

MANUAL PARA LA SOLUCIÓN DE ERRORES EN LAS HERRAMIENTAS XAMPP, PENTAHO Y SAIKU.



Fundación Universitaria Konrad Lorenz.
Iván Felipe Molano Ardila.
Facultad de Matemáticas e Ingenierías.
2014.

TABLA DE CONTENIDO.

Tabla de Imágenes.	2
Versiones de Software.	3
1. Introducción.	4
1.1 Propósito del Documento.	4
2. Definición de Conceptos.	5
3. Herramientas de Trabajo.	7
3.1 XAMPP.....	7
3.1.1 Modificación de Puertos en los Archivos de Configuración.	7
3.2 Pentaho.	13
3.2.1 Configuración Variables de Entorno.	14
3.2.2 Errores en la consola de Administración.....	17
3.2.3 Errores a la hora de publicar un cubo.....	24
3.3 Saiku.....	30
3.3.1 Configuración Variables de Entorno.	30
3.3.2 Creación de un archivo de fuente de Datos.....	31
3.3.3 Abrir Cubos de Pentaho en Saiku.	34
3.3.4 Roles de Usuario.	39
3.4 Herramientas de Trabajo Adicionales.	45
4. Referencias.....	46

Tabla de Imágenes.

<i>Imagen 1: Interfaz de Usuario XAMPP.</i>	8
<i>Imagen 2: Error presentado al momento de subir servicio de Tomcat.</i>	8
<i>Imagen 3: Error Presentado al momento de subir servicio de Apache</i>	9
<i>Imagen 4: Módulos de XAMPP funcionando adecuadamente</i>	10
<i>Imagen 5: Error en el puerto HTTPS.</i>	11
<i>Imagen 6: Nuevo puerto HTTPS predeterminado.</i>	12
<i>Imagen 7: Verificación de la versión disponible de Java.</i>	14
<i>Imagen 8: Error al intentar subir los servicios de Pentaho.</i>	15
<i>Imagen 9: Configuración de Variables de Entorno.</i>	16
<i>Imagen 10: Inicio de Servicios en Pentaho.</i>	17
<i>Imagen 11: Login para acceder a la consola de administración de Pentaho</i>	18
<i>Imagen 12: Ingresando en la consola de Administración de Pentaho.</i>	18
<i>Imagen 13: Error al cargar página web de Pentaho en la consola de Administración.</i>	19
<i>Imagen 14: Error al realizar comunicación con la consola de Usuario de Pentaho.</i>	19
<i>Imagen 15: Estado del Servidor Inactivo.</i>	20
<i>Imagen 16: Estado del servidor Activado.</i>	21
<i>Imagen 17: Menú de la pestaña "Administration" Consola de administración.</i>	22
<i>Imagen 18: Formulario para crear una nueva conexión de BD.</i>	22
<i>Imagen 19: Ejemplo configuración cadena de conexión.</i>	24
<i>Imagen 20: Asociar una cadena de Conexión.</i>	25
<i>Imagen 21: Configuración de conexión en Schema Workbench.</i>	25
<i>Imagen 22: Conexión configurada exitosamente.</i>	26
<i>Imagen 23: Definición de contraseña para la publicación de cubos.</i>	27
<i>Imagen 24: Introduciendo la contraseña de publicación.</i>	27
<i>Imagen 25: Error al realizar la publicación de un cubo.</i>	28
<i>Imagen 26: Publicación de un cubo.</i>	28
<i>Imagen 27: Publicación de cubo exitosa.</i>	29
<i>Imagen 28: Inicio de Servicio en Saiku</i>	30
<i>Imagen 29: Saiku almacenado en el directorio principal C:\</i>	31
<i>Imagen 30: Creación de un Archivo de Fuente de Datos.</i>	32
<i>Imagen 31: Ejemplo Archivo de fuente de Datos.</i>	33
<i>Imagen 32: Creación carpeta "Modelos" en Pentaho.</i>	34
<i>Imagen 33: Creación carpeta "Foodmart" dentro de los archivos de Saiku.</i>	35
<i>Imagen 34: Habilitando Líneas de Texto en fuente de datos "foodmart".</i>	36
<i>Imagen 35: Configuración para presentar el cubo "FoodMart" en la interfaz de Saiku.</i>	37
<i>Imagen 36: Login Saiku.</i>	38

<i>Imagen 37: Interfaz Usuario Saiku.....</i>	<i>38</i>
<i>Imagen 38: Cubos disponibles para visualizar por medio de Saiku.</i>	<i>39</i>
<i>Imagen 39: Directorio para crear nuevos usuarios.</i>	<i>40</i>
<i>Imagen 40: Usuarios y Roles.....</i>	<i>40</i>
<i>Imagen 41: Configuración de Seguridad en el cubo “foodmart”.....</i>	<i>41</i>
<i>Imagen 42: Habilitando Configuraciones de Seguridad.</i>	<i>41</i>
<i>Imagen 43: Asignando un "Alias" a los roles predeterminados de Saiku.....</i>	<i>42</i>
<i>Imagen 44: Sintaxis para asignar de permisos en los roles de usuario.</i>	<i>43</i>
<i>Imagen 45: El usuario "postgres" no puede ver ningún cubo.....</i>	<i>44</i>
<i>Imagen 46: Rol privado con acceso al cubo FoodMart.</i>	<i>44</i>

Versiones de Software.

Los nombres y las versiones de los respectivos programas que se utilizarán a lo largo del presente manual son definidos en la siguiente tabla:

NOMBRE DEL SOFTWARE	VERSIÓN
XAMPP	3.2.1
Pentaho BI Server.	4.5.0
Pentaho Schema Workbench	3.5.0
Pentaho Data Integration	4.1.0
Saiku.	2.5

1. Introducción.

1.1 Propósito del Documento.

Hoy en día, encontramos muchas herramientas (software) diseñadas específicamente para el área de inteligencia de Negocios (Business Intelligence), las cuales permiten diseñar e implementar modelos de datos en la organización, con el fin de conocer la información de sus actividades, y de esta manera, evaluar los resultados para tomar decisiones que beneficien a la organización.

En ocasiones, al adquirir software, el usuario solamente realiza la instalación de este, suponiendo que todo va a funcionar de manera adecuada; por lo que raramente el usuario tendrá que realizar alguna configuración del programa de forma manual. En las ocasiones en las que se presenta esta situación, el usuario debe acceder a los archivos del programa, y realizar las respectivas modificaciones para que el programa proporcione una funcionalidad adecuada y es específicamente de lo que se hablara en el presente documento.

En esta ocasión, se trabaja con tres herramientas enfocadas al campo de Business Intelligence, con las que se pueden diseñar, implementar y visualizar todos los elementos de un cubo, y de igual manera gestionar conexiones entre las bases de datos. Este manual está diseñado con el fin de presentar las configuraciones adecuadas para instalar y configurar las herramientas XAMPP, Pentaho y Saiku. Además se pretende mostrar las soluciones de algunos errores observados al instalarlas en Windows; De esta manera, se garantiza la funcionalidad adecuada de cada una de las herramientas.

Estas herramientas, generalmente poseen una configuración predeterminada para su funcionamiento dentro del sistema operativo; pero en ciertos casos, estas configuraciones no son las correctas, por lo que las herramientas no funcionan adecuadamente y presentan constantemente errores. En una situación de este tipo, se debe modificar esta configuración, y ajustarla dependiendo de las necesidades, con el objetivo de que funcionen correctamente en el sistema.

Finalmente, se pretende definir los conceptos más importantes, para ofrecer un mejor entendimiento en las explicaciones de los diversos procesos que se definirán a lo largo del documento.

2. Definición de Conceptos.

Business Intelligence – Es un conjunto de estrategias, metodologías y capacidades enfocadas a la creación y administración de información por medio del análisis de todos los datos existentes en una organización, con el fin de que esta tome decisiones acorde los resultados obtenidos durante el análisis de la información. [12].

Data Warehouse – Consiste en una base de datos caracterizada por ofrecer un repositorio de la información de una o diversas fuentes de datos, con el fin de procesarla permitiendo el análisis desde varios puntos de vista. [8].

Data Mart – Es un subconjunto de un Data Warehouse, se trata de una base de datos departamental, especializada en el almacenamiento de los datos de un departamento laboral específico. [7].

Staging Area - Es un sistema que tiene por objetivo facilitar la extracción de los datos desde la fuente de estos, con el fin de mejorar la calidad de los datos. Permanece entre las fuentes de datos y el Data Warehouse. [14].

Proceso ETL – Es un proceso que consiste en transportar el conjunto de datos necesarios, desde las diversas fuentes de datos originales al Data Warehouse, pasando por una serie de fases que permitan realizar cambios que cumplan con las necesidades del modelo de Business Intelligence [4]. Este a su vez se divide en tres fases:

- I. **Extracción (*Extraction*)** – Se extraen los datos necesarios desde las fuentes de datos originales.
- II. **Transformación (*Transformation*)** – Se pulen los datos, de manera que no haya información incoherente con el fin de ajustarla a las necesidades que requiera el modelo de Business Intelligence.
- III. **Carga (*Load*)** – Se toman los datos transformados y se almacenan en una base de datos, desde la cual se realizarán los diferentes análisis para cumplir con cada uno de los requerimientos.

OLAP (On-Line Analytical Processing) - Es una solución que tiene como objetivo agilizar la consulta de grandes cantidades de datos. Para esto, utiliza estructuras multidimensionales (o cubos OLAP) que contienen datos resumidos desde diversas fuentes de datos. [3].

Cubo - Es un concepto desde un punto de vista gráfico, que solamente es aplicable para un modelo de tres dimensiones. Un modelo de más de tres dimensiones suele denominarse “Híper cubo”. Una base de datos multidimensional. [13].

Dimensión - Es la representación de un punto de vista para cierto proceso de negocio. [13].

Métrica - Son todos los indicadores cuantificables que permiten medir un proceso de negocio. [13].

Tabla de Hechos - Representación de todos los procesos de negocio de la organización. [13].

Java Virtual Machine (JVM) - Es un programa, que puede ser ejecutado bajo una plataforma específica, este es capaz de interpretar y ejecutar instrucciones expresadas en un código binario especial conocido como el Java bytecode, el cual es generado por el compilador del lenguaje Java. [9].

Java Development Kit (JDK) - Básicamente, consiste en un conjunto de programas y librerías que permiten realizar tareas como las de compilar, ejecutar y documentar programas codificados bajo el lenguaje Java. [10].

Java Runtime Environment (JRE) - Es una máquina virtual de Java que tiene como función principal, actuar como intermediario entre una aplicación programada bajo el lenguaje Java y el sistema operativo que el usuario este utilizando. [1].

Variable de Entorno - Son variables globales del sistema, que son accesibles por todos los procesos que ejecuta el sistema operativo y que tienen como funcionalidad, el almacenamiento de los valores de todo el sistema. [6].

Sistema Operativo - Es un programa que actúa como intermediario entre un usuario y el hardware de un computador; así mismo, es considerado como un conjunto de programas que han sido especialmente diseñados para la ejecución de varias tareas. [5].

HTTPS (Hypertext Transfer Protocol Secure) - Es un protocolo basado en el protocolo HTTP, el cual tiene como función principal, la transferencia segura de datos de Hipertexto, en pocas palabras, es la versión segura de HTTP. [11].

3. Herramientas de Trabajo.

Se utilizarán varias herramientas, las cuales permiten realizar todas las labores necesarias para diseñar e implementar el modelo de Business Intelligence (XAMPP, Pentaho y Saiku). Primeramente, se aclara que todas las herramientas trabajan sobre el sistema operativo de Windows, por lo tanto, el desarrollo e implementación del modelo de Business Intelligence se llevará a cabo en este.

3.1 XAMPP.

La primera herramienta a definir, es conocida bajo el nombre de XAMPP. Su nombre está basado según la siguiente especificación:

- X – (Tiene Funcionalidad en cualquier sistema Operativo).
- A – (Servidor Web Apache)
- M – (Base de Datos MySQL)
- P – (Intérprete de Lenguajes PHP)
- P – (Intérprete de Lenguajes Perl)

Esta herramienta funciona como un servidor de plataforma libre que consiste en la unificación de un servidor web “Apache”, intérpretes de lenguaje de scripts “PHP”, servidor de base de datos “MySQL” en una sola aplicación. Además, es una herramienta que permite realizar pruebas a un software o página web, sin tener que acceder a internet. [2].

En esta ocasión, se trabaja con la versión 3.2.1 de XAMPP para definir las soluciones posteriormente presentadas. Se utilizará el motor de base de datos MySQL, el cual va a contener la base de datos, desde la cual se realizará una staging area de los datos necesarios para cumplir con la elaboración de los requerimientos asignados.

3.1.1 Modificación de Puertos en los Archivos de Configuración.

Como se definió en la introducción del documento, algunos programas tienen una serie de configuraciones predeterminadas las cuales suponen que funcionarán adecuadamente dentro del sistema y permitirán ejecutar las aplicaciones sin ningún inconveniente. A continuación se presenta un error basado en este tipo de configuraciones predeterminadas, las cuales generan conflictos entre uno o varios módulos del programa, en los cuales se está en la obligación de modificar esta configuración con el fin garantizar la funcionalidad de la herramienta.

