brinda permisos de superusuario al usuario sobre plataforma.
- **AUTHENTICATED:** Brinda permiso de login al usuario asociado.
- **ANONYMOUS:** Permite al usuario loguearse como invitado.

Para crear un usuario nos dirigimos a la solapa **Users&Roles:**

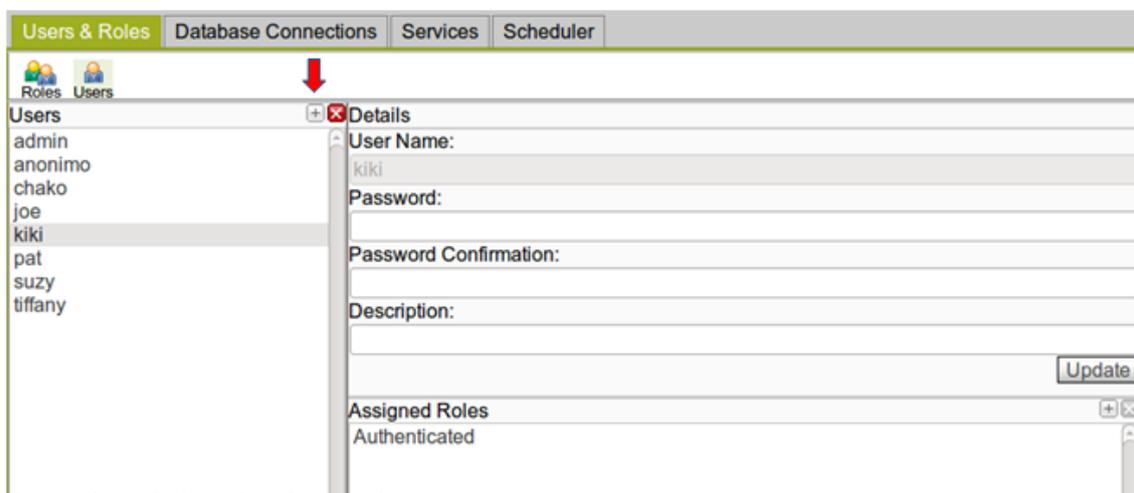
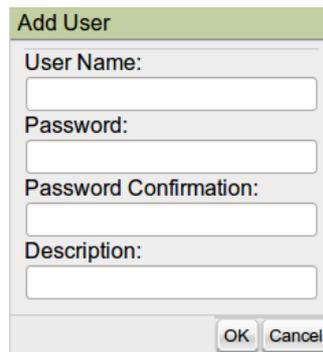


Ilustración 46: Creación de usuarios

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Allí se da clic en el botón “+”, lo que despliega una interfaz para ingresar los datos del usuario:



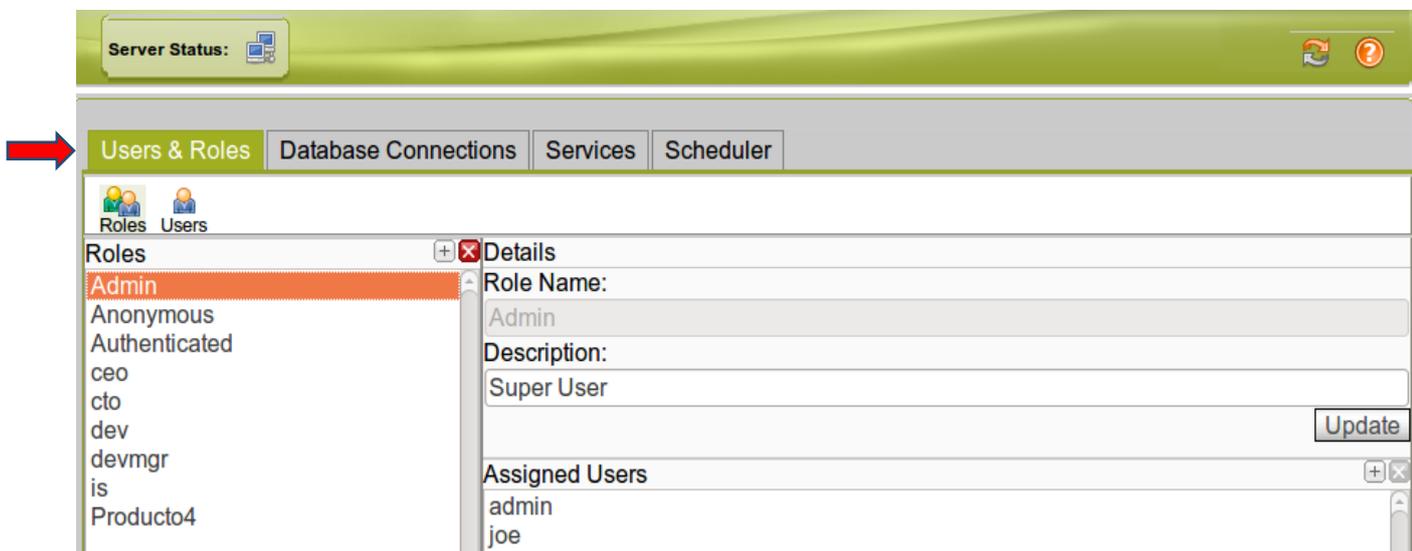
*Ilustración 47: Interfaz agregar usuario*

Finalizada la creación del usuario se puede proceder a agregar uno o más roles a este. Para ello debe seleccionarse el usuario en el panel de nuestra izquierda y presionar el botón “+” que aparece en el panel de la derecha.



*Ilustración 48: Roles asignados a un usuario*

Por defecto, un nuevo usuario se crea con el permiso “**Authenticated**”. De la misma manera se puede dar de alta o eliminar un rol presionando en el botón “**Roles**”.



*Ilustración 49: Pestaña usuarios y roles*

A modo de ejemplo se ha creado un usuario llamado “**UsuarioPrueba**”, al que luego se le asignara un rol específico, el cual tendrá permisos asociados desde la herramienta “**Mondrian Schema Workbench**”

### DISEÑADOR

Luego de crear un usuario, se hace uso del mismo para loguearse en el diseñador. Para ingresar al mismo, primero se debe lanzar el servidor BI. Para ello se debe ejecutar el siguiente script situado en el path “*\$PentahoHome/biserver-ce/*”:

```
./start-pentaho.sh
```

Dicho script inicia el BI server de pentaho en el puerto 8080, se accede a él ingresando la siguiente url en el navegador:

<http://localhost:8080/pentaho>



*Ilustración 50: Login pentaho*

Al loguearse, ya se dispone la herramienta para hacer uso de la misma.

En la barra superior se encontramos los menús, que permiten realizar diversas acciones. Debajo de ello, se encuentran algunos iconos que hacen referencias a los plugins y aplicaciones embebidas que soporta pentaho. La mayoría de las aplicaciones/plugins pueden instalarse desde el “Marketplace” que ofrece pentaho, accesible desde el menú a través de la pestaña Herramientas -> Marketplace.



*Ilustración 51: Barra de herramientas, Marketplace*

El próximo pasó a llevar a cabo será la modelación de la estructura olap mediante la herramienta grafica “**Mondrian Schema Workbench**”.

### MONDRIAN

Mondrian es una de las aplicaciones más importantes de la plataforma Pentaho BI. Es un servidor OLAP, ó ROLAP más correctamente ya que no tiene su propio almacenamiento de datos, sino que trabaja con los datos almacenados en una base de datos relacional (cualquiera la que pueda conectarse por JDBC).

Su función consiste en traducir las consultas de lenguaje dimensional (MDX, XMLA, etc) a SQL estándar y arrojarlas contra la base relacional, actuando como “JDBC para OLAP”. Es decir, es el encargado de gestionar la comunicación entre una aplicación OLAP (escrita en Java) y la base de datos con los datos fuente.

Permite crear cubos de información para el análisis multidimensional que se componen de archivos XML y en ellos se definen las dimensiones y las conexiones de los datos. Los archivos XML por lo general son complejos de realizar manualmente por lo que es común utilizar herramientas gráficas para realizar la edición de estos.

En sí es un archivo JAR, y viene preinstalado en la plataforma Pentaho, por lo que no es necesario instalar otros paquetes para utilizarlo.

Una de las grandes ventajas que aporta es el uso de la cache para mejorar el tiempo de respuesta al no tener que volver a realizar una consulta que ya ha hecho. Además es quien hace un perfecto manejo de roles y permisos de usuarios junto con la aplicación BI de Pentaho.

Dado los fines de este proyecto, solo se utilizarán las capacidades de análisis de Mondrian desde el punto de vista de usuarios, no desarrolladores.

### MONDRIAN SCHEMA WORKBENCH

Mondrian Schema Workbench (también llamado Pentaho Schema Workbench, en adelante PSW) crear esquema de información de forma gráfica y sencilla para realizar el análisis multidimensional.

Los cubos se generan como archivos XML y en ellos se definen las Dimensiones y las conexiones de los datos. Los XML son necesarios para servir como fuente de datos del Front-End “Saiku”. Generalmente dichos esquemas son complejos de realizar manualmente por lo que es común utilizar este tipo de herramientas gráficas para realizar la edición de los mismos. Este motor es necesario ya que las preguntas que podría hacer un usuario serían muy costosas si se tratasen de realizar contra una base de datos relacional.

En este punto se hace referencia a la presencia de operaciones matriciales o “Joins” entre varias tablas que pueden llegar a tener millones de entradas, lo que se traduciría en consultas que demoren mucho en ser respondidas, sin hablar del nivel de concurrencia que podría tener esta base con más de un usuario haciendo este tipo de consultas y sin tener en cuenta de que debería contarse con el conocimiento avanzado tanto del lenguaje SQL y de la estructura de la base relacional para llevar a cabo dicha tarea, algo no esperable para todos los tipos de usuarios.

Es por ello que se requiere de una estructura olap, que soporte y permita este tipo de operaciones. Para ello es que será utilizado el “**Mondrian Schema Workbench**”.

USO

Para iniciar PSW se ejecuta el script “workbench.sh” ubicado en la carpeta raíz del directorio.

El siguiente paso consta en configurar la conexión a la fuente de datos, para esto se accede a “Options → Connection” del siguiente modo:

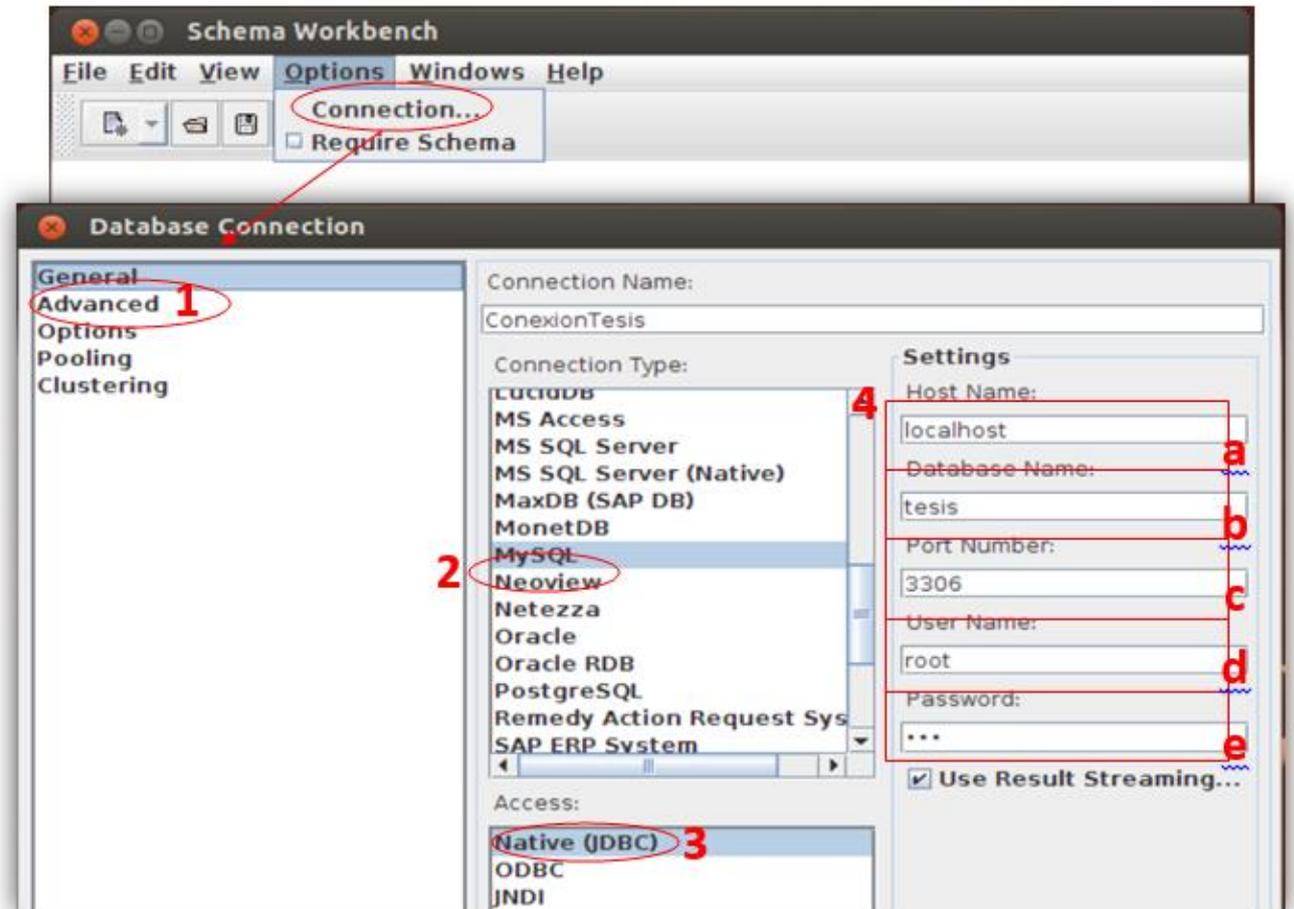


Ilustración 52: Pestaña conexiones, mondrian

Una vez dentro de la configuración de la conexión se procede a ingresar los datos de la base de datos que se utilizará como fuente de datos.

