**Cierre de Conexiones**

Hibernate

**Baufest**

**noviembre de 2016**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fecha** | **Descripción** | **Autor** | **Versión** | **Revisor** |
| 01/11/2016 | Documento Inicial | Israel Calderón | 1.0 |  |
|  |  |  |  |  |

Contenido

[1. Objetivo de este documento 3](#_Toc466539223)

[2. Problemática 3](#_Toc466539224)

[2.1 Descripción 3](#_Toc466539225)

[2.2 Como nos conectamos actualmente a la capa de datos 3](#_Toc466539226)

[2.3 Como reproducir el error 3](#_Toc466539227)

[2.3.1 Ejemplo 3](#_Toc466539228)

[2.3.1.1 Data Source - Configuración 3](#_Toc466539229)

[2.3.1.2 Guardado Exitoso 4](#_Toc466539230)

[2.3.1.3 Guardado Erróneo 4](#_Toc466539231)

[3. Acciones realizadas al día de hoy 4](#_Toc466539232)

[3.1 Validaciones 4](#_Toc466539233)

[3.2 Aumento de Recursos 4](#_Toc466539234)

[3.3 Procesos Alternos 4](#_Toc466539235)

[4. Soluciones Propuestas 5](#_Toc466539236)

[4.1 Tipos de Controladores JDBC 5](#_Toc466539237)

[4.1.1 XA 5](#_Toc466539238)

[Java, tiene soporte a este protocolo mediante el uso de: 5](#_Toc466539239)

[4.1.2 No XA 6](#_Toc466539240)

[4.2 Uso adecuando de cada Controladores JDBC 6](#_Toc466539241)

[4.2.1 Un único recurso 6](#_Toc466539242)

[4.2.2 Múltiples recurso 6](#_Toc466539243)

[4.3 Ejemplos 7](#_Toc466539244)

[4.3.1 Hibernate escenario 1 (Exitoso) 7](#_Toc466539245)

[4.3.1.1 Datos correctos (OK) 7](#_Toc466539246)

[4.3.1.2 Datos incorrectos (OK) 7](#_Toc466539247)

[4.3.2 Hibernate escenario 2 (Error) 7](#_Toc466539248)

[4.3.2.1 Datos correctos (OK) 8](#_Toc466539249)

[4.3.2.2 Datos incorrectos (KO) 8](#_Toc466539250)

[4.3.3 JTA escenario 3 (Exitoso) 9](#_Toc466539251)

[4.3.3.1 Datos correctos (OK) 9](#_Toc466539252)

[4.3.3.2 Datos incorrectos (OK) 9](#_Toc466539253)

[5. Referencias 10](#_Toc466539254)

# Objetivo de este documento

El presente documento pretende identificar la causa del error ocurrido al momento de consumir conexiones de base de datos y que por alguna razón (de datos identificada hasta este momento), se genere una excepcion, obligando a un cierre de la conexión, sin éxito (no liberando el recurso), provocando la falta de recursos de conexión en el servidor de aplicaciones.

# Problemática

## Descripción

Hasta este momento se ha tenido un problema recurrente con el cierre de conexiones a base de datos. Específicamente al hacer uso del ORM – Hibernate integrado con el Servidor de Aplicaciones Weblogic. Pero esto no descarta se

## Como nos conectamos actualmente a la capa de datos

El mecanismo de conexión empleado hasta este momento es mediante el uso de DataSource. Generados en el servidor de aplicaciones.

## Como reproducir el error

Se han identificado varios escenarios de reproducción, los más comunes:

* Al momento de que se genera una Excepción en el método que engloba la transacción sea cual sea el tipo, ejemplo: NullPointerException, NumberFormatException, IndexOutOfBoundsException, etc.
* Al intentar guardar datos con longitud mayor a la permitida por el campo en base de datos.

### Ejemplo

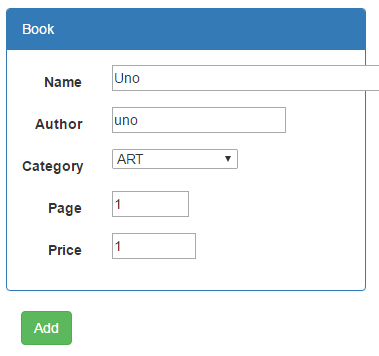
### Data Source - Configuración



La configuración con la que se creó el DataSource es la default. Solo cabe mencionar que el tipo de controlador seleccionado fue el *OracleXADataSource*.

