

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

10 de mayo de 2013
Autor: José Luis Cunillera

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Contenido

INTRODUCCIÓN	2
INSTALACIÓN DE PENTAHO BI SERVER 4.8 EN MYSQL	3
PASO 1. CREACIÓN DE LA UBICACIÓN DEL PRODUCTO	3
PASO 2. DESCARGAR Y DESEMPAQUETAR EL PRODUCTO BI SERVER Y SAIKU.....	3
PASO 3. VERIFICAR QUE JAVA ESTÁ BIEN INSTALADO Y REFERENCIADO.	4
PASO 4. CREACIÓN DE LOS USUARIOS Y LAS BASES DE DATOS EN MYSQL.....	5
PASO 5. VARIABLES DE ENTORNO	6
PASO 6. ACTUALIZAR FICHERO: APPLICATIONCONTEXT-<FRAMEWORK>-SECURITY- HIBERNATE.PROPERTIES	7
PASO 7. ACTUALIZAR FICHERO: APPLICATIONCONTEXT-<FRAMEWORK>-SECURITY- JDBC.XML	8
PASO 8. ACTUALIZAR FICHERO: PUBLISHER_CONFIG.XML.....	8
PASO 9. ACTUALIZAR FICHERO: HIBERNATE-SETTINGS.XML.....	9
PASO 10. ACTUALIZAR FICHERO: WEB.XML.....	9
PASO 11. ACTUALIZAR FICHERO: CONTEXT.XML	10
PASO 11. ACTUALIZAR SEGUNDO FICHERO DE CONTEXTO: CONTEXT.XML.....	11
PASO 12. VERIFICAR FICHERO: HQL_DATASOURCE_MYSQL.CFG.XML.....	12
PASO 13. VERIFICAR FICHERO: APPLICATIONCONTEXT-SPRING-SECURITY-HIBERNATE.PROPERTIES....	13
PASO 14. VERIFICAR FICHERO: MYSQL5.HIBERNATE.CFG.XML	13
PASO 15. VERIFICAR FICHERO OLAP: DATASOURCES.XML.....	14
PASO 16. VERIFICAR FICHERO DE CONEXIONES: JDBC.PROPERTIES.....	15
PASO 17. CAMBIAR EL PASSWORD DE LA CONSOLA DE ADMINISTRACIÓN:.....	16
PASO 18. ARRANCAR LA CONSOLA DE ADMINISTRACIÓN.....	17
PASO 19. CAMBIAR EL USUARIO "JOE" POR EL USUARIO "PUBLICADOR" EN EL ARCHIVO: CONSOLE.XML	18
PASO 20. REINICIAR LA CONSOLA DE ADMINISTRACIÓN E INICIAR PENTAHO BI SERVER.....	19
PASO 21. LIMPIAR LOS CASOS DE EJEMPLO EN CASO DE UN ENTORNO PRODUCTIVO	20
ANEXOS.....	21
ANEXO 1. SCRIPT DE CREACIÓN DE LA BASE DE DATOS 'SAMPLEDATA' PARA MYSQL: CREATE_SAMPLEDATA_MYSQL.SQL.....	21
ANEXO 1. SHELL PARA EL ARRANQUE Y PARADA DE LA CONSOLA DE ADMINISTRACIÓN EN LINUX:.....	23
ANEXO 1. SHELL PARA EL ARRANQUE Y PARADA DEL SERVICIO BI SERVER EN LINUX:	24

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Introducción

Este documento se presenta como una guía para la instalación del producto BI Server de PENTAHO en su versión más actual 4.8 con la base de datos MySQL 5.x.

El objetivo es ser breve, conciso y completo por consiguiente se da paso a las instrucciones de instalación.

Se incluirá un plugin adicional para mejorar las capacidades del BI Server. El plugin de Saiku.

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Instalación de PENTAHO BI Server 4.8 en MySQL

En la instalación del producto se harán referencia a una serie de directorios y una serie de productos que pueden variar según el sistema operativo, o la voluntad del instalador, o la versión del producto a instalar. Por ejemplo si la versión estable del producto PENTAHO BI Server es otra que la 4.8.0, más nueva, se puede utilizar la más nueva. En cualquier caso es sólo una guía de referencia.

PASO 1. Creación de la ubicación del producto

Para situar el producto en algún lugar creamos un directorio donde se ubicará el producto. A modo de ejemplo crearemos el directorio siguiente para Linux:

En Linux sería una buena opción ubicarlo en /usr/share

```
/usr/share/pentaho/app/product
```

O en windows

```
C:\Pentaho\app\Product
```

PASO 2. Descargar y desempaquetar el producto BI Server y Saiku

En <http://sourceforge.net/projects/pentaho/files> se suelen encontrar los ficheros empaquetados del servidor, para descargarlos e instalarlos. Si hay que buscar una versión nueva es conveniente verlo desde este punto.

En nuestro caso la última versión es la 4.8.0 que se encuentra en :

<http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/4.8.0-stable/biserver-manual-ce-4.8.0-stable.zip>

o para Linux

<http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/4.8.0-stable/biserver-ce-4.8.0-stable.tar.gz>

desde la línea de comandos usaríamos wget para descargar el producto y desempaquetarlo:

```
$> cd /usr/share/pentaho/app/product
$> wget
http://sourceforge.net/projects/pentaho/files/Business%20Intelligence%20Server/4.8.0-stable/biserver-ce-4.8.0-stable.tar.gz
$> tar -zxvf biserver-ce-4.8.0-stable.tar.gz
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

En la versión para Windows sería descargar el archivo biserver-manual-ce-4.8.0-stable.zip con un navegador, copiar el archivo en el directorio C:\Pentaho\app\Product y desempaquetarlo en ese mismo lugar.

Para incluir el plugin de **Saiku** realizaremos lo siguiente:

En Linux:

```
$> cd /usr/share/pentaho/app/product
$> wget --no-check-certificate https://raw.githubusercontent.com/pmalves/ctools-installer/master/ctools-installer.sh