1. La pestaña General es la única a configurar ya que las demás poseen opciones más avanzadas que no serán utilizadas en este trabajo de grado.
2. Se selecciona el Tipo de Conexión “MySQL”
3. Se selecciona el Acceso “JDBC”
4. Se ingresan los datos de la conexión
  - a. Nombre del host
  - b. Nombre de la base de datos
  - c. Numero de puerto
  - d. Nombre de usuario
  - e. Contraseña

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### SCHEMA

Primero se debe crear un Schema, para esto vamos a File → New → Schema

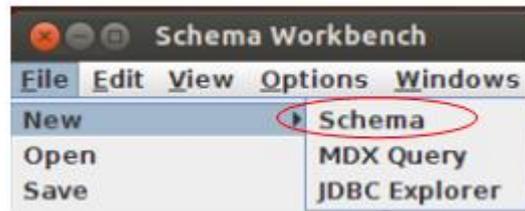


Ilustración 53: Creación Schema

Luego se le asigna un nombre al Schema quedando del siguiente modo:

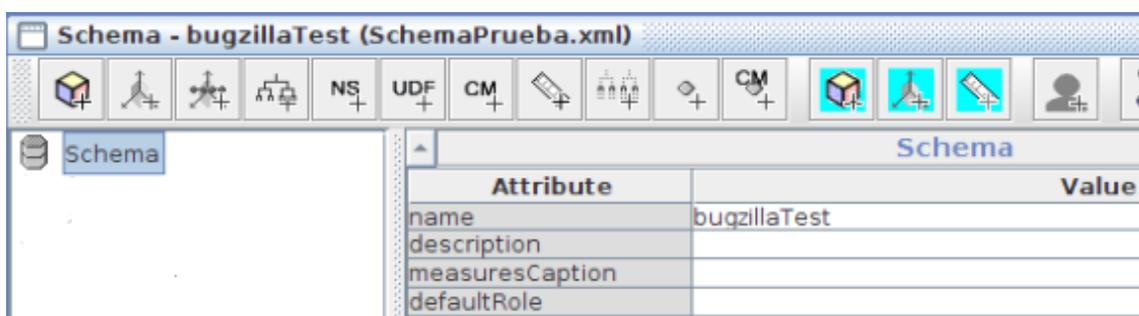


Ilustración 54: Asignación de nombre a un Schema

Se parte del esquema en estrella que se presentó anteriormente a partir del cual se modelará el cubo:

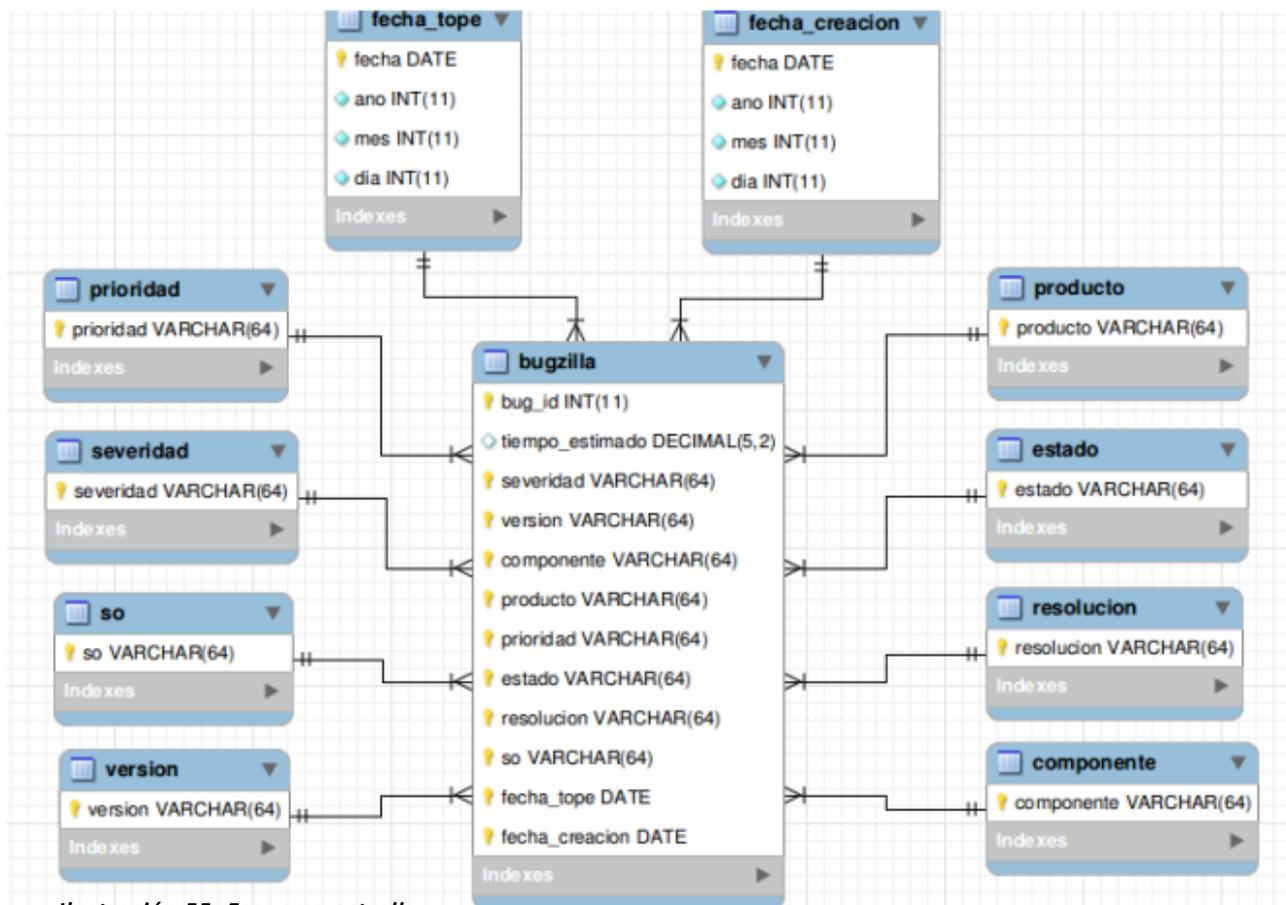


Ilustración 55: Esquema estrella

CUBO

Una vez que el Schema fue creado se procede a crear un nuevo Cubo haciendo click derecho sobre el Schema y eligiendo la opción “Add Cube”:

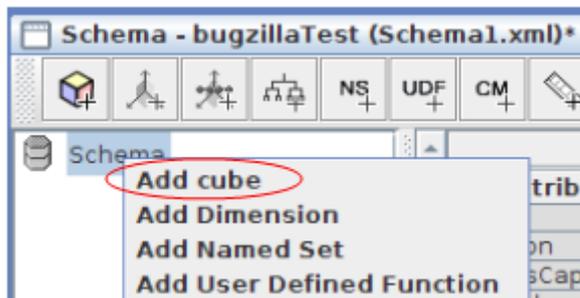


Ilustración 56: Agregar cubo

Ahora es necesario asignar un nombre al cubo creado

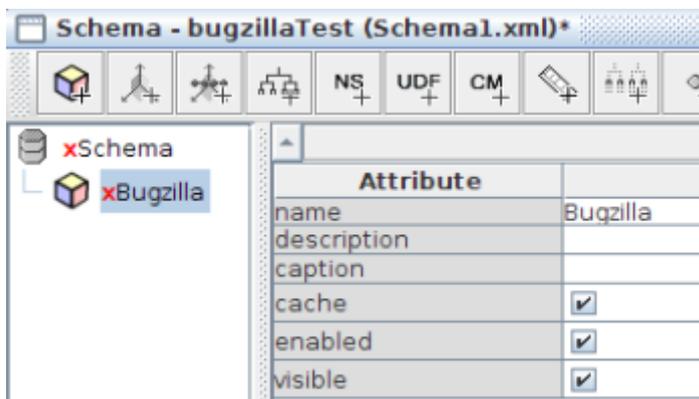


Ilustración 57: Asignar nombre al cubo

TABLA DE HECHOS

PSW avisa que no se ha definido la tabla de hechos aún mediante un mensaje en la parte inferior de la interfaz.

Fact name must be set

Para definirla se debe hacer click derecho sobre el cubo y seleccionar la opción “Add Table”

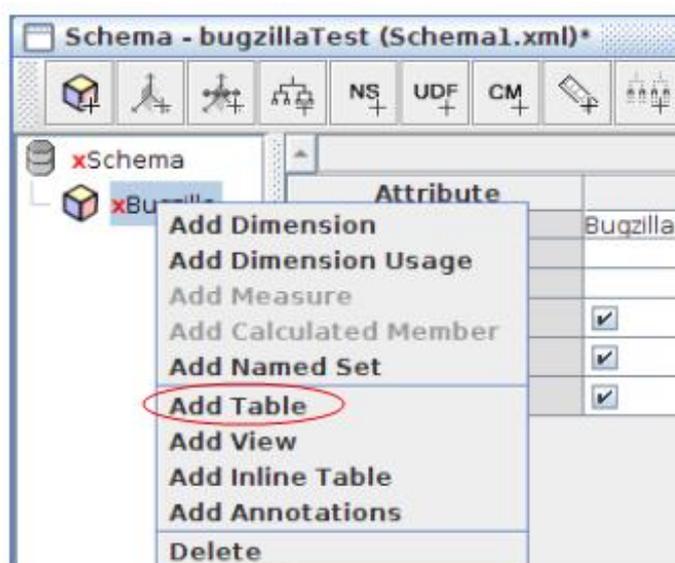


Ilustración 58: Agregar tabla

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

En el campo Name es posible elegir la tabla del listado que PSW genera a partir de la conexión configurada anteriormente:

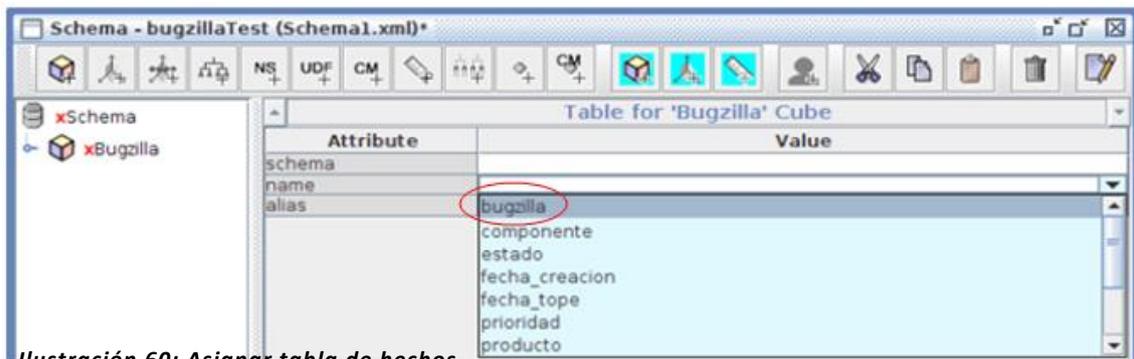


Ilustración 60: Asignar tabla de hechos

Una vez seleccionada la tabla de hechos “bugzilla” se procede a crear las dimensiones.

### DIMENSIONES

Se denomina “dimensiones” a aquellos datos que nos permiten filtrar, agrupar o seccionar la información.

Para crear una nueva dimensión se debe realizar click derecho sobre el cubo, y luego seleccionar la opción “Add Dimension”:

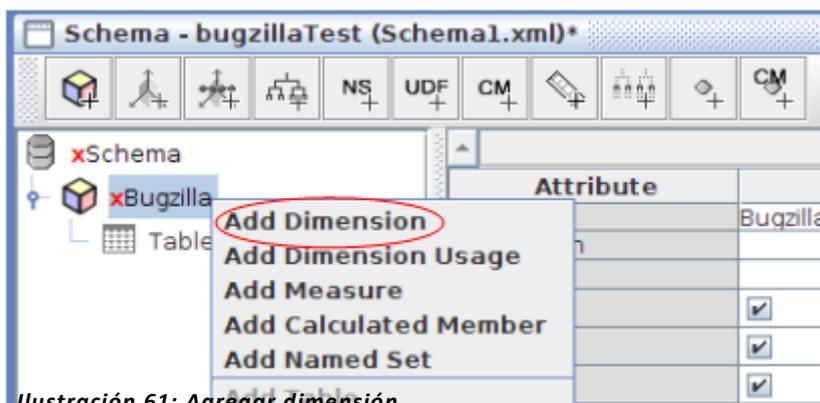


Ilustración 61: Agregar dimensión

Al igual que en los casos anteriores se debe asignar un nombre a la dimensión. A modo de ejemplo se realizara la implementación completa de la dimensión “Estado” ya que el procedimiento es muy similar para las restantes dimensiones excepto en las de tiempo cuyas diferencias serán explicadas.

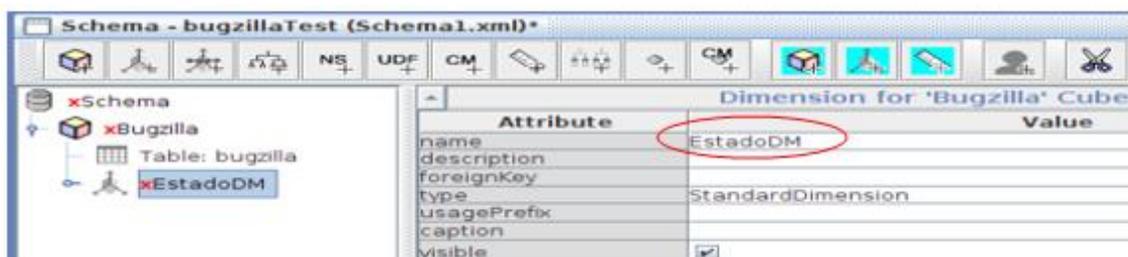


Ilustración 62: Establecer nombre dimensión

**\*NOTA:** En las dimensiones de tiempo el campo type debe ser del tipo “TimeDimension”

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Una vez definida la Dimensión se procede a especificar las jerarquías que esta poseerá:

### JERARQUIAS

Para poder definir la Jerarquía es necesario antes seleccionar la tabla correspondiente a ésta:

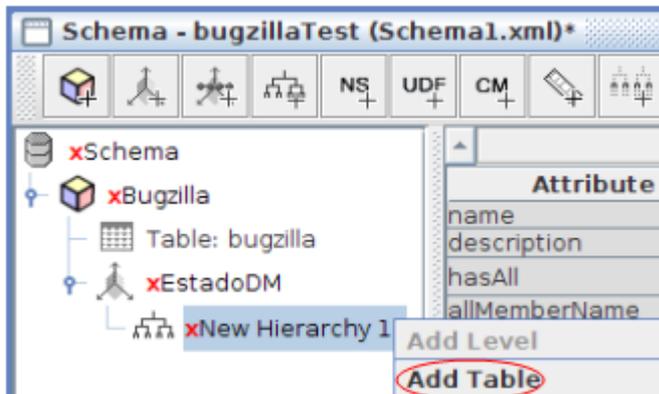


Ilustración 63: Agregar una tabla a una jerarquía

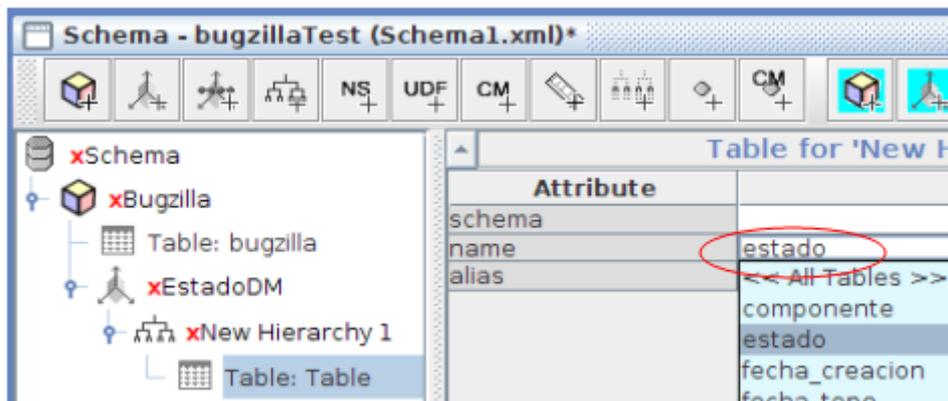


Ilustración 64: Mapear tabla a una jerarquía

Una vez seleccionada la tabla correspondiente a la jerarquía PSW nos permite seleccionar la clave primaria la jerarquía definida.

