

## CURSO DE JAVA CON JDBC

# MANEJO DE JDBC



Por el experto: Ing. Ubaldo Acosta



### CURSO DE JAVA CON JDBC

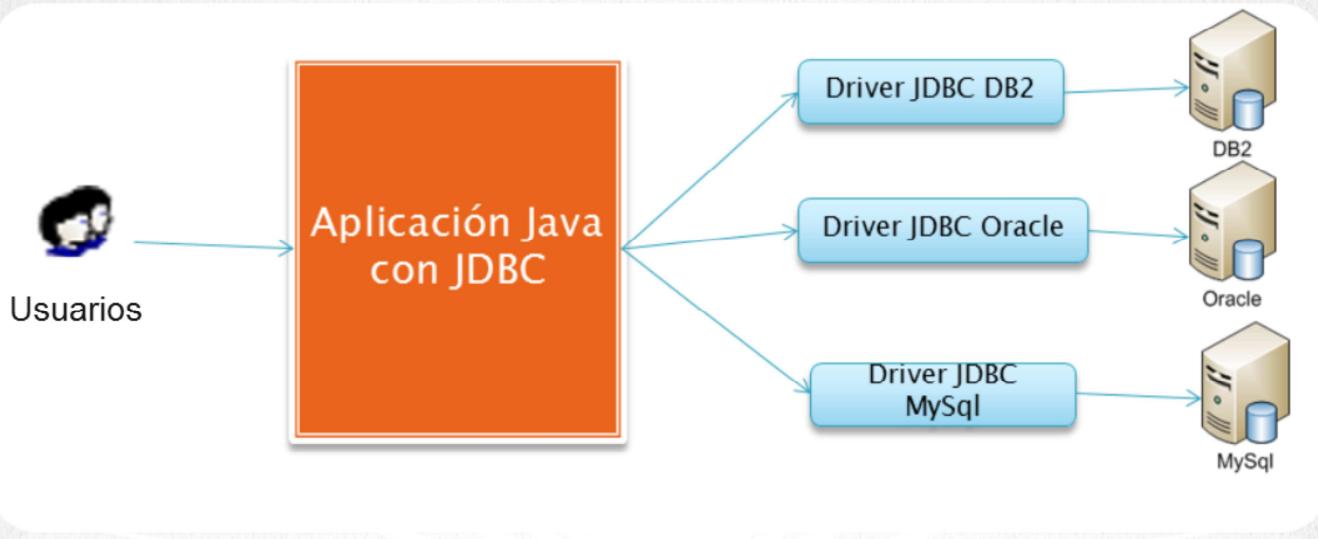
[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Hola, te saluda Ubaldo Acosta. Bienvenida o bienvenido nuevamente. Espero que estés listo para comenzar con esta lección.

Vamos a estudiar el tema cómo manejar el API de JDBC.

¿Estás listo? Ok, ¡Vamos!

# DRIVERS DE JDBC



## CURSO DE JAVA CON JDBC

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

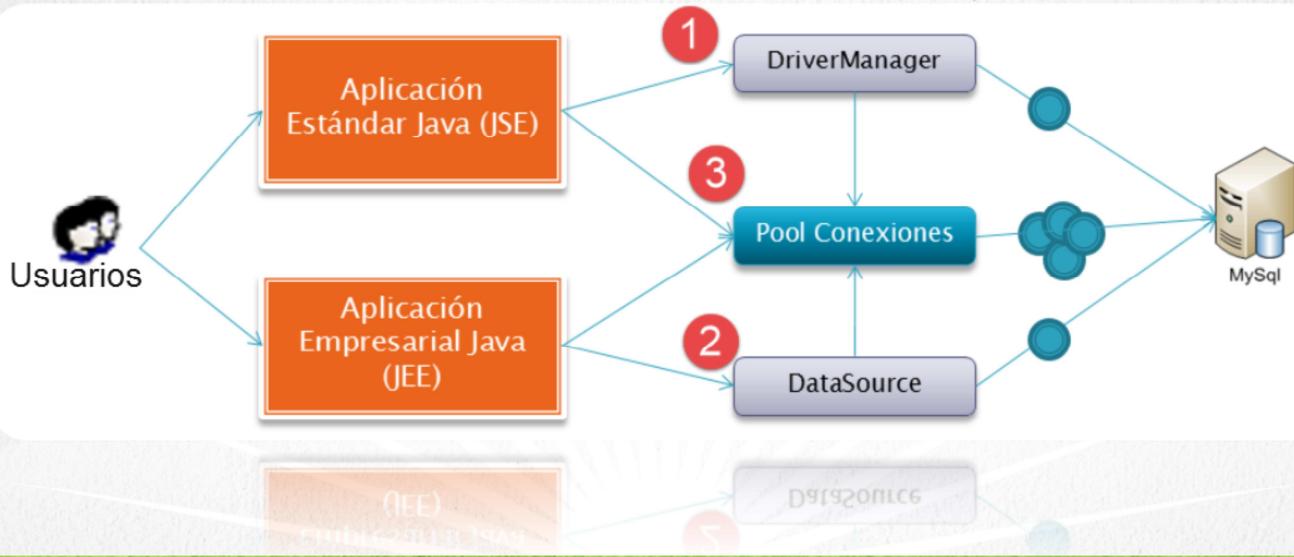
Vamos a revisar a continuación el tema de Drivers JDBC.