Este programa puede ser descargado de manera gratuita desde la página web oficial. <http://www.apachefriends.org/es/download.html>. (Se aclara que XAMPP ya debe estar instalado en el equipo para poder realizar los pasos siguientes).

Al iniciar el programa, se observa una interfaz como la siguiente:

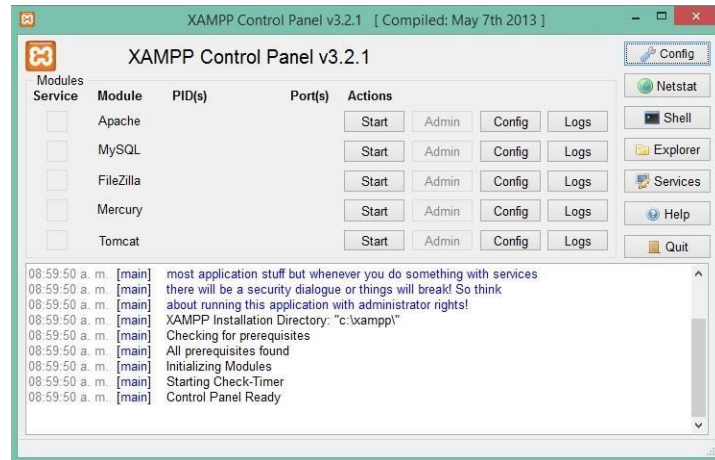


Imagen 1: Interfaz de Usuario XAMPP.

La interfaz de usuario y funcionalidad de la herramienta, depende de la versión de XAMPP con la que se esté trabajando, ya que algunas ofrecen menos opciones e interfaces mucho más resumidas.

3.1.1.1 Error del Puerto Predeterminado.

Inicialmente, se quiere activar los servicios necesarios para la ejecución de los procesos que se necesitan para realizar las diferentes tareas, pero surge el siguiente inconveniente:

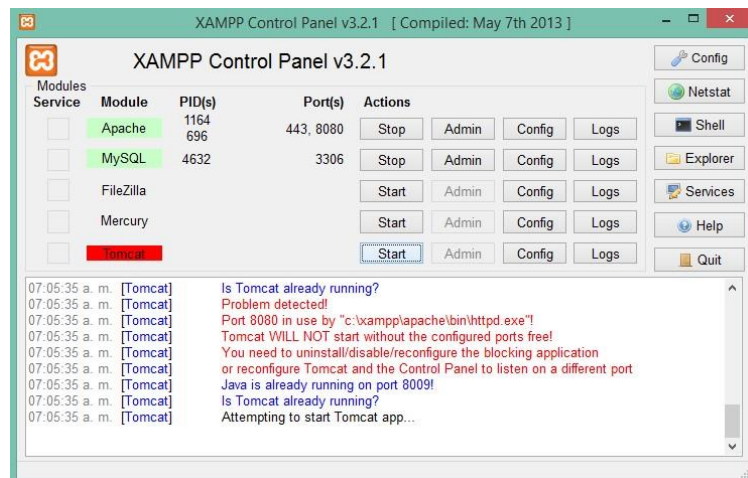


Imagen 2: Error presentado al momento de subir servicio de Tomcat

Si se observa la imagen anterior, se puede ver que el módulo de Tomcat presenta un color rojo en el estado de activación, indicando que el servicio no se ha subido adecuadamente y seguidamente se presenta el error por el cual no se puede activar; este error informa que el puerto 8080 está siendo utilizado por otra aplicación y por esta causa no se ha podido activar el servicio.

El módulo de Apache se encuentra funcionando correctamente, ya que su color de estado de activación aparece en verde, pero así mismo, es la aplicación que está utilizando el puerto 8080 y la que impide que el servicio pueda ser activado en el módulo de Tomcat.

Esto se debe a que en las configuraciones predeterminadas del programa, se asigna este puerto pensando que solo una aplicación en todo el sistema operativo hará uso de este, pero en el caso de que dos aplicaciones estén configuradas para utilizar el mismo puerto, la primera aplicación que se ejecute será la que tenga el control total para el uso de este puerto, mientras que la otra aplicación a la que le fue asignada ese mismo puerto tendrá que esperar a que el puerto se encuentre libre y posteriormente pueda tomarlo para realizar sus actividades. A continuación se presenta un ejemplo de lo anteriormente escrito:



Imagen 3: Error Presentado al momento de subir servicio de Apache

En esta imagen se observa que al activar primeramente el módulo de Tomcat, este presenta un color verde en su estado de activación, haciendo conocer que el servicio fue activado correctamente; si se observa al lado siguiente, se puede ver que está utilizando tres puertos, dentro de estos el 8080. Seguidamente se procede a la activación de los módulos de MySQL y Apache, de los cuales Apache no se ejecuta correctamente, haciendo que el programa muestre una serie de errores relacionados a que el puerto 8080 ya está siendo utilizado por el módulo de Tomcat. En comparación de la imagen 2, primeramente se activó el módulo de Apache, y

este se ejecutó correctamente, pero al momento de activar el módulo de Tomcat, como el puerto ya se encontraba siendo utilizado, no se ejecutó y presento el error por el cual no podía activar el servicio.

Para solucionar este error, es necesario modificar las configuraciones por defecto que trae este programa. Lo que se debe hacer cuando se presente un inconveniente como el anteriormente mencionado, es modificar el puerto con el que va a trabajar el programa, para esto se deben seguir los siguientes pasos:

- I. Abrir el siguiente directorio: “C:\xampp\apache\conf”.
- II. Buscar y abrir con Notepad++, el archivo de configuración bajo el nombre de “httpd.conf”
- III. Una vez abierto el documento, se busca una línea que contenga la palabra “Listen 80” y se modifica el puerto 80, por el puerto que se elija para asignar al programa, en este caso el puerto 8084. (La línea “Listen 80” debe estar habilitada, es decir, que no debe estar con el carácter #).
- IV. De igual manera, se realiza una búsqueda de la palabra “ServerName localhost 80” y se modifica el puerto, asignando el mismo puerto del paso anterior. (La línea “ServerName localhost 80” debe estar habilitada, es decir, que no debe estar con el carácter #).
- V. Guardar los cambios realizados en el archivo y reiniciar XAMPP.

Si todo quedo configurado adecuadamente, al momento de activar todos los módulos de XAMPP, estos deben presentar el estado de activación en verde. De esta manera el error es solucionado.



Imagen 4: Módulos de XAMPP funcionando adecuadamente

Haciendo una comparación de la imagen anterior con la imagen 2, se puede observar que en el módulo de apache presenta la modificación anteriormente

realizada, permitiendo conocer que ya no trabaja con el puerto predeterminado 8080, si no que ahora trabaja con el puerto asignado 8084.

3.1.1.2 Error del Puerto HTTPS.

Otro error presentado en esta herramienta, consiste en el puerto HTTPS configurado dentro de los archivos de XAMPP.

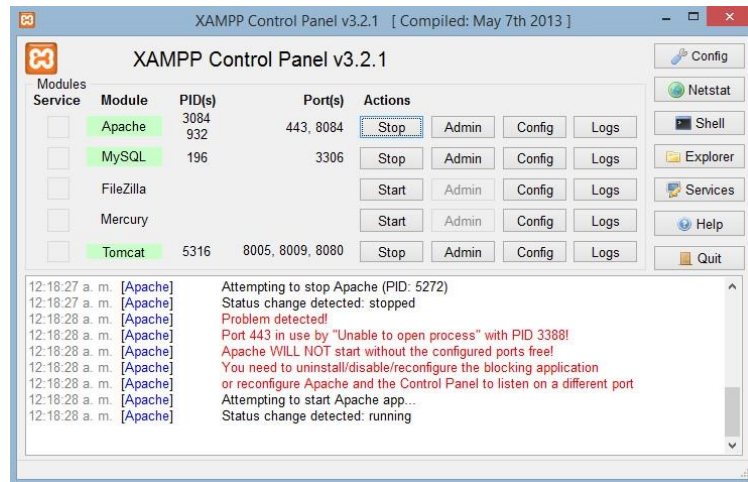


Imagen 5: Error en el puerto HTTPS.

En varias ocasiones cuando se quiere activar el módulo de apache, resulta que el puerto HTTPS definido en los archivos de configuración de XAMPP está siendo utilizado por otro programa, impidiendo que el servicio de apache se active correctamente. Para solucionar este problema de raíz, e impedir que en ocasiones futuras se vuelva a presentar, se está en la obligación de cambiar el puerto definido por la herramienta dentro del archivo de configuración. Para esto se deben de seguir los siguientes pasos:

- I. Abrir el siguiente directorio: `C:\xampp\apache\conf\extra`
- II. Buscar y abrir el archivo de configuración bajo el nombre de `"httpd-ssl.conf"`.
- III. Una vez abierto el documento, se busca una línea que contenga la palabra `"Listen 443"` y se modifica el puerto 443, por el puerto que se elija para asignar al programa, en este caso el puerto 4430. (La línea `"Listen 443"` debe estar habilitada, es decir, que no debe estar con el carácter #).
- IV. De igual manera, se realiza una búsqueda de la palabra `"ServerName"` y seguidamente se reemplaza `"www.example.com"` por `"localhost"`, además se modifica el puerto, asignando el mismo puerto del paso anterior. (La línea `"ServerName localhost:443"` debe estar habilitada, es decir, que no debe estar con el carácter #).

V. Guardar los cambios realizados en el archivo y reiniciar XAMPP.

Si la configuración se realizó adecuadamente, al activar el módulo de apache, no se debe visualizar el error presentado en la imagen 5. De esta manera el error es solucionado.

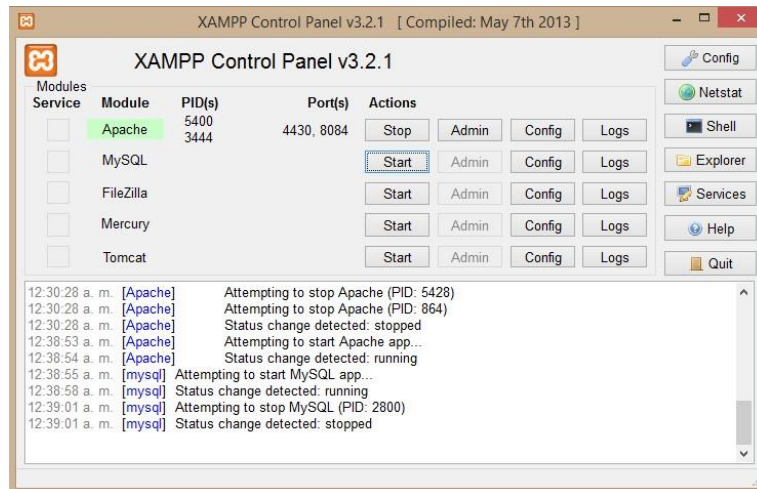


Imagen 6: Nuevo puerto HTTPS predeterminado.

Como se puede observar en la imagen anterior, al ejecutar nuevamente XAMPP, este ejecuta la configuración realizada anteriormente, haciendo saber que está trabajando por medio del puerto 4430.

3.2 Pentaho.

Ofrece una suite de programas que proporcionan varias utilidades, dentro de las cuales se puede mencionar el diseño e implementación de cubos, realización de procesos ETL a las fuentes de datos (en este caso desde la base de datos MySQL), presentación de los cubos bajo una interfaz de usuario, en la cual se puede consultar información y por último, la creación de reportes con los datos que el usuario considere necesarios. No está en la intención del documento hacer la explicación sobre cómo instalar o de alguna manera presentar ejemplos del funcionamiento de cada uno de estos programas, ya que el documento está orientado a definir soluciones de algunos errores en las herramientas de trabajo.

Dentro de los programas con los cuales trabaja Pentaho, encontramos el “**Data Integration**”, mediante el cual se realizan los procesos de extracción de datos de la fuente de datos, creación de dimensiones y tabla de hechos, procesos ETL en los datos, almacenamiento de los datos en un Data Warehouse entre otras. Esta herramienta es la base fundamental para la creación de los modelos de Business Intelligence.

Por otro lado, se encuentra el “**Schema Workbench**”, en el cual se crea el cubo con sus respectivas dimensiones, métricas y tabla de hechos, tomando los datos que han sido manipulados anteriormente desde “Data Integration”. Por medio de esta herramienta se realizan las publicaciones de los cubos a la consola de usuario (será definida más adelante), en la cual puede realizar las consultas que considere pertinentes. De igual manera, otra de las herramientas que se encuentran en la suite de Pentaho es el “**Report Designer**”, en el cual el usuario puede crear reportes tomando como referencia los datos que se encuentran almacenados en el Data Warehouse.

Finalmente, encontramos el “**BI Server**”, en el que se incluye la consola de administración y de usuario de Pentaho. Primeramente, la consola de administración, realiza la gestión de roles y usuarios asignando los permisos correspondientes en cada una de las herramientas anteriormente definidas, de igual manera, controla la configuración de las conexiones de las diferentes bases de datos con las que se esté trabajando. Es importante establecer estas conexiones desde el inicio de la implementación de un modelo de Business Intelligence, de lo contrario, se presentarán conflictos a la hora de realizar actividades bajo las herramientas de Pentaho.