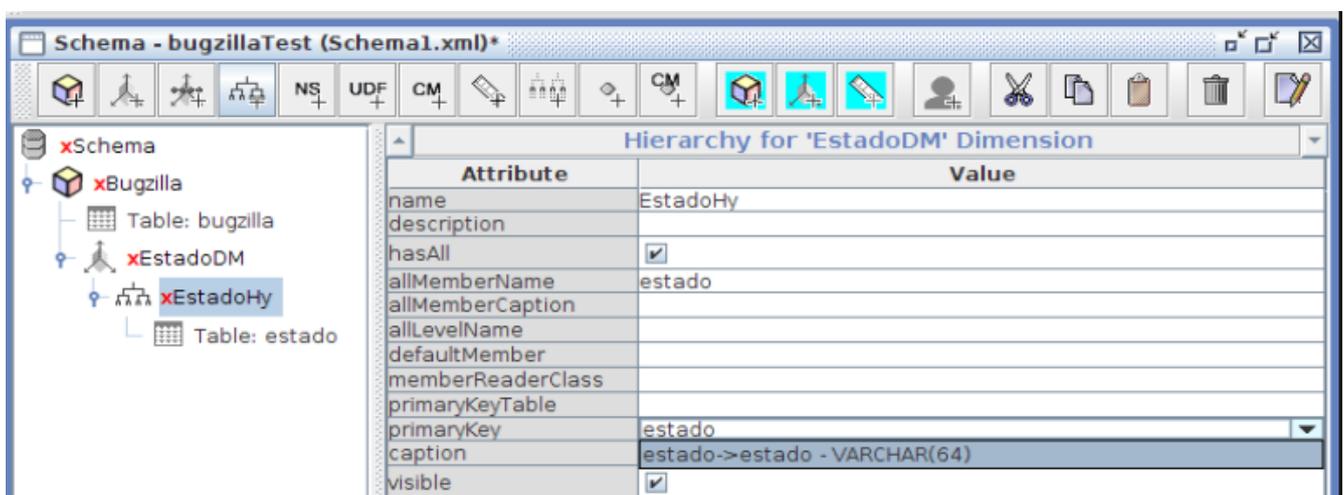


Ilustración 65: Asignar clave primaria para una jerarquía

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Por ultimo se setea el campo caption el cual permite definir que nombre figurará en el menu de selecccion en la interfaz de Pentaho una vez publicado el cubo, en este caso “Estado”.

caption Estado

A continuacion se deben definir los niveles que se desean implementar.

### NIVELES

Para preacisar los niveles se realiza click derecho en la jerarquia y luego se selecciona la opcion “Add Level”

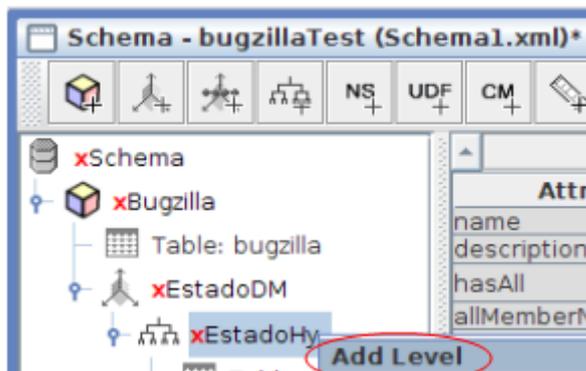


Ilustración 66: Agregar nivel a una jerarquía

Una vez creado el nuevo nivel se definen los siguientes campos:

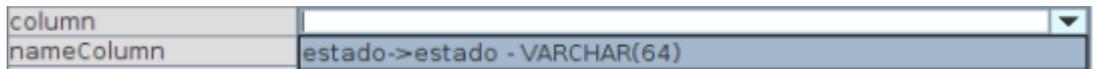
Level for 'EstadoHy' Hierarchy	
Attribute	Value
name	EstadoLv <b>1</b>
description	
table	
column	estado <b>2</b>
nameColumn	
parentColumn	
nullParentValue	
ordinalColumn	
type	String <b>3</b>
internalType	String
uniqueMembers	<input checked="" type="checkbox"/>
levelType	Regular <b>4</b>
hideMemberif	Never
approxRowCount	
caption	estado <b>5</b>
captionColumn	
formatter	
visible	<input checked="" type="checkbox"/>

Ilustración 67: Definición de atributos para un nivel

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

1. Nombre del Nivel, en este caso “EstadoLv”
2. Columna a la cual pertenece el nivel dentro de la tabla seleccionada como jerarquía.



*Ilustración 68: Columna que representa al nivel*

3. Tipo de Dato, en este caso “String”

**\*NOTA:** Para Dimensiones de tiempo se utiliza “Integer” y luego se le asigna en el campo LevelType el tipo.

4. Tipo de Nivel, en este caso “Regular”



*Ilustración 69: Diferentes tipos de nivel*

**\*NOTA:** Para Dimensiones de tiempo dependiendo el tipo de nivel aquí se definirá el tipo que se desee, por ejemplo “TimeYears” para un nivel de tipo “Años”.

5. “Caption” es donde se define el título del campo a ser mostrado en Pentaho.

EJEMPLO –FECHACREACION

A continuación se puede apreciar la estructura de un ejemplo donde existe más de un nivel y la dimensión es de tipo fecha:

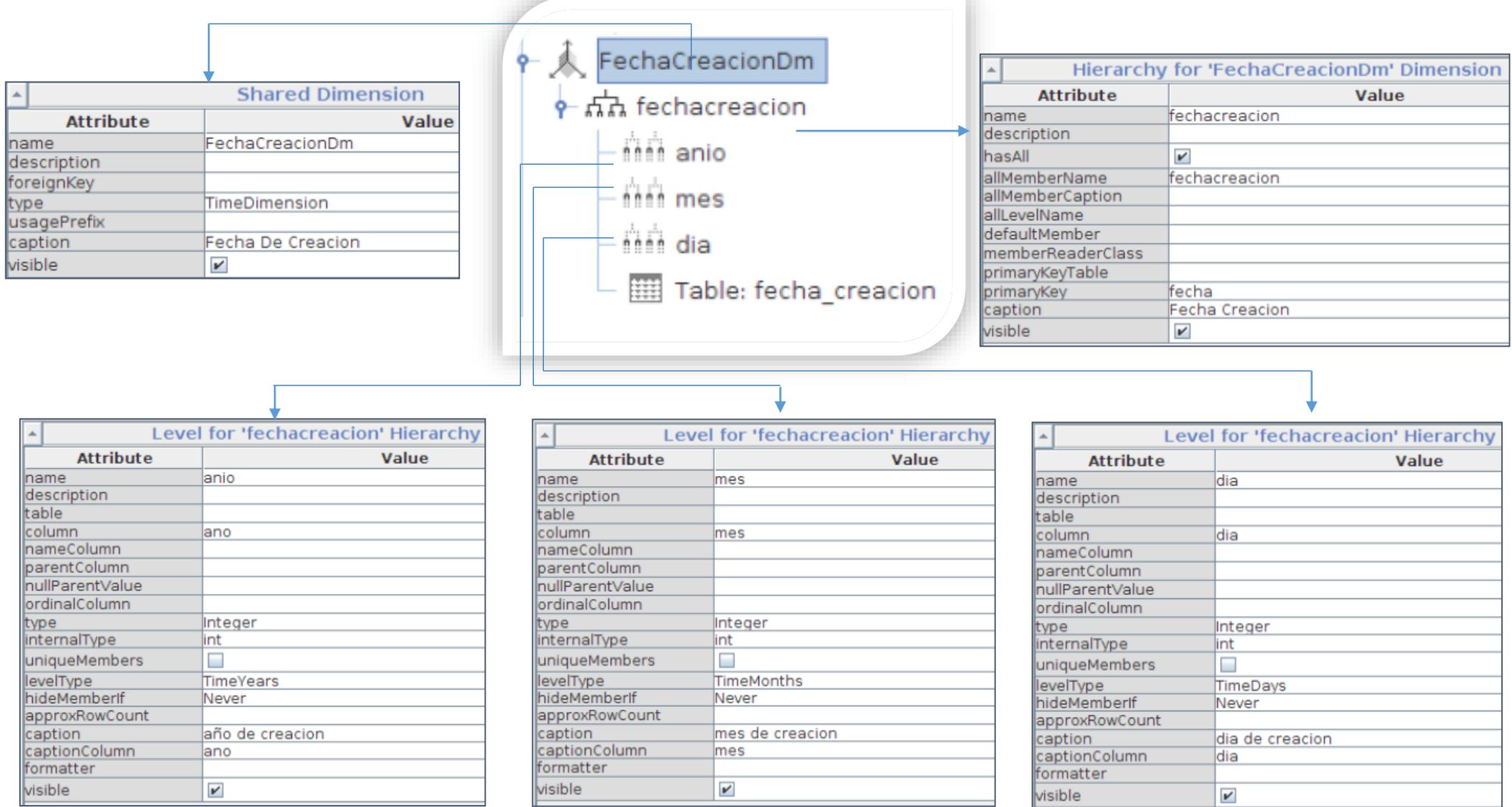


Ilustración 70: Ejemplo completo, dimensión Fecha

MEDIDAS

Una vez creada una dimensión, sus jerarquías y sus niveles se puede apreciar que PSW nos indica con un mensaje que, es necesario definir al menos una medida:

Cube must contain measures

Las medidas son básicamente “el que analizar”, son numéricas y sumables a diferente nivel de detalle.

Para definir una nueva medida se debe hacer click derecho sobre el cubo → Add Measure

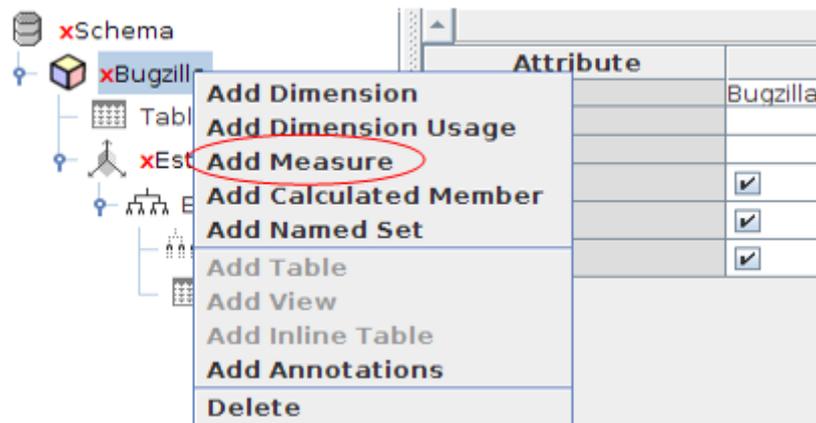


Ilustración 71: Agregar medida

Luego se procede a rellenar los campos que PSW solicita como se aprecia a continuación:

Una vez creada, el sistema no arroja más alertas y estamos listos para el guardado y la publicación del cubo.

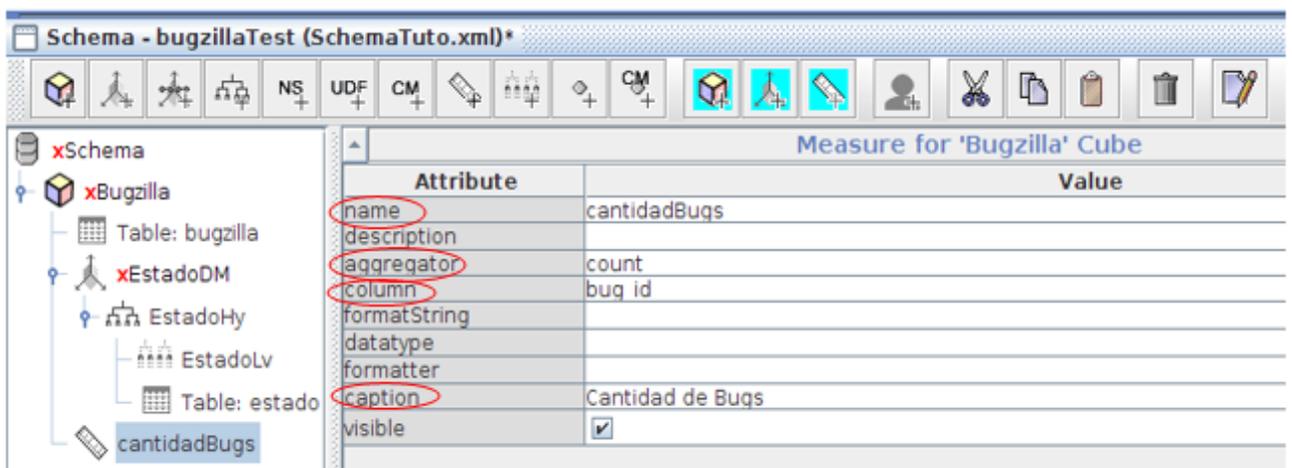


Ilustración 72: Atributos de una medida

### GUARDADO Y PUBLICACIÓN

Para guardar el cubo se debe hacer click en File → Save, y como se puede ver en la imagen PSW creará un archivo de extensión .xml.



*Ilustración 73: Guardado de un cubo*

Para publicar el cubo se debe acceder a File → Publish



*Ilustración 74: Publicación de un cubo*

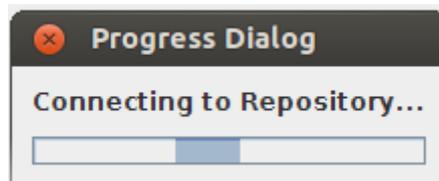
Se puede apreciar que PSW solicita los datos de la conexión:

- **Server**
  - **URL:** Url del donde se ejecuta Pentaho
  - **Publish Password:** Es la contraseña que se definió en la instalación al comienzo de esta sección.
- **Pentaho Credentials**
  - **User:** El usuario de Pentaho con el cual publicaremos el cubo
  - **Password:** La contraseña del usuario de Pentaho con el cual publicaremos el cubo.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Una vez completados los datos, se debe aceptar haciendo click en OK, y se puede ver que PSW informa que está conectándose con el repositorio.



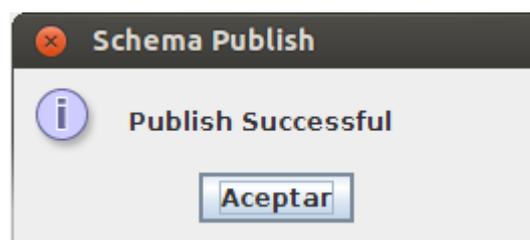
*Ilustración 75: Dialogo de progreso de la conexión a un*

A continuación se solicita la ubicación para guardar la publicación del esquema y los datos el nombre de la fuente de datos, la cual debe ser definida previamente y se explica el proceso en la siguiente sección.



*Ilustración 76: Opciones de guardado para la publicación de un schema*

Una vez completada la información solicitada, se debe seleccionar "Publish", y PSW informa que se publicó correctamente mediante el siguiente mensaje:



*Ilustración 77: Dialogo de publicación exitosa*

### CREACION DE DATA SOURCE EN LA CONSOLA ADMINISTRATIVA.

Para publicar el cubo en PSW se debe previamente crear un Data Source desde la consola administrativa.

Una vez logueado en la consola se debe acceder a la segunda pestaña y allí configurar los datos solicitados.