### Guardado Exitoso

##### Pantalla

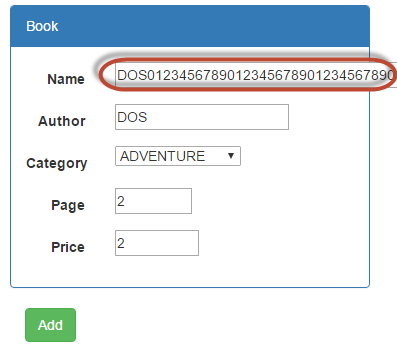


##### Estatus de Origen de Datos (JDBC)



### Guardado Erróneo

##### Pantalla



##### Estatus de Origen de Datos (JDBC)



# Acciones realizadas al día de hoy

## Validaciones

Agregar las validaciones necesarias a nivel de negocio y de datos. Evitando de esta forma la generación de excepciones durante la transacción en proceso.

## Aumento de Recursos

Incremento en el Pool de Conexiones, permitiendo tener un mayor número de recursos disponibles. Que en un tiempo determinado finalmente se consumirán.

## Procesos Alternos

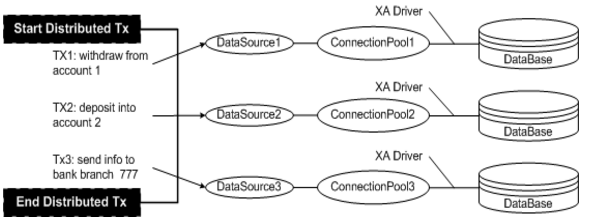
Mecanismos alternos (script manuales) de liberación de recursos en el servidor de aplicaciones.

# Soluciones Propuestas

En base al análisis y pruebas realizadas, se identificó que el problema es generado por el tipo de controlador utilizado en la configuración del Origen de Datos (DataSource).

## Tipos de Controladores JDBC

### XA



Este protocolo es utilizado para administrar la transacción entre varios recursos. Es un gestor de transacciones.

### Java, tiene soporte a este protocolo mediante el uso de:

**JTA (Java Transaction API)**

Tal Transaction Manager está incluido en un servidor de aplicaciones Java EE.

* Glassfish
* JBoss
* WebSphere
* Weblogic

**Componente independiente**

Con posibilidad de comunicarse con recursos mediante el protocolo XA.

* Atomikos
* Java Open Transaction Manager (JOTM)
* Bitronix

### No XA



Este protocolo es utilizado para administrar una única transacción de un solo recurso.

Java, tiene soporte a este protocolo mediante el uso de:

* JBDC
* Hibernate
* JPA (Java Persistence API)

## Uso adecuando de cada Controladores JDBC

### Un único recurso

En este caso vamos a utilizar Hibernate, para realizar la gestiona.

**Pasos realizados…**

*Servidor de Aplicaciones*

Crear el *DataSource* con el tipo de controlador oracle.jdbc.OracleDriver

*Aplicación*

Configurar el *Transaction Manager* con org.springframework.orm.hibernate4.HibernateTransactionManager

### Múltiples recurso

En este caso vamos a utilizar JTA, para realizar la gestiona.

**Pasos realizados…**

*Servidor de Aplicaciones*

Crear el *DataSource* con el tipo de controlador oracle.jdbc.xa.client.OracleXADataSource

*Aplicación*

Configurar el *Transaction Manager* con org.springframework.transaction.jta.JtaTransactionManager

Agregar en el *applicationContext.xml*

<tx:jta-transaction-manager/>

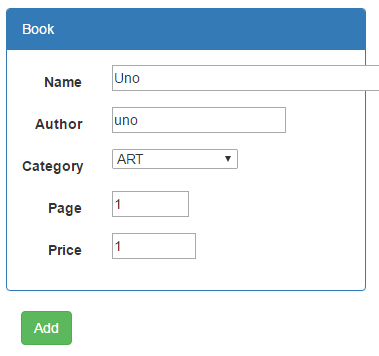
## Ejemplos

### Hibernate escenario 1 (Exitoso)

* DataSource (*jdbc/BookTestDat*) con el controlador oracle.jdbc.OracleDriver
* Transaction Manager org.springframework.orm.hibernate4.HibernateTransactionManager