La consola de usuario, es la encargada de presentar el cubo implementado al usuario, donde este podrá tomar decisiones de acuerdo a los resultados que arrojen las consultas de datos que este formule. Este visor OLAP tiene una funcionalidad aceptable, ya que no es muy atractiva visualmente, y es un poco molesta al interactuar y/o realizar consultas de datos del cubo. Para presentar los cubos de una forma más elegante, se utilizara el visor OLAP Saiku.

3.2.1 Configuración Variables de Entorno.

Pentaho es un programa que está desarrollado bajo el entorno de desarrollo de Java, por lo que se tienen que agregar las variables de entorno necesarias para que este programa pueda funcionar adecuadamente. A continuación se presentan una serie de pasos para verificar o configurar cada una de estas variables.

El primer paso a seguir, consiste en verificar si el computador tiene instalada alguna versión de Java.

- I. Para esto, en la interfaz de Windows, se hace clic en “Inicio”.
- II. Se selecciona la opción “Ejecutar...”.
- III. A continuación, se abre una nueva ventana, en la cual se introduce la palabra “cmd” con el fin de abrir la consola de comandos de Windows.
- IV. Utilizar el comando “*java -version*”, con el fin de que el sistema, informe si tiene una versión instalada de Java o si es necesario instalarlo.

En caso de tener una versión de Java instalada en el equipo, el sistema presenta al usuario una breve información acerca de la versión que el equipo tiene disponible. Se debería visualizar algo muy similar de lo que se observa a continuación:



```
ca Símbolo del sistema
Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\toshiba>java -version
java version "1.7.0_51"
Java(TM) SE Runtime Environment (build 1.7.0_51-b13)
Java HotSpot(TM) 64-Bit Server VM (build 24.51-b03, mixed mode)

C:\Users\toshiba>
```

Imagen 7: Verificación de la versión disponible de Java.

Si no aparece ningún tipo respuesta mediante la utilización de este comando, es porque el equipo no tiene ninguna versión de Java disponible; para solucionar este problema, se puede descargar gratuitamente cualquier versión de Java desde la página web oficial - <https://www.java.com/es/download/>. (Se aclara que este documento no entrará en detalles sobre cómo se debe realizar la instalación de los paquetes de programas de Java).

Una vez verificados los pasos anteriores, procedemos a subir los servicios de Pentaho. Si las variables de entorno adecuadas están bien configuradas, los servicios en esta herramienta deben ejecutarse sin ningún problema. Si se intenta subir alguno de los servicios de cualquiera de esta herramienta (Consola de Administración o Consola de Usuario), y se presenta un pantallazo que aparece y desaparece rápidamente, lo más probable es que sea muy parecido al que se mostrara a continuación:



Imagen 8: Error al intentar subir los servicios de Pentaho.

Este error se debe a que no se tienen configuradas las variables de entorno en el sistema, provocando que los servicios no se ejecuten correctamente. Para solucionar este error, se deben seguir los siguientes pasos desde la interfaz del sistema operativo con el fin de configurar o verificar estas variables de entorno:

- I. Clic en el botón “Inicio”.
- II. Clic derecho en la opción “Equipo”.
- III. Seleccionar la opción “Configuración avanzada del sistema”.
- IV. Clic en el botón “Variables de entorno...”.
- V. En el panel “Variables de usuario” se crean dos nuevas variables con la siguiente información (dependiendo de la ruta en la que esté instalado Java):

- Para la primera variable:
 - i. Nombre de la Variable : **CLASSPATH**
 - ii. Valor de la Variable : C:\Program Files (x86)\Java\jre7\bin
- Para la segunda variable:
 - i. Nombre de la Variable: **JAVA_HOME**
 - ii. Valor de la Variable: C:\Program Files (x86)\Java\jdk1.7.0_51

Las dos variables de entorno anteriormente creadas, trabajan específicamente en aplicaciones hechas bajo el entorno de Java. La primera variable de entorno creada bajo el nombre de “CLASSPATH”, se encarga de mantener una lista de directorios, los cuales contienen varias clases propias de Java y a su vez un archivo JAR. De esta manera, al iniciar algún programa asociado a esta variable, el compilador de Java, busca en su lista de directorios, todas las clases de Java referenciadas en el programa, para finalmente ejecutar la aplicación.

La segunda variable, creada bajo el nombre de “JAVA_HOME”, tiene como labor almacenar el directorio en el cual se ha instalado el JDK, y por medio de este, acceder a las diversas herramientas utilizadas por Java.

El valor de cada una de las variables anteriormente mencionadas depende de la ruta en la que se haya instalado Java dentro del equipo, por lo que la ruta del directorio sería diferente a la que se presenta en el ejemplo anterior. Después de haber configurado y creado las dos nuevas variables de entorno, se podrá visualizar lo siguiente:

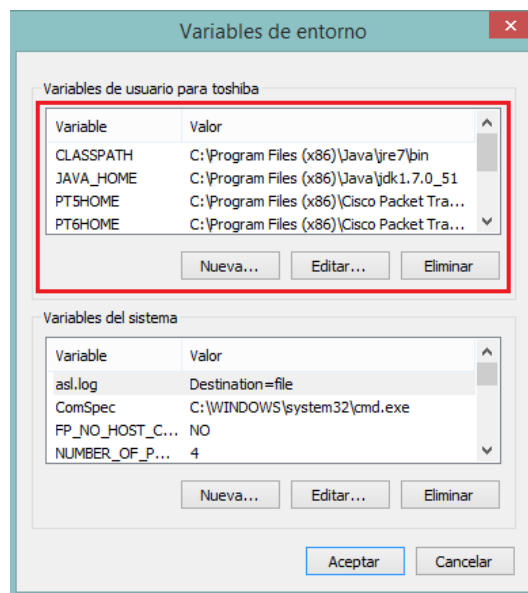


Imagen 9: Configuración de Variables de Entorno.

Finalmente, se hace clic en el botón “Aceptar”, y la nueva configuración quedará guardada en el sistema. Se recomienda reiniciar el computador para que la nueva configuración sea tomada por el sistema operativo, ya que si se intenta subir alguno de los servicios de Pentaho sin reiniciarlo, se visualizará el mismo error que se presentó desde un inicio (Ver Imagen 8).

Al activar los servicios de Pentaho, se puede observar lo siguiente:

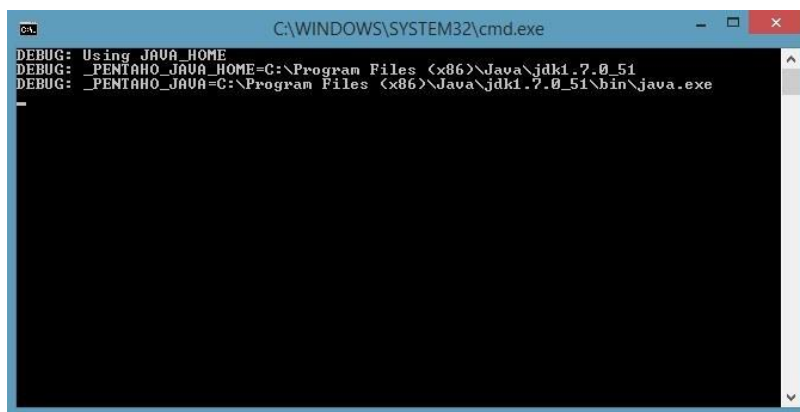


Imagen 10: Inicio de Servicios en Pentaho.

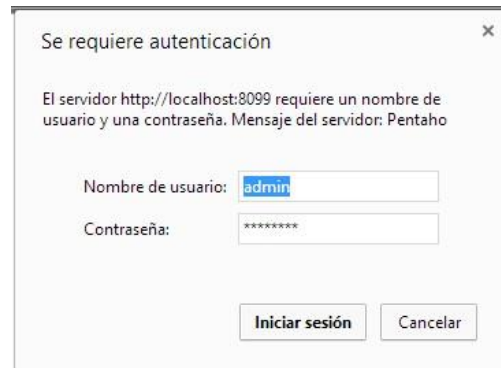
3.2.2 Errores en la consola de Administración.

En esta sección del documento se habla un poco más del BI Server; en el cual se encuentra la consola de administración y la consola de usuario de Pentaho. Como se especificó anteriormente, es importante definir inicialmente una cadena de conexión asociada a la base de datos con la cual se va a trabajar, ya que los otros programas de Pentaho, necesitan conocer esta conexión para que puedan realizar sus respectivas funciones sin ningún inconveniente.

Lo primero que se debe hacer consiste en lo siguiente:

- I. Abrir la carpeta contenedora de la consola de administración de Pentaho: “C:\Pentaho\biserver-ce-4.5.0-stable\administration-console”.
- II. Activar el servicio, para acceder a la consola de administración. (Abrir Archivo “start-pac.bat”). Si la aplicación se ejecuta correctamente, podrá seguir con el siguiente paso; si no se ejecutó, revisar la sección 3.2.1 y configurar las variables de entorno para que la aplicación pueda ser ejecutada).
- III. Desde el explorador de internet de su preferencia, ingrese la siguiente URL: “localhost:8099” (La configuración de la consola de administración viene con este puerto por defecto).

- IV. Al acceder a esta URL, el explorador deberá mostrar un cuadro de Login como el que se presenta a continuación:



A login dialog box titled "Se requiere autenticación" (Authentication is required). It contains the text: "El servidor http://localhost:8099 requiere un nombre de usuario y una contraseña. Mensaje del servidor: Pentaho". Below this, there are two input fields: "Nombre de usuario:" with the text "admin" entered, and "Contraseña:" with "*****" entered. At the bottom, there are two buttons: "Iniciar sesión" (Log in) and "Cancelar" (Cancel).

Imagen 11: Login para acceder a la consola de administración de Pentaho

En cualquier explorador de internet, aparecerá una interfaz parecida a la que se observa en la anterior imagen. Para acceder a la consola de administración, se utiliza un nombre de usuario y contraseña predeterminados en la configuración de Pentaho.

- V. En el campo de nombre de usuario ingresar “admin” y en el campo de contraseña ingresar “password”.
- VI. Al ingresar los datos de logueo, se podrá acceder a una interfaz como la que se presenta a continuación:

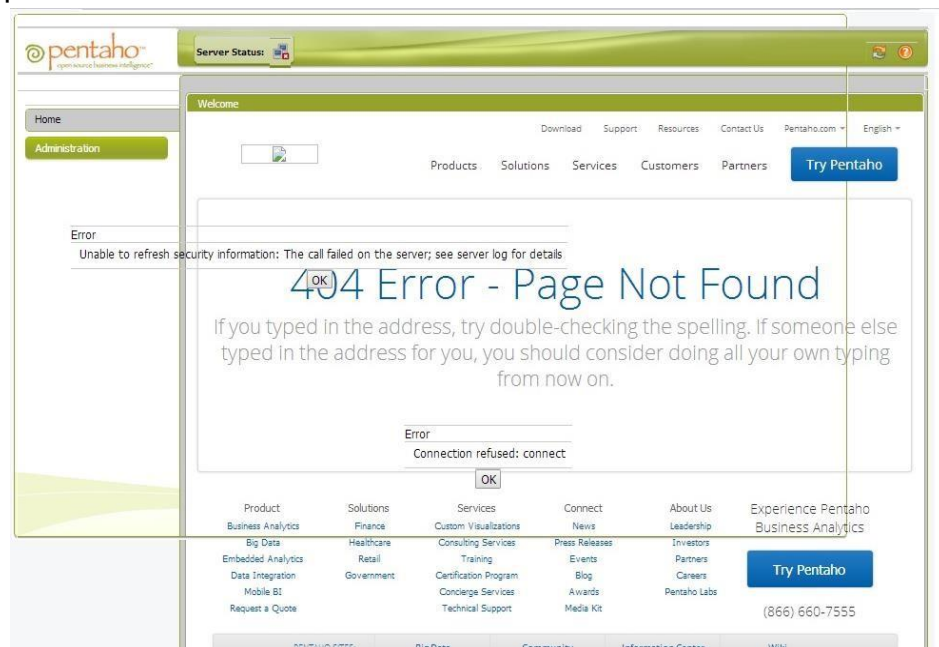


Imagen 12: Ingresando en la consola de Administración de Pentaho.

Como se observa, al entrar en esta interfaz aparecerán varios errores, los cuales serán mencionados más adelante. Por otra parte, al seleccionar la pestaña “Administration”, se podrá visualizar otra interfaz, en la cual se realiza la configuración de roles, usuarios y gestión de conexiones de las bases de datos. A continuación se muestran las imágenes correspondientes a los errores presentados en la interfaz de administración, con el fin de definir su causa y presentar su solución.

Error de Presentación.



Imagen 13: Error al cargar página web de Pentaho en la consola de Administración.

Este error es el primero que se visualiza al momento de acceder a la consola de administración. Realmente, no afecta de ninguna manera la funcionalidad del programa. Básicamente, cuando se accede a esta interfaz, se carga una página web propia de Pentaho, en la cual se ofrece información sobre la plataforma y una variedad de menús con los que puede interactuar el usuario. Cuando se presenta este error, es porque la consola de administración, dentro de sus configuraciones predeterminadas, intenta cargar un enlace establecido para abrir la página web, pero la página a la cual intenta acceder no está disponible, por lo cual, envía el anterior error a la interfaz, informando al usuario que la página no pudo ser encontrada.

