

Somos su empresa de Soporte a Desarrollo Informático

Ese apoyo que siempre quiso tener

- Desarrollo de componentes y proyectos a medida.
- Auditoría de código y recomendaciones de mejora.
- Arranque de proyectos basados en nuevas tecnologías.
- Curso de Formación
 - Dirección de Proyectos Informáticos.
 - Gestión eficaz del Tiempo.
 - Arquitecturas de desarrollo Web: Web, J2EE, SOA, WebServices, BPM, etc.
 - Java/ J2EE a todos los niveles: JSPs, Servlets, EJBs, JMS, JNI, etc.
 - Análisis y diseño orientado a objeto.
 - UML y patrones de diseño.
 - Buenas prácticas en el desarrollo de aplicaciones
 - Técnicas avanzadas: Lucene, Hibernate, Spring, JSF, Struts, etc.

*Nuestra mejor referencia son los conocimientos que
compartimos en nuestro web*

www.adictosaltrabajo.com

Decenas de entidades cuentan ya con nosotros

Para más información visítenos en www.autentia.com

Tel. 91 675 33 06 - info@autentia.com

Autentia Real Business Solutions

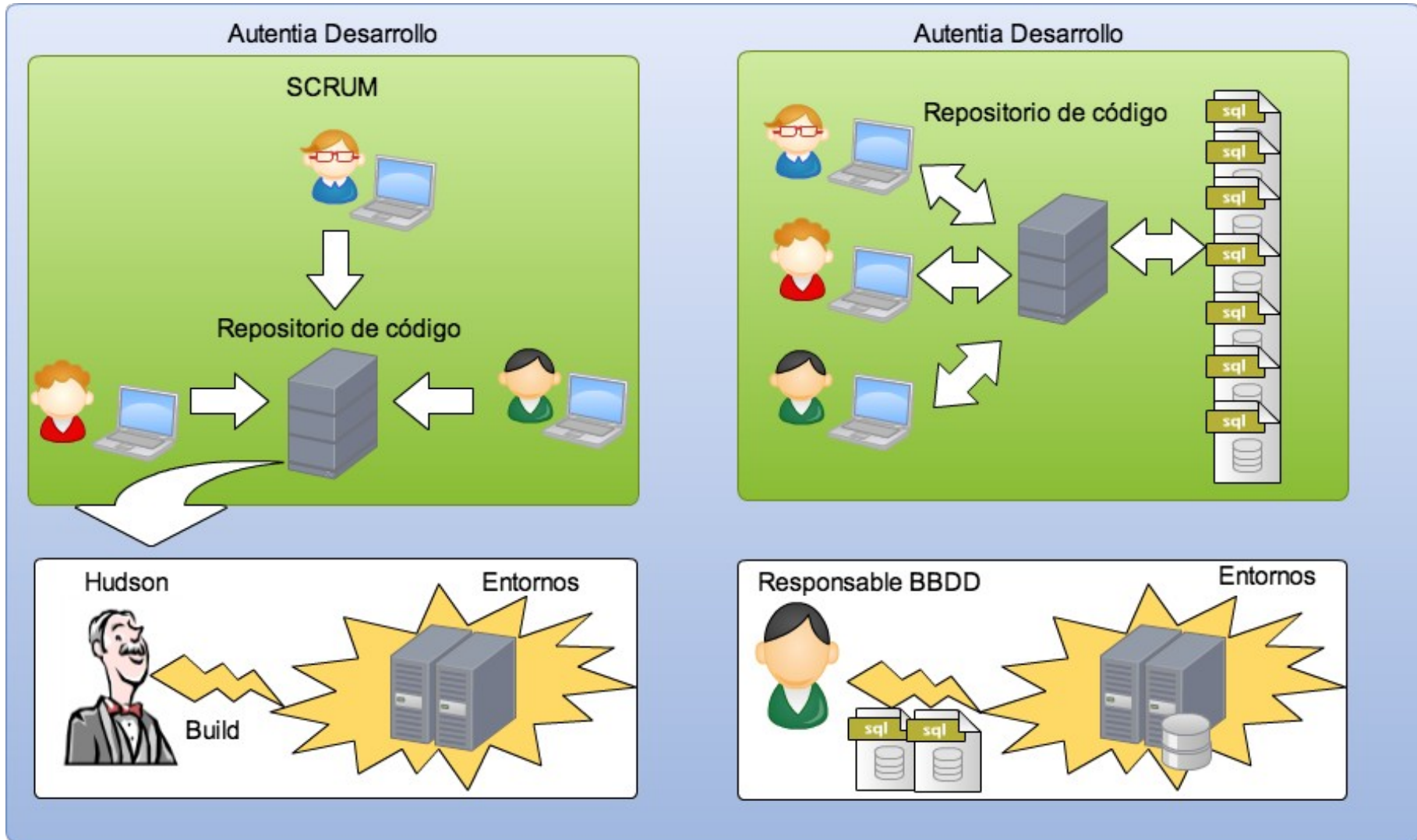


www.autentia.com

LIQUI BASE

Saul García Díaz
Consultor tecnológico de desarrollo de proyectos informáticos.

En contexto



Antes de Liquibase



- Ejecución de cambios en el modelo de datos manual.
- No podíamos garantizar que una base de datos se encontrase en una versión concreta.
- Nos era demasiado costoso revertir la bbdd a estados anteriores.
- No podíamos garantizar la sincronía entre el modelo de datos y el modelo de entidades de la aplicación.
- Generación manual de documentación sobre el modelo de datos.
- Las incorporaciones de un nuevo miembro al equipo eran mas costosas.

Bases de datos soportadas



www.autentia.com

Base de datos	Tipo / Nombre
MySQL	mysql
PostgreSQL	postgresql
Oracle	oracle
MS-SQL	mssql
Sybase Enterprise	sybase
Sybase Anywhere	asany
DB2	db2
Apache Derby	derby
HSQL	hsqldb
H2	h2
InterSystems Caché	cache
Firebird	firebird
MaxDB / SAPDB	maxdb
SQLite	sqlite

Puesta en marcha



- <http://www.liquibase.org/download>

```
Terminal — bash — 79x23
MacBook-Pro-de-Saul-Garcia-Diaz:~ sgdiarz$ liquibase
Errors:
  Command not passed

Usage: java -jar liquibase.jar [options] [command]

Standard Commands:
  update                Updates database to current version
  updateSQL             Writes SQL to update database to current
                       version to STDOUT
  updateCount <num>    Applies next NUM changes to the database
  updateSQL <num>      Writes SQL to apply next NUM changes
