



El futuro digital  
es de todos

MinTIC

# Ciclo 2: Programación Básica

## Sesión 22: JDBC

Programa Ciencias de la Computación e Inteligencia Artificial  
Escuela de Ciencias Exactas e Ingeniería  
Universidad Sergio Arboleda  
Bogotá



UNIVERSIDAD  
SERGIO ARBOLEDA

Misión  
TIC 2022



# Contenido

- Introducción JDBC
- JDBC instalación
- CRUD usando JDBC





El futuro digital  
es de todos

MinTIC

# Introducción JDBC



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC 2022’

# Java DataBase Connectivity (JDBC)



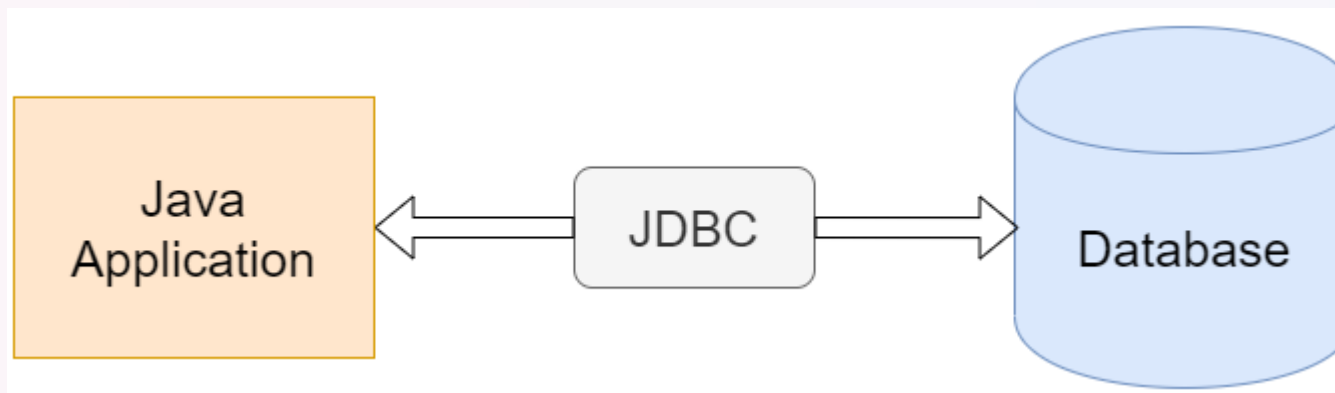
El futuro digital  
es de todos

MinTIC

Es una API que permite acceder desde Java a cualquier motor de bases de datos.

JDBC permite escribir aplicaciones Java que gestionan estas tres actividades de programación:

1. Conectarse a una fuente de datos, como una base de datos
2. Enviar consultas y actualizar registros a la base de datos.
3. Recuperar y procesar los resultados recibidos de la base de datos en respuesta a su consulta