Error de Conexión.

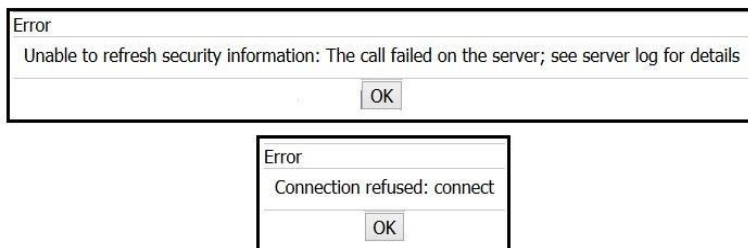


Imagen 14: Error al realizar comunicación con la consola de Usuario de Pentaho.

Los siguientes mensajes de error se pueden visualizar en la imagen 12, solo que por motivos de visibilidad se exponen mediante la anterior imagen. Estos errores se derivan de un problema principal, el cual será explicado a continuación.

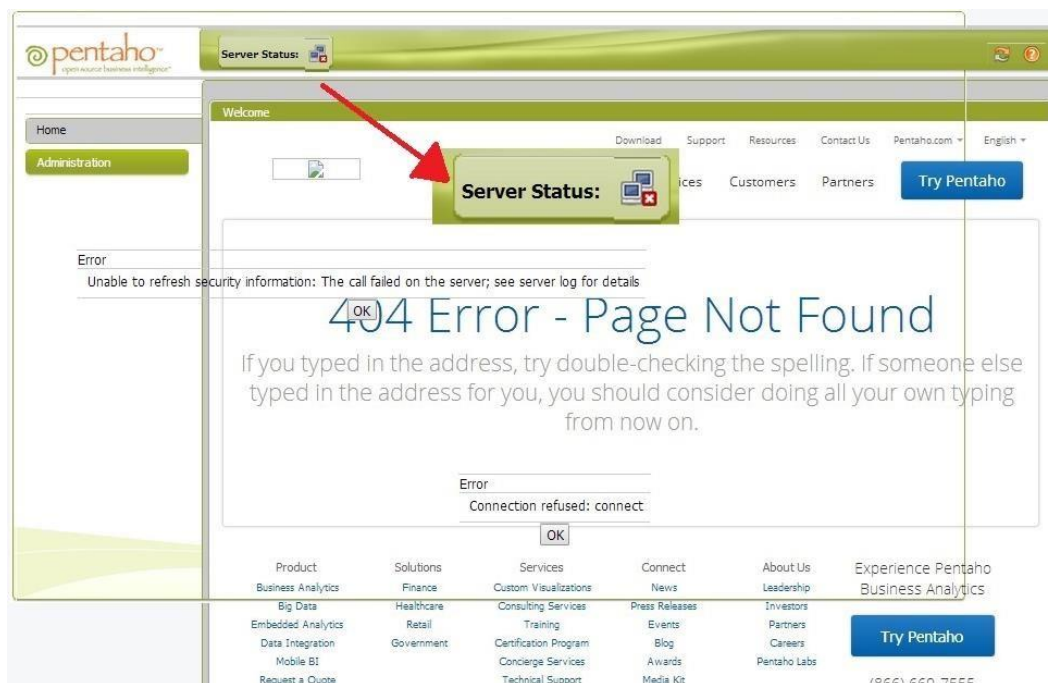


Imagen 15: Estado del Servidor Inactivo.

Como se puede observar, la consola de administración posee una ventana en la cual se muestra el estado en el que se encuentra el servidor. En esta ventana, se observa que hay un icono de dos computadores con una letra “x”, que permite saber, que el servidor está inactivo y por esta razón se visualizan los errores presentados en la imagen 14.

Se supone que con los pasos anteriormente definidos en el documento, se activa correctamente el servicio de la consola de administración, esto es cierto; solo que hace falta un paso importante y que es necesario tener en cuenta a la hora de realizar la publicación de un cubo en la consola de usuario.

Para activar el servicio de la consola de administración en su totalidad, es necesario igualmente activar desde un principio la consola de usuario. Estas dos consolas, deben ser activadas inicialmente para permitir la comunicación entre estas, y de esta manera cambiar el estado de la conexión. Para esto se debe hacer lo siguiente:

- I. Abrir la carpeta contendora de la consola de usuario de Pentaho :
“C:\Pentaho\biserver-ce-4.5.0-stable\biserver-ce”
- II. Activar el servicio, para acceder a la consola de Usuario. (Abrir Archivo “startpentaho.bat”). Si la aplicación se ejecuta correctamente, podrá seguir con el siguiente paso; si no se ejecutó, revisar la sección 3.2.1 y configurar las variables de entorno para que la aplicación pueda ser ejecutada).

- III. Una vez activado el servicio de la consola de usuario, ir al explorador de internet y actualizar la página en la cual se tiene la consola de administración.

Como resultado deberíamos tener la siguiente interfaz:

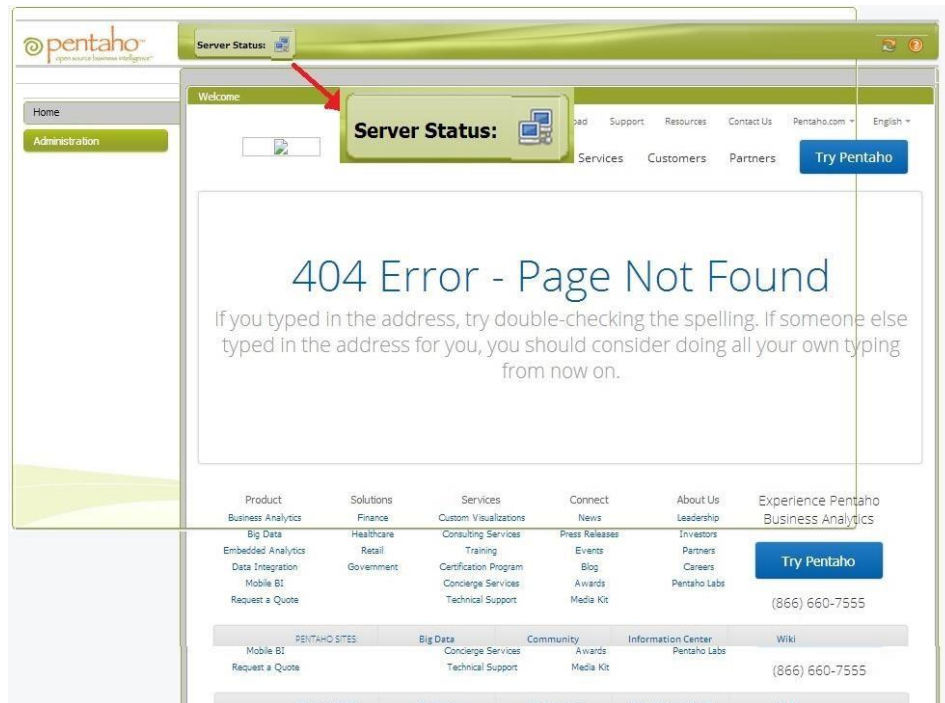


Imagen 16: Estado del servidor Activado.

Se observa, que la ventana de estado muestra otro icono, el cual indica que el servidor está activado y que se puede acceder a las configuraciones de la pestaña “Administración”. De igual manera, se observa que no apareció ninguno de los errores de la imagen 14, ya que lo anteriormente especificado, es la solución para corregir estos tipos de errores. Igualmente se sigue visualizando el error de la imagen 13, ya que la consola de administración intenta buscar un enlace web predefinido, pero el enlace que intenta buscar puede que no exista o que este deshabilitado.

Una vez se haya seleccionado la pestaña “Administración” se podrá observar lo siguiente:

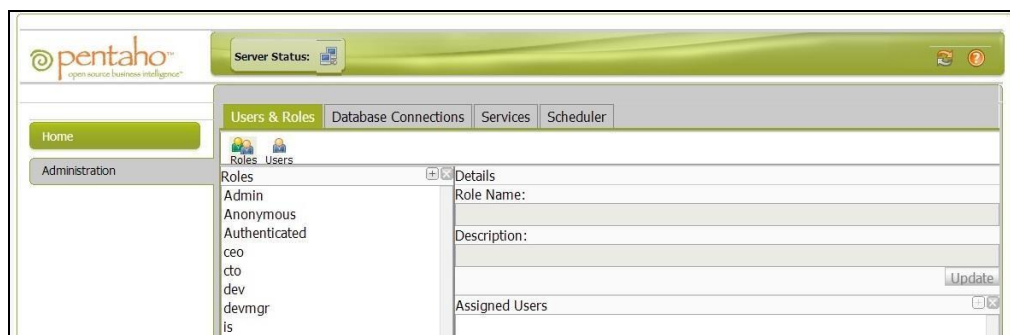


Imagen 17: Menú de la pestaña “Administration” Consola de administración.

Anteriormente, bajo la interfaz de la pestaña “Administration”, no se tenía la visualización de ningún menú de algunos datos; pero al cambiar el estado del servidor, se puede visualizar el conjunto de datos que Pentaho trae por defecto. Desde este momento se enfoca más en la pestaña de “Database Connections”, ya que el propósito de esta sección consiste en definir la creación de una cadena de conexión a la base de datos que se emplee para almacenar los datos en un Data Warehouse.

Lo primero que se tiene que hacer es seleccionar el menú “Database Connections” y crear una cadena de conexión asociada a la base de datos a la cual se esté haciendo referencia. Se observa que se presenta una lista de las conexiones que han sido creadas anteriormente. En la parte superior, se encuentra un pequeño botón con el símbolo “+” con el que se podrá crear una nueva conexión. Seguidamente lo seleccionamos y se presenta un formulario exactamente igual a este:

Imagen 18: Formulario para crear una nueva conexión de BD.

Los parámetros a configurar son los siguientes:

- **Name:** Nombre que recibirá la nueva conexión.

- **Driver Class:** Se elige el driver del motor de base de datos con el cual se vaya a trabajar. En este caso, se utilizará postgreSQL como motor de base de datos.
- **User Name & Password:** Es el nombre de usuario y la clave, que debe estar anteriormente creada en el motor de base de datos, la cual permite el acceso a la base de Datos a definir en esta conexión. Por lo contrario, si no se maneja un nombre de usuario y clave en la base de datos a la cual se quiere acceder, no es necesario llenar estos campos.
- **URL:** Es la ruta que se define, para que la conexión pueda asociarse a la base de datos que se vaya a especificar.

“(Driver que se utiliza para gestionar la conexión)://localhost: (Numero del Puerto con el que se está trabajando)/(Nombre de la base de datos a enlazar)”

En postgreSQL, la conexión se crea utilizando la siguiente sintaxis:

“Jdbc:postgresql://localhost: (Puerto)/(Nombre BD)”

Al trabajar con el motor de base de datos postgreSQL, es necesario descargar el driver que permita conectar a las herramientas de Pentaho con las bases de datos referenciadas en cada una de las configuraciones. Para esto se siguen los siguientes pasos:

- I. Abrir la página web <http://jdbc.postgresql.org/download.html>
- II. Descargar la versión del driver de postgreSQL de su preferencia.
- III. Copiar el driver descargado.
- IV. Pegar el driver en la siguiente ruta: “C:\Pentaho\biserver-ce-4.5.0stable\biserver-ce\tomcat\lib”.

De esta manera, las herramientas de Pentaho podrán acceder sin ninguna complicación a las bases de datos referenciadas en las configuraciones. Para cada motor de base de datos, existe una sintaxis diferente, por lo que se debe tener en cuenta el tipo de driver que utilizará para poder asociar la base de datos. Luego de haber llenado la anterior información, hacer clic en la palabra “Test” para que la consola de administración informe al usuario si la cadena de conexión está bien configurada, o no. Si se presenta algún error al hacer esta configuración, verificar que la información ingresada en el formulario sea adecuada, y verificar si se está trabajando o no, con un nombre de usuario o contraseña desde el motor de base de datos.

Para un ejemplo que será tomado en la sección siguiente, se definiría una cadena de conexión suponiendo que estamos trabajando con una base de datos llamada “Foodmart”, por lo cual se tiene:



Imagen 19: Ejemplo configuración cadena de conexión.

3.2.3 Errores a la hora de publicar un cubo.

La publicación de un cubo, es el paso final luego de haber pasado por una serie de procesos a través de las herramientas de Data Integration y Schema Workbench en las que se realiza el diseño y la implementación del cubo con sus respectivas dimensiones, métricas y tabla de hechos.

Se enfoca un poco más en el Schema Workbench, ya que es el encargado de realizar la publicación de un cubo a la interfaz de usuario. Para abrir el archivo ejecutable de este programa, se debe seguir los siguientes pasos:

- I. Abrir la carpeta del schema workbench: “C:\Pentaho\schema-workbench”.
- II. Abrir el archivo “Workbench.bat” y esperar a que se abra el programa.