Database Connections

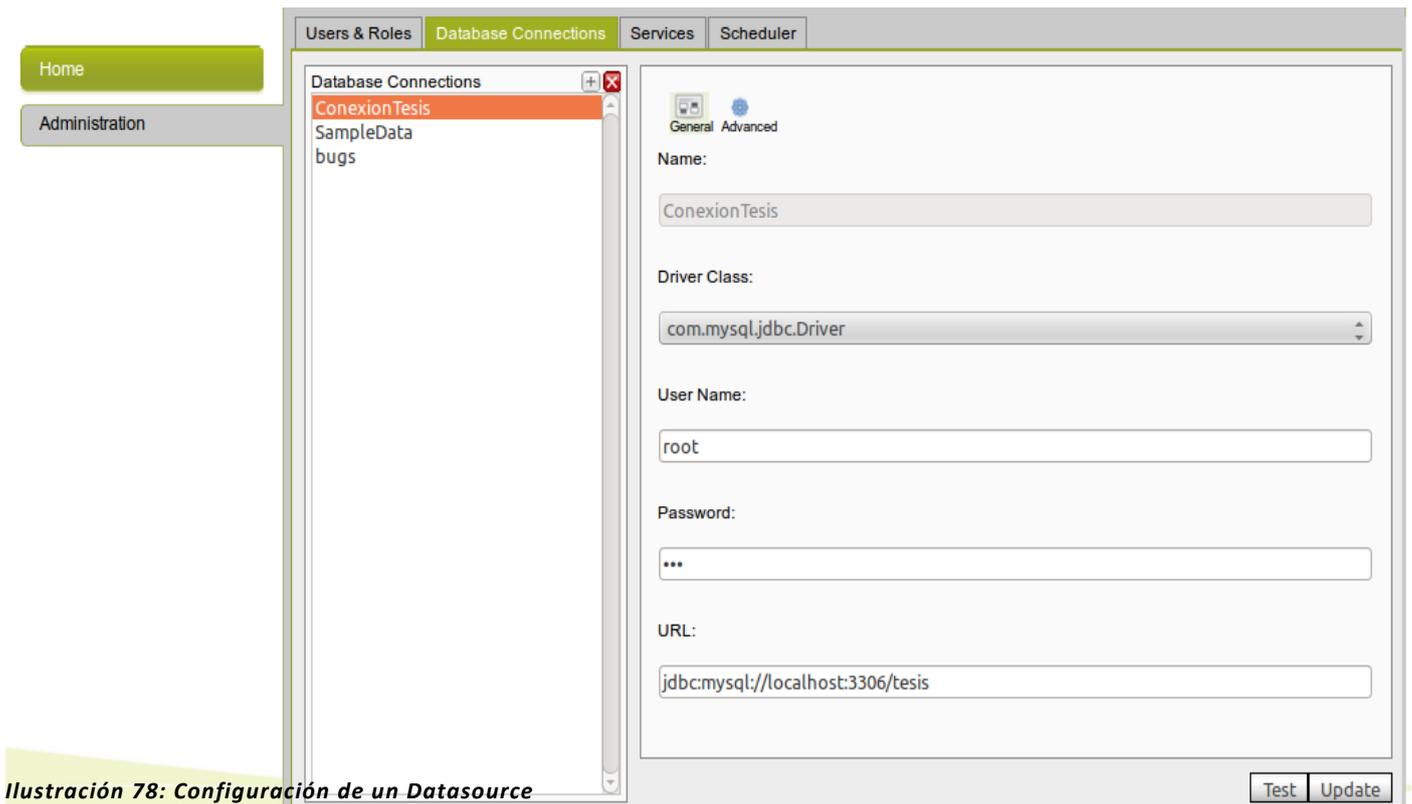


Ilustración 78: Configuración de un Datasource

- **Nombre de la Conexión**
- **Clase de Driver**
- **Nombre de Usuario**
- **Contraseña**
- **Url de la Conexión a la base de datos.**

Si se desea testear la conexión creada se puede hacer click en "Test" en el borde derecho inferior, a lo que la consola debe responder con el siguiente mensaje:

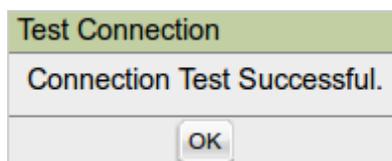
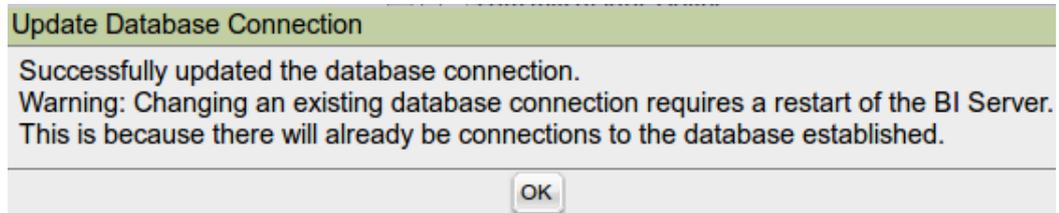


Ilustración 79: Dialogo de conexión exitosa al

Finalmente se selecciona Update para guardar la nueva conexión y la consola informa que ha sido exitosa la actualización y que para ver los cambios reflejados se debe reiniciar BIService.



*Ilustración 80: Dialogo de guardado exitoso de un*

El paso siguiente será la instalación de la herramienta “Saiku Analytics” a través del Marketplace de **Pentaho** para la posterior creación y visualización de los cuadros de mando, tal y como se detalló en dicha sección.

### SAIKU ANALITICS

En este apartado se desarrolla la integración de saiku con el resto de las herramientas. Se detallara su forma de uso así como las diversas características que ofrece al usuario. La instalación de dicha herramienta se encuentra detallada en el apartado Anexo de este documento.

### INTRODUCCION

Saiku es la herramienta que se utilizara para el apoyo en la toma de decisiones. Como se señaló anteriormente, la misma fue seleccionada por un método de “scoring multicriterio” comparándose el desempeño de esta con una herramienta de dashboard (CDE).

A modo de resumen, se puede decir que saiku permite a los usuarios construir dinámicamente sus propias consultas, similares a las que se pueden obtener de los cuadros de mando, con la gran ventaja de ser más simple y flexible ante cambios que los dashboards. A través de una intuitiva interfaz, esta herramienta permite a los clientes combinar medidas y dimensiones según su criterio. De esta manera quien hace uso de la herramienta puede, en cualquier momento y sin necesidad de introducir cambios en la herramienta, generar nuevos indicadores del negocio que le sean útiles.

### DESARROLLO

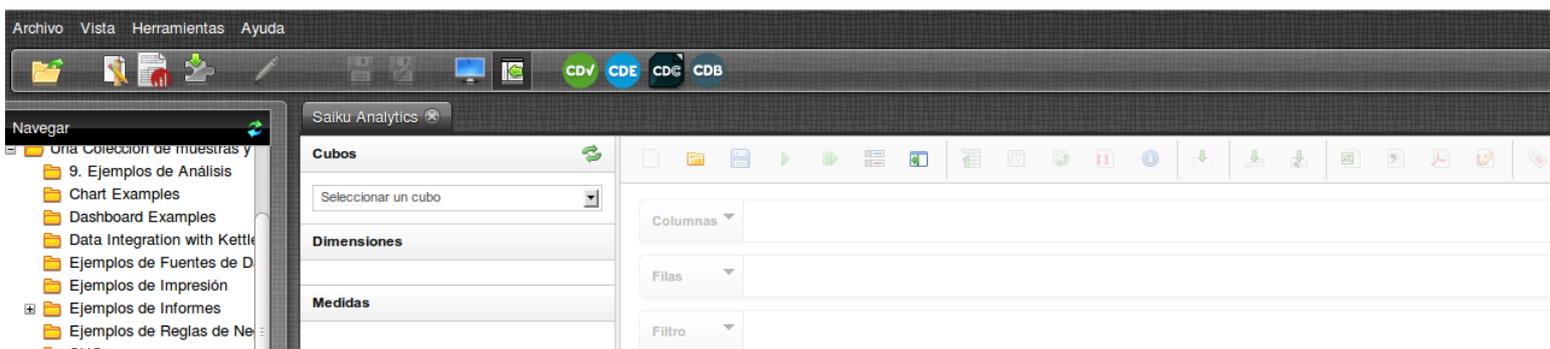
Una vez que se realizó la instalación de la herramienta, se procede a ingresar al Bi Server para hacer uso de la misma.

Ya dentro del sistema, se puede acceder a saiku mediante el icono ubicado en la barra de plugins.



*Ilustración 81: Saiku en la barra de herramientas de*

Al abrir la herramienta, se despliega una pestaña, donde en primera instancia se debe seleccionar la fuente de datos para empezar a operar. Esta fuente de datos (en este caso) viene publicada desde mondrian con permisos por defecto para todos los usuarios.



*Ilustración 82: Interfaz Saiku Analytics*

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Luego se selecciona el cubo deseado, esto desplegara las medidas y dimensiones disponibles en la fuente de datos y se habilitan las demás opciones, que permiten al usuario personalizar e interactuar con su cuadro de mando.

En la columna de la izquierda se pueden visualizar y seleccionar las dimensiones y medidas disponibles. Para generar un indicador valido debe seleccionarse al menos una medida y una o más dimensiones por la que se pueda “filtrar” o agrupar esa medida.

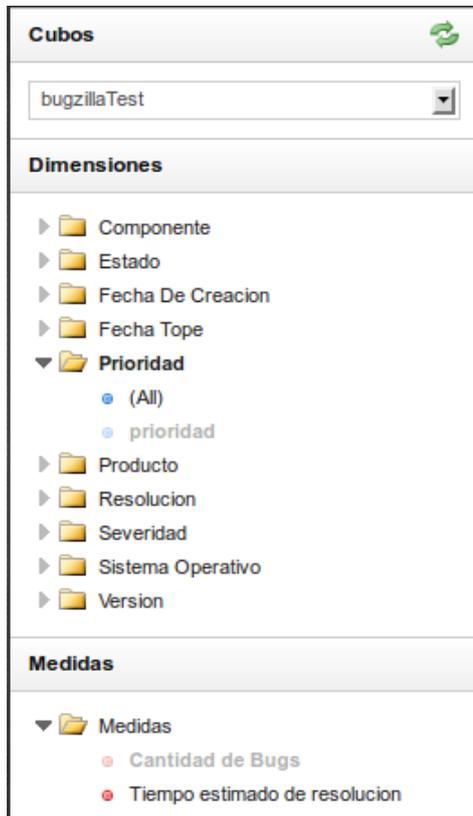


Ilustración 83: Dialogo de guardado exitoso de un

Cabe destacar que la herramienta permite intercambiar el lugar que ocupan las medidas y dimensiones para personalizar su posición en los ejes a la hora de convertir la tabla que se genera en un gráfico. Si ubicamos al elemento como fila, representara el eje de las “X” y como columnas el eje de las “Y”.



prioridad	Cantidad de Bugs
---	446
High	449
Highest	432
Low	418
Lowest	427
Normal	430

Ilustración 84: Representación tabular de consultas

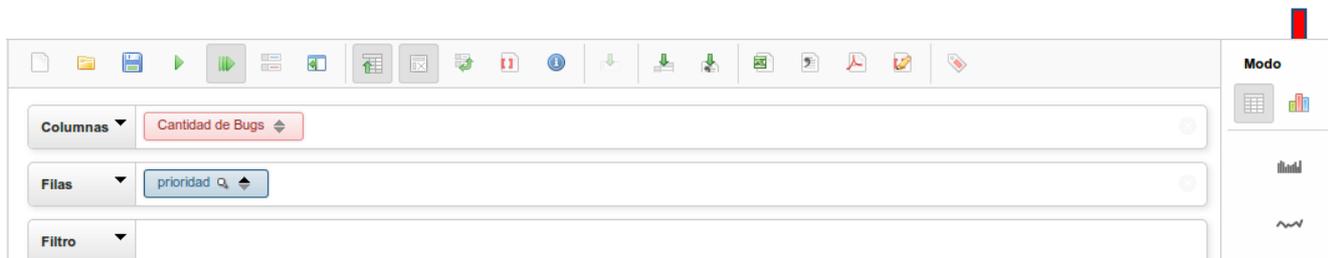
## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

La herramienta permite dentro de cada dimensión seleccionada filtrar por valores disponibles, esto brinda una mayor granularidad a las consultas. Por ejemplo, en el caso de la prioridad, se puede apreciar la posibilidad de seleccionar entre los tipos de prioridades.



Con un simple clic se puede cambiar la salida al modo gráfico, la cual nos permite seleccionar entre diversos tipos. Para ello se debe presionar el botón de **“Modo gráfico”** situado en la el menú derecho de la interfaz de saiku.



Al presionar el botón, **“Modo Gráfico”** se despliegan más opciones permitiendo al usuario seleccionar el tipo de grafico que se ajuste a su consulta.

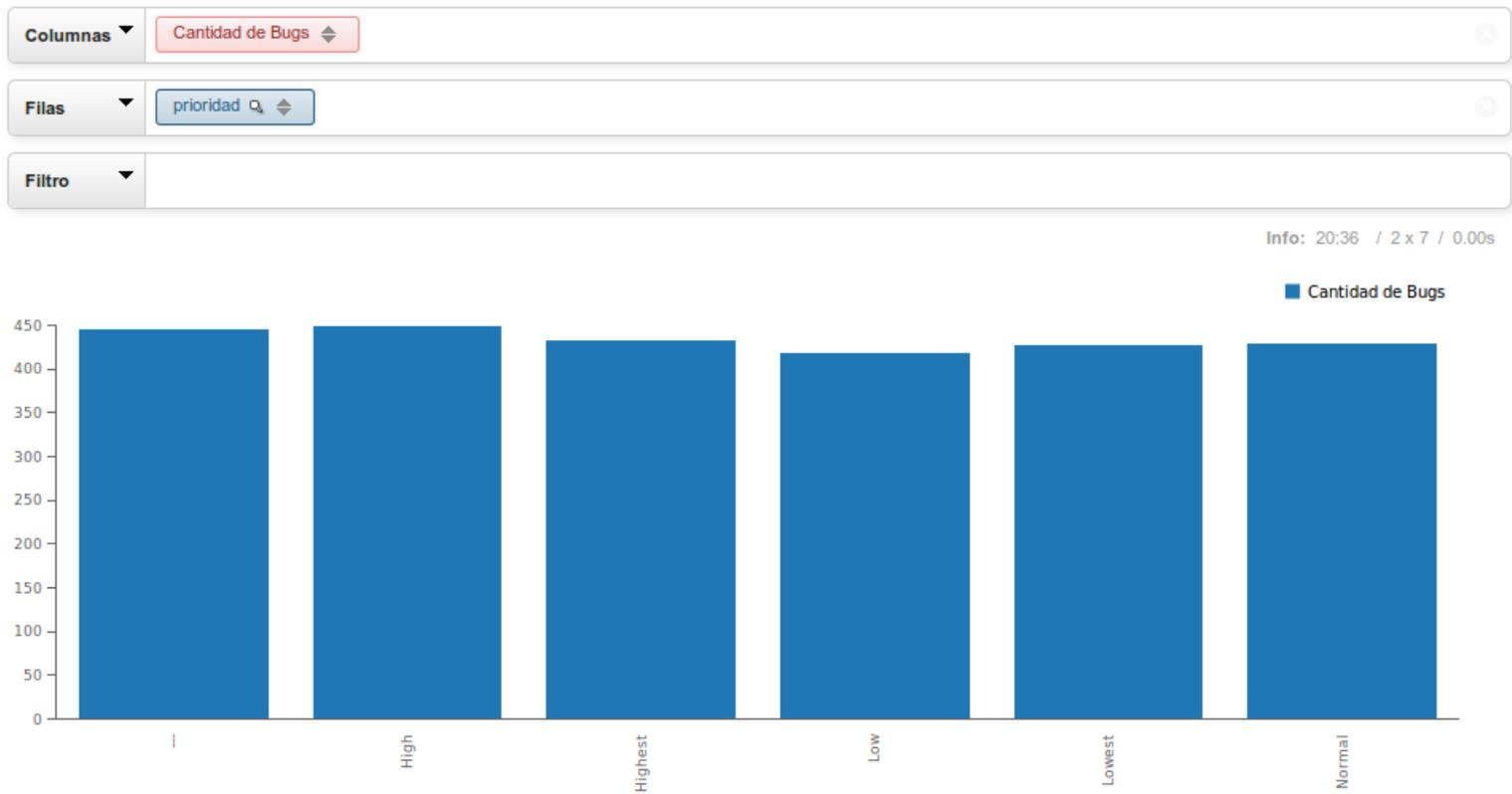


## Trabajo Final De Grado

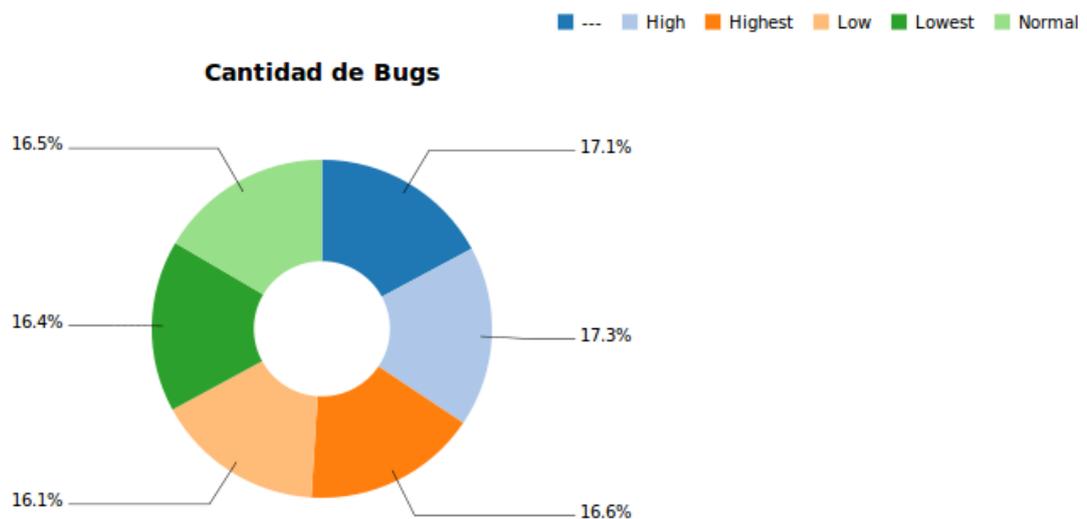
Argañaraz Francisco Jose – Mirizio Esteban Ricardo

La solapa “**Exportar**” permite al usuario guardar ese grafico en un formato que permita transportar ese grafico fuera del contexto de saiku para poder compartir el mismo con otras personas si así lo desea. Los formatos disponibles son JPEG, PNG, PDF, SGV.