                       to the database
  rollback <tag>        Rolls back the database to the the state is was
                       when the tag was applied
  rollbackSQL <tag>    Writes SQL to roll back the database to that
                       state it was in when the tag was applied
                       to STDOUT
  rollbackToDate <date/time> Rolls back the database to the the state is was
                       at the given date/time.
                       Date Format: yyyy-MM-dd HH:mm:ss
  rollbackToDateSQL <date/time> Writes SQL to roll back the database to that
                       state it was in at the given date/time version
```

Construyendo Change-Logs



- changelog = Conjunto de cambios que se aplicaran en base de datos.

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9 http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">

  <preConditions>
    <dbms type="mysql"/>
    <sqlCheck expectedResult="1">select 1</sqlCheck>
    <runningAs username="liquibase"/>
  </preConditions>

  <changeSet id="1" author="sgdiaz">
    <comment>
      Podemos y debemos añadir comentarios a los changeset.
      Incluso pueden ser multiples si se desea.
      No se utilizan para calcular el MD5Sum del changeset, de tal manera que podamos
      modificarlos sin causar problemas
    </comment>
    <createTable tableName="Users">
      <column name="id" type="int" autoIncrement="true">
        <constraints primaryKey="true" nullable="false"/>
      </column>
      <column name="firstname" type="varchar(50)"/>
      <column name="lastname" type="varchar(50)">
        <constraints nullable="false"/>
      </column>
    </createTable>
  </changeSet>
</databaseChangeLog>
```

Liquibase Tags



- `<databaseChangeLog>`

Es la raíz de cualquier conjunto de cambios que definamos para la base de datos.

<code><preConditions></code>	Condiciones requeridas para aplicar los cambios
<code><changeSet></code>	Cambios que se aplicaran al modelo de datos
<code><property></code>	Propiedades comunes a los change set
<code><include></code>	Conjunto de cambios adicionales u otros recursos

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9 http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">
  </databaseChangeLog>
```


Liquibase Tags



- `<databaseChangeLog>`

Es la raíz de cualquier conjunto de cambios que definamos para la base de datos.