$> chmod +x ctools-installer.sh
$> ./ctools-installer.sh -s /usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/ -w /usr/share/pentaho/app/product/biserver-ce/tomcat/webapps/pentaho

Aceptando todas las opciones
```

Dado que en esta versión el driver de MySQL se encuentra en el directorio de administration-console pero no en el tomcat de bi server hay que copiarlo con:

```
$> cp /usr/share/pentaho/app/product/administration-console/jdbc/mysql-connector-java-5.1.17.jar /usr/share/pentaho/app/product/biserver-ce/tomcat/lib/.
```

PASO 3. Verificar que Java está bien instalado y referenciado.

La instalación del entorno Java no es objetivo de este manual y se deja al creador del JAVA la explicación de dicha instalación. No obstante en lo que respecta a Linux podremos verificar que las variables de entorno están correctamente informadas:

En Linux informar la variable de entorno sería el siguiente comando:

```
$> echo 'JRE_HOME="/usr/lib/jvm/jdk1.7.0/jre"' >> /etc/environment
```

Suponiendo que el entorno Java se encuentra en dicho directorio. Este comando puede variar según la versión de Unix / Linux que se use.

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

PASO 4. Creación de los usuarios y las bases de datos en MySQL

Damos por supuesto que se ha creado una instancia de MySQL y que está operativa en el servidor local. Si estuviera en otro servidor se debería de proceder en consecuencia.

El administrador de MySQL debe crear los usuarios y las bases de datos que se vayan a necesitar para la instalación. Los usuarios necesarios para PENTAHO son hibuser, quartz y pentaho_user. Las bases de datos que tienen que crearse para que PENTAHO funcione son: quartz, hibernate y sampledata. La base de datos sampledata es para que funcionen los ejemplos.

Los comandos son los siguientes:

Creación de los usuarios

```
$> mysql -uroot -ppassword
```

```
mysql> create user 'hibuser'@'localhost' identified by 'password';
mysql> create user 'quartz'@'localhost' identified by 'password';
mysql> create user 'pentaho_user'@'localhost' identified by 'password';
mysql> create user 'hibuser'@'%' identified by 'password';
mysql> create user 'quartz'@'%' identified by 'password';
mysql> create user 'pentaho_user'@'%' identified by 'password';
mysql> exit
```

Creación de las bases de datos

```
$> cd /usr/share/pentaho/app/product/biserver-ce/data/mysql5
$> mysql -uroot -ppassword < create_quartz_mysql.sql
$> mysql -uroot -ppassword < create_repository_mysql.sql
$> mysql -uroot -ppassword < create_sample_datasource_mysql.sql
```

Obtenemos el script **create_sampledate_mysql.sql** del anexo 1 de este documento y creamos la base de datos sampledata

```
$> mysql -uroot -ppassword < create_sampledata_mysql.sql
```

Se modifica una fila en la base de datos hibernate para que pueda conectarse PENTAHO.

```
$> use hibernate
$> UPDATE DATASOURCE
      SET URL='jdbc:mysql://localhost:3306/sampledata'
        , DRIVERCLASS='com.mysql.jdbc.Driver'
        , QUERY='select 1;'
      WHERE NAME='SampleData';
$> commit;
$> exit
```

Para cargar datos de prueba en sampledata podemos hacer lo siguiente:

```
$> cd /usr/share/pentaho/app/product/biserver-ce/data/hsqldb
$> grep "^INSERT" sampledata.script > /tmp/sampledata.sql
$> cd /tmp
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

```
$> mysql -uroot -ppassword -D sampledata < sampledata.sql
$> rm sampledata.sql

Ahora se le dan permisos de acceso a los usuarios creados a las bases de datos creadas:

$> mysql -uroot -ppassword

mysql> grant all on `hibernate`.* to 'hibuser'@'localhost' identified by 'password';
mysql> grant all on `hibernate`.* to 'quartz'@'localhost' identified by 'password';
mysql> grant all on `hibernate`.* to 'pentaho_user'@'localhost' identified by 'password';
mysql> grant all on `hibernate`.* to 'hibuser'@'%' identified by 'password';
mysql> grant all on `hibernate`.* to 'quartz'@'%' identified by 'password';
mysql> grant all on `hibernate`.* to 'pentaho_user'@'%' identified by 'password';

mysql> grant all on `quartz`.* to 'hibuser'@'localhost' identified by 'password';
mysql> grant all on `quartz`.* to 'quartz'@'localhost' identified by 'password';
mysql> grant all on `quartz`.* to 'pentaho_user'@'localhost' identified by 'password';
mysql> grant all on `quartz`.* to 'hibuser'@'%' identified by 'password';
mysql> grant all on `quartz`.* to 'quartz'@'%' identified by 'password';
mysql> grant all on `quartz`.* to 'pentaho_user'@'%' identified by 'password';

mysql> grant all on `sampledata`.* to 'hibuser'@'localhost' identified by 'password';
mysql> grant all on `sampledata`.* to 'quartz'@'localhost' identified by 'password';
mysql> grant all on `sampledata`.* to 'pentaho_user'@'localhost' identified by 'password';
mysql> grant all on `sampledata`.* to 'hibuser'@'%' identified by 'password';
mysql> grant all on `sampledata`.* to 'quartz'@'%' identified by 'password';
mysql> grant all on `sampledata`.* to 'pentaho_user'@'%' identified by 'password';

Si para nuestra instalación necesitamos un usuario adicional con bases de datos adicionales se pueden crear tambien

$> mysql -uroot -ppassword

mysql> create user 'production_user'@'localhost' identified by 'password';
mysql> create user 'production_user'@'%' identified by 'password';
mysql> create database `production_database` character set utf8;
mysql> grant all on `production_database`.* to 'production_user'@'localhost' identified by 'password';
mysql> grant all on `production_database`.* to 'production_user'@'%' identified by 'password';
mysql> exit
```

PASO 5. Variables de entorno

Se deben fijar las variables de entorno siguientes:

Para LINUX

VARIABLE	VALOR
CATALINA_BASE	/usr/share/pentaho/app/product/biserver-ce/tomcat
CATALINA_HOME	/usr/share/pentaho/app/product/biserver-ce/tomcat

