

JDBC – Parte 2

Carrera: **COMPUTACIÓN E INFORMÁTICA**

Semestre: **2016 - II**

Nombre de Unidad Didáctica: **TALLER DE PROGRAMACION WEB**

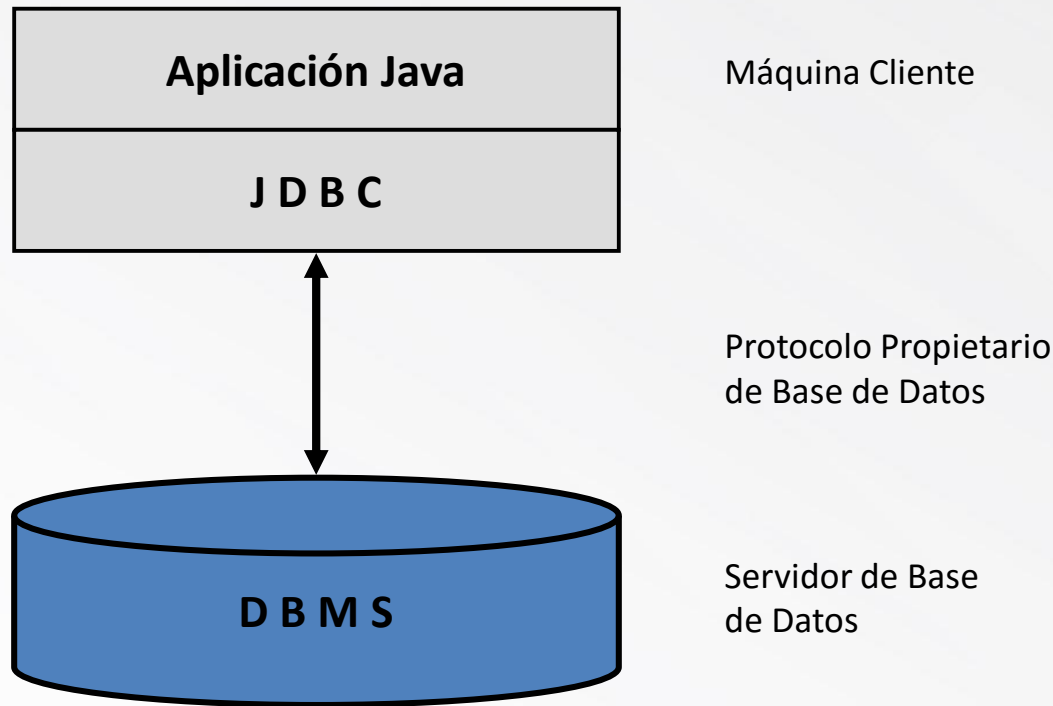
Profesor: **ERIC GUSTAVO CORONEL CASTILLO**

Blog: **gcoronelc.blogspot.com**



- Objetivo
- Archivo de Propiedades
- Transacción
- Transacciones Controladas Desde el Cliente
- Procedimientos Almacenados
- CallableStatement
- Transacciones de Base de Datos

- Utilizar Archivo de Propiedades.
- Programar transacciones.



datos.properties

```
# Parámetros de conexión
```

```
Driver=oracle.jdbc.OracleDriver
```

```
URL=jdbc:oracle:thin:@192.168.52.129:1521:orcl
```

```
User=eurekabank
```

```
Password=admin
```

```
package parametros;

import java.io.InputStream;
import java.util.Properties;

/**
 *
 * @author Eric Gustavo Coronel Castillo
 */
public class Parametros extends Properties {

    public Parametros() throws Exception {
        super();
        InputStream in = Parametros.class.getResourceAsStream("datos.properties");
        this.load(in); // Cargamos el contenido del flujo
        in.close();
    }

}
```