<code><preConditions></code>	Condiciones requeridas para aplicar los cambios
<code><changeSet></code>	Cambios que se aplicaran al modelo de datos
<code><property></code>	Propiedades comunes a los change set
<code><include></code>	Conjunto de cambios adicionales u otros recursos

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9 http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">
  </databaseChangeLog>
```

Liquibase Tags



- `<preConditions>`

Atributos disponibles:

<code>onFail</code>	Que hacer si alguna precondición falla
<code>onError</code>	Que hacer si alguna precondición produce un error

Valor de los atributos:

<code>HALT</code>	Suspende inmediatamente la ejecución de la totalidad del conjunto de cambios.[Default]
<code>WARN</code>	Nos marcara un warning pero la ejecución del conjunto de cambios continua.

Precondiciones disponibles:

<code><dbms></code>	Pasa si la bases de datos con la que se trabaja coincide con la especificada
<code><runningAs></code>	Pasa si el usuario de base de datos coincide con el usuario especificado

Liquibase Tags



- `<preConditions>`

Precondiciones disponibles:

<code><changeSetExecuted></code>	Pasa si el chageset ya ha sido ejecutado
<code><columnExists></code>	Pasa si la columna especificada existe en la base de datos
<code><tableExists></code>	Pasa si la tabla especificada existe en la base de datos
<code><viewExists></code>	Pasa si la vista especificada existe en base de datos
<code><foreignKeyConstraintExists></code>	Pasa si la foreign key especificada existe en la base de datos
<code><indexExists></code>	Pasa si el indice especificado existe en base de datos
<code><sequenceExists></code>	Pasa si la secuencia especificada existe en base de datos
<code><primaryKeyExists></code>	Pasa si la primary key especificada existe en base de datos
<code><sqlCheck></code>	Pasa si el valor devuelto por la ejecución de una sentencia SQL es el especificado.
<code><customPrecondition></code>	Condiciones previas personalizadas.

Liquibase Tags



- `<preConditions>`

Algunos ejemplos :

```
<preConditions onFail="WARN">
  <dbms type="oracle" />
  <runningAs username="SYSTEM" />
</preConditions>
```

```
<preConditions>
  <or>
    <dbms type="oracle" />
    <dbms type="mysql" />
  </or>
</preConditions>
```

```
<preConditions>
  <dbms type="oracle" />
  <dbms type="mysql" />
</preConditions>
```

```
<preConditions>
  <or>
    <and>
      <dbms type="oracle" />
      <runningAs username="SYSTEM" />
    </and>
    <and>
      <dbms type="mysql" />
      <runningAs username="sa" />
    </and>
  </or>
</preConditions>
```

Liquibase Tags



- `<changeSet>`

Atributos disponibles:

id	Identificador alfanumérico[Requerido]
author	Creador del change set [Requerido]
dbms	Tipo de bbdd que para este chageset
runAlways	Siempre se ejecuta el change set
runOnChange	Se ejecuta el change set la primera vez y siempre que este se modifique
failOnError	Nos sirve para indicar si se debe parar o no la ejecución
context	Indica el entorno de ejecucion del changeset

Liquibase Tags



- `<changeSet>` Sub Tags disponibles:

- `<createTable>`

```
<createTable tableName="Users">
  <column name="id" type="int" autoIncrement="true">
    <constraints primaryKey="true" nullable="false"/>
  </column>
  <column name="firstname" type="varchar(50)"/>
  <column name="lastname" type="varchar(50)">
    <constraints nullable="false"/>
  </column>
  <column name="phone" type="varchar(50)"/>
</createTable>
```

- `<dropTable>`

```
<dropTable tableName="Users" schemaName="name_schema"/>
```

- `<addColumn tableName="Users"><column name="movilphone" type="varchar(50)"/></addColumn>`

- `<renameColumn tableName="Users" oldColumnName="movilphone" newColumnName="movilnumber"/>`

Liquibase Tags



- `<changeSet>` Sub Tags disponibles:

- `<modifyColumn>`

```
<modifyColumn tableName="Users">  
  <column name="phone" type="INT" defaultValueNumeric="000000000"/>  
</modifyColumn>
```

- `<dropColumn>`

```
<dropColumn tableName="Users" columnName="phone"/>
```

- `<addForeignKeyConstraint>`

```
<addForeignKeyConstraint constraintName="fk_address_person"  
  baseTableName="address"  
  baseColumnNames="person_id"  
  referencedTableName="person"  
  referencedColumnNames="id"/>
```

- `<createIndex>`