Se presenta la interfaz de usuario, en la cual podrá definir las dimensiones, métricas y tabla de hechos que el modelo de Business Intelligence requiera. Lo primero que se debe hacer antes de crear el cubo, es asignar la conexión de base de datos siguiendo los pasos de la sección anterior (3.2.2).

Para asociar la conexión de la base de datos desde la cual se tomarán los datos para definir cada una de las características de un cubo, se seguirán los siguientes pasos:

- I. Seleccionar el menú “Options”, luego se selecciona la opción “Connection...”



Imagen 20: Asociar una cadena de Conexión.

Seguidamente aparece una nueva ventana en la cual se configuran los parámetros de la conexión de la base de datos. Siguiendo el ejemplo señalado anteriormente en la sección 3.2.2, imagen 19 se tiene:

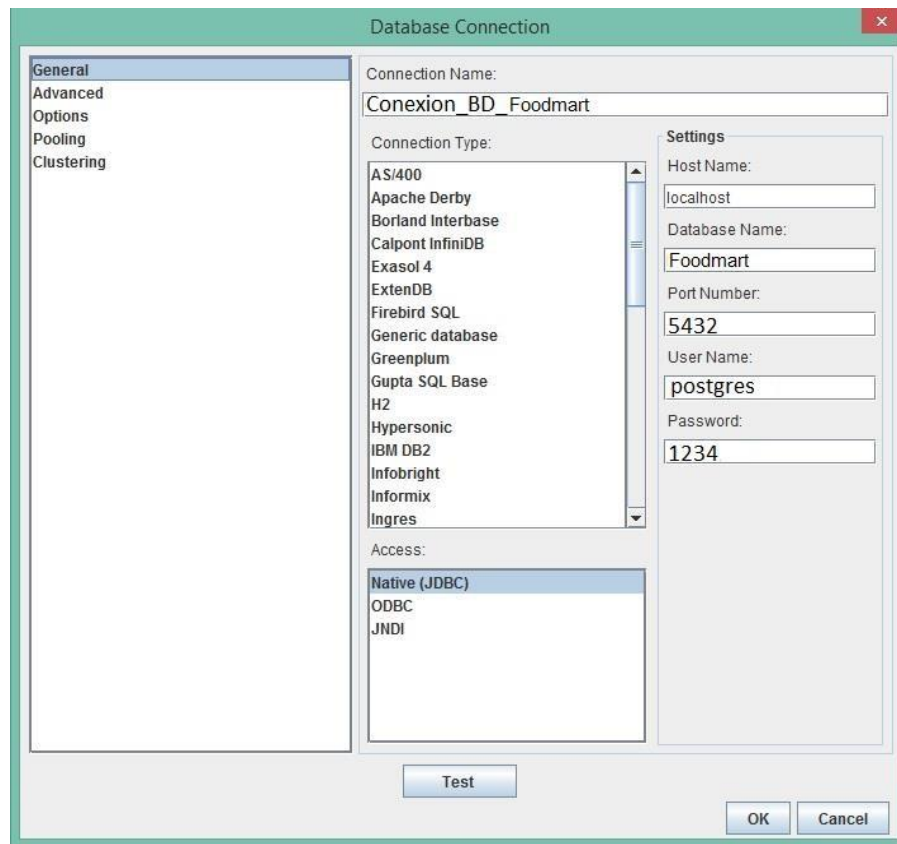


Imagen 21: Configuración de conexión en Schema Workbench.

Se procede a hacer el Test de la configuración, con el fin de verificar si la conexión es válida. Si es así, se debería visualizar un mensaje como el siguiente:



Imagen 22: Conexión configurada exitosamente.

En caso de que no se visualice esto, verificar la conexión creada anteriormente en la consola de administración de Pentaho; de igual manera comprobar que los datos ingresados sean los correctos y realizar nuevamente el test. La conexión debe de estar configurada apropiadamente para poder tomar los datos desde la base de datos a la cual se hace referencia al momento de crear la conexión, de lo contrario, no se podrá implementar el cubo ni ninguna de sus características.

Suponiendo que ya se han creado las dimensiones, métricas y la tabla de hechos correspondientes a un modelo de Business Intelligence, lo que se quiere es publicar el cubo. Para publicar un cubo, se siguen los siguientes pasos:

- I. Seleccionar en la barra de menú, la opción “File”.
- II. Una vez seleccionada, de la lista de opciones seleccionar la opción “Publish...”
- III. A continuación aparece una nueva ventana, en la que se configuran los datos de publicación. La interfaz maneja dos grandes paneles, llamados “Server” y “Pentaho credentials” respectivamente. Inicialmente, solo se cuenta con los datos del panel “Pentaho credentials”, ya que corresponden al nombre de usuario y contraseña con el que se accede a la consola de usuario; pero en el panel de “Server” se debe configurar una contraseña de publicación desde los archivos de configuración predeterminados de Pentaho. Para este fin se seguirán los siguientes pasos:
 - a. Ir al siguiente directorio: “C:\Pentaho\biserver-ce-4.5.0stable\biserver-ce\pentaho-solutions\system”.
 - b. Abrir el archivo “Publisher_config.xml”. Para este fin podemos utilizar la herramienta Notepad++.
 - c. La contraseña será definida por medio del comando “<publisherpassword>”, en el cual escribirá la contraseña de su preferencia luego de haberse definido el comando. Finalmente, guardar la configuración del archivo. Se obtendría algo parecido a la imagen que se presenta a continuación:

```
<publisher-config>  
  <publisher-password>password</publisher-password>  
</publisher-config>
```

Imagen 23: Definición de contraseña para la publicación de cubos.

En este caso, y para presentar un ejemplo, la contraseña de publicación que se ha definido es “password”.

- IV. Una vez modificado este archivo, se procede a introducir la contraseña de publicación en el panel “Server”. Es recomendable reiniciar el Schema Workbench para que sea cargada la nueva configuración.

Realizados esta serie de pasos, se procede nuevamente a abrir la ventana de publicación en la cual encontramos los dos paneles anteriormente mencionados. En el panel “Server”, se muestra una URL predeterminada en la configuración de Pentaho, y seguidamente un cuadro de texto en el que se introducirá la contraseña anteriormente creada. Para presentar un ejemplo, se toma la contraseña de publicación creada en la imagen 23.

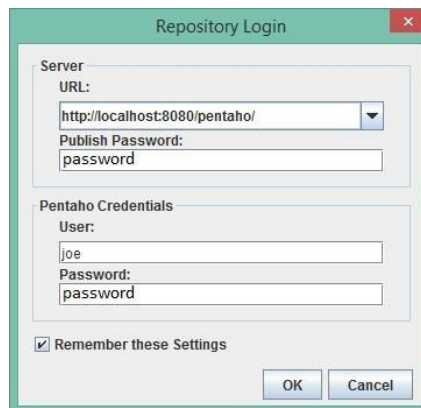


Imagen 24: Introduciendo la contraseña de publicación.

Es muy importante en este paso tener en cuenta lo que se definió anteriormente (Sección 3.2.2) sobre la activación de la consola de administración y de usuario; (ambas deben estar activadas antes de realizar la publicación de un cubo) de lo contrario, a la hora de oprimir el botón “OK” de la anterior imagen, la respuesta del Schema workbench sería la siguiente:

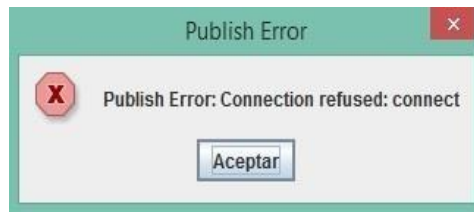


Imagen 25: Error al realizar la publicación de un cubo.

Si se fija detenidamente, es el mismo error que se podía observar inicialmente desde la consola de administración (Ver Imagen 14), y la solución que se planteó para este error, consistió en activar las consolas de administración y de usuario para permitir la comunicación entre estas. Los pasos para evitar este error son:

- I. Abrir la siguiente ruta en el equipo: “C:\Pentaho\biserver-ce-4.5.0stable\administration-console\start-pac”.
- II. Abrir la siguiente ruta en el equipo: “C:\Pentaho\biserver-ce-4.5.0stable\biserver-ce\start-pentaho”.
- III. En caso de que se tenga el Schema Workbench abierto, se recomienda cerrarlo, y seguir los dos pasos anteriores.
- IV. Abrir nuevamente el schema workbench: “C:\Pentaho\schema-workbench\workbench” y seleccionar nuevamente la opción para publicar un cubo.

Al intentar publicar de nuevo el cubo, el error no se debe presentar, y por lo tanto se debe visualizar una ventana como la siguiente:

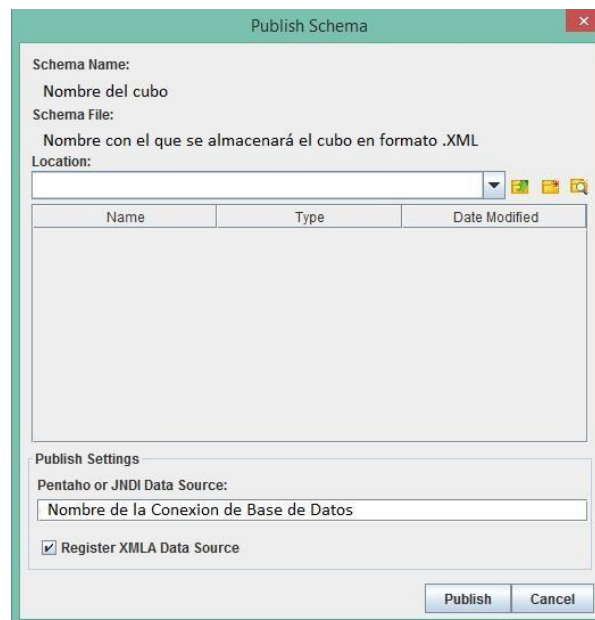


Imagen 26: Publicación de un cubo.

El “Schema name” es el nombre que se le asigna al cubo. El “Schema File” muestra el nombre que se asignó al momento de guardar el cubo y finalmente en “Pentaho or JNDI Data Source”, se introduce el nombre que se asignó a la cadena de conexión anteriormente definida: Luego se hace clic en el botón “Publish”, y si todo quedo bien configurado se debería mostrar el siguiente mensaje:



Imagen 27: Publicación de cubo exitosa.

3.3 Saiku.

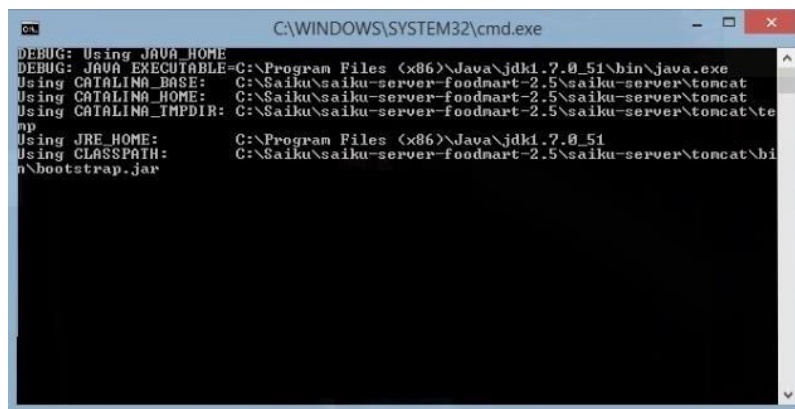
Es un visor OLAP, el cual permite obtener los cubos realizados en Pentaho y presentarlos de una forma amigable y sencilla de manejar, ya que posee elementos gráficos como la interacción visual con las dimensiones y sus respectivos atributos, diversidad de gráficos a la hora de presentar datos, la forma en la cual se puede realizar una consulta entre otras. Mediante esta herramienta, se podrá simplificar la forma en la que el usuario final consulta la información que está contenida en el cubo.

Cabe aclarar que el manual corresponde a la instalación de Saiku como servidor independiente, es decir no se integra con Pentaho. Lo que se describe en esta sección del documento, consiste en cómo se puede visualizar un cubo realizado en Pentaho a través del visor OLAP de Saiku, y de igual manera como crear permisos para los distintos roles de usuario, con el fin de permitir o denegar permisos sobre un cubo determinado.

3.3.1 Configuración Variables de Entorno.

Al igual que Pentaho, Saiku también necesita el entorno de desarrollo de Java para funcionar adecuadamente. En esta sección del documento, se asume que ya se configuraron las variables de entorno especificadas en la sección 3.2.1, ya que son las mismas con las que trabaja Saiku. En caso de no ser así, revisar nuevamente la sección 3.2.1 y seguir cada uno de los pasos allí definidos para solucionar este error.