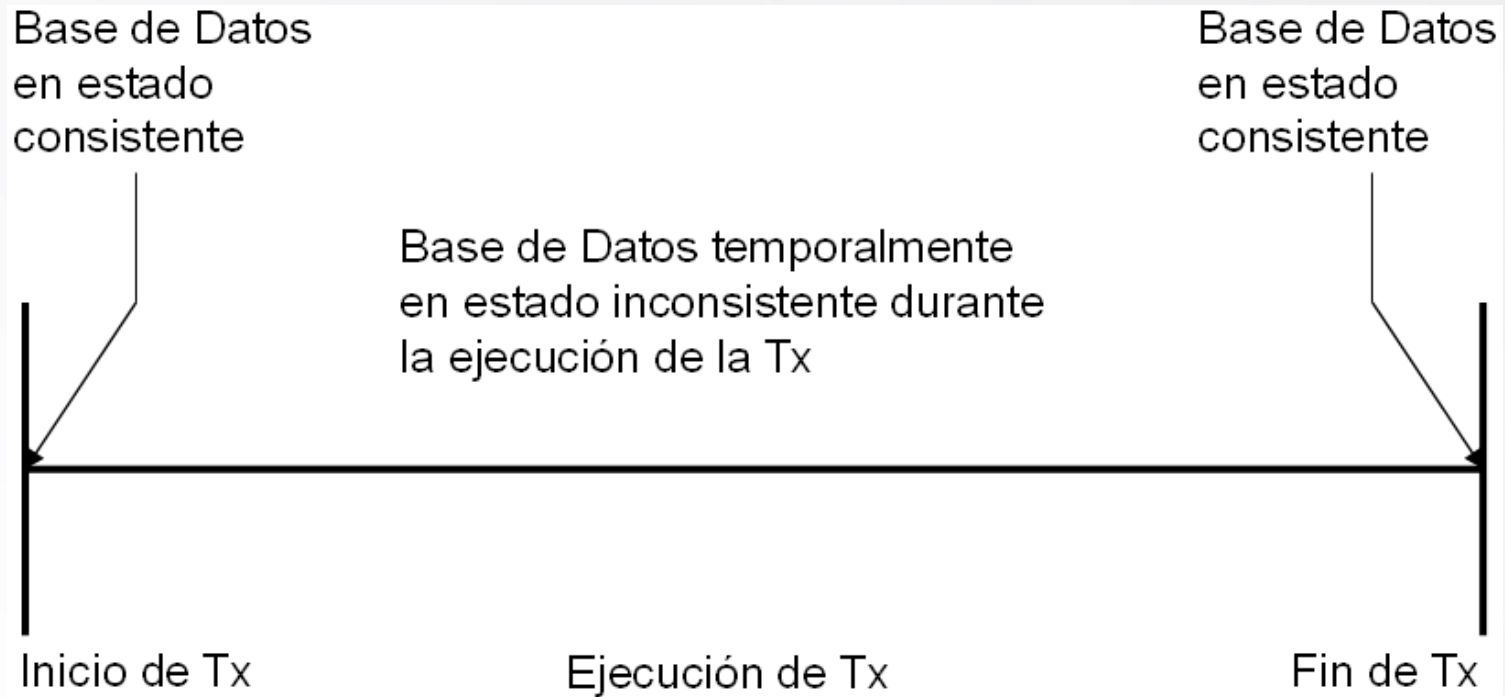
```
package pruebas;

import parametros.Parametros;

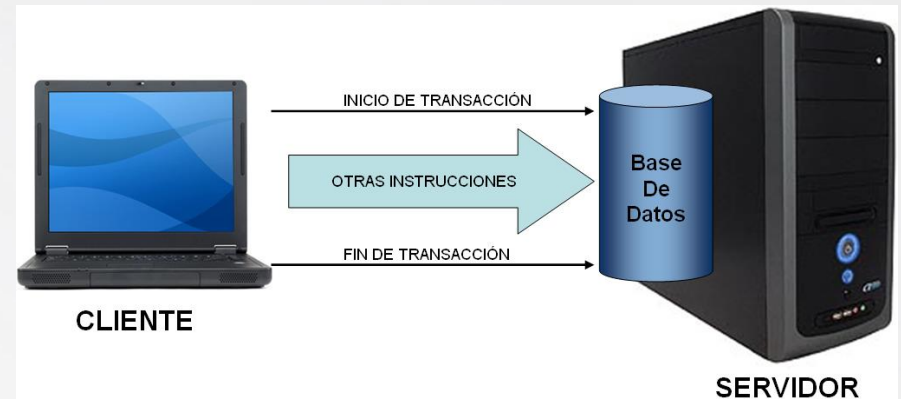
public class Prueba01 {

    public static void main(String[] args) {
        try {
            Parametros param = new Parametros();
            System.out.println("Driver: " + param.getProperty("Driver"));
            System.out.println("URL: " + param.getProperty("URL"));
            System.out.println("User: " + param.getProperty("User"));
            System.out.println("Password: " + param.getProperty("Password"));
        } catch (Exception ex) {
            System.out.println(ex.getMessage());
        }
    }
}
```