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

CATALINA_OPTS	<code>-Xms256m -Xmx768m -XX:MaxPermSize=256m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000</code>
CATALINA_TMPDIR	<code>/usr/share/pentaho/app/product/biserver-ce/tomcat/temp</code>
JRE_HOME	<code>/usr/lib/jvm/jdk1.7.0/jre</code>

O para WINDOWS:

VARIABLE	VALOR
CATALINA_BASE	<code>C:\Pentaho\app\Product\biserver-ce\tomcat</code>
CATALINA_HOME	<code>C:\Pentaho\app\Product\biserver-ce\tomcat</code>
CATALINA_OPTS	<code>-Xms256m -Xmx768m -XX:MaxPermSize=256m -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000</code>
CATALINA_TMPDIR	<code>C:\Pentaho\app\Product\biserver-ce\tomcat\temp</code>
JRE_HOME	<code>C:\Archivos de programa\Java\jdk1.6.0_14\jre</code>

Los valores de las variables deben ajustarse al sistema operativo y a la ubicación de los productos.

No obstante si no se definen estas variables, los archivos de lanzamiento del producto las define dinámicamente.

PASO 6. Actualizar FICHERO: `applicationContext-<FRAMEWORK>-security-hibernate.properties`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\pentaho-solutions\system\`

o en:

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/`

y que puede tener diferentes nombres pero normalmente se nombra como:

`applicationContext-acegi-security-hibernate.properties`

o como:

`applicationContext-spring-security-hibernate.properties`

Contenido anterior

```
jdbc.driver=org.hsqldb.jdbcDriver
jdbc.url=jdbc:hsqldb:hsqldb://localhost:9001/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.HSQLDialect
```


Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Nuevo contenido

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.MySQL5InnoDBDialect
```

Suponiendo que en el localhost se encuentre el servidor de MySQL y el puerto por el que escucha es el 3306. Si no es así se debe cambiar el servidor y el puerto según corresponda.

PASO 7. Actualizar FICHERO: `applicationContext-<FRAMEWORK>-security-jdbc.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\pentaho-solutions\system\`

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system`

y que puede tener diferentes nombres pero normalmente se nombra como:

`applicationContext-acegi-security-jdbc.xml`

o como:

`applicationContext-spring-security-jdbc.xml`

Contenido anterior

```
<!-- This is only for Hypersonic. Please update this section for any other database you are using -->
<bean id="dataSource"
      class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="org.hsqldb.jdbcDriver" />
  <property name="url"
            value="jdbc:hsqldb:hsqldb://localhost:9001/hibernate" />
  <property name="username" value="hibuser" />
  <property name="password" value="password" />
</bean>
```

Nuevo contenido

```
<bean id="dataSource"
      class="org.springframework.jdbc.datasource.DriverManagerDataSource">
  <property name="driverClassName" value="com.mysql.jdbc.Driver" />
  <property name="url"
            value="jdbc:mysql://localhost:3306/hibernate" />
  <property name="username" value="hibuser" />
  <property name="password" value="password" />
</bean>
```

PASO 8. Actualizar FICHERO: `publisher_config.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\pentaho-solutions\system\`

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system`

y se nombra como:

`publisher_config.xml`

Contenido anterior

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

Nuevo contenido

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

PASO 9. Actualizar FICHERO: `hibernate-settings.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C : \Pentaho\app\Product\biserver-ce\pentaho-solutions\system\hibernate\`

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system /hibernate`

y se nombra como:

`hibernate-settings.xml`

Contenido anterior

```
<config-file>system/hibernate/hsqldb.hibernate.cfg.xml</config-file>
```

Nuevo contenido

```
<config-file> system/hibernate/mysql5.hibernate.cfg.xml</config-file>
```

PASO 10. Actualizar FICHERO: `web.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C : \Pentaho\app\Product\biserver-ce\tomcat\webapps\pentaho\WEB-INF\`

O en

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

`/usr/share/pentaho/app/product/biserver-ce/tomcat/webapps/pentaho/WEB-INF`

y se nombra como:

`web.xml`

Contenido anterior

```
...
<context-param>
  <param-name>base-url</param-name>
  <param-value>http://localhost:8080/pentaho/</param-value>
</context-param>
...
<param-name>TrustedIpAddr</param-name>
<param-value>127.0.0.1</param-value>
<description>Comma separated list of IP addresses of a trusted hosts.</description>
...
<!-- insert additional resource-refs -->
```

Nuevo contenido

```
...
<context-param>
  <param-name>solution-path</param-name>
  <param-value>/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions</param-value>
</context-param>
...
<context-param>
  <param-name>base-url</param-name>
  <param-value>http://172.16.4.101:8080/pentaho/</param-value>
</context-param>
...
<param-name>TrustedIpAddr</param-name>
<param-value>127.0.0.1,172.16.4.101</param-value>
<description>Comma separated list of IP addresses of a trusted hosts.</description>
...
<!-- insert additional resource-refs -->
<security-role>
  <description>security role assignment</description>
  <role-name>PENTAHO_ADMIN</role-name>
</security-role>
...
```

Donde se ha supuesto para este caso que la instalación es en LINUX, el host de PENTAHO BI SERVER se encuentra en la dirección IP 172.16.4.101. Si el BI Server estuviera ubicado en otra dirección habría que cambiarla, y si la instalación fuera en otro directorio habría que cambiar el path de la tercera línea.

PASO 11. Actualizar FICHERO: `context.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\tomcat\webapps\pentaho\META-INF\`

O en

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

`/usr/share/pentaho/app/product/biserver-ce/tomcat/webapps/pentaho/META-INF`

y se nombra como:

`context.xml`

Contenido anterior

```
...
<Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="hibuser" password="password"
  driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:hsqldb://localhost/hibernate"
  validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES" />
<Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="pentaho_user" password="password"
  driverClassName="org.hsqldb.jdbcDriver" url="jdbc:hsqldb:hsqldb://localhost/quartz"
  validationQuery="select count(*) from INFORMATION_SCHEMA.SYSTEM_SEQUENCES"/>
```

Nuevo contenido