Si al intentar subir el servicio de Saiku, aparece un error como el que es presentado en la imagen 8, quiere decir que las variables de entorno aún no han sido configuradas. De lo contrario, si al ejecutar el servicio de Saiku y aparece algo similar a la siguiente imagen:



```
C:\WINDOWS\SYSTEM32\cmd.exe
DEBUG: Using JAVA_HOME
DEBUG: JAVA_EXECUTABLE=C:\Program Files (x86)\Java\jdk1.7.0_51\bin\java.exe
Using CATALINA_BASE: C:\Saiku\saiku-server-foodmart-2.5\saiku-server\tomcat
Using CATALINA_HOME: C:\Saiku\saiku-server-foodmart-2.5\saiku-server\tomcat
Using CATALINA_TMPDIR: C:\Saiku\saiku-server-foodmart-2.5\saiku-server\tomcat\temp
Using JRE_HOME: C:\Program Files (x86)\Java\jdk1.7.0_51
Using CLASSPATH: C:\Saiku\saiku-server-foodmart-2.5\saiku-server\tomcat\bin\bootstrap.jar
```

Imagen 28: Inicio de Servicio en Saiku

Se puede dar cuenta que el servicio se ejecuta correctamente, y se ofrece acceso a la consola de usuario, en la cual se podrá observar y manipular cada uno de los cubos implementados por medio de las herramientas ofrecidas en Pentaho.

3.3.2 Creación de un archivo de fuente de Datos.

Antes de definir el proceso para visualizar un cubo implementado bajo la suite de programas de Pentaho en el visor OLAP de Saiku, es necesario crear una serie de configuraciones para enlazar y presentar los archivos XML que contienen la codificación de cada cubo en la interfaz de usuario.

Un archivo de fuente de datos, es aquel que contiene las configuraciones y parámetros adecuados, que permiten a Saiku leer los archivos en formato .XML de los cubos realizados anteriormente en Pentaho, y de esta manera presentarlos en la interfaz de usuario, para que este pueda interactuar con la información de un cubo determinado.

Se definen los siguientes pasos a seguir, los cuales son de gran importancia para poder continuar con los temas que se explican posteriormente.

- I. Descargar el visor OLAP de Saiku, este puede ser descargado desde la página oficial <http://meteorite.bi/saiku/download>. En este caso, se utiliza la versión “Saiku Server 2.5”.
- II. Una vez descargado, se extrae la carpeta del programa y se guarda en la raíz principal del disco duro. (En este caso C:\)

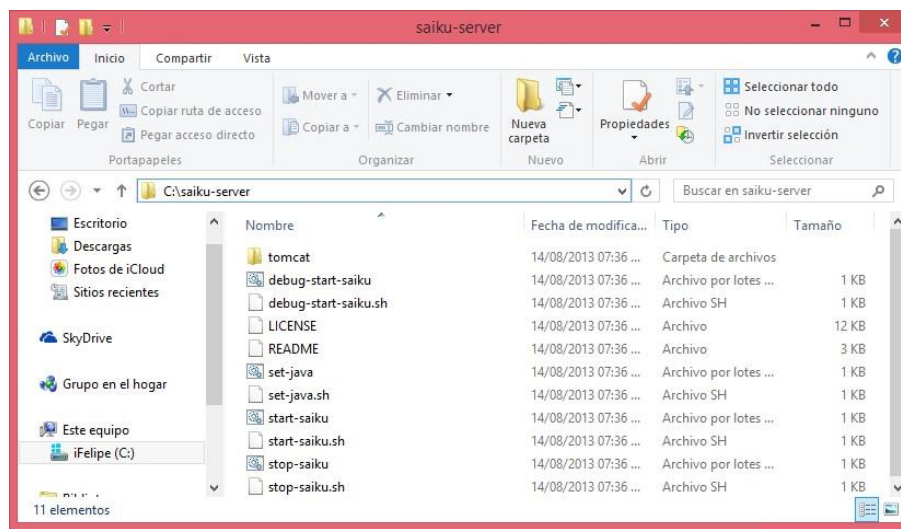


Imagen 29: Saiku almacenado en el directorio principal C:\

Una vez realizados los pasos anteriores, se debe visualizar lo que se presenta en la anterior imagen. Por otro lado, se especifican una serie de pasos para crear un archivo de fuente de datos:

- I. Dirigirse a la siguiente ruta “C:\saiku-server\tomcat\webapps\saiku\WEB-INF\classes”.
- II. Abrir la carpeta “saiku-datasources”.
- III. Copiar el archivo llamado “README” y pegarlo en el mismo directorio bajo otro nombre. (En este caso “README –copia”).
- IV. Utilizando el programa Notepad++, abrir la copia del archivo anteriormente mencionado.

En este archivo, se presentan algunas configuraciones desde el punto de vista de varios motores de bases de datos, pero específicamente, se va a utilizar el que viene definido para MySQL.

- V. Buscar una línea de texto que contenga “# SAMPLE DATASOURCES” y a continuación, buscar otra línea que contenga “Mondrian on MySQL”.

Una vez realizado el paso anterior, se observa un conjunto de líneas de texto definidas para el motor de base de datos MySQL.

- VI. Crear un nuevo archivo en el programa Notepad++, “Archivo – Nuevo”. (En este caso se llama “Ejemplo.txt”)
- VII. Copiar todas las líneas de texto correspondientes a MySQL y pegarlas en el archivo anteriormente creado.

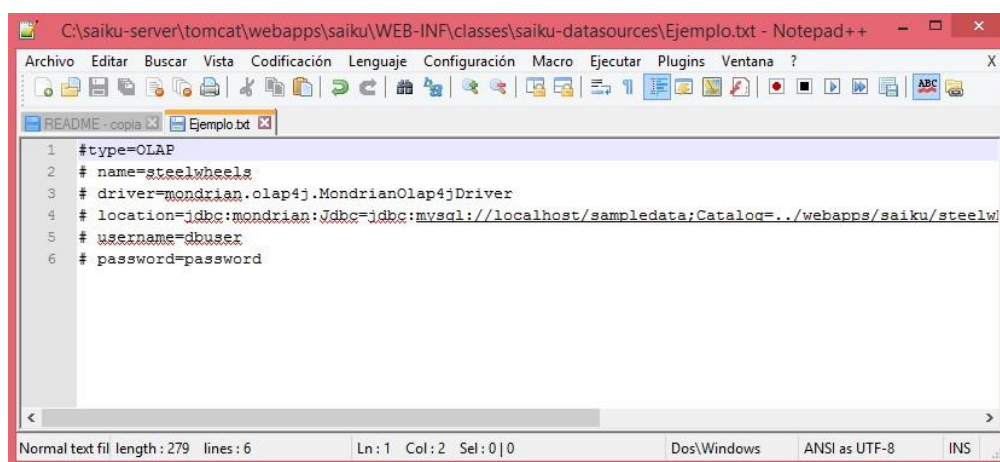
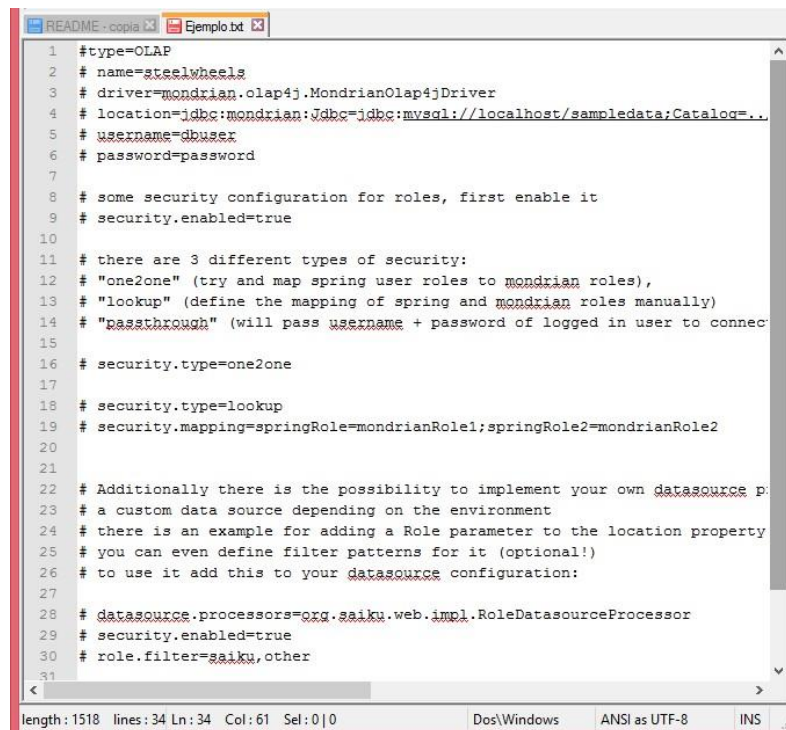


Imagen 30: Creación de un Archivo de Fuente de Datos.

Hasta el momento, se debería tener algo muy parecido a lo presentado en la anterior imagen.

- VIII. Regresar al archivo copia de “README”, y buscar la siguiente línea de texto “SECURITY CONFIGURATION”.
- IX. Copiar el conjunto de líneas de texto que se muestran seguidamente de la línea de texto del paso anterior.
- X. Pegar el conjunto de líneas de texto en el archivo nuevo que se había creado pasos atrás.(Ejemplo.txt)

Asumiendo que se realizaron todos los pasos anteriormente mencionados, se debería tener algo como lo que se presenta seguidamente:



```
1 #type=OLAP
2 # name=stealwheels
3 # driver=mondrian.olap4j.MondrianOlap4jDriver
4 # location=jdbc:mondrian:jdbc:mysql://localhost/sampledata;Catalog=...
5 # username=dbuser
6 # password=password
7
8 # some security configuration for roles, first enable it
9 # security.enabled=true
10
11 # there are 3 different types of security:
12 # "oneZone" (try and map spring user roles to mondrian roles),
13 # "lookup" (define the mapping of spring and mondrian roles manually)
14 # "passthrough" (will pass username + password of logged in user to connect)
15
16 # security.type=oneZone
17
18 # security.type=lookup
19 # security.mapping=springRole=mondrianRole1;springRole2=mondrianRole2
20
21
22 # Additionally there is the possibility to implement your own datasource processor
23 # a custom data source depending on the environment
24 # there is an example for adding a Role parameter to the location property
25 # you can even define filter patterns for it (optional!)
26 # to use it add this to your datasource configuration:
27
28 # datasource.processors=org.saiku.web.impl.RoleDatasourceProcessor
29 # security.enabled=true
30 # role.filter=saiiku,other
31
```

Imagen 31: Ejemplo Archivo de fuente de Datos.

- XI. Se debe copiar toda la información que contiene el archivo en el cual se pegaron todas las configuraciones anteriormente mencionadas (En este caso “Ejemplo.txt”), y pegarla en la copia del archivo “README”, con el fin de guardar la configuración en un formato de archivo adecuado que el cual pueda reconocer Saiku.

De esta manera, se crea un archivo de fuente de datos. Posteriormente se habla sobre la configuración de los parámetros necesarios para enlazar un archivo XML (El cual contiene la codificación del cubo) y presentarlo en la interfaz de usuario de Saiku.

3.3.3 Abrir Cubos de Pentaho en Saiku.

Siendo Saiku un visor OLAP, solo se encarga de presentar en la interfaz de usuario todos los datos que están dentro de un cubo determinado, por lo que esta herramienta solo está enfocada a la parte visual de la información y por lo tanto, no se pueden crear cubos como lo ofrece la suite de programas de Pentaho.

Suponiendo que anteriormente se ha creado un cubo mediante el Schema Workbench de Pentaho se inicia la explicación de la presente sección. La presente definición, se realizara por medio de un ejemplo, con el fin de que el lector pueda entender de una forma más sencilla la serie de procesos que se desarrollaran a continuación. Se implementó un cubo llamado “FoodMart” por medio del Schema Workbench, el cual va a ser guardado en formato XML y publicado en la consola de usuario por medio de este programa.

Usualmente, los archivos de los cubos bajo formato XML, son almacenado en la carpeta de Pentaho-solutions, “C:\Pentaho\biserver-ce-4.5.0-stable\biserver-ce\pentahosolutions”. Estos archivos no pueden ser almacenados dentro de la carpeta raíz de Pentaho, por lo que será necesario almacenarlos en una carpeta creada dentro de este directorio. Para este ejemplo, el cubo “FoodMart” será almacenado en una carpeta llamada “Modelos”.

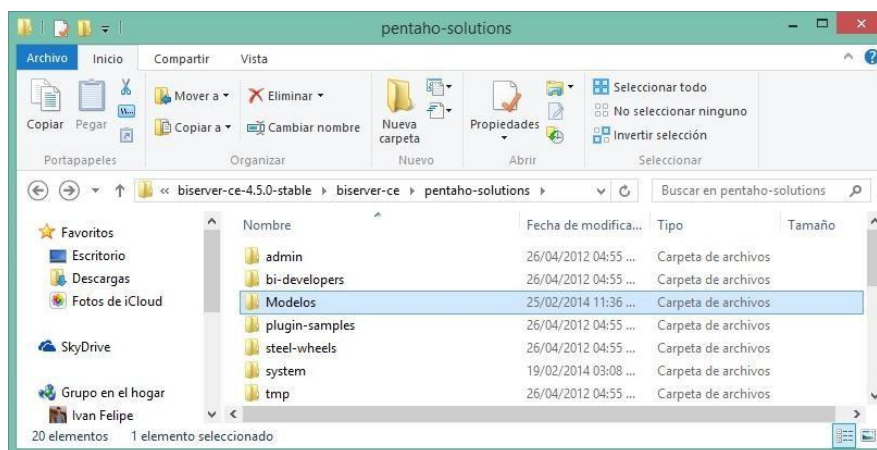


Imagen 32: Creación carpeta “Modelos” en Pentaho.