Un driver es la implementación de la especificación del API de JDBC definido por Java. Como hemos comentado antes, cada proveedor de base de datos implementa una o varias versiones para que desde nuestras aplicaciones Java podamos conectarnos a su base de datos.

La figura nos muestra como una aplicación Java puede conectarse a una base de datos por ejemplo DB2, ORACLE, MySQL entre otras. Y también podemos observar como cada proveedor crea su propio driver para conectarse a su base de datos respectiva.

Existen varias versiones de drivers JDBC, por lo que debemos tener en cuenta que dependiendo de la versión de JDK es la versión que podemos utilizar del driver de JDBC, así que normalmente para poder utilizar la última del driver de JDBC de la base de datos que seleccionemos, también tenemos que utilizar una versión reciente del JDK.

# DRIVERS DE JDBC



## CURSO DE JAVA CON JDBC

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Vamos a revisar ahora el tema de tipo de conexiones JDBC. Como podemos observar en la figura, existen 2 formas para conectarnos a una base de datos utilizando el driver de JDBC.

La primera forma que se utilizó en las primeras versiones de JDBC es la opción de `DriverManager`. Esta clase se encuentra en el paquete `java.sql` y nos permite crear una nueva conexión al apoyarnos del método `Class.forName`.

En la segunda forma podemos utilizar el concepto de `DataSource`. Esta opción es la más recomendada, aunque normalmente debemos de apoyarnos de un WebServer o un application server como puede ser Tomcat, Jboss, Glassfish, entre otros.

Una de las ventajas de utilizar el concepto de `DataSource` es que debido a que la configuración se hace vía JNDI (Java Naming and Directory Interface) la cual es un API de Java para el manejo de un directorio de recursos, esta configuración permite que el Application Server podamos cambiar los valores de la conexión a base de datos al vuelo y sin necesidad de modificar nuestro código Java. Veremos más ejemplos de la configuración de este tipo de conexiones en el curso de JSPs y Servlets.

Por último existe una tercera forma de conectarnos a la base de datos y es por medio de un pool de conexiones. Este pool de conexiones es un conjunto de objetos los cuales ya tienen una conexión viva a la base de datos, esto nos permite minimizar el costo y tiempo de creación de los objetos debido a que cada conexión a la base de datos es un proceso costoso, y por lo tanto, si ya tenemos disponibles varios objetos de conexión a bases de datos, minimizaremos los recursos y aumentaremos el performance de nuestra aplicación en Java.

Por último, podemos observar en la figura que normalmente una aplicación estándar hace uso de lo que es el driver manager y una aplicación empresarial hace uso de un `DataSource`, en ambos casos podemos combinar las tecnologías para utilizar un pool de conexiones y así eficientar cada uno de los procesos según la tecnología que estemos utilizando.

## TIPOS DE STATEMENTS (SENTENCIAS) EN JDBC

- **La interface statement de jdbc tiene distintos tipos:**
- **Statement:** se utiliza para cualquier tipo de sentencia SQL, pero no hace cache del SQL ejecutado.
- **PreparedStatement:** Se usa para hace cache del query a ejecutar, evitando la re-compilación de la sentencia SQL.
- **CallableStatement:** se utiliza para llamar a procedimientos almacenados en una Base de Datos.

### CURSO DE JAVA CON JDBC

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Vamos a revisar a continuación la interface Statement (Sentencia), que es la más básica de las interfaces para ejecutar una sentencia SQL.

Al ejecutar una sentencia SQL en repetidas ocasiones, un objeto statement compila en cada ocasión la consulta SQL que se ejecuta.

La interfaz PrepareStatement hereda de la interfaz Statement y puede almacenar de manera precompilada una sentencia SQL, por lo que es una opción más eficiente si es que necesitamos ejecutar una sentencia SQL en repetidas ocasiones, a su vez también podemos especificar parámetros a ejecutar en nuestro query, esto lo veremos en ejercicios posteriores.

Por último, la interface CallableStatement se utiliza para mandar a llamar un procedimiento almacenado, el cual puede regresar información en forma de *cursor*, un tipo de dato específico o en alguna otra forma según el resultado que se necesite procesar.,

A continuación vamos a crear ejemplos relacionados con la interface Statement.