A continuación se muestran varios gráficos de la misma consulta elaborados por saiku.



**Ilustración 88: Representación grafico de barras mediante**

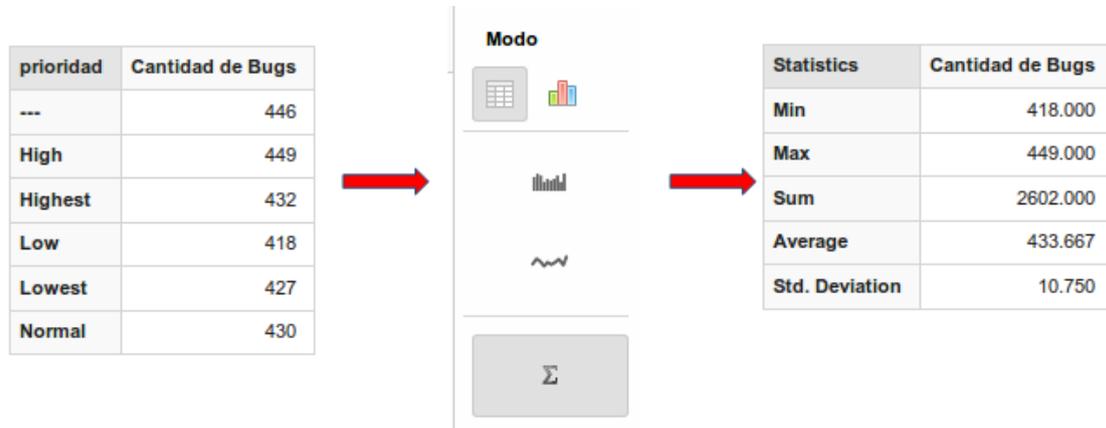


**Ilustración 89: Representación grafico de torta mediante**

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

A través del botón “**Estadísticas**” saiku provee algunos datos interesantes acerca de la consulta. Dicha característica muestra el promedio, el valor mínimo, el valor máximo y la desviación estándar sobre los datos que surgieron de la consulta.



*Ilustración 90: Estadísticas sobre consultas*

Como se puede apreciar saiku es muy fácil de utilizar y dota al usuario con una herramienta potente para el apoyo de toma de decisiones la cual a su vez le da libertad para elaborar sus propios indicadores del negocio.

Saiku posee algunas pequeñas funcionalidades adicionales, las cuales serán detalladas en el apartado de anexo de este documento. A continuación se detallara la implementación de los permisos, y su funcionamiento.

### PERMISOS

En este apartado se detalla cómo y con qué herramientas fue realizada la implementación de permisos en la arquitectura que se desarrolló en el actual trabajo.

### INTRODUCCION

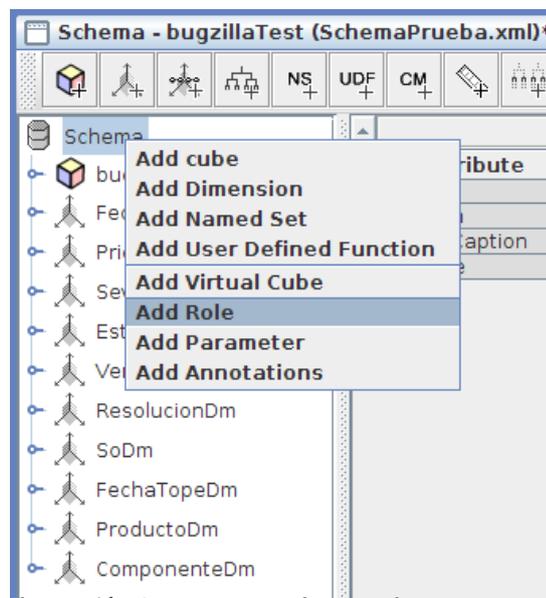
La solución de permisos implementada en este trabajo, combina varias herramientas dando como resultados una estructura flexible que permite independizar a los permisos del dashboard generado. Así, pudiéndose aplicar los mismos directamente sobre las medidas y dimensiones se quita la necesidad de generar estructuras de dashboards particulares para cada rol o usuario. De esta manera el uso de la herramienta para cada usuario es igual para todos, solo difiere en aquellas dimensiones y medidas a las que podrá tener acceso dependiendo del rol que posea. El nivel de granularidad en los permisos que involucra esta solución llega a tal punto que el administrador puede introducirse dentro de las dimensiones y restringir la visión de elementos presentes en dichas dimensiones. Así por ejemplo, si tomamos la dimensión **“Producto”** se puede restringir, el acceso a la información del producto **“Simulador de vuelo”** solo para los usuarios que están a cargo de dicho producto.

### DESAROLLO

Para mejorar la comprensión de este apartado, se llevara a cabo la generación de un permiso particular durante la explicación. Este permiso involucra la limitación de la vista de un determinado producto de la dimensión “Producto” para un rol.

Para la implementación de permisos, primero deben ser generados los mismos en el perfil del cubo desde la herramienta mondrian.

Para esto se debe lanzar la herramienta y abrir el archivo .xml de nuestro cubo. Paso siguiente se hace clic derecho en el schema y se desplegara un menú, en el que se debe seleccionar la opción “**Add Role**”.



*Ilustración 91: Agregar rol a un schema*

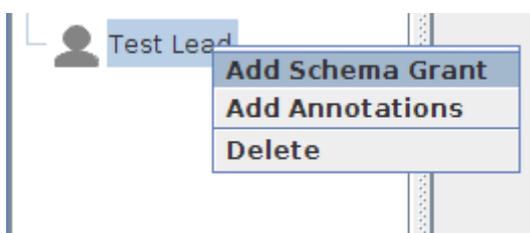
Automáticamente se agrega un nuevo rol al esquema al cual se le debe asignar un nombre.

Attribute	Value
name	Test Lead

*Ilustración 92: Agregar nombre a un rol*

### PERMISOS HACIA EL ESQUEMA

En el siguiente paso se agregan los permisos del rol hacia el schema. Para ello, debe hacerse clic derecho en el icono del nuevo rol que se acaba de crear y luego en la opción “**Add Schema Grant**”.



*Ilustración 93: Agregar permisos hacia un schema*

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

En el cuadro de la derecha se pueden observar las opciones acerca de los permisos hacia el esquema.

Attribute	Value
access	all
	all
	custom
	none
	all_dimensions

Ilustración 94: Permisos hacia un esquema

- **All:** Acceso a todas las dimensiones y medidas de todos los cubos que posea el esquema.
- **Custom:** Permite heredar los permisos solo de aquellos cubos que poseen reglas explícitas para dicho usuario.
- **None:** No permite el acceso a ninguna dimensión ni medida de ningún cubo.
- **All\_dimensions:** El rol tiene acceso a todas las dimensiones y medidas pero necesita permiso explícito sobre el acceso a los cubos.

Para este caso se eligió la opción **custom**, lo que va a permitir que solo tenga acceso a los cubos que explícitamente indiquemos.

### PERMISOS HACIA UN CUBO

El próximo paso es agregar los permisos para un cubo particular del schema. Clic derecho sobre **“Schema Grant”** y seleccionamos la opción **“Cube Grant”**.



Ilustración 95: Agregar Cubo

En el cuadro de la derecha se pueden observar las opciones acerca de los permisos hacia el Cubo

Attribute	Value
access	all
cube	all
	custom
	none

Ilustración 96: Permisos hacia un cubo

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

### Access

- **All:** Acceso a todas las dimensiones y medidas del cubo.
- **Custom:** Permite heredar los permisos solo de aquellos dimensiones de dicho cubo que poseen reglas explicitas para el rol.
- **None:** No permite el acceso a ninguna dimensión ni medida del cubo.

### Cube

Indica el cubo sobre el cual se establecen los permisos.

Para este caso las opciones elegidas fueron:

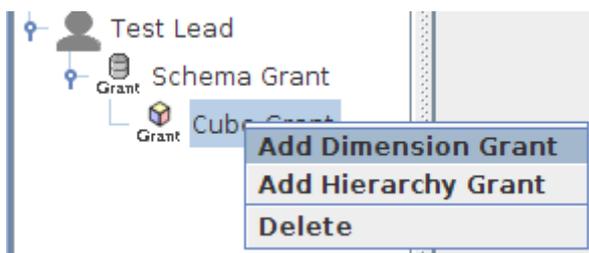
Attribute	Value
access	all
cube	bugzillaTest

*Ilustración 97: Permisos seleccionados para el cubo*

El atributo all, va a habilitar el acceso a todas las dimensiones y medidas de ese cubo a menos que le indiquemos lo contrario.

## PERMISOS HACIA UNA DIMENSION

Luego deben establecer los permisos para una dimensión en particular de cubo particular del schema. Clic derecho sobre **“Cube Grant”** y seleccionamos la opción **“Add Dimension Grant”**.



*Ilustración 98: Agregar permisos hacia una dimensión*

En el cuadro de la derecha se pueden observar las opciones acerca de los permisos hacia la dimensión.

Attribute	Value
access	custom
dimension	all
	custom
	none

*Ilustración 99: Permisos hacia una dimensión*

### Access

- **All:** Acceso a todas las jerarquías de la dimensión declarada.
- **Custom:** Permite heredar los permisos solo de aquellas jerarquias de dicha dimensión que posean reglas explícitas para el rol.
- **None:** No permite el acceso a ninguna jerarquía de la dimensión declarada.

**Nota:** Si un rol posee el tipo de acceso “**ALL**” a un cubo, este va a tener acceso implícitamente a todas las dimensiones del mismo. Si los permisos del esquema están establecidos como “**ALL\_DIMENSIONS**”, esta dimensión heredara el tipo de acceso “**ALL**”

### Dimensión

Indica la dimensión o medida sobre el cual se establecen los permisos.

En este caso se dejara que el acceso sea a todas las dimensiones por lo cual no se incluirá un miembro “**Dimension Grant**”, dejando implícito el acceso a todas las dimensiones del cubo para este rol.

## PERMISOS HACIA UNA JERARQUIA

Por ultimo pueden establecerse permisos sobre una jerarquía en particular de una dimensión del cubo. Clic derecho sobre “**Cube Grant**” y seleccionamos la opción “**Add Hierarchy Grant**”.



*Ilustración 100: Agregar permisos hacia una jerarquía*

En el cuadro de la derecha se pueden observar las opciones acerca de los permisos hacia una jerarquía.

Attribute	Value
access	all
hierarchy	
topLevel	
bottomLevel	

*Ilustración 101: Permisos hacia una jerarquía*

## Access

- **All:** Acceso a todos los miembros y niveles de la jerarquía declarada.
- **Custom:** Permite especificar el “topLevel” y “bottomLevel” de la jerarquía. Solicita obligatoriamente señalar el miembro determinado sobre el cual se va o no dar el acceso.
- **None:** No permite el acceso a ningún miembro ni nivel de la jerarquía declarada.

## Hierarchy

Indica el nombre de la jerarquía sobre la cual se establecen los permisos

### Top Level

Indica el nivel máximo de la jerarquía en la cual, el rol tiene permitido ver los miembros.

### Bottom Level

Indica el nivel mínimo de la jerarquía en la cual, el rol tiene permitido ver los miembros.

En este caso la configuración elegida es la siguiente.

Attribute	Value
access	custom
hierarchy	[Producto.ProductoHy]
topLevel	[Producto.ProductoHy].[ProductoLv]
bottomLevel	[Producto.ProductoHy].[ProductoLv]

Ilustración 102: Permisos seleccionados para la jerarquía

Esta configuración establece que solo se otorgaran permisos a miembros particulares de la jerarquía “Productos”. Esto va a permitir luego establecer el permiso para que el rol solo tenga acceso al miembro de la jerarquía que se eligió.

## PERMISOS HACIA UN MIEMBRO

Por último pueden establecerse permisos sobre un miembro particular en una jerarquía en. Clic derecho sobre “Hierarchy Grant” y seleccionamos la opción “Add Member Grant”.

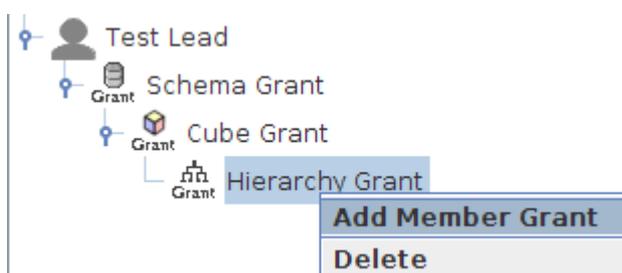


Ilustración 103: Agregar permisos hacia un miembro

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

En el cuadro de la derecha se pueden observar las opciones acerca de los permisos hacia un miembro.

Attribute	Value
access	all
member	

*Ilustración 104: Permisos hacia un miembro*

### Access

- **All:** Acceso al miembro declarada.
- **None:** No permite el acceso al miembro declarado.

### Member

Aquí debe especificarse, a través de una sintaxis, el miembro al que se hace referencia en los permisos.

Sintaxis

**[Dimension.Jerarquia].[Miembro]**

Attribute	Value
access	all
member	[Producto.ProductoHy].[Prod0]

*Ilustración 105: Sintaxis de permisos hacia un miembro*

Finalizada esta tarea, primero se deben guardar los cambios accediendo al menú **File -> Save**, de otra manera los cambios no serán reflejados. Luego deberá publicarse el cubo nuevamente para que los cambios impacten en pentaho. Se recuerda que debe ser llevado a cabo mediante el menú de la herramienta **File -> Publish**.

A continuación se muestra la configuración final para limitar el acceso del rol al miembro "Prod0" de la jerarquía producto.

### USO DEL ROL

Antes de hacer uso de este rol, el mismo debe ser dado de alta desde la consola administrativa de pentaho bajo el mismo nombre con el que fue creado. Debido a que ya se explicó el cómo realizar esto, solo se mostraran algunas imágenes relacionadas al ejemplo.

Una vez creado el rol, se puede comenzar a mapear usuarios con dicho rol. Como puede apreciarse en la imagen, se ha asignado un usuario ya existente al nuevo rol. Podría también haberse creado un nuevo usuario y mapearlo con el nuevo rol.



**Ilustración 106: Mapeo de usuarios a roles**

Resta loguearse con el usuario en el servidor pentaho, ingresar a saiku y hacer una consulta que involucre a la dimensión Producto.