Se presentan una serie de pasos, los cuales permiten visualizar un cubo realizado en Pentaho desde la interfaz de usuario de Saiku:

- I. Copiar el archivo XML “FoodMart” almacenado en la carpeta “Modelos” en el directorio de Pentaho.

- II. Crear una carpeta dentro de la carpeta principal de archivos de Saiku, con el objetivo de almacenar el archivo XML “FoodMart”. Específicamente en el siguiente directorio: “C:\saiku-server\tomcat\webapps\saiku”. (En este ejemplo, la carpeta se llamara “Foodmart”).

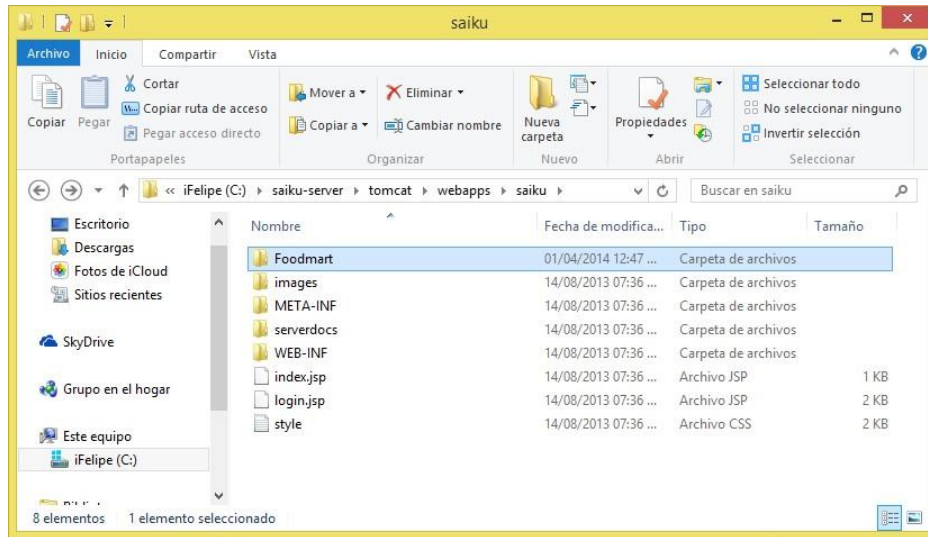


Imagen 33: Creación carpeta “Foodmart” dentro de los archivos de Saiku.

- III. Pegar el archivo en “FoodMart” en la carpeta creada anteriormente.
- IV. Dirigirse a la ruta: “C:\saiku-server\tomcat\webapps\saiku\WEB-INF\classes\saiku-datasources”.

Esta carpeta contiene los archivos de las fuentes de datos, de los que se habló en la sección 3.3.2, los cuales son necesarios para visualizar un cubo determinado en la interfaz de usuario de Saiku.

- V. Copiar el archivo en el cual se guardó la configuración de la fuente de datos creada en la sección 3.3.2. (En este caso el archivo se llama “README copia”).
- VI. Guardar la copia del archivo en este mismo directorio asignando un nombre relacionado al archivo XML que esté utilizando, en este caso la copia del archivo se llamara “foodmart”.

Una vez realizados los pasos anteriores, se procede a modificar el archivo “foodmart”, en el cual se van a introducir los parámetros necesarios para enlazar el archivo del cubo “FoodMart”. Primero que todo se deben tener en cuenta los siguientes pasos.

- I. Abrir el archivo “foodmart” con la herramienta Notepad++.

Al ser una copia del archivo de fuente de datos creado en la sección 3.3.2, se debe visualizar exactamente lo que se presenta en la imagen 31.

- II. Se debe quitar el símbolo “#” de las primeras seis líneas del documento, con el fin de habilitar las líneas de texto.

De esta manera, Saiku reconoce las respectivas configuraciones de cada una de las líneas de texto de este archivo. Por lo tanto, en las primeras seis líneas del documento se debe tener lo que se presenta a continuación:

```
type=OLAP
name=steelwheels
driver=mondrian.olap4j.MondrianOlap4jDriver
location=jdbc:mondrian:Jdbc=jdbc:mysql://localhost/sampledata;Catalog=../w
username=dbuser
password=password
```

Imagen 34: Habilitando Líneas de Texto en fuente de datos “foodmart”.

Una vez realizado el paso anterior, se procede a configurar estas seis primeras líneas de texto, con los parámetros reales de la base de datos que se está utilizando para obtener los datos que se presentan en el cubo (Generalmente un Data Warehouse).

Algunas de estas líneas de texto, vienen con una configuración predeterminada, la cual no es necesaria modificar, estos campos son:

- “type” y “driver”.

Las líneas de texto en las que se ve obligado a modificar los parámetros son las siguientes:

- **“name”** – Nombre con el cual se guardó el cubo dentro del Schema Workbench.
- **“location”** – Contiene la URL por la cual enlaza a la base de datos correspondiente para cada cubo. Si se presta atención, es la misma URL, que se introduce en la consola de administración, cuando se crea una nueva cadena de conexión (Sección 3.2.2).

- **“Catalog”** – Es la ruta que toma Saiku para abrir el archivo en formato XML que contiene la información del cubo y de esta manera presentarlo en la interfaz de usuario de Saiku.
- **Username & password** – Si el motor de base de datos con el que se está trabajando requiere un nombre de usuario y contraseña para acceder a esta, es necesario ingresarlos en estos campos. Si no se trabaja con un logueo, los campos pueden quedar vacíos.

Realizando la respectiva modificación de los parámetros definidos anteriormente se tiene:

```
type=OLAP
name=foodmart
driver=mondrian.olap4j.MondrianOlap4jDriver
location=jdbc:mondrian:jdbc=jdbc:postgresql://localhost:5432/Foodmart: Catalog=../webapps/saiku/Foodmart/FoodMart.xml;
username=postgres
password=1234
```

Imagen 35: Configuración para presentar el cubo "FoodMart" en la interfaz de Saiku.

En este ejemplo, se trabaja con una base de datos implementada en el motor de base de datos PostgreSQL, por lo cual la línea de texto “location” debe ser modificada con el parámetro asociado a este motor de base de datos. En el campo “Jdbc”, el parámetro asignado es parecido a lo que se definió anteriormente en la sección 3.2.2 (Ver especificaciones en la imagen 18), en la cual se tiene:

Driver de postgresQL://localhost(Numero del Puerto)/(Nombre de la Base de Datos).

De igual manera, es necesario descargar el driver de este motor de base de datos accediendo a la página web <http://jdbc.postgresql.org/>. Este driver, permite realizar la conexión a las bases de datos referenciadas en los archivos de configuración, Una vez descargado el driver realizar los siguientes pasos:

- I. Copiar el archivo descargado desde la página web anteriormente mencionada.
- II. Pegarlo en la siguiente ruta: “C:\saiku-server\tomcat\lib”

Por otra parte, la línea de texto “catalog” contiene la ruta necesaria para poder acceder el archivo “FoodMart.XML”, por lo que en el parámetro, se asignó la ruta en la cual se guardó este archivo en pasos anteriores. De igual forma, en las líneas de texto “username” y “password” se define el usuario y la contraseña que tiene la base de datos “Foodmart”. Como se definió anteriormente, estos campos pueden ser vacíos, en caso de que una base de datos determinada no requiera autenticación de usuario.

Una vez configurados estos campos, se guarda el archivo, y posteriormente se procede a visualizar el cubo desde la interfaz de usuario de Saiku. Para esto se deben seguir los siguientes pasos:

- I. Dirigirse a la siguiente ruta: “C:\saiku-server”.
- II. Abrir el archivo “start-saiku.bat”. (Si la aplicación se ejecuta correctamente, podrá seguir con el siguiente paso; si no se ejecutó, revisar la sección 3.2.1 y configurar las variables de entorno para que la aplicación pueda ser ejecutada).
- III. Desde el explorador de internet de su preferencia, ingrese la siguiente URL: “localhost:8080” (La configuración de la consola de administración viene con este puerto por defecto; Verificar que el servicio de la consola de usuario de Pentaho no este activo, si lo está, desactivar el servicio).
- IV. Al acceder a esta URL, el explorador deberá mostrar un cuadro de Login como el que se presenta a continuación:

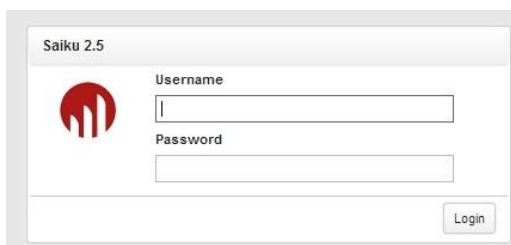


Imagen 36: Login Saiku.

- V. En el campo de nombre de usuario ingresar “admin” y en el campo de contraseña ingresar “admin”.(Datos precargados en los archivos de configuración de Saiku)
- VI. Al ingresar los datos de logueo, se podrá acceder a una interfaz como la que se presenta a continuación:

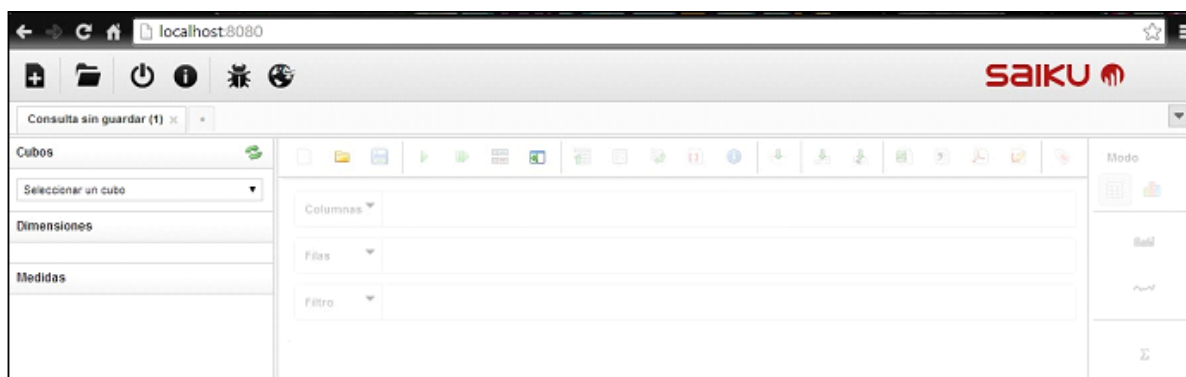


Imagen 37: Interfaz Usuario Saiku.

Para visualizar el cubo de “FoodMart”, este debe ser elegido desde la lista de opciones en la cual dice “Seleccionar un Cubo”



Imagen 38: Cubos disponibles para visualizar por medio de Saiku.

Finalmente, se elige el cubo con el cual se quiere trabajar, y Saiku presentara todos los datos que este contenga. De esta forma, el usuario puede visualizar un cubo hecho con las herramientas de Pentaho en Saiku. Para cada cubo que se desee visualizar en Saiku, se deben realizar todos los pasos mencionados en esta sección.

3.3.4 Roles de Usuario.

Se tiene la necesidad en algunas ocasiones, restringir los cubos a personas o departamentos laborales de una empresa, dado que el cubo puede ser visto por gran variedad de personas, y se podría presentar algún problema respecto a la seguridad y privacidad de la información que estén contenidos en estos.

Saiku permite crear roles de usuario precisamente para dar permisos en determinados cubos, siendo totalmente seguro y confiable. Para lograr esto se deberá seguir los siguientes pasos.

- I. Abrir la siguiente ruta: “C:\saiku-server\tomcat\webapps\saiku\WEB-INF”
- II. Abrir el archivo “users.properties” utilizando Notepad++.

Este archivo contiene la información de los usuarios que pueden acceder mediante un Login a la interfaz de Saiku.

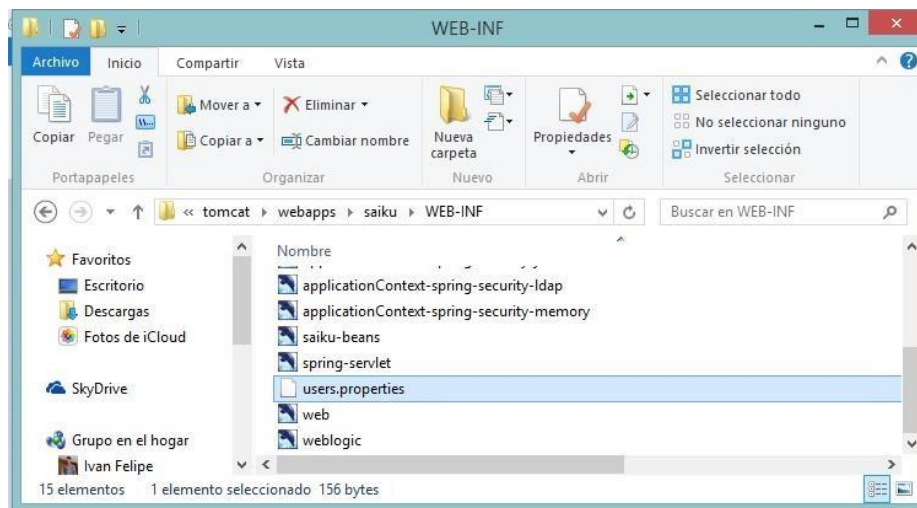


Imagen 39: Directorio para crear nuevos usuarios.