```
...
<Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="hibuser" password="password"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/hibernate"
  validationQuery="select 1" />
<Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="pentaho_user" password="password"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/quartz"
  validationQuery="select 1"/>
<Resource name="jdbc/sampledata" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="pentaho_user" password="password"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/sampledata"
  validationQuery="select 1"/>
<Resource name="jdbc/myproduccion_bi1" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="myusuario" password="password"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/myproduccion_bi1"
  validationQuery="select 1"/>
<Resource name="jdbc/myproduccion_bi2" auth="Container" type="javax.sql.DataSource"
  factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
  maxWait="10000" username="myusuario" password="password"
  driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/myproduccion_bi2"
  validationQuery="select 1"/>
...
```

En donde myproduccion_bi1 y myproduccion_bi2 son dos bases de datos con las que pentaho tendrá que trabajar. Se encuentran definidas aquí a modo de ejemplo.

PASO 11. Actualizar segundo fichero de contexto: `context.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\tomcat\conf\`

O en

`/usr/share/pentaho/app/product/biserver-ce/tomcat/conf`

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

y se nombra como:

context.xml

Contenido anterior

```
<Context>
  <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <!-- Uncomment this to disable session persistence across Tomcat restarts -->
  <Manager pathname="" />
</Context>
```

Nuevo contenido

```
<Context>
  <!-- Default set of monitored resources -->
    <WatchedResource>WEB-INF/web.xml</WatchedResource>
  <!-- Uncomment this to disable session persistence across Tomcat restarts -->
  <Manager pathname="" />

  <Resource name="jdbc/Hibernate" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="hibuser" password="password"
    driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/hibernate"
    validationQuery="select 1" />

  <Resource name="jdbc/Quartz" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="pentaho_user" password="password"
    driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/quartz"
    validationQuery="select 1"/>

  <Resource name="jdbc/sampleddata" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="pentaho_user" password="password"
    driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/sampleddata"
    validationQuery="select 1"/>

  <Resource name="jdbc/myproduccion_bi1" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="myusuario" password="password"
    driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/myproduccion_bi1"
    validationQuery="select 1"/>

  <Resource name="jdbc/myproduccion_bi2" auth="Container" type="javax.sql.DataSource"
    factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20" maxIdle="5"
    maxWait="10000" username="myusuario" password="password"
    driverClassName="com.mysql.jdbc.Driver" url="jdbc:mysql://localhost:3306/myproduccion_bi2"
    validationQuery="select 1"/>

</Context>
```

En donde myproduccion_bi1 y myproduccion_bi2 son dos bases de datos con las que pentaho tendrá que trabajar. Se encuentran definidas aquí a modo de ejemplo.

PASO 12. Verificar FICHERO: HQL_Datasource_MySQL.cfg.xml

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

C : \Pentaho\app\Product\biserver-ce\ pentaho-solutions\bi-developers\datasources

O en

/usr/share/pentaho/app/product/biserver-ce/ pentaho-solutions/bi-developers/datasources

y se nombra como:

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

HQL_Datasource_MySQL.cfg.xml

Contenido anterior

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
<session-factory >

    <property name="cache.provider_class">org.hibernate.cache.EhCacheProvider</property>
    <property name="hibernate.generate_statistics">true</property>
    <property name="hibernate.cache.use_query_cache">true</property>

    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/hibernate</property>
    <property name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
    <property name="connection.username">hibuser</property>
    <property name="connection.password">password</property>
    <property name="connection.pool_size">10</property>
    <property name="show_sql">>false</property>
    <property name="hibernate.jdbc.use_streams_for_binary">true</property>
    <mapping_resource="hibernate/mysql5innodb.hbm.xml" />
</session-factory>
</hibernate-configuration>
```

Verificar que el usuario y la configuración general es correcta.

PASO 13. Verificar FICHERO: applicationContext-spring-security-hibernate.properties

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

C : \Pentaho\app\Product\biserver-ce\pentaho-solutions\system\dialects\mysql5

O en

/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/dialects/mysql5/

y se nombra como:

applicationContext-spring-security-hibernate.properties

Contenido anterior

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

Verificar que el usuario y la configuración general es correcta.

PASO 14. Verificar FICHERO: mysql5.hibernate.cfg.xml

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

C : \Pentaho\app\Product\biserver-ce\pentaho-solutions\system\hibernate

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/hibernate/`

y se nombra como:

`mysql5.hibernate.cfg.xml`

Contenido anterior

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration
PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>

    <property name="cache.provider_class">org.hibernate.cache.EhCacheProvider</property>

    <property name="hibernate.generate_statistics">true</property>
    <property name="hibernate.cache.use_query_cache">true</property>

    <!-- MySQL Configuration -->
    <property name="connection.driver_class">com.mysql.jdbc.Driver</property>
    <property name="connection.url">jdbc:mysql://localhost:3306/hibernate</property>
    <property name="dialect">org.hibernate.dialect.MySQL5InnoDBDialect</property>
    <property name="connection.username">hibuser</property>
    <property name="connection.password">password</property>
    <property name="connection.pool_size">10</property>
    <property name="show_sql">>false</property>
    <property name="hibernate.jdbc.use_streams_for_binary">true</property>
    <!-- replaces DefinitionVersionManager -->
    <property name="hibernate.hbm2ddl.auto">update</property>
    <!-- load resource from classpath -->
    <mapping resource="hibernate/mysql5innodb.hbm.xml" />
    <!-- This is only used by Pentaho Administration Console. Spring Security will not use these mapping files -->
    <mapping resource="PentahoUser.hbm.xml" />
    <mapping resource="PentahoRole.hbm.xml" />
    <mapping resource="PentahoUserRoleMapping.hbm.xml" />