# Java DataBase Connectivity (JDBC)



El futuro digital  
es de todos

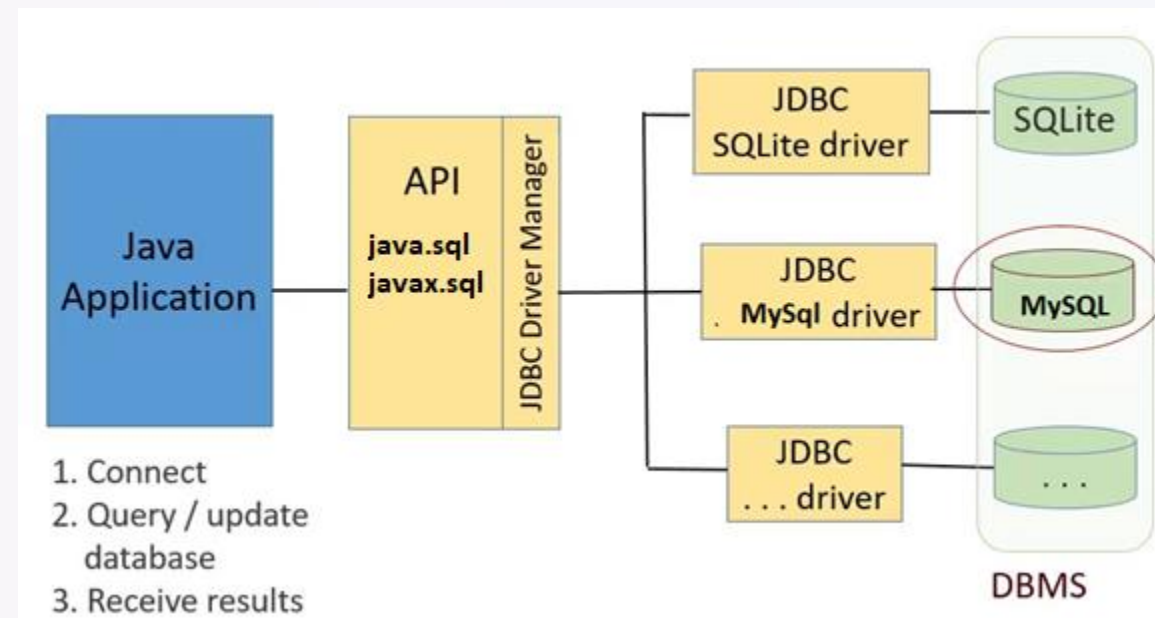
MinTIC

La API de JDBC se compone de dos paquetes: `java.sql` y `javax.sql`.

La API de JDBC consta de dos partes:

La primera parte de la API JDBC es parte de los paquetes java estándar en el paquete `java.sql`. Usamos la API del paquete `java.sql` para acceder y procesar datos almacenados en una fuente de datos (generalmente una base de datos relacional) usando el lenguaje de programación Java.

La segunda parte es la API de bajo nivel para conectarse a un servidor de base de datos (JDBC Driver Manager).



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022



# API – JDBC Clases Importante del Paquete java.sql



El futuro digital  
es de todos

MinTIC

No.	Clase	Descripción
1	java.sql.Connection	Representa una conexión a la base de datos. Abstrae el detalle de cómo comunicarse con el servidor de base de datos.
2	java.sql.DriverManager	Administra los drivers JDBC usados por la aplicación. En conjunto con el URL y la autenticación apropiada, puede proveer aplicaciones con objetos Connection válidos.
3	java.sql.Statement	Provee métodos para que los desarrolladores puedan ejecutar sentencias SQL.
	java.sql.PreparedStatement	Subclase de java.sql.Statement, la cual va a contener una consulta a la cual se le pueden enviar parámetros.
4	java.sql.ResultSet	Representa los resultados de una sentencia SQL. Estos objetos son retornados usualmente por métodos del objeto Statement o del objeto PreparedStatement.