```
<createIndex tableName="Users" indexName="idx_person_name">  
  <column name="firstname"/>  
  <column name="lastname"/>  
</createIndex>
```

Liquibase Tags



- `<changeSet>` Sub Tags disponibles:

- `<insert>`

```
<insert tableName="Users">  
  <column name="firstname" value="Fred"/>  
  <column name="lastname" value="Johnson"/>  
  <column name="phone" valueNumeric="918876541"/>  
</insert>
```

-

```
<update tableName="Users">  
  <column name="firstname" value="Jose"/>  
  <column name="lastname" value="Perez"/>  
  <where>id=2</where>  
</update>
```

- `<<delete tableName="Users"><where>id=2</where></delete>`

- `<sql>insert into Users (firstname, lastname) values ('Fred', 'Johnson')</sql>`

- `<sqlFile path="classpath:liquibase/insertPackardbellMasterProducts.sql" />`

- `<comment>Descripción del cambio</comment>`

Liquibase Tags



- `<changeSet>` Sub Tags disponibles:

- Rollback

```
<changeSet id="rollback-create" author="sgdiaz">
  <createTable tableName="rollback-create">
    <column name="id" type="int"/>
  </createTable>
</changeSet>
```

```
<changeSet id="rollback-create" author="sgdiaz">
  <createTable tableName="rollback-create">
    <column name="id" type="int"/>
  </createTable>
  <rollback>
    <dropTable tableName="rollback-create"/>
  </rollback>
</changeSet>
```

```
<changeSet id="rollback-drop" author="sgdiaz">
  <dropTable tableName="rollback-create"/>
  <rollback changeSetId="rollback-create" changeSetAuthor="sgdiaz"/>
</changeSet>
```

```
<changeSet id="rollbackMultiple" author="sgdiaz">
  <createTable tableName="rollbackMultiple1">
    <column name="id" type="int"/>
  </createTable>
  <createTable tableName="rollbackMultiple2">
    <column name="id" type="int"/>
  </createTable>
  <createTable tableName="rollbackMultiple3">
    <column name="id" type="int"/>
  </createTable>
  <rollback>
    drop table rollbackMultiple1;
    drop table rollbackMultiple2;
  </rollback>
  <rollback>drop table rollbackMultiple3</rollback>
</changeSet>
```

Liquibase Tags



- `<property>`

```
<property name="shortVarchar" value="varchar(20)"/>
```

- `<include>`

```
<?xml version="1.0" encoding="UTF-8"?>
<databaseChangeLog
  xmlns="http://www.liquibase.org/xml/ns/dbchangelog/1.9"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.liquibase.org/xml/ns/dbchangelog/1.9 http://www.liquibase.org/xml/ns/dbchangelog/dbchangelog-1.9.xsd">

  <include file="/Users/sgdiaz/liquibase/changeLogs/autentia_1.0.xml" />
  <include file="/Users/sgdiaz/liquibase/changeLogs/autentia_1.1.xml" />
  <include file="/Users/sgdiaz/liquibase/changeLogs/autentia_1.2.xml" />