# MÉTODOS DE LA INTERFACE STATEMENTS EN JDBC

- La interface statement tiene varios métodos, como son:

Para una sentencia SELECT:

`executeQuery( String sql )`: Regresa un objeto ResultSet para procesar los registros

Para una sentencia DML/DDL:

`executeUpdate( String sql )`: Regresa un int

Para cualquier sentencia SQL:

`execute( String )`: Regresa un boolean

**CURSO DE JAVA CON JDBC**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

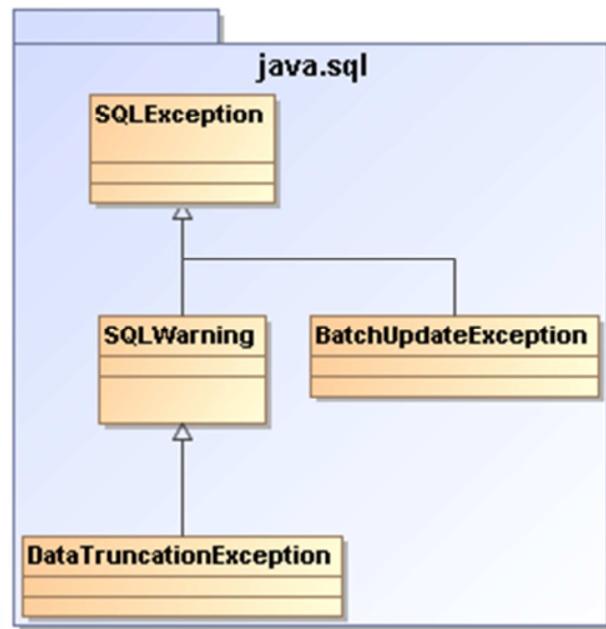
Ahora vamos a revisar el uso de Statement en JDBC. La interface de statement tiene varios métodos. Existen métodos para hacer un SELECT, para ejecutar alguna instrucción DML(Data Manipulation Language), DDL (Data Definition Language) o para cualquier otra sentencia de tipo SQL.

El método de ***executeQuery*** se utiliza para ejecutar sentencias de tipo select y por ello el método regresa un objeto ResultSet, el cual almacena el resultado en forma de una matriz de dos dimensiones, es decir, en renglones y columnas, así podemos procesar el resultado del query sin ningún problema.

El método ***executeUpdate*** se utiliza para ejecutar sentencias DML (Data Manipulation Language) como son las sentencias insert, update y delete. También nos va a permitir ejecutar sentencias de tipo DDL (Data Definition Language) como son las sentencias create table, truncate table, entre otras. La función `executeUpdate` regresa un entero, indicando el número de registros afectados como resultado de ejecutar el query deseado.

Por último, el método de ***execute*** nos sirve para ejecutar cualquier tipo de sentencia SQL. Este método regresa un valor booleano, el cual regresa true si el resultado es un ResulSet, esto es, si se ejecutó un select, y regresa false si se ejecutó alguna otra sentencia, como insert, update, delete, o alguna otra.

# EXCEPCIONES DE JDBC



**CURSO DE JAVA CON JDBC**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En la figura mostrada podemos observar algunas de las clases de excepciones en el API de JDBC.

La clase `SQLException` es la más utilizada para procesar excepciones al utilizar JDBC. Existen más clases de excepción, las cuales se encuentran en el paquete `java.sql`.

Debido a que la clase `SQLException` desciende de la clase `Exception` estamos obligados a procesar dicha excepción por un bloque `try/catch` o debemos declararla como parte de la firma del método que hace uso del API JDBC.

Poco a poco vamos a ir revisando el manejo de excepciones en JDBC, pero en los ejercicios que vamos a realizar estaremos manejando este concepto para que vayamos aclarándolo poco a poco.

## EJERCICIOS CURSO PROGRAMACIÓN CON JAVA

- **ABRIR LOS ARCHIVOS DE EJERCICIOS EN PDF.**
- **EJERCICIO:** Ejercicio Manejo JDBC.

**CURSO DE JAVA CON JDBC**

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

# CURSO ONLINE

# JAVA CON JDBC

Por: Ing. Ubaldo Acosta



## CURSO DE JAVA CON JDBC

[www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

En Global Mentoring promovemos la Pasión por la Tecnología Java. Te invitamos a visitar nuestro sitio Web donde encontrarás cursos Java Online desde Niveles Básicos, Intermedios y Avanzados, y así te conviertas en un experto programador Java.

Además agregamos nuevos cursos para que continúes con tu preparación como programador Java profesional. A continuación te presentamos nuestro listado de cursos:

- |                          |                                     |
|--------------------------|-------------------------------------|
| ✓ Lógica de Programación | ✓ Hibernate Framework               |
| ✓ Fundamentos de Java    | ✓ Spring Framework                  |
| ✓ Programación con Java  | ✓ JavaServer Faces                  |
| ✓ Java con JDBC          | ✓ Java EE (EJB, JPA y Web Services) |
| ✓ HTML, CSS y JavaScript | ✓ JBoss Administration              |
| ✓ Servlets y JSP's       | ✓ Android con Java                  |
| ✓ Struts Framework       | ✓ HTML5 y CSS3                      |

### Datos de Contacto:

Sitio Web: [www.globalmentoring.com.mx](http://www.globalmentoring.com.mx)

Email: [informes@globalmentoring.com.mx](mailto:informes@globalmentoring.com.mx)