Columns: Cantidad de Bugs, Tiempo estimado de resolucion

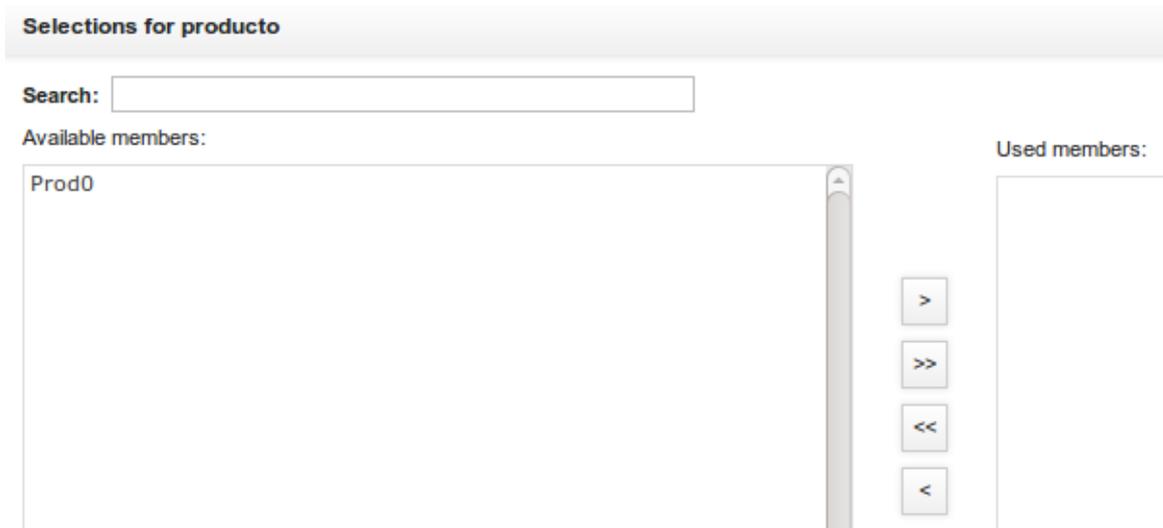
Filas: producto

Filtro:

producto	Cantidad de Bugs	Tiempo estimado de resolucion
Prod0	504	6.020,4

**Ilustración 107: Consulta con restricción de permisos**

En la siguiente imagen se puede apreciar cómo se encuentra limitada la vista solo para el Producto 0, siéndole imposible al usuario visualizar los demás miembros de la jerarquía.

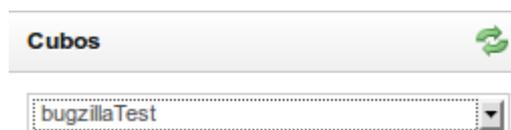


*Ilustración 108: Elementos de la dimensión producto limitado por permisos*

### Nota:

En caso de que los cambios no se reflejen puede ser necesario

- Limpiar la cache del navegador
- Ingresar con un usuario del tipo administrador y ejecutar las siguientes acciones en el menú de herramientas:
  - Ejecutar la opción **Herramientas-> Actualizar -> Vaciar cache de esquemas de mondrian.**
  - Ejecutar la opción **Herramientas-> Actualizar -> CDA Cache**
  - Refrescar los cubos



*Ilustración 109: Refrescar cubos disponibles*

A modo de resumen, se puede decir que para manejar los permisos se deben tener en claros dos aspectos:

- El tipo de acceso **“All”** brinda acceso completo al elemento sobre el que estamos trabajando y los demás elementos que están por debajo de los mismos, a menos que se especifique lo contrario.

Estos elementos pueden ser:

- Schema
- Cubo
- Dimensión
- Jerarquía
- Miembro

Es decir, si se otorgan permisos de acceso ALL a una dimensión, esto permitirá que todos sus jerarquías y miembros sean accesibles también.

- El tipo de acceso **“Custom”** permite especificar aquellos elementos a los cuales queremos darle un trato especial sobre los permisos. Se utiliza para realizar acciones tales como, dar acceso a solo algunos cubos, dimensiones, jerarquías o miembros.

Ej. En un cubo con acceso custom, deberán especificarse cuales dimensiones **SI** van a estar disponibles para el rol.

- Combinado con el tipo de acceso **“None”**, el cual niega el acceso a algún elemento, se puede completar la tarea de restringir los permisos según lo desee el administrador.

Ej. En una dimensión con acceso ALL, se puede restringir el acceso a una jerarquía declarando dicho elemento haciendo uso del tipo **“None”** en el mismo.

## CASOS DE PRUEBA

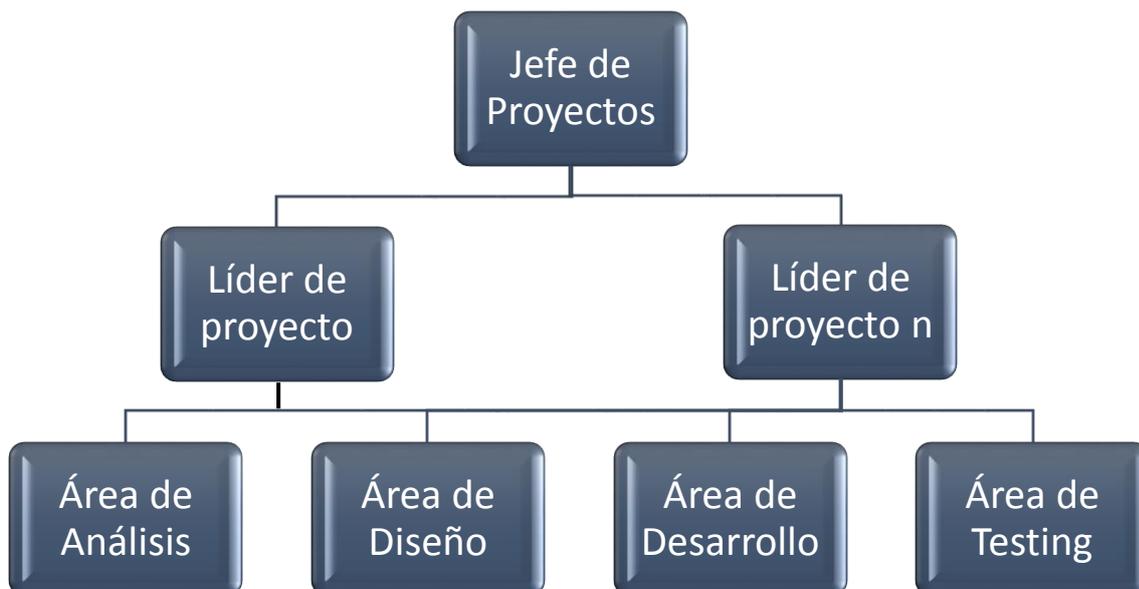
A continuación se demostrara la pertinencia y eficacia de la solución mediante la elaboración de casos de prueba.

Para llevar a cabo el desarrollo de dichos casos de prueba, ha sido necesaria la población ficticia de la fuente de datos de la herramienta Bugzilla simulando un uso real. Dicha tarea fue realizada a través de la codificación de un programa en java, encargado de generar datos aleatorios coherentes, los cuales luego se encarga de insertar en la base de datos de la herramienta.

Dicho script se adjunta en el anexo de este trabajo.

## PLANTEO DEL ESCENARIO

Una organización que se dedica a desarrollar software posee una distribución representada mediante el siguiente organigrama dentro de su departamento de informática.



**Ilustración 110: Distribución funcional grupo de desarrollo**

En dicha organización existe un solo jefe de proyectos, al cual responden los diferentes líderes de proyectos. El jefe de proyecto posee un cargo de nivel directivo, pudiendo el mismo tomar decisiones sin necesidad de consultar al resto. Un proyecto tiene asignado un solo líder y se pueden llevar adelante más de un proyecto simultáneamente. Las personas que trabajan en las diferentes áreas pueden trabajar en más de un proyecto al mismo tiempo.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Una de las formas más comunes de establecer una jerarquía de permisos en una primera instancia es la de mapear dichos permisos según los roles o funciones que se cumplen en la organización.

Siguiendo esta premisa, y suponiendo que se está trabajando en 'X' cantidad de proyectos en la organización, una posible distribución de permisos que se acople a la arquitectura presentada anteriormente podría ser similar a la siguiente:

### **Estructura de Permisos según funciones:**

#### **Jefe\_Proyecto**

El jefe de proyecto tendrá acceso a todas las medidas y dimensiones del cubo.

Dicho rol podrá consultar y disponer de toda la información que ofrece el cubo. El mismo tiene control total sobre la información, esto le permitirá construir una amplia gama de indicadores que puedan responder a sus preguntas para apoyar sus decisiones como jefe.

#### **Lider\_Proyecto\_x**

Tendrá acceso a todas las medidas y dimensiones del producto "x".

Dicho rol se encuentra limitado solo al alcance del producto y por ello todos los componentes que lo integran. Esto permite reducir el dominio de la información accesible al usuario evitando conflictos de intereses entre líderes y apreciaciones subjetivas al tener la posibilidad de comparar con el rendimiento de otros productos que están a cargo de otros líderes. Esto permitirá entre otras cosas transmitir al grupo, en que componentes deben focalizar su esfuerzo.

#### **Desarrollador\_Comp\_x\_Proyecto\_x**

Tendrá acceso a ciertos componentes del producto "x" en los que se encuentren trabajando.

Puede ser pensado para agrupar los usuarios del área de testing y desarrollo. Esto permitirá visualizar de una manera más tangible y en tiempo real el estado y avance de las incidencias en el los componentes software. Quizá para estos usuarios el uso de la arquitectura no sea para fines analíticos como puede serlo para los anteriores roles, sino más bien un uso que le brinde visibilidad al trabajo que están realizando sobre las incidencias.

#### **Usuario\_Proposito\_Gral**

Tendrá acceso a todas las medidas y dimensiones, excepto la dimensión producto y la dimensión componente. Este permiso será destinado a usuarios en los cuales no interese que recaben información específica sobre un producto o componente sino sobre el estado y avance de las incidencias en general.

<b>Caso de prueba: Nº 1</b>	
<b>Nombre:</b> Asignacion_permisos_01	
<b>Descripción:</b> Verificar que un usuario afectado al rol "Lider_proyecto_1" solo pueda acceder al elemento "Producto1" de la dimensión "Productos".	
<b>Prerrequisitos:</b>	<ul style="list-style-type: none"><li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en el rol "Lider_proyecto_1".</li><li>- El rol "Lider_proyecto_1" debe haber sido dado de alta.</li><li>- Deben haber sido asignados los permisos correspondientes al rol "Lider_proyecto_1" desde la herramienta Mondrian Schema Workbench.</li></ul>
<b>Datos de Prueba:</b>	<ul style="list-style-type: none"><li>- Usuario: UsuarioDemo_01</li><li>- Password: ab12cd23</li></ul>
<b>Pasos:</b>	<ul style="list-style-type: none"><li>- El usuario se loguea en pentaho con sus credenciales</li><li>- Presiona el botón que inicia "Saiku Reporting"</li><li>- Elige el cubo Bugzilla</li><li>- Despliega el menú de medidas y dimensiones en el panel izquierdo</li><li>- Elige la dimensión Producto</li><li>- Elige alguna de las medidas disponibles</li><li>- Realiza la consulta que desee</li></ul>
<b>Resultado Esperado:</b> El usuario afectado al rol "Lider_proyecto_1" solo accede al elemento "Producto1" de la dimensión "Productos".	
<b>Resultado Actual:</b> El usuario afectado al rol "Lider_proyecto_1" solo accede al elemento "Producto1" de la dimensión "Productos".	
<b>Estado:</b> Aprobado	
<b>Comentarios:</b>	

<b>Caso de prueba: Nº 2</b>	
<b>Nombre:</b> Asignacion_de_permisos_02	
<b>Descripción:</b> Verificar que un usuario no autorizado no pueda ejecutar satisfactoriamente una consulta de una dimensión a la cual no posee permisos.	
<b>Prerrequisitos:</b>	
	<ul style="list-style-type: none"><li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en algún rol que permita tener acceso al sistema.</li><li>- El rol que permite el acceso al sistema debe haber sido dado de alta.</li><li>- Deben haber sido asignados los permisos correspondientes al rol que permite la autenticación en el sistema desde la herramienta Mondrian Schema Workbench excluyéndole el acceso a la medida Producto → "Producto1".</li><li>- Debe haber sido configurado el "Cron-Job" que permite que se ejecute automáticamente cada cierto intervalo de tiempo la tarea realizada por Keetle encargada de extraer los datos de la base "Bugzilla" para su posterior transformación.</li></ul>
<b>Datos de Prueba:</b>	
	<ul style="list-style-type: none"><li>- Usuario: UsuarioDemo_02</li><li>- Password: 012ghi654</li></ul>
<b>Pasos:</b>	
	<ul style="list-style-type: none"><li>- El usuario se loguea en pentaho con sus credenciales</li><li>- Presiona el botón que inicia "Saiku Reporting"</li><li>- Elige el cubo Bugzilla.</li><li>- Despliega el menú de medidas y dimensiones en el panel izquierdo</li><li>- Ingresa al filtro de la medida Producto</li><li>- El usuario no visualiza el miembro: "Producto1"</li></ul>
<b>Resultado Esperado:</b> El usuario no puede realizar la consulta deseada ya que no posee permisos para visualizar el miembro "Producto1" de la dimensión Producto.	
<b>Resultado Actual:</b> El usuario no puede realizar la consulta deseada ya que no posee permisos para visualizar el miembro "Producto1" de la dimensión Producto.	
<b>Estado:</b> Aprobado	
<b>Comentarios:</b> -	

<b>Caso de prueba: Nº 3</b>	
<b>Nombre:</b> Asignacion_permisos_02	
<b>Descripción:</b> Verificar que un usuario afectado al rol "Jefe_Proyecto" puede acceder a toda la información disponible en el cubo (dimensiones y medidas).	
<b>Prerrequisitos:</b>	<ul style="list-style-type: none"><li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en el rol "Jefe_Proyecto".</li><li>- El rol "Jefe_Proyecto" debe haber sido dado de alta.</li><li>- Deben haber sido asignados los permisos correspondientes al rol "Jefe_Proyecto" desde la herramienta Mondrian Schema Workbench.</li></ul>
<b>Datos de Prueba:</b>	<ul style="list-style-type: none"><li>- Usuario: JefeDemo</li><li>- Password: ab32cd10</li></ul>
<b>Pasos:</b>	<ul style="list-style-type: none"><li>- El usuario se loguea en pentaho con sus credenciales</li><li>- Presiona el botón que inicia "Saiku Reporting"</li><li>- Elige el cubo Bugzilla</li><li>- Despliega el menú de medidas y dimensiones en el panel izquierdo</li><li>- Realiza una consulta seleccionando las 8 dimensiones disponibles y la medida "Cantidad de Bugs".</li><li>- Realiza una consulta seleccionando las 8 dimensiones disponibles y la medida "Tiempo de Resolución".</li></ul>
<b>Resultado Esperado:</b> Las dos consultas realizadas se muestran exitosamente	
<b>Resultado Actual:</b> Las dos consultas realizadas se mostraron exitosamente	
<b>Estado:</b> Aprobado	
<b>Comentarios:</b>	