</databaseChangeLog>
```

Liquibase - Funcionalidad



- Database Update Commands

update	Actualiza la base de datos a la versión actual
updateCount <value>	Aplica el número de change set indicados por <value>
updateSQL	Genera un scrit.sql para actualizar la bbdd a la versión actual
updateCountSQL <value>	Genera un scrit.sql para actualizar la bbdd con el numero de change set indicado por <value>

- Database Rollback Commands

rollback <tag>	La bbdd vuelve al estado en que se encontraba cuando la etiqueta se aplico.
rollbackToDate <date/time>	Revierte la bbdd al estado en que se encontraba en la fecha dada.
rollbackCount <value>	Revierte la bbdd el número de chage set indicados por <value>
rollbackSQL <tag>	Genera un scrit.sql para revertir la bbdd al estado en el que se encontraba cuando la etiqueta se aplicó.
rollbackToDateSQL <date/time>	Genera un script.sql para revertir la bbdd al estado en el que se encontraba en la fecha dada

Liquibase - Funcionalidad



- Database Rollback Commands

rollbackCountSQL <value>	Genera un script.sql para revirtir la bbdd el número de change set indicado por <value>
futureRollbackSQL	Genera un script.sql para hacer retroceder la bbdd a la situación actual después de que los cambios en el changelog se han aplicado

- Maintenance Commands

tag <tag>	Nos permite etiquetar un conjunto de cambios para rollbacks futuros
status	Nos que cambios hay en nuestro conjunto de cambios que no han sido aplicados en base de datos
validate	Chequea el el changelog en busca de errores
changelogSync	Marca todos los cambios que seran ejecutados en la base de datos
changelogSyncSQL	Genera un script.sql con todos los cambios que serán ejecutados en base de datos
markNextChangeSetRan	Nos marca el siguiente cambio ha ejecutar en base de datos

Liquibase - Funcionalidad



- Maintenance Commands

listLocks	Nos indica quien tiene bloqueado el histórico
releaseLocks	Libera todos los bloqueos de base de datos
dropAll	Elimina todas las tablas de base de datos
clearCheckSums	Elimina los checkSum de base de datos. En la siguiente ejecución se generan de nuevo

- Parámetros requeridos

changeLogFile=<path and filename>	Conjunto de cambios que utilizaremos para aplicar los cambios en base de datos
username=<value>	Usuario de base de datos
password=<value>	Password de base de datos
url=<value>	URL de conexión a bse de datos
driver=<jdbc.driver.ClassName>	Nombre driver de base de datos usada
classpath=<value>	Indicamos la ubicación del driver de base de datos

Liquibase - Funcionalidad



- Parámetros Opcionales

contexts=<value>	Contexto en el que se ejecutaran los distintos change set
defaultSchemaName=<schema>	Especifica el schema de bbdd predeterminado
databaseClass=<DatabaseImpl>	Especifica una implementación personalizada de Interface Database
defaultsFile=</path/to/file>	Fichero que contiene los valores por defecto para cualquier migración (./liquibase.properties)
includeSystemClasspath=<boolean>	Incluye el classpath del sistema en el classpath de liquibase (default true)
promptForNonLocalDatabase=<boolean>	Pregunta si las bases de datos no son locales (Default false)
logLevel=<level>	Define el nivel de log(finest, finer, fine, info, warning, severe)
logFile=<path/to/file>	Fichero que contendrá las trazas de error en caso de que una migración falle

Liquibase - Funcionalidad



- Diffs Commands

<code>diff [diff parameters]</code>	Muestra las diferencias entre dos bbdd por consola
<code>diffChangeLog [diff parameters]</code>	Genera un conjunto de cambios en base a las diferencias existentes entras dos bases de datos comparadas

- Parámetros requeridos

<code>baseUsername=<value></code>	Usuario de la bbdd que se compara
<code>basePassword=<value></code>	Password de la bbdd que se compara
<code>baseUrl=<value></code>	URL de la bbdd que se compara

- Change Log properties

<code>D <property.name>=<property.value></code>	Permite pasar propiedades al changelog siendo usadas en este al sustituyendo por <code>\${property.name}</code>
---	---

- Database Documentation

<code>dbDoc <outputDirectory></code>	Nos permite generar documentación al estilo JavaDoc en base a los comentarios de la versión actual de base de datos
--	---

Integración con Spring



- pom.xml

```
<dependency>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-core</artifactId>
  <version>1.9.5</version>
</dependency>
```

- applicationContext.xml