  </session-factory>
</hibernate-configuration>
```

Verificar que el usuario y la configuración general es correcta.

PASO 15. Verificar FICHERO OLAP: `datasources.xml`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\biserver-ce\pentaho-solutions\system\olap\`

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/olap/`

y se nombra como:

`datasources.xml`

Contenido anterior

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSources>
  <DataSource>
    <DataSourceName>Provider=Mondrian;DataSource=Pentaho</DataSourceName>
    <DataSourceDescription>Pentaho BI Platform Datasources</DataSourceDescription>
    <URL>http://localhost:8080/pentaho/Xmla?userid=joe&password=password</URL>
    <DataSourceInfo>Provider=mondrian</DataSourceInfo>
    <ProviderName>PentahoXMLA</ProviderName>
    <ProviderType>MDP</ProviderType>
    <AuthenticationMode>Unauthenticated</AuthenticationMode>
  </DataSource>
</DataSources>
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

```
<Catalog name="SteelWheels">
  <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
  <Definition>solution:steel-wheels/analysis/steeltwheels.mondrian.xml</Definition>
</Catalog>
<Catalog name="SampleData">
  <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
  <Definition>solution:steel-wheels/analysis/SampleData.mondrian.xml</Definition>
</Catalog>
</Catalogs>
</DataSource>
</DataSources>
```

Nuevo contenido

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSources>
  <DataSource>
    <DataSourceName>Provider=Mondrian;DataSource=Pentaho</DataSourceName>
    <DataSourceDescription>Pentaho BI Platform Datasources</DataSourceDescription>
    <URL>http://localhost:8080/pentaho/Xmla?userid=joe&password=password</URL>
    <DataSourceInfo>Provider=mondrian</DataSourceInfo>
    <ProviderName>PentahoXMLA</ProviderName>
    <ProviderType>MDP</ProviderType>
    <AuthenticationMode>Unauthenticated</AuthenticationMode>
  </DataSource>
  <Catalogs>
    <Catalog name="SteelWheels">
      <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
      <Definition>solution:steel-wheels/analysis/steeltwheels.mondrian.xml</Definition>
    </Catalog>
    <Catalog name="SampleData">
      <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
      <Definition>solution:steel-wheels/analysis/SampleData.mondrian.xml</Definition>
    </Catalog>
  </Catalogs>
</DataSources>
```

En un entorno de producción normalmente no deberá dejarse el usuario joe/password, pero este será el usuario que deberá actualizarse en la consola de administración y será el usuario que podrá tener acceso a todo lo relacionado con el servicio de BI de PENTAHO. Cuando se actualice en la consola de administración se deberá actualizar aquí también.

PASO 16. Verificar FICHERO de conexiones: `jdbc.properties`

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

```
C:\Pentaho\app\Product\biserver-ce\pentaho-solutions\system\simple-jndi\
```

O en

```
/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/simple-jndi/
```

y se nombra como:

```
jdbc.properties
```

Contenido anterior

```
# Copyright 2008 Pentaho Corporation. All rights reserved.
# This software was developed by Pentaho Corporation and is provided under the terms
# of the Mozilla Public License, Version 1.1, or any later version. You may not use
# this file except in compliance with the license. If you need a copy of the license,
# please go to http://www.mozilla.org/MPL/MPL-1.1.txt. The Original Code is the Pentaho
# BI Platform. The Initial Developer is Pentaho Corporation.
#
# Software distributed under the Mozilla Public License is distributed on an "AS IS"
```


Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

```
# basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. Please refer to
# the license for the specific language governing your rights and limitations.
SampleData/type=javax.sql.DataSource
SampleData/driver=org.hsqldb.jdbcDriver
SampleData/url=jdbc:hsqldb:hsq://localhost/sampledata
SampleData/user=pentaho_user
SampleData/password=password
Hibernate/type=javax.sql.DataSource
Hibernate/driver=org.hsqldb.jdbcDriver
Hibernate/url=jdbc:hsqldb:hsq://localhost/hibernate
Hibernate/user=hibuser
Hibernate/password=password
Quartz/type=javax.sql.DataSource
Quartz/driver=org.hsqldb.jdbcDriver
Quartz/url=jdbc:hsqldb:hsq://localhost/quartz
Quartz/user=pentaho_user
Quartz/password=password
Shark/type=javax.sql.DataSource
Shark/driver=org.hsqldb.jdbcDriver
Shark/url=jdbc:hsqldb:hsq://localhost/shark
Shark/user=sa
Shark/password=
SampleDataAdmin/type=javax.sql.DataSource
SampleDataAdmin/driver=org.hsqldb.jdbcDriver
SampleDataAdmin/url=jdbc:hsqldb:hsq://localhost/sampledata
SampleDataAdmin/user=pentaho_admin
SampleDataAdmin/password=password
```

Nuevo contenido

```
SampleData/type=javax.sql.DataSource
SampleData/driver=com.mysql.jdbc.Driver
SampleData/url=jdbc:mysql://localhost:3306/sampledata
SampleData/user=pentaho_user
SampleData/password=password
Hibernate/type=javax.sql.DataSource
Hibernate/driver=com.mysql.jdbc.Driver
Hibernate/url=jdbc:mysql://localhost:3306/hibernate
Hibernate/user=hibuser
Hibernate/password=password
Quartz/type=javax.sql.DataSource
Quartz/driver=com.mysql.jdbc.Driver
Quartz/url=jdbc:mysql://localhost:3306/quartz
Quartz/user=pentaho_user
Quartz/password=password
SampleDataAdmin/type=javax.sql.DataSource
SampleDataAdmin/driver=com.mysql.jdbc.Driver
SampleDataAdmin/url=jdbc:mysql://localhost:3306/sampledata
SampleDataAdmin/user=pentaho_admin
SampleDataAdmin/password=password
icroms_bi/type=javax.sql.DataSource
icroms_bi/driver=com.mysql.jdbc.Driver
icroms_bi/url=jdbc:mysql://localhost:3306/myproduccion_bi1
icroms_bi/user=myusuario
icroms_bi/password=passwrod
icroms_web/type=javax.sql.DataSource
icroms_web/driver=com.mysql.jdbc.Driver
icroms_web/url=jdbc:mysql://localhost:3306/myproduccion_bi2
icroms_web/user=myusuario
icroms_web/password=passwrod
```

En un entorno de producción normalmente no deberán dejarse las credenciales de los usuarios con password.