- Una transacción es un grupo de acciones que hacen transformaciones consistentes en las tablas preservando la consistencia de la base de datos.



```
1.  try {  
2.      // Inicio de la Transacción  
3.      cn.setAutoCommit(false);  
  
4.      // Otras instrucciones  
  
5.      // Confirmar Transacción  
6.      cn.commit();  
7.  } catch (Exception e) {  
8.      // Cancelar transacción  
9.      cn.rollback();  
10.     // Otras instrucciones de control  
11. }
```



Esquema 1

```
DELIMITER $$
```

```
CREATE PROCEDURE nombreProcedimiento
```

```
( Parámetros)
```

```
BEGIN
```

```
-- Instrucción SELECT
```

```
END$$
```

```
DELIMITER ;
```



Esquema 2

```
delimiter //  
  
CREATE PROCEDURE nombreProcedimiento  
( parámetros )  
BEGIN  
  
    -- Iniciar Transacción  
    START TRANSACTION;  
  
    -- Instrucciones INSERT, UPDATE ó DELETE  
  
    -- Confirmar Transacción  
    COMMIT;  
  
END //  
  
delimiter;
```

- Permite realizar llamadas a procedimientos almacenados de la base de datos.
- Es posible utilizar parámetros de entrada y salida.
- Cada parámetro está definido por un signo de interrogación (?).
- Antes de ejecutarse la sentencia se debe especificar un valor para cada uno de los parámetros de entrada a través de los métodos **setXXX** apropiados y registrar los parámetros de salida.

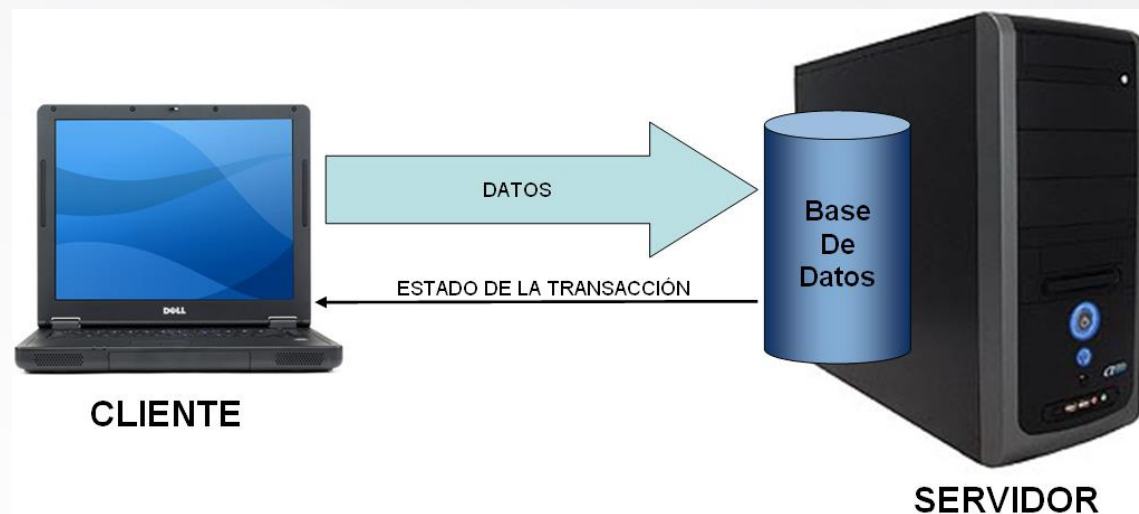
Sintaxis

{call nombre_del_procedimiento}

{call nombre_del_procedimiento[(?,?,...)]}

{?=call nombre_del_procedimiento[(?.?...)]}

```
1.  try {  
2.      // Habilitamos el autocommit  
3.      cn.setAutoCommit(true);  
  
4.      // Ejecutamos el procedimiento almacenado  
  
5.  } catch (Exception e) {  
6.      // Instrucciones de control  
7.  }
```



- Desarrollando Soluciones con Java y MySQL Server
Eric Gustavo Coronel Castillo
- Piensa en Java
Bruce Eckel
- Como Programar en Java
Deitel y Deitel
- Java 2
Steven Holzner
- La Biblia de Java 2 v5.0
Herbert Schildt
- Acceso a Bases de Datos con Java-JDBC
Ángel Esteban