```
<!-- Liquibase -->
<bean id="liquibase" class="liquibase.spring.SpringLiquibase">
  <property name="dataSource" ref="dataSource" />
  <property name="changeLog" value="classpath:/liquibase/autentia_liquibase.xml" />
  <property name="contexts" value="test, production"/>
</bean>
```


Integración con Maven

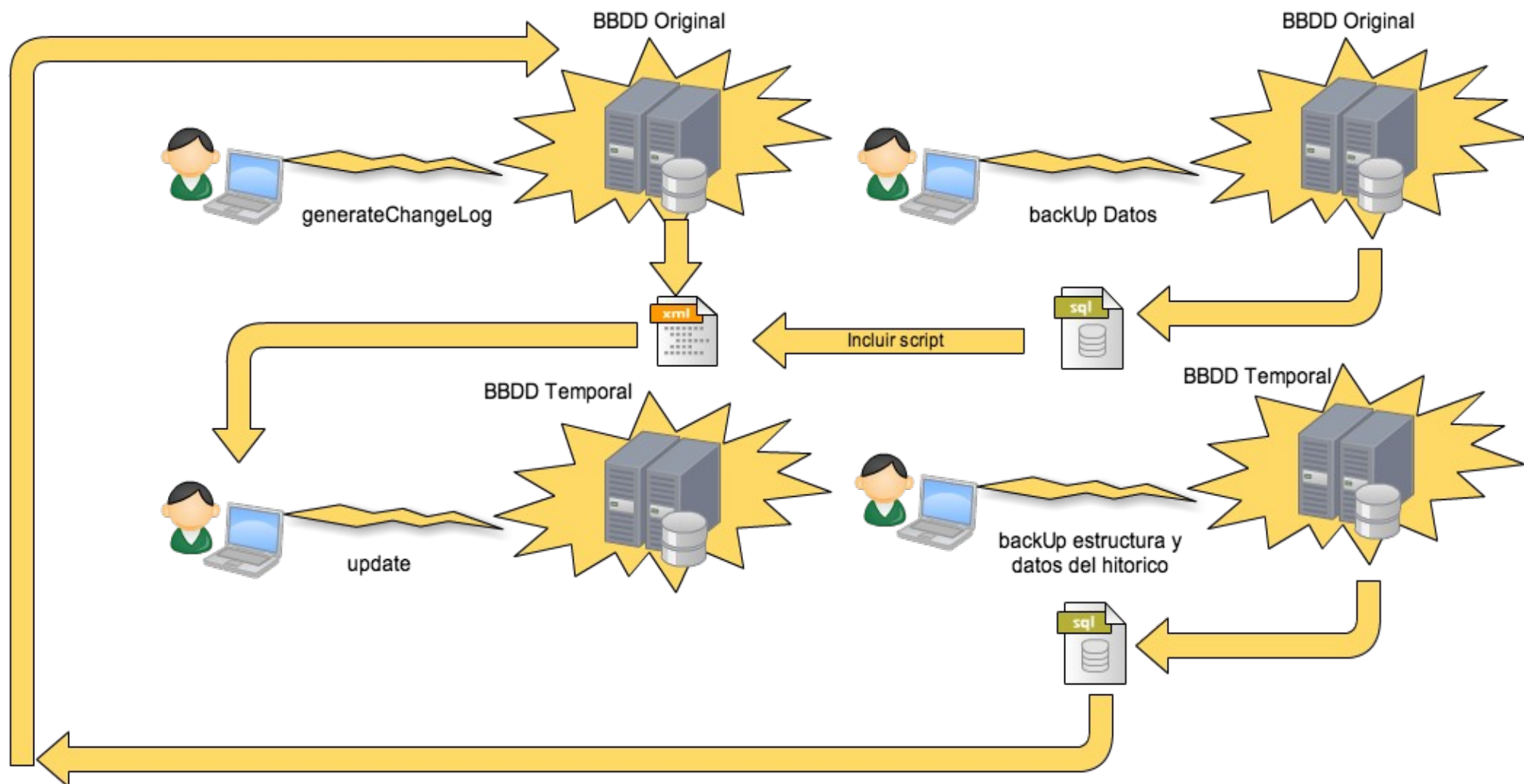


- pom.xml

```
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <!-- Replace with the version of the MySQL driver you want to use -->
  <version>5.0.5</version>
</dependency>

<plugin>
  <groupId>org.liquibase</groupId>
  <artifactId>liquibase-plugin</artifactId>
  <version>1.6.1.0</version>
  <executions>
    <execution>
      <phase>package</phase>
      <configuration>
        <changeLogFile>src/main/resources/liquibase/autentia_liquibase.xml</changeLogFile>
        <driver>com.mysql.jdbc.Driver</driver>
        <url>jdbc:mysql://localhost:3306/liquibase</url>
        <username>liquibase</username>
        <password>liquibase</password>
        <tag>${project.version}</tag>
      </configuration>
      <goals>
        <goal>update</goal>
        <goal>tag</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

Incorporando Liquibase BBDD existente



Después de Liquibase



- Ejecución automática de cambios en el modelo de datos.
- Podemos garantizar que una base de datos se encontrase en una versión concreta.
- Nos es mucho menos costoso revertir la bbdd ha estados anteriores.
- Podemos garantizar la sincronía entre el modelo de datos y el modelo de entidades de la aplicación.
- Generación automática de documentación sobre el modelo de datos.
- Las incorporaciones de un nuevo miembro al equipo son menos costosas.

Recursos obtenidos



www.autentia.com

- <http://www.liquibase.org>

Podeis seguirnos en :



www.autentia.com

- <http://www.autentia.com>
- <http://www.adictosaltrabajo.com>

¿ Preguntas ?



www.autentia.com