Al abrir archivo se puede observar la siguiente información de los usuarios, en el que se definen los nombres de usuario con su respectiva contraseña de logueo, y el rol que posee dentro de Saiku. (Estos roles viene por defecto de Saiku).

```

1 #Username,password,role
2 bob=dyllan,ROLE_USER
3 krishna=krish2341,ROLE_USER
4 smith=pravah@001,ROLE_USER
5 admin=admin,ROLE_USER,ROLE_ADMIN
6 postgres=1103,ROLE_USER
  
```

Imagen 40: Usuarios y Roles.

Dentro de este archivo, se ha adicionado un nuevo usuario llamado “postgres” con contraseña “1103” y rol de usuario dentro de Saiku. Si se necesita adicionar nuevos usuarios o roles, por medio de este archivo se puede realizar este tipo de configuraciones.

A continuación, se explica cómo habilitar la seguridad para la asignación de permisos en cada uno de los roles.

- I. Ir a la siguiente dirección: “C:\saiku-server\tomcat\webapps\saiku\WEB-INF\classes\saiku-datasources”.
- II. Abrir el archivo de configuración “foodmart”, utilizando Notepad++.

Inicialmente, se observa que se trabaja sobre el mismo archivo en los que se realizan las configuraciones para presentar un cubo de Pentaho en Saiku. Estas configuraciones se dejaran totalmente quietas, y solo se enfocara en la parte del archivo que se presenta a continuación:

```
# some security configuration for roles, first enable it
# security.enabled=true

# there are 3 different types of security:
# "one2one" (try and map spring user roles to mondrian roles),
# "lookup" (define the mapping of spring and mondrian roles manually)
# "passthrough" (will pass username + password of logged in user to connection, e.g. jdbc user + password)

# security.type=one2one

# security.type=lookup
# security.mapping=springRole=mondrianRole1;springRole2=mondrianRole2
```

Imagen 41: Configuración de Seguridad en el cubo "foodmart".

Primeramente, todas las líneas de texto vienen con el símbolo “#” al inicio de cada definición de una línea de texto. Cuando el sistema de Saiku reconoce estas líneas, toma todo lo que inicie con el símbolo “#” como código deshabilitado por lo que no realizara ninguna función importante por el momento.

Para habilitar las líneas de texto, se debe eliminar este símbolo. En este caso, solo se elimina este símbolo de las siguientes líneas de texto:

1. **“security.enabled=true”** - Se define que la seguridad en Saiku ha sido habilitada
2. **“security.type=lookup”** – Define el tipo de seguridad que utiliza Saiku.
3. **“security.mapping”** - Se asignan los permisos para cada tipo de rol.

Según lo anterior se debe tener lo siguiente:

```
# some security configuration for roles, first enable it
security.enabled=true

# there are 3 different types of security:
# "one2one" (try and map spring user roles to mondrian roles),
# "lookup" (define the mapping of spring and mondrian roles manually)
# "passthrough" (will pass username + password of logged in user to connection, e.g. jdbc user + password)

# security.type=one2one

security.type=lookup
security.mapping=springRole=mondrianRole1;springRole2=mondrianRole2
```

Imagen 42: Habilitando Configuraciones de Seguridad.

Luego de haber habilitado las líneas de texto anteriormente mencionadas, se procede a configurar los roles de usuario por medio de la línea de texto “security.mapping”. Como se observa en la imagen 39, existen dos tipos de roles definidos en las configuraciones predeterminadas de Saiku, estos son “ROLE_USER” y “ROLE_ADMIN”. Para cada uno de estos roles, se crea un “alias” con el cual van a ser reconocidos al momento de permitir o denegar el acceso a un cubo. ROLE_USER, será nombrado como “publico” y ROLE_ADMIN, será

nombrado como “privado”, estos nombres se asignan, para que Saiku pueda diferenciar un rol de otro, ya que no se puede implícitamente utilizar el nombre que tienen por defecto en el archivo de configuración de usuarios.

```
# some security configuration for roles, first enable it
security.enabled=true

# there are 3 different types of security:
# "one2one" (try and map spring user roles to mondrian roles),
# "lookup" (define the mapping of spring and mondrian roles manually)
# "passthrough" (will pass username + password of logged in user to connection, e.g. jdbc user + password)

# security.type=one2one

security.type=lookup
security.mapping=ROLE USER=publico;ROLE ADMIN=privado
```

Imagen 43: Asignando un "Alias" a los roles predeterminados de Saiku.

En la anterior imagen se define que para el rol global de Saiku “ROLE_USER” su nombre para asignar permisos en un cubo será “publico”, de igual manera para “ROLE_ADMIN” bajo el nombre de “privado”.

Se guarda la nueva configuración del archivo, y se siguen los siguientes pasos:

- I. Buscar el archivo XML, relacionado con el archivo al cual se realizó la nueva configuración (buscar el archivo XML del cubo FoodMart el cual se encuentra en la siguiente ruta) “C:\saiku-server\tomcat\webapps\saiku\Foodmart”.
- II. Abrir el archivo XML utilizando Notepad++, se debe visualizar algo parecido a la imagen que se presenta a continuación:

Básicamente, lo que se presenta en este archivo es todo lo que se hizo anteriormente desde la suite de programas de Pentaho, solo que se presenta de forma codificada cada dimensión, métrica y tabla de hechos. Ahora se agregaran unas nuevas líneas de código, las cuales definirán los permisos que tiene cada rol sobre este cubo.

- III. Con el archivo XML abierto, buscar una línea que contenga la siguiente sintaxis: “</cube>”.

Después de esta línea, se definen los roles para los usuarios. La sintaxis </cube>, permite saber que hay termina lo codificación de un cubo, y que a partir de ahí se pueden realizar otras configuraciones, de tal manera que no dañamos la codificación del cubo.

- IV. Configurar los roles, de acuerdo a la siguiente imagen:

```

</Cube>
  <Role name="publico">
    <SchemaGrant access="none">
      <cubeGrant cube="FoodMart" access="none">
      </cubeGrant>
    </SchemaGrant>
  </Role>
  <Role name="privado">
    <SchemaGrant access="all">
      <cubeGrant cube="FoodMart" access="none">
      </cubeGrant>
    </SchemaGrant>
  </Role>
</Schema>

```

Imagen 44: Sintaxis para asignar de permisos en los roles de usuario.

Hasta el momento solo se tienen dos tipos de roles definidos: “publico”=ROLE_USER y “privado”=ROL_ADMIN, por lo que en la anterior imagen solo se visualiza el código para estos dos roles, si hubiese más de dos roles, esta misma codificación se debería agregar en el respectivo espacio, modificando el nombre del rol y el acceso que tendrá al cubo.

Por medio del comando “**Role**”, se crea un nuevo rol, al que se le asignaran o negaran permisos sobre el cubo. Por otra parte, el comando “**SchemaGrant**” define el acceso al esquema, por eso seguidamente se observa el comando “**access=**””, cuando este comando es igual a “**none**” se define que un rol determinado no podrá visualizar el esquema especificado, por otra parte, cuando el valor que toma el comando “**access=**” es igual a “**all**” el rol podrá acceder a todos el contenido del esquema.

El otro comando que puede ser observado más adelante (“**cubeGrant**”), tiene como función, asignar los permisos a un cubo, al que tendrá acceso el usuario. Siendo así, cuando a este comando se le asigna el valor de “**none**”, se especifica que el usuario no podrá acceder al cubo, por lo contrario, cuando el valor asignado es “**all**”, el usuario podrá acceder a todos los componentes de un cubo sin ningún inconveniente.

Finalmente, al modificar la anterior información en el archivo XML, es necesario guardar los cambios realizados y reiniciar el servicio de Saiku, con el fin de que este aplique las nuevas configuraciones.

A continuación se presenta la prueba del funcionamiento de las nuevas configuraciones realizadas.

- I. Entrar a la consola de Saiku (URL: localhost: 8080, desde cualquier explorador de internet).

- II. Ingresar a Saiku, utilizando alguno de los datos de usuario presentados en el archivo de configuración de usuarios. (Empezando por cualquiera que tenga un rol predeterminado “ROLE_USER”, en este ejemplo se utiliza el usuario creado bajo el nombre de “postgres” y contraseña “1103”).
- III. Según la imagen 44, el cubo “FoodMart” no debe ser visible a los usuarios que se encuentren bajo un rol “publico”. Esto fue lo que apareció a la hora de ver los cubos que puede observar “postgres” bajo el rol “publico”:



Imagen 45: El usuario "postgres" no puede ver ningún cubo.

Por otro lado, si se accede bajo el alias de “privado”, el cual fue declarado bajo el rol predeterminado “ROL_ADMIN” se debería ver el cubo. Esto fue lo que se presentó:



Imagen 46: Rol privado con acceso al cubo FoodMart.

De esta manera se realiza la asignación de permisos en los roles de usuario de Saiku. Cabe aclarar, que un usuario bajo el rol global “ROL_ADMIN” siempre va a tener acceso a todos los cubos, así se intente negar el permiso a este.

Se debe realizar todo este procedimiento para los cubos a los que se quiere permitir o negar el acceso, presenta un poco de complicación, pero al final si garantiza la seguridad de los cubos ante las personas que no deben de tener acceso a este.

3.4 Herramientas de Trabajo Adicionales.

En lo largo de todo el documento, se habla de otras herramientas que son utilizadas, pero no serán detalladas debido a que el documento se centra en XAMPP, Pentaho y Saiku:

Estas herramientas son:

- **Notepad++** - Es la herramienta que permite modificar algunos de los archivos en formato XML y de configuración de Pentaho y Saiku, con el fin de adaptarlos a las configuraciones que sean necesarias para que funcionen adecuadamente.
- **Motor de Base de Datos PostgreSQL** – En este motor de base de datos, se creará el Data Warehouse que contendrá toda la información del modelo después de haber realizado el proceso de ETL de la fuente de datos original.

4. Referencias.

[1]. AYUDA BITTORRENT. “JRE (Java Runtime Environment)”. {En Línea}. {10 de Marzo de 2014} disponible en: (<http://www.ayudabittorrent.com/jre>).

[2]. BLOGSPOT. “XAMPP”. {En Línea}. {11 de Marzo de 2014} disponible en: (<http://myu-charly.blogspot.com>).

[3]. BUSINESS INTELLIGENCE. “¿Qué es OLAP?” {En Línea}. {21 de Marzo de 2014} disponible en: (<http://www.businessintelligence.info/definiciones/que-es-olap.html>).

[4]. DATA WAREHOUSES. “ETL Process”. {En Línea}. {13 de Marzo de 2014} disponible en: (<http://data-warehouses.net/architecture/etlprocess.html>)

[5]. ECURED. “Sistema Operativo”. {En Línea}. {21 de Marzo de 2014} disponible en: (http://www.ecured.cu/index.php/Sistema_operativo)

[6]. NANYANG TECHNOLOGICAL UNIVERSITY. “Environment Variables for Java Applications”. {En Línea}. {10 de Marzo de 2014} disponible en: (http://www3.ntu.edu.sg/home/ehchua/programming/howto/Environment_Variables.html)

[7]. SINEXUSS. “Datamart”. {En Línea}. {20 de Marzo de 2014} disponible en: (http://www.sinnexus.com/business_intelligence/datamart.aspx)

[8]. SINNEXUS. “Datawarehouse”. {En Línea}. {21 de Marzo de 2014} disponible en: (http://www.sinnexus.com/business_intelligence/datawarehouse.aspx)

[9]. UNIVERSIDAD DE ALCALÁ. “Manual de Instalación de la Máquina Virtual de Java (JVM)”. {En Línea}. {10 de Marzo de 2014} disponible en: (<http://www3.uah.es/ice/AV/Documentos/Manuales/JVM.pdf>)

[10]. UNIVERSIDAD DE NAVARRA. “Que es JDK (Java Development Kit)”. {En Línea}. {10 de Marzo de 2014} disponible en: (<http://www.esi.unav.es/Asignaturas/Informat2/Clases/Clases9899/Clase01/JavaEntorno/tsld003.htm>)

[11]. WIKIPEDIA. “Hypertext Transfer Protocol Secure”. {En Línea}. {31 de Marzo de 2014} disponible en: (http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol_Secure)

- [12]. WIKIPEDIA. “Inteligencia Empresarial”. {En Línea}. {21 de Marzo de 2014} disponible en: (http://es.wikipedia.org/wiki/Inteligencia_empresarial)
- [13]. WIKISPACES. “Modelado Dimensional”. {En Línea}. {13 de Marzo de 2014} disponible en: (<http://bddimensionales.wikispaces.com/Modelado+Dimensional>)
- [14]. WORDPRESS. “¿Qué es una Staging Area?”. {En Línea}. {21 de Marzo de 2014} disponible en: (<http://informationmanagement.wordpress.com/2007/10/15/%C2%BFque-es-una-staging-area/staging-area>)