<b>Caso de prueba: Nº 4</b>	
<b>Nombre:</b> Ejecucion_consulta_01	
<b>Descripción:</b> Verificar que un usuario afectado a algún rol pueda realizar y visualizar una consulta en el sistema en forma de grafico de barras.	
<b>Prerrequisitos:</b>	<ul style="list-style-type: none"><li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en algún rol que permita tener acceso al sistema.</li><li>- El rol que permite el acceso al sistema debe haber sido dado de alta.</li><li>- Deben haber sido asignados los permisos correspondientes al rol que permite la autenticación en el sistema desde la herramienta Mondrian Schema Workbench.</li></ul>
<b>Datos de Prueba:</b>	<ul style="list-style-type: none"><li>- Usuario: UsuarioDemo</li><li>- Password: 012ghi654</li></ul>
<b>Pasos:</b>	<ul style="list-style-type: none"><li>- El usuario se loguea en pentaho con sus credenciales</li><li>- Presiona el botón que inicia "Saiku Reporting"</li><li>- Elige el cubo Bugzilla Despliega el menú de medidas y dimensiones en el panel izquierdo</li><li>- Realiza una consulta combinando alguna de las medidas y dimensiones que le son accesibles.</li><li>- Hace clic en el botón de "Modo Grafico" ubicado en el margen derecho.</li><li>- Selecciona el tipo de grafico "Grafico de barras"</li></ul>
<b>Resultado Esperado:</b> Las consulta es realizada exitosamente representada en forma de grafico de barras.	
<b>Resultado Actual:</b> Las consulta es realizada exitosamente representada en forma de grafico de barras.	
<b>Estado:</b> Aprobado	
<b>Comentarios:</b>	

<b>Caso de prueba: Nº 5</b>	
<b>Nombre:</b> Ejecucion_consulta_02	
<b>Descripción:</b> Verificar que un usuario afectado a algún rol pueda realizar y visualizar una consulta en el sistema y que esta esté actualizada con la información en tiempo real.	
<b>Prerrequisitos:</b>	<ul style="list-style-type: none"><li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en algún rol que permita tener acceso al sistema.</li><li>- El rol que permite el acceso al sistema debe haber sido dado de alta.</li><li>- Deben haber sido asignados los permisos correspondientes al rol que permite la autenticación en el sistema desde la herramienta Mondrian Schema Workbench.</li><li>- Debe haber sido configurado el "Cron-Job" que permite que se ejecute automáticamente cada cierto intervalo de tiempo la tarea realizada por Keetle encargada de extraer los datos de la base "Bugzilla" para su posterior transformación.</li></ul>
<b>Datos de Prueba:</b>	<ul style="list-style-type: none"><li>- Usuario: UsuarioDemo</li><li>- Password: 012ghi654</li></ul>
<b>Pasos:</b>	<ul style="list-style-type: none"><li>- El usuario se loguea en pentaho con sus credenciales</li><li>- Presiona el botón que inicia "Saiku Reporting"</li><li>- Elige el cubo Bugzilla</li><li>- Despliega el menú de medidas y dimensiones en el panel izquierdo</li><li>- Realiza una consulta combinando alguna de las medidas y dimensiones que le son accesibles.</li></ul>
<b>Resultado Esperado:</b> La consulta es realizada exitosamente y posee los últimos datos cargados en Bugzilla.	
<b>Resultado Actual:</b> La consulta es realizada exitosamente y posee los últimos datos cargados en Bugzilla.	
<b>Estado:</b> Aprobado	
<b>Comentarios:</b>	

Caso de prueba: Nº 6
<b>Nombre:</b> Ejecucion_con_filtros_01
<b>Descripción:</b> Verificar que un usuario autorizado pueda ejecutar satisfactoriamente una consulta personalizada con multiples filtros.
<b>Prerrequisitos:</b> <ul style="list-style-type: none"> <li>- El usuario debe contar con las credenciales necesarias para autenticarse en el sistema mediante un usuario que se encuentre mapeado en algún rol que permita tener acceso al sistema.</li> <li>- El rol que permite el acceso al sistema debe haber sido dado de alta.</li> <li>- Deben haber sido asignados los permisos correspondientes al rol que permite la autenticación en el sistema desde la herramienta Mondrian Schema Workbench.</li> <li>- Debe haber sido configurado el "Cron-Job" que permite que se ejecute automáticamente cada cierto intervalo de tiempo la tarea realizada por Keetle encargada de extraer los datos de la base "Bugzilla" para su posterior transformación.</li> </ul>
<b>Datos de Prueba:</b> <ul style="list-style-type: none"> <li>- Usuario: UsuarioDemo</li> <li>- Password: 012ghi654</li> </ul>
<b>Pasos:</b> <ul style="list-style-type: none"> <li>- El usuario se loguea en pentaho con sus credenciales</li> <li>- Presiona el botón que inicia "Saiku Reporting"</li> <li>- Elige el cubo Bugzilla.</li> <li>- Despliega el menú de medidas y dimensiones en el panel izquierdo</li> <li>- Realiza una consulta combinando:               <ul style="list-style-type: none"> <li>- Medidas                   <ul style="list-style-type: none"> <li>- Cantidad de bugs</li> </ul> </li> <li>- Dimensiones                   <ul style="list-style-type: none"> <li>- Año de creación                       <ul style="list-style-type: none"> <li>- Filtrar → "2012"</li> </ul> </li> <li>- Prioridad                       <ul style="list-style-type: none"> <li>- Filtrar → "Highest"</li> </ul> </li> <li>- Sistema Operativo                       <ul style="list-style-type: none"> <li>- Filtrar → "Linux"</li> </ul> </li> </ul> </li> <li>- Severidad</li> </ul> </li> </ul>
<b>Resultado Esperado:</b> La consulta es realizada exitosamente y posee los últimos datos cargados en Bugzilla de Bugs del Año "2012", con Prioridad "Highest", Sistema operativo "Linux" y de todas las Severidades.
<b>Resultado Actual:</b> La consulta es realizada exitosamente y posee los últimos datos cargados en Bugzilla de Bugs del Año "2012", con Prioridad "Highest", Sistema operativo "Linux" y de todas las Severidades.
<b>Estado:</b> Aprobado
<b>Comentarios:</b> -

### CONCLUSION

Finalmente se ha llegado a la conclusión del presente proyecto, con excelentes resultados obtenidos y una gran cantidad de experiencia adquirida, logrando como recompensa la satisfacción de los tesistas y el grupo de Investigación y Desarrollo del Instituto Universitario Aeronáutico, quienes han demostrado una clara aceptación del trabajo final de grado. En una primera instancia se produjo el acercamiento al grupo de I+D, alcanzando una apreciación general sobre su realidad actual y la situación problemática a la cual se enfrentaban y comparando sus características con grupos de desarrollo genéricos para lograr un producto/sistema final adaptable a distintos entornos. A partir de ello, se realizó una planificación para llevar a cabo la ejecución del proyecto, estableciendo objetivos y el alcance. Durante el Marco Teórico se analizaron, estudiaron y dominaron los conceptos necesarios para plantear una solución a la problemática existente. Para la implementación del proyecto se utilizaron diversas herramientas open source que permitieron cumplir con los objetivos planteados al comienzo del presente trabajo los cuales fueron corroborados a través del uso de casos de prueba.

## ANEXO

## PENTAHO



Pentaho es una suite de productos de BI (Inteligencia de Negocios) con la cual se puede realizar análisis, minería y reporte de datos, como también procesos ETL (extracción, transformación y carga) entre otros.

A continuación se procede a analizar el proceso de instalación de “Pentaho Business Intelligence Server” o “BiServer”, el cual es el servidor de la suite, y la consola administrativa que viene incluida en dicho paquete.

---

**INSTALACIÓN**

1. Como primera medida se debe descargar el software. Se recomienda el siguiente link:

- <http://sourceforge.net/projects/pentaho/>

La versión utilizada para realizar el presente trabajo de grado es la 4.8.0 estable.

2. En el archivo descargado se pueden encontrar dos carpetas:

```
usuario@tfg:~/Descargas$ tar -ztf biserver-ce-4.8.0-stable.tar.gz --exclude '*/*'
administration-console/
biserver-ce/
```

- **Administration Console:** La consola administrativa se utiliza para la creación de usuarios, roles, fuentes de datos, etc.
- **Biserver:** Es el servidor propiamente dicho, es administrado desde la consola previamente mencionada.

3. Se descomprimen los contenidos del archivo obteniendo las dos carpetas antes mencionadas.

4. Para iniciar el servidor simplemente se navega hasta su carpeta, y se ejecuta el script “start-pentaho.sh”

```
usuario@tfg:~$ cd pentaho/biserver-ce/
/home/usuario/pentaho/biserver-ce
usuario@tfg:~/pentaho/biserver-ce$ ./start-pentaho.sh
/home/usuario/pentaho/biserver-ce
/home/usuario/pentaho/biserver-ce
DEBUG: Using PENTAHO_JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-7-oracle
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-7-oracle/bin/java
Using CATALINA_BASE:   /home/usuario/pentaho/biserver-ce/tomcat
Using CATALINA_HOME:   /home/usuario/pentaho/biserver-ce/tomcat
Using CATALINA_TMPDIR: /home/usuario/pentaho/biserver-ce/tomcat/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /home/usuario/pentaho/biserver-ce/tomcat/bin/bootstrap.jar
usuario@tfg:~/pentaho/biserver-ce$ █
```

NOTA: Para que Pentaho se ejecute de manera satisfactoria Java debe estar instalado en el sistema, con las variables de entorno correctamente definidas.

```
usuario@tfg:~/pentaho/administration-console$ java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b15)
Java HotSpot(TM) Client VM (build 23.25-b01, mixed mode)
```

```
usuario@tfg:/etc$ cat environment
PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games"
PENTAHO_JAVA_HOME=/usr/lib/jvm/java-7-oracle
```

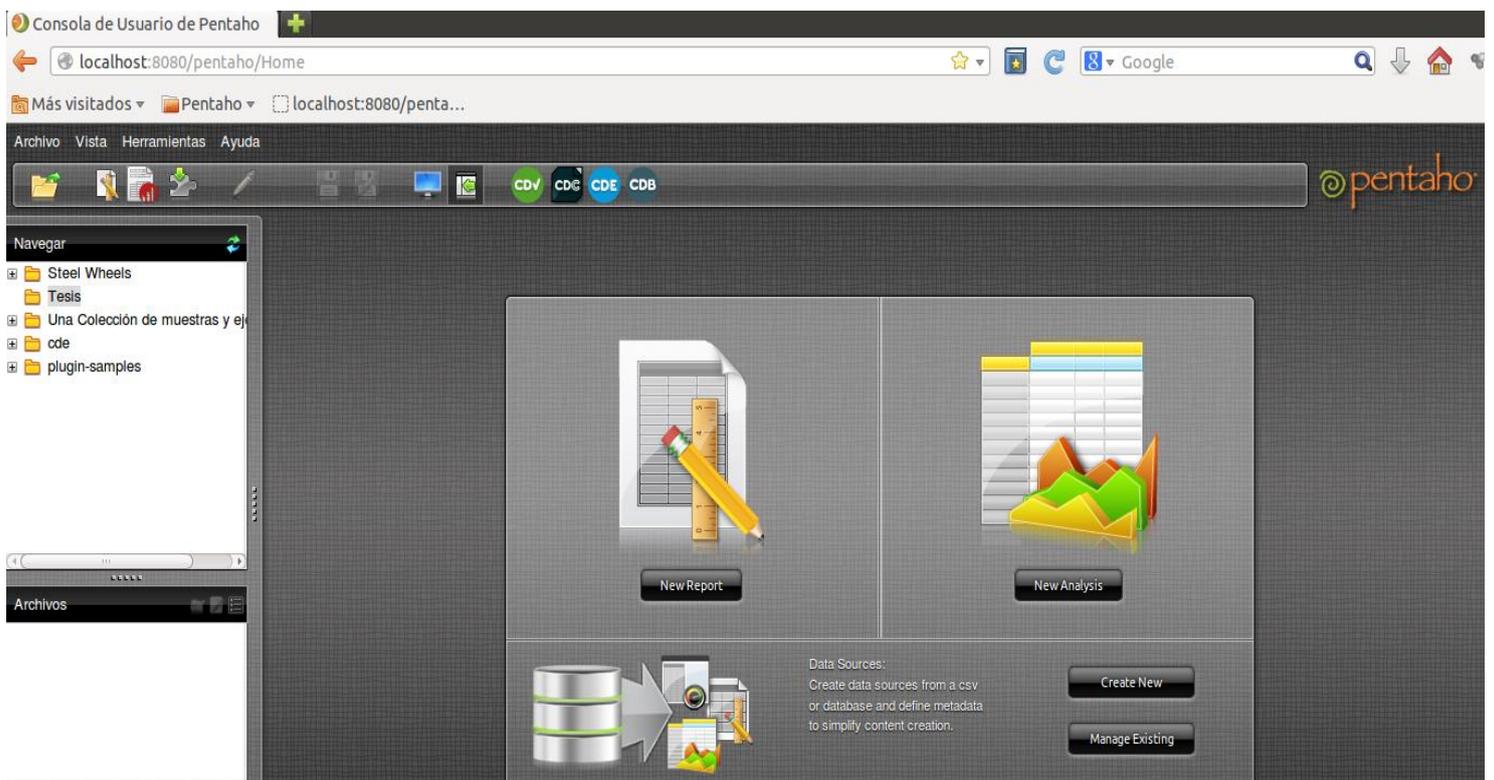
## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

5. Solo resta acceder a la interfaz mediante algún navegador, ingresando a la siguiente URL:
  - <http://localhost:8080/pentaho>
6. Una vez allí se procede a loguearse en el servidor, utilizando el usuario y contraseñas por defecto:
  - **Usuario:** "Joe"
  - **Password:** "password"



7. Una vez logueado ya está listo para ser utilizado:



8. A continuación se procede a iniciar la consola administrativa para realizar las configuraciones necesarias.

Para esto se navega hasta el directorio y se ejecuta el script “start-pac.sh”:

```
usuario@tfg:~$ cd pentaho/administration-console/  
/home/usuario/pentaho/administration-console  
usuario@tfg:~/pentaho/administration-console$ ./start-pac.sh  
/home/usuario/pentaho/administration-console  
/home/usuario/pentaho/administration-console  
DEBUG: Using PENTAHO_JAVA_HOME  
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-7-oracle  
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-7-oracle/bin/java  
11:14:43,814 INFO [JettyServer] Console is starting  
11:14:43,938 INFO [/] org.pentaho.pac.server.BrowserLocaleServlet-16378580: init  
11:14:43,943 INFO [/] org.pentaho.pac.server.DefaultConsoleServlet-6339859: init  
11:14:43,945 INFO [/] org.pentaho.pac.server.PacServiceImpl-19755963: init  
11:14:43,946 INFO [/] org.pentaho.pac.server.SchedulerServiceImpl-29909967: init  
11:14:43,948 INFO [/] org.pentaho.pac.server.SolutionRepositoryServiceImpl-30489209: init  
11:14:43,949 INFO [/] org.pentaho.pac.server.SubscriptionServiceImpl-32474996: init  
11:14:43,950 INFO [/] org.pentaho.pac.server.common.HibernateConfigurationServiceImpl-292427: init  
11:14:43,950 INFO [/] org.pentaho.pac.server.common.JdbcDriverDiscoveryServiceImpl-24683752: init  
11:14:44,018 INFO [JettyServer] Console is now started. It can be accessed using http://tfg:8099 or http://127.0.1.1:8099
```

9. En la anterior imagen se puede apreciar que la consola informa de la url a la cual se debe acceder para utilizar la consola.

En un navegador entonces se abre la url y se utilizan los datos por defecto de autenticación:

- Usuario: “admin”
- Contraseña: “password”



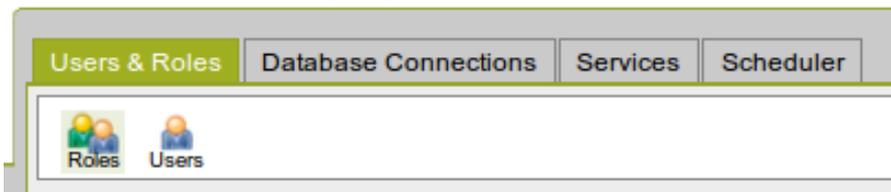
## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

- Una vez logueado en la consola, si BIServer está corriendo al posicionarse sobre el icono de las computadoras el sistema debería informarle que BIServer se encuentra disponible del siguiente modo:



- En la pestaña "Administration" se puede luego configurar roles y usuarios, conexiones a bases de datos, servicios y un planificador de tareas.