PASO 17. Cambiar el password de la consola de administración:

La consola de administración tiene puesto su usuario y password en el fichero `login.properties` ubicado en

`C:\Pentaho\app\Product\administration-console\resource\config\`

O en

`/usr/share/pentaho/app/product/administration-console/resource/config/`

Para cambiar dicho fichero hay que realizar las siguientes acciones:

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Se lanzan los siguientes comandos recreando la contraseña del usuario admin:

```
$> cd /usr/share/pentaho/app/product/administration-console
$> java -cp lib/jetty-6.1.2.jar:lib/jetty-util-6.1.9.jar org.mortbay.jetty.security.Password
admin <nuevopassword>

<nuevopassword>
OBF:1w241sov1w1e
MD5:d9729feb74992cc3482b350163a1a010
CRYPT:ad6dbST2WsEV

$> vi /usr/share/pentaho/app/product/administration-console/resource/config/login.properties

admin: OBF:1v2jluum1xtvlzejlzerlxtnluvk1vlv,admin

editamos este fichero y lo cambiamos por:

admin: OBF:1w241sov1w1e,admin
```

PASO 18. Arrancar la consola de administración.

Arrancar la consola de administración.

Se lanzan los siguientes comandos desde una ventana de comandos:

```
$> cd /usr/share/pentaho/app/product/administration-console
$> ./start-pac.sh
```

Abrir la consola de administración:

Normalmente, tras el último comando lanzado para iniciar la consola de administración, aparece una dirección URL a través de la cual el usuario se puede conectar y operar. El usuario de acceso es admin y el password el que hemos fijado en el paso 17.

Realizamos las siguientes acciones:

Accedemos a:

- > Administration
- > Database Connections

y actualizamos las conexiones de la base de datos:

myproduccion_bi1:

Name: myproduccion_bi1

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Driver class: com.mysql.jdbc.Driver
UserName: myusuario
Password: <password>
URL: jdbc:mysql://127.0.0.1:3306/myproduccion_bi1
Maximum Active Connections: 30
Number of Idle Connections: 5
Validation Query: select 1;
Wait: (milliseconds) : 1000

myproduccion_bi2:
Name: myproduccion_bi2
Driver class: com.mysql.jdbc.Driver
UserName: myusuario
Password: <password>
URL: jdbc:mysql://127.0.0.1:3306/myproduccion_bi2
Maximum Active Connections: 30
Number of Idle Connections: 5
Validation Query: select 1;
Wait: (milliseconds) : 1000

SampleData:
Name: SampleData
Driver class: com.mysql.jdbc.Driver
UserName: pentaho_user
Password: <password>
URL: jdbc:mysql://127.0.0.1:3306/sampledata
Maximum Active Connections: 20
Number of Idle Connections: 5
Validation Query: select 1;
Wait: (milliseconds) : 1000

Accedemos a:

- > Administration
- > Users & Roles

Y:

- eliminar todos los usuarios
- crear usuarios de uso de la plataforma
- crear usuario "publicador" para publicar los informes. A este usuario se le asignan todos los perfiles de seguridad que haya para que no tenga problemas en la publicación.

PASO 19. Cambiar el usuario "joe" por el usuario "publicador" en el archivo: console.xml

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C:\Pentaho\app\Product\administration-console\resource\config\`

O en

`/usr/share/pentaho/app/product/administration-console/resource/config/`

y se nombra como:

`console.xml`

Contenido anterior

```
<?xml version="1.0" encoding="UTF-8"?>
<console>
  <solution-path>/solution-path>
  <war-path>/war-path>
  <platform-username>joe</platform-username>
  <biserver-status-check-period-millis>30000</biserver-status-check-period-millis>
  <homepage-url>http://www.pentaho.com/console_home</homepage-url>
  <homepage-timeout-millis>15000</homepage-timeout-millis>
  <!-- comma separated list of roles (no spaces) -->
  <default-roles>Authenticated</default-roles>
  <default-server-dir>biserver-ce</default-server-dir>
</console>
```

Nuevo contenido

```
<?xml version="1.0" encoding="UTF-8"?>
<console>
  <solution-path>/solution-path>
  <war-path>/war-path>
  <platform-username>publicador</platform-username>
  <biserver-status-check-period-millis>30000</biserver-status-check-period-millis>
  <homepage-url>http://www.pentaho.com/console_home</homepage-url>
  <homepage-timeout-millis>15000</homepage-timeout-millis>
  <!-- comma separated list of roles (no spaces) -->
  <default-roles>Authenticated</default-roles>
  <default-server-dir>biserver-ce</default-server-dir>
</console>
```

PASO 20. Reiniciar la consola de administración e iniciar Pentaho BI Server

Realizar los comandos siguientes:

Se lanzan los siguientes comandos desde una ventana de comandos:

```
$> cd /usr/share/pentaho/app/product/biserver-ce
$> ./start-pentaho.sh

$> cd /usr/share/pentaho/app/administration-console
$> ./stop-pac.sh
$> ./start-pac.sh
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

PASO 21. Limpiar los casos de ejemplo en caso de un entorno productivo

En este paso se deben cambiar algunas líneas del archivo que se encuentra en el directorio

`C : \Pentaho\app\Product\biserver-ce\pentaho-solutions\system\olap\`

O en

`/usr/share/pentaho/app/product/biserver-ce/pentaho-solutions/system/olap/`

y se nombra como:

`datasources.xml`

Contenido anterior

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSources>
  <DataSource>
    <DataSourceName>Provider=Mondrian;DataSource=Pentaho</DataSourceName>
    <DataSourceDescription>Pentaho BI Platform Datasources</DataSourceDescription>
    <URL>http://localhost:8080/pentaho/Xmla?userid=joesamp;password=password</URL>
    <DataSourceInfo>Provider=mondrian</DataSourceInfo>
    <ProviderName>PentahoXMLA</ProviderName>
    <ProviderType>MDP</ProviderType>
    <AuthenticationMode>Unauthenticated</AuthenticationMode>
    <Catalogs>
      <Catalog name="SteelWheels">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/steelwheels.mondrian.xml</Definition>
      </Catalog>
      <Catalog name="SampleData">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/SampleData.mondrian.xml</Definition>
      </Catalog>
    </Catalogs>
  </DataSource>
</DataSources>
```