### Datos correctos (OK)

##### Pantalla

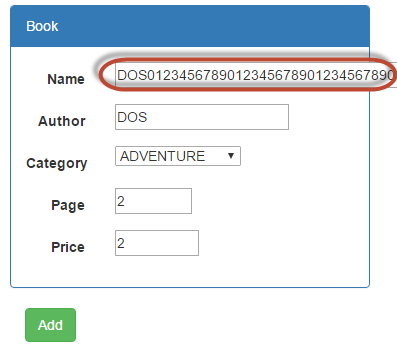


##### Estatus de Origen de Datos (JDBC)



### Datos incorrectos (OK)

##### Pantalla



##### Estatus de Origen de Datos (JDBC)

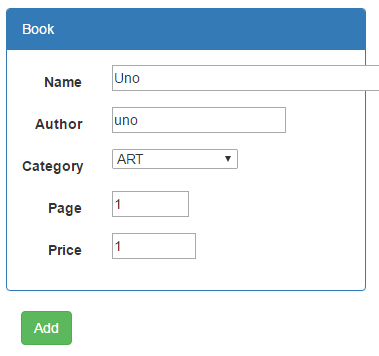


### Hibernate escenario 2 (Error)

* DataSource (*jdbc/BookTestDat*) con el controlador oracle.jdbc.xa.client.OracleXADataSource
* Transaction Manager org.springframework.orm.hibernate4.HibernateTransactionManager

### Datos correctos (OK)

##### Pantalla

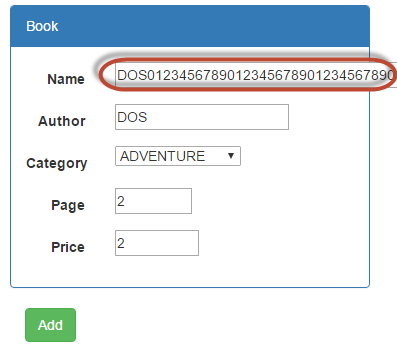


##### Estatus de Origen de Datos (JDBC)



### Datos incorrectos (KO)

##### Pantalla



##### Estatus de Origen de Datos (JDBC)

Presionar Add 🡪 Por primera vez



Presionar Add 🡪 Por segunda vez



Presionar Add 🡪 Por tercera vez

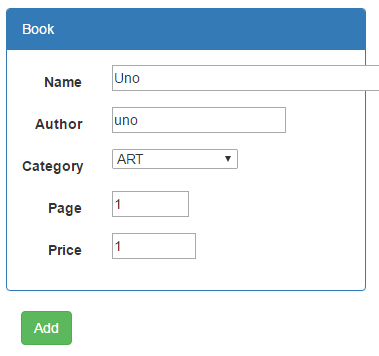
org.springframework.transaction.CannotCreateTransactionException: Could not open Hibernate Session for transaction; nested exception is org.hibernate.exception.GenericJDBCException: Could not open connection at org.springframework.orm.hibernate4.HibernateTransactionManager.doBegin(HibernateTransactionManager.java:517)

### JTA escenario 3 (Exitoso)

* DataSource (*jdbc/BookTestDatXA*) con el controlador oracle.jdbc.xa.client.OracleXADataSource
* Transaction Manager org.springframework.transaction.jta.JtaTransactionManager
* En el applicationContext.xml agregamos<tx:jta-transaction-manager/>

### Datos correctos (OK)

##### Pantalla

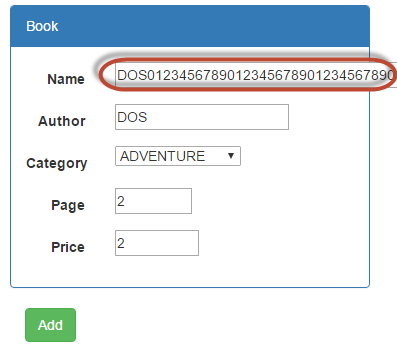


##### Estatus de Origen de Datos (JDBC)



### Datos incorrectos (OK)

##### Pantalla



##### Estatus de Origen de Datos (JDBC)



**No hay bloqueo de conexiones!**

# Referencias

1. <https://docs.oracle.com/cd/E13218_01/wlp/docs81/sp2/db/5XA.html>
2. <https://docs.oracle.com/cd/B28359_01/appdev.111/b28424/adfns_xa.htm#BGBDAJCI>
3. <https://fizalihsan.github.io/technology/transaction.html>