- Por ultimo para parar el servidor BiServer se navega hasta su directorio y se ejecuta el script "stop-pentaho.sh"

```
usuario@tfg:~/pentaho/biserver-ce$ ./stop-pentaho.sh
/home/usuario/pentaho/biserver-ce
/home/usuario/pentaho/biserver-ce
DEBUG: Using PENTAHO_JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-7-oracle
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-7-oracle/bin/java
Using CATALINA_BASE: /home/usuario/pentaho/biserver-ce/tomcat
Using CATALINA_HOME: /home/usuario/pentaho/biserver-ce/tomcat
Using CATALINA_TMPDIR: /home/usuario/pentaho/biserver-ce/tomcat/temp
Using JRE_HOME: /usr
Using CLASSPATH: /home/usuario/pentaho/biserver-ce/tomcat/bin/bootstrap.jar
```

- Igualmente para parar la consola administrativa se navega hasta su directorio y se ejecuta el script "stop-pac.sh"

```
usuario@tfg:~$ cd pentaho/administration-console/
/home/usuario/pentaho/administration-console
usuario@tfg:~/pentaho/administration-console$ ./stop-pac.sh
/home/usuario/pentaho/administration-console
/home/usuario/pentaho/administration-console
DEBUG: Using PENTAHO_JAVA_HOME
DEBUG: _PENTAHO_JAVA_HOME=/usr/lib/jvm/java-7-oracle
DEBUG: _PENTAHO_JAVA=/usr/lib/jvm/java-7-oracle/bin/java
```

## INSTALACIÓN

La instalación de Pentaho Data Integration, o Kettle consta simplemente en descargar el paquete y ejecutar el script de la herramienta que se desea utilizar.

1. Se procede a descargar el paquete. Se recomienda el siguiente link:

- <http://sourceforge.net/projects/pentaho/files/Data%20Integration/>

La versión utilizada para realizar el presente trabajo de grado es la 4.4.0 estable.

**pdi-ce-4.4.0-stable.tar.gz**

2. Una vez descomprimido se navega hasta la carpeta para ver su contenido:

```

usuario@tfg:~/pentaho/data-integration$ ls
Carte.bat          import.sh         plugins           set-pentaho-env.sh
carte.sh           Kitchen.bat      pwd              simple-jndi
Data Integration 32-bit.app kitchen.sh        README_INFOBRIGHT.txt Spoon.bat
Data Integration 64-bit.app launcher        README_LINUX.txt   spoon.ico
docs              lib              README_OSX.txt     spoon.png
Encr.bat          libext           README_UNIX_AS400.txt run_kettle_cluster_example.bat ui
encr.sh           libswt          runSamples.sh     ui
generateClusterSchema.sh null             samples
Import.bat        Pan.bat          set-pentaho-env.bat
import-rules.xml  pan.sh
usuario@tfg:~/pentaho/data-integration$

```

Kettle se encuentra compuesto por tres módulos, los cuales pueden ser ejecutados por línea de comando simplemente llamándolos ya que son scripts:

- **Spoon**  
Es una interfaz de usuario gráfica que permite diseñar transformaciones y trabajos que pueden ser ejecutados con los otros módulos de Kettle: Pan y Kitchen
- **Pan**  
Pan es un motor de transformación de datos capaz de ejecutar una gran cantidad de funciones como leer, manipular y escribir datos desde varias fuentes de datos.
- **Kitchen**  
Es el programa encargado de ejecutar los trabajos (Jobs) diseñados en Spoon en extensión XML o en un repositorio de bases de datos.  
Usualmente los trabajos son programados por lotes para que sean ejecutados en intervalos regulares de tiempo.
- **Carte**  
Es un pequeño servidor web que permite la ejecución remota de transformaciones y jobs.

---

### INSTALACIÓN

El proceso de instalación consta simplemente en descargar un archivo de formato zip y descomprimirlo.

Se puede descargar PSW accediendo a la siguiente dirección:

- <http://sourceforge.net/projects/mondrian/files/schema%20workbench/>

Una vez allí se puede seleccionar la última versión estable, al día de hoy es la 3.6.1-stable por lo que se selecciona para el presente trabajo el archivo psw-ce-3.6.1.zip.

Una vez descargado, se procede a descomprimirlo, obteniendo un directorio con la siguiente estructura:

```
usuario@tfg:~/pentaho/schema-workbench$ ls
cpappend.bat  ivy.xml          mondrian.properties  Schema_Workbench.app  workbench.bat
demo          lib              PentahoSchemaWorkbench480-GA-licenses.html  set-pentaho-env.bat   workbench.sh
doc           LICENSE.html    plugins              set-pentaho-env.sh
drivers       log4j.xml       readme.txt           src
```

---

### CONFIGURACIÓN

Se debe añadir al directorio “~/schema-workbench/drivers/” el driver para conectarse a la base de datos, en el caso de este proyecto se utilizó MySQL por lo que el contenido final de dicho directorio se puede apreciar a continuación:

```
usuario@tfg:~/pentaho/schema-workbench/drivers$ ls
mysql-connector-java-5.1.26-bin.jar  readme.txt
```

El próximo paso a realizar es el mapeo de roles entre mondrian y la consola administrativa de pentaho.

El mapeador de roles conecta restricciones de roles de usuarios definidos en un Schema mondrian a roles de usuario definidos en la consola administrativa de pentaho. Esto permite a los desarrolladores BI definir controles de accesos en un solo lugar. Dicho mapeador se configura a través de un archivo xml ubicado en el archivo */pentaho-solutions/system/pentahoObjects.spring.xml*.

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Existen tres implementaciones posibles del mapeador:

- **One to One User Roler Mapper:** Mapea cada nombre de rol de la consola de administración a roles definidos en un schema ROLAP.
- **Sample Lookup Map Roler Mapper :** Este mapeador provee una tabla de “traducción” utilizando un elemento <map> , que permite asociar roles de la consola de administración con roles de un Schema ROLAP.
- **Sample User Session Roler Mapper :** Este mapeador recupera los roles de esquema ROLAP de una variable de sesión HTTP.

En nuestra solución se utilizara el método “**One to one User Role Mapper**”. Para su implementación debemos editar el archivo *pentahoObjects.spring.xml* agregando la siguiente línea:

```
<bean id="Mondrian-UserRoleMapper"
      name="Mondrian-SampleUserSession-UserRoleMapper"
      class="org.pentaho.platform.plugin.action.mondrian.mapper.
      MondrianUserSessionUserRoleListMapper"
      scope="singleton">
    <property name="sessionProperty" value="MondrianUserRoles" />
</bean>
```

## INDICE DE ILUSTRACIONES

Ilustración 1: Saiku reporting (Front-End) .....	35
Ilustración 2: Mondrian Schema Workbench.....	36
Ilustración 3: Visualizacion de una consulta mediante "Saiku Reporting" .....	37
Ilustración 4: Combinación de medidas y dimensiones .....	38
Ilustración 5: Selección de dimensiones .....	38
Ilustración 6: Selección cubo.....	39
Ilustración 7: Selección de dimensión y medidas.....	39
Ilustración 8: Ejecución consulta modo "Chart".....	39
Ilustración 9: Presentación Community Dashboard Edition.....	41
Ilustración 10: Configuración Front-End CDE.....	42
Ilustración 11: Configuración de un componente CDE.....	43
Ilustración 12: Logueo consola pentaho .....	44
Ilustración 13: Visualizacion Dashboard.....	44
Ilustración 14: Dashboard con interacción para el usuario .....	45
Ilustración 15: Propiedades del dashboard.....	46
Ilustración 16: Permisos sobre el dashboard .....	46
Ilustración 17: Tabla ponderación multicriterio.....	48
Ilustración 18: Estado de un bug.....	50
Ilustración 19: Pantalla de inicio Bugzilla .....	51
Ilustración 20: Carga de un bug.....	52
Ilustración 21: Explorador de bugs.....	54
Ilustración 22: Visualización de bugs en el explorador .....	54
Ilustración 23: Bugzilla, búsqueda simple .....	56
Ilustración 24: Bugzilla, búsqueda avanzada.....	56
Ilustración 25: Modificación de un bug.....	57
Ilustración 26: Tipos de reportes en bugzilla .....	58
Ilustración 27: Selección de ejes en reportes.....	58
Ilustración 28: Reporte generado por bugzilla .....	58
Ilustración 29: DER, base auxiliar .....	64
Ilustración 30: Trabajo Kettle completo.....	70
Ilustración 31: Programación de trabajos Kettle.....	71
Ilustración 32: Script "Dropfks" .....	71
Ilustración 33: Script "Fill_BugsDB" .....	72
Ilustración 34: Script "Update_NullDates" .....	72
Ilustración 34: Script "Update_Null_Res .....	73
Ilustración 35: "Fill" tablas externas .....	73
Ilustración 36: Transformación "AddFks".....	74
Ilustración 37: Ejecución trabajo kettle por línea de comandos .....	77

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Ilustración 38: Servicio “Cron” .....	77
Ilustración 39: Visualización del crontab.....	77
Ilustración 40: Linux “Cron” scripts .....	78
Ilustración 41: Parámetro Hourly .....	78
Ilustración 42: Parámetro Daily.....	78
Ilustración 44: Parámetro Monthly.....	79
Ilustración 43: Parámetro Weekly.....	79
Ilustración 45: Seteo script en “Cron Hourly” .....	79
Ilustración 44: Servidor BI .....	80
Ilustración 45: Consola Administrativa.....	80
Ilustración 46: Creación de usuarios .....	81
Ilustración 48: Roles asignados a un usuario .....	82
Ilustración 49: Pestaña usuarios y roles .....	82
Ilustración 50: Loguin pentaho.....	83
Ilustración 51: Barra de herramientas, Marketplace .....	83
Ilustración 52: Pestaña conexiones, mondrian .....	85
Ilustración 53: Creación Schema .....	86
Ilustración 54: Asignación de nombre a un Schema .....	86
Ilustración 55: Esquema estrella .....	86
Ilustración 56: Agregar cubo .....	87
Ilustración 57: Asignar nombre al cubo.....	87
Ilustración 58: Agregar tabla.....	87
Ilustración 60: Asignar tabla de hechos .....	88
Ilustración 61: Agregar dimensión .....	88
Ilustración 62: Establecer nombre dimensión.....	88
Ilustración 64: Mapear tabla a una jerarquía .....	89
Ilustración 63: Agregar una tabla a una jerarquía.....	89
Ilustración 65: Asignar clave primaria para una jerarquía.....	89
Ilustración 66: Agregar nivel a una jerarquía .....	90
Ilustración 67: Definición de atributos para un nivel .....	90
Ilustración 68: Columna que representa al nivel.....	91
Ilustración 69: Diferentes tipos de nivel.....	91
Ilustración 70: Ejemplo completo, dimensión Fecha Creación .....	92
Ilustración 71: Agregar medida .....	93
Ilustración 72: Atributos de una medida.....	93
Ilustración 73: Guardado de un cubo.....	94
Ilustración 74: Publicación de un cubo .....	94
Ilustración 75: Dialogo de progreso de la conexión a un repositorio.....	95
Ilustración 76: Opciones de guardado para la publicación de un schema .....	95
Ilustración 77: Dialogo de publicación exitosa .....	95

## Trabajo Final De Grado

Argañaraz Francisco Jose – Mirizio Esteban Ricardo

Ilustración 78: Configuración de un Datasource .....	96
Ilustración 79: Dialogo de conexión exitosa al datasouce.....	96
Ilustración 80: Dialogo de guardado exitoso de un datasource .....	97
Ilustración 81: Saiku en la barra de herramientas de pentaho .....	98
Ilustración 82: Interfaz Saiku Analytics.....	98
Ilustración 83: Dialogo de guardado exitoso de un datasource .....	99
Ilustración 84: Representación tabular de consultas mediante Saiku.....	99
Ilustración 85: Filtro de elementos en una dimensión .....	100
Ilustración 86: Selección de modo grafico Saiku .....	100
Ilustración 87: Tipos de gráficos ofrecidos por saiku .....	100
Ilustración 88: Representación grafico de barras mediante Saiku .....	101
Ilustración 89: Representación grafico de torta mediante Saiku .....	101
Ilustración 90: Estadísticas sobre consultas .....	102
Ilustración 91: Agregar rol a un schema.....	104
Ilustración 92: Agregar nombre a un rol .....	104
Ilustración 93: Agregar permisos hacia un schema .....	104
Ilustración 94: Permisos hacia un esquema .....	105
Ilustración 96: Permisos hacia un cubo.....	105
Ilustración 97: Permisos seleccionados para el cubo .....	106
Ilustración 98: Agregar permisos hacia una dimensión .....	106
Ilustración 99: Permisos hacia una dimensión .....	106
Ilustración 100: Agregar permisos hacia una jerarquía.....	107
Ilustración 101: Permisos hacia una jerarquía .....	107
Ilustración 102: Permisos seleccionados para la jerarquía .....	108
Ilustración 103: Agregar permisos hacia un miembro .....	108
Ilustración 104: Permisos hacia un miembro .....	109
Ilustración 105: Sintaxis de permisos hacia un miembro .....	109
Ilustración 106: Mapeo de usuarios a roles .....	110
Ilustración 107: Consulta con restricción de permisos.....	110
Ilustración 108: Elementos de la dimensión producto limitado por permisos del rol .....	111
Ilustración 109: Refrescar cubos disponibles .....	111
Ilustración 110: Distribución funcional grupo de desarrollo .....	113

### BIBLIOGRAFIA

#### Indicadores, Métricas y Medidas

- Ingeniería del Software – Un Enfoque Practico (Roger S. Pressman)
- <http://www.noqualityinside.com/nqi/nqifiles/M%C3%A9tricasDeCalidadDeSW.pdf>

#### Cuadro de mando

- Cuadro de Mando Integral – 2da edición (Kaplan y Norton)
- Balanced Scorecard - <http://www.3w3search.com/Edu/Merc/Es/GMerc094.htm>

#### CDE

- <http://www.webdetails.pt/ctools/cde.html>
- <http://type-exit.org/adventures-with-open-source-bi/2011/06/creating-dashboards-with-cde/>

#### SAIKU + MONDRIAN

- <http://meteorite.bi/saiku>
- <http://dev.analytical-labs.com/>
- <http://codeissue.com/articles/a04edc3096d9c7d/introduction-to-mondrian-olap-schema>
- <http://mondrian.pentaho.com/documentation/>

#### Medidas, Dimensiones, Jerarquías y Niveles

- <http://exa.unne.edu.ar/depar/areas/informatica/SistemasOperativos/OLAPMonog.pdf>

#### Bugzilla

- <http://www.bugzilla.org/docs/4.2/en/pdf/Bugzilla-Guide.pdf>
- <http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=debianBugzilla>

#### Pentaho

- <http://etl-tools.info/pentaho-tutorial.html>
- <http://infocenter.pentaho.com/help/index.jsp>

#### Kettle

- <http://wiki.pentaho.com/display/EAI/Latest+Pentaho+Data+Integration+%28aka+Kettle%29+Documentation>
- <http://goo.gl/OEiRRR>

#### CRON

- <http://blog.desdelinux.net/cron-crontab-explicados/>
- [http://www.linuxtotal.com.mx/?cont=info\\_admon\\_006](http://www.linuxtotal.com.mx/?cont=info_admon_006)

#### Bi server y Administration console

- <http://wiki.pentaho.com/display/ServerDoc2x/Introduction+to+the+BI+Platform>