Nuevo contenido

```
<?xml version="1.0" encoding="UTF-8"?>
<DataSources>
  <DataSource>
    <DataSourceName>Provider=Mondrian;DataSource=Pentaho</DataSourceName>
    <DataSourceDescription>Pentaho BI Platform Datasources</DataSourceDescription>
    <URL>http://localhost:8080/pentaho/Xmla?userid=publicador&password=password</URL>
    <DataSourceInfo>Provider=mondrian</DataSourceInfo>
    <ProviderName>PentahoXMLA</ProviderName>
    <ProviderType>MDP</ProviderType>
    <AuthenticationMode>Unauthenticated</AuthenticationMode>
    <Catalogs>
      <!--
      <Catalog name="SteelWheels">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/steelwheels.mondrian.xml</Definition>
      </Catalog>
      <Catalog name="SampleData">
        <DataSourceInfo>Provider=mondrian;DataSource=SampleData</DataSourceInfo>
        <Definition>solution:steel-wheels/analysis/SampleData.mondrian.xml</Definition>
      </Catalog>
      -->
    </Catalogs>
  </DataSource>
</DataSources>
```

En donde se ha cambiado el usuario que publica los informes por el usuario **publicador**, su contraseña y se han comentado las líneas de los catálogos de SampleData y SteelWhells

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

Ahora se pueden eliminar las carpetas que hay en Pentaho BI Server

ANEXOS

ANEXO 1. SCRIPT DE CREACIÓN DE LA BASE DE DATOS 'sampledata' para MySQL: create_sampledata_mysql.sql

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/ `sampledata` /*!40100 DEFAULT CHARACTER SET latin1 */;

USE `sampledata`;

/*=====*/
/* Table: CUSTOMERS */
/*=====*/
create table CUSTOMERS
(
  CUSTOMERNUMBER      int(11) not null,
  CUSTOMERNAME        varchar(50) not null,
  CONTACTLASTNAME     varchar(50) not null,
  CONTACTFIRSTNAME    varchar(50) not null,
  PHONE               varchar(50) not null,
  ADDRESSLINE1        varchar(50) not null,
  ADDRESSLINE2        varchar(50) default NULL,
  CITY                varchar(50) not null,
  STATE               varchar(50) default NULL,
  POSTALCODE          varchar(15) default NULL,
  COUNTRY              varchar(50) not null,
  SALESREPEMPLYEENUMBER int(11) default NULL,
  CREDITLIMIT         int(17) default NULL,
  primary key (CUSTOMERNUMBER)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: CUSTOMER_W_TER */
/*=====*/
create table CUSTOMER_W_TER
(
  CUSTOMERNUMBER      int(11) default NULL,
  CUSTOMERNAME        varchar(50) default NULL,
  CONTACTLASTNAME     varchar(50) default NULL,
  CONTACTFIRSTNAME    varchar(50) default NULL,
  PHONE               varchar(50) default NULL,
  ADDRESSLINE1        varchar(50) default NULL,
  ADDRESSLINE2        varchar(50) default NULL,
  CITY                varchar(50) default NULL,
  STATE               varchar(50) default NULL,
  POSTALCODE          varchar(15) default NULL,
  COUNTRY              varchar(50) default NULL,
  EMPLOYEENUMBER      int(11) default NULL,
  CREDITLIMIT         decimal(30,17) default NULL,
  TERRITORY           varchar(10) default NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: DEPARTMENT MANAGERS */
/*=====*/
create table DEPARTMENT_MANAGERS
(
  REGION              varchar(50) not null,
  MANAGER_NAME        varchar(50) not null,
  EMAIL               varchar(50) default NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: DIM_TIME */
/*=====*/
create table DIM_TIME
(
  TIME_ID             varchar(10) default NULL,
  MONTH_ID            int(11) default NULL,
  QTR_ID              int(11) default NULL,
  YEAR_ID             int(11) default NULL,
  MONTH_NAME          char(3) default NULL,
  MONTH_DESC          varchar(9) default NULL,
  QTR_NAME            varchar(4) default NULL,
  QTR_DESC            varchar(9) default NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: EMPLOYEES */
/*=====*/
create table EMPLOYEES
(
  EMPLOYEENUMBER      int(11) not null,
  LASTNAME             varchar(50) not null,
  FIRSTNAME           varchar(50) not null,
  EXTENSION           varchar(10) not null,
  EMAIL               varchar(100) not null,
  OFFICECODE          varchar(20) not null,
  REPORTSTO           int(11) default NULL,
  JOBTITLE             varchar(50) not null,
  primary key (EMPLOYEENUMBER)
)
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

```
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: OFFICES */
/*=====*/
create table OFFICES
(
  OFFICECODE          varchar(50) not null,
  CITY                varchar(50) not null,
  PHONE               varchar(50) not null,
  ADDRESSLINE1        varchar(50) not null,
  ADDRESSLINE2        varchar(50) default NULL,
  STATE               varchar(50) default NULL,
  COUNTRY              varchar(50) not null,
  POSTALCODE          varchar(10) not null,
  TERRITORY            varchar(10) not null,
  ACTIVE              tinyint(1) not null,
  primary key (OFFICECODE)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: ORDERDETAILS */
/*=====*/
create table ORDERDETAILS
(
  ORDERNUMBER         int(11) not null,
  PRODUCTCODE         varchar(50) not null,
  QUANTITYORDERED    int(11) not null,
  PRICEEACH           int(17) not null,
  ORDERLINENUMBER     smallint(6) not null,
  primary key (ORDERNUMBER, PRODUCTCODE)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: ORDERFACT */
/*=====*/
create table ORDERFACT
(
  ORDERNUMBER         int(11) default NULL,
  PRODUCTCODE         varchar(50) not null,
  QUANTITYORDERED    int(11) default NULL,
  PRICEEACH           double(30,17) default NULL,
  ORDERLINENUMBER     int(11) default NULL,
  TOTALPRICE          double default NULL,
  ORDERDATE           timestamp not null default CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  REQUIREDDATE        timestamp not null default '0000-00-00 00:00:00',
  SHIPPEDDATE         timestamp not null default '0000-00-00 00:00:00',
  STATUS              varchar(15) default NULL,
  COMMENTS            text,
  CUSTOMERNUMBER      int(11) default NULL,
  TIME_ID             varchar(10) default NULL,
  QTR_ID              bigint(20) default NULL,
  MONTH_ID            bigint(20) default NULL,
  YEAR_ID             bigint(20) default NULL,
  key SYS_FK_1 (PRODUCTCODE)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: ORDERS */
/*=====*/
create table ORDERS
(
  ORDERNUMBER         int(11) not null,
  ORDERDATE           timestamp not null default CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  REQUIREDDATE        timestamp not null default '0000-00-00 00:00:00',
  SHIPPEDDATE         timestamp not null default '0000-00-00 00:00:00',
  STATUS              varchar(15) not null,
  COMMENTS            text,
  CUSTOMERNUMBER      int(11) not null,
  primary key (ORDERNUMBER)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: PAYMENTS */
/*=====*/
create table PAYMENTS
(
  CUSTOMERNUMBER      int(11) not null,
  CHECKNUMBER         varchar(50) not null,
  PAYMENTDATE         timestamp not null default CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  AMOUNT              int(17) not null,
  primary key (CUSTOMERNUMBER, CHECKNUMBER)
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: PRODUCTS */
/*=====*/
create table PRODUCTS
(
  PRODUCTCODE         varchar(50) not null,
  PRODUCTNAME         varchar(70) not null,
  PRODUCTLINE         varchar(50) not null,
  PRODUCTSCALE        varchar(10) not null,
  PRODUCTVENDOR       varchar(50) not null,
  PRODUCTDESCRIPTION  text not null,
  QUANTITYINSTOCK     smallint(6) not null,
  BUYPRICE            int(17) not null,
  MSRP                int(17) not null,
  primary key (PRODUCTCODE)
```

Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

```
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: QUADRANT_ACTUALS */
/*=====*/
create table QUADRANT_ACTUALS
(
  REGION          varchar(50) not null,
  DEPARTMENT      varchar(50) not null,
  POSITIONTITLE    varchar(50) not null,
  ACTUAL          decimal(18,4) default NULL,
  BUDGET          decimal(18,4) default NULL,
  VARIANCE        decimal(18,4) default NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*=====*/
/* Table: TRIAL_BALANCE */
/*=====*/
create table TRIAL_BALANCE
(
  Type           char(1) default NULL,
  Account_Num    int(11) default NULL,
  Category       varchar(57) default NULL,
  Category2      varchar(57) default NULL,
  Detail         varchar(57) default NULL,
  Amount         int(11) default NULL
)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

alter table ORDERFACT add constraint SYS_FK_1 foreign key (PRODUCTCODE)
references PRODUCTS (PRODUCTCODE) on delete restrict on update restrict;
```

ANEXO 1. Shell para el arranque y parada de la consola de administración en LINUX:

```
#!/bin/sh

filelog=/usr/share/pentaho/app/product/administration-console/log/pentaho-admconsole_`date +%Y%m%d_%H%M%S`.log
proceso=ps -ef | grep "org.pentaho.pac.server.JettyServer" | grep -v "grep" | awk '{print $2;}'
proceso2=ps -ef | grep "\/usr\/share\/pentaho\/app\/product\/administration-console" | grep -v "grep" | awk '{print $2;}'

case "$1" in
  start)
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
    if [ "${proceso2}" != "" ]; then
      kill -9 $proceso2
    fi
    cd /usr/share/pentaho/app/product/administration-console
    nohup /usr/share/pentaho/app/product/administration-console/start-pac.sh 1>$filelog 2>&1 &
  ;;
  stop)
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
    if [ "${proceso2}" != "" ]; then
      kill -9 $proceso2
    fi
  ;;
  restart)
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
    if [ "${proceso2}" != "" ]; then
      kill -9 $proceso2
    fi
    cd /usr/share/pentaho/app/product/administration-console
    nohup /usr/share/pentaho/app/product/administration-console/start-pac.sh 1>$filelog 2>&1 &
  ;;
  *)
    echo "uso $0 [start|stop|restart]"
  esac
exit 0
```


Instalación de PENTAHO BI Server 4.8 en MySQL

Para versiones de MySQL 5.x

ANEXO 1. Shell para el arranque y parada del servicio BI Server en LINUX:

```
#!/bin/sh
logfile=/usr/share/pentaho/app/product/biserver-ce/log/pentaho-biserver_`date +%Y%m%d_%H%M%S`.log
proceso=`ps -ef | grep "/usr/share/pentaho/app/product/biserver-ce/tomcat" | grep -v "grep" | awk '{print $2;}'`
case "$1" in
  start)
    /usr/share/pentaho/app/product/biserver-ce/stop-pentaho.sh > $logfile 2>&1
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
    /usr/share/pentaho/app/product/biserver-ce/start-pentaho.sh >> $logfile 2>&1
  ;;
  stop)
    /usr/share/pentaho/app/product/biserver-ce/stop-pentaho.sh > $logfile 2>&1
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
  ;;
  restart)
    /usr/share/pentaho/app/product/biserver-ce/stop-pentaho.sh > $logfile 2>&1
    if [ "${proceso}" != "" ]; then
      kill -9 $proceso
    fi
    /usr/share/pentaho/app/product/biserver-ce/start-pentaho.sh >> $logfile 2>&1
  ;;
  *)
    echo "uso $0 [start|stop|restart]"
  ;;
esac
exit 0
```