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC 2022

# Java DataBase Driver Manager

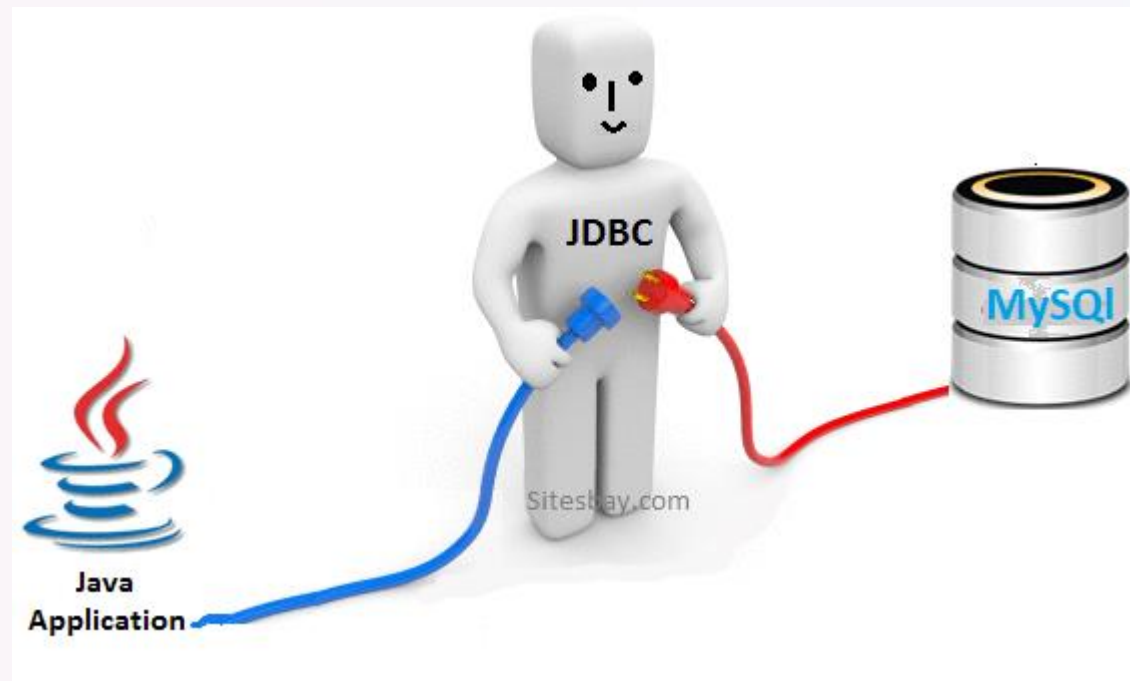


El futuro digital  
es de todos

MinTIC

Para poder utilizar JDBC:

- Es necesario conseguir el Driver adecuado (Archivo .jar)
- Necesitamos el Driver Java – MySQL (Archivo **mysql-connector-java-8.X.XX.jar**)
- El JDK no trae todos los drivers, ya que sería muy pesado.





El futuro digital  
es de todos

MinTIC

# JDBC Descarga y Uso Driver



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC 2022’



# Java DataBase Connectivity (JDBC) - Instalación



El futuro digital  
es de todos

MinTIC

- Es posible que el conector ya esté descargado, para confirmar buscar en las siguientes carpetas posibles:

C:/Program Files/MySQL/Connector J 8/

o en

C:/Program Files(x86)/MySQL/Connector J 8/

El archivo en esta carpeta debería llamar: **mysql-connector-java-8.XX.XX**



UNIVERSIDAD  
SERGIO ARBOLEDA



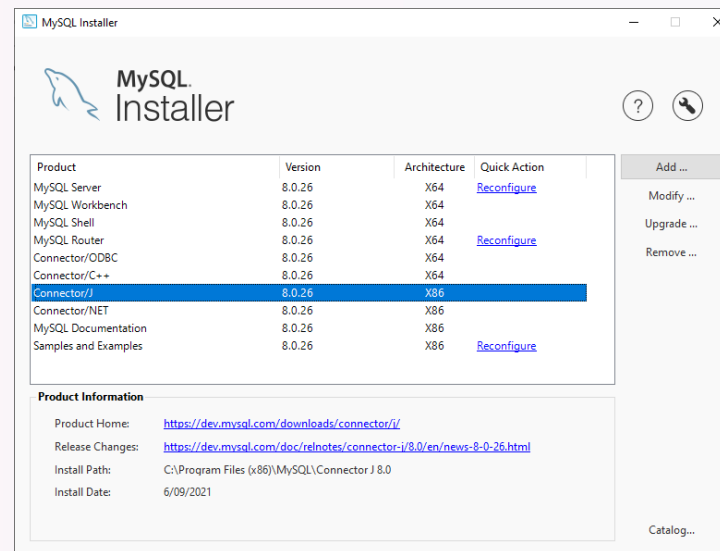
# Java DataBase Connectivity (JDBC) - Instalación



El futuro digital  
es de todos

MinTIC

- Si no se consigue en la carpeta se descarga la última versión en:  
<https://mvnrepository.com/artifact/mysql/mysql-connector-java>
- Otra forma un poco más larga es ejecutar nuevamente el instalador de MySQL (mysql-installer-community-8.0.26.0.msi) y con el asistente añadir (add) el conector de Java.



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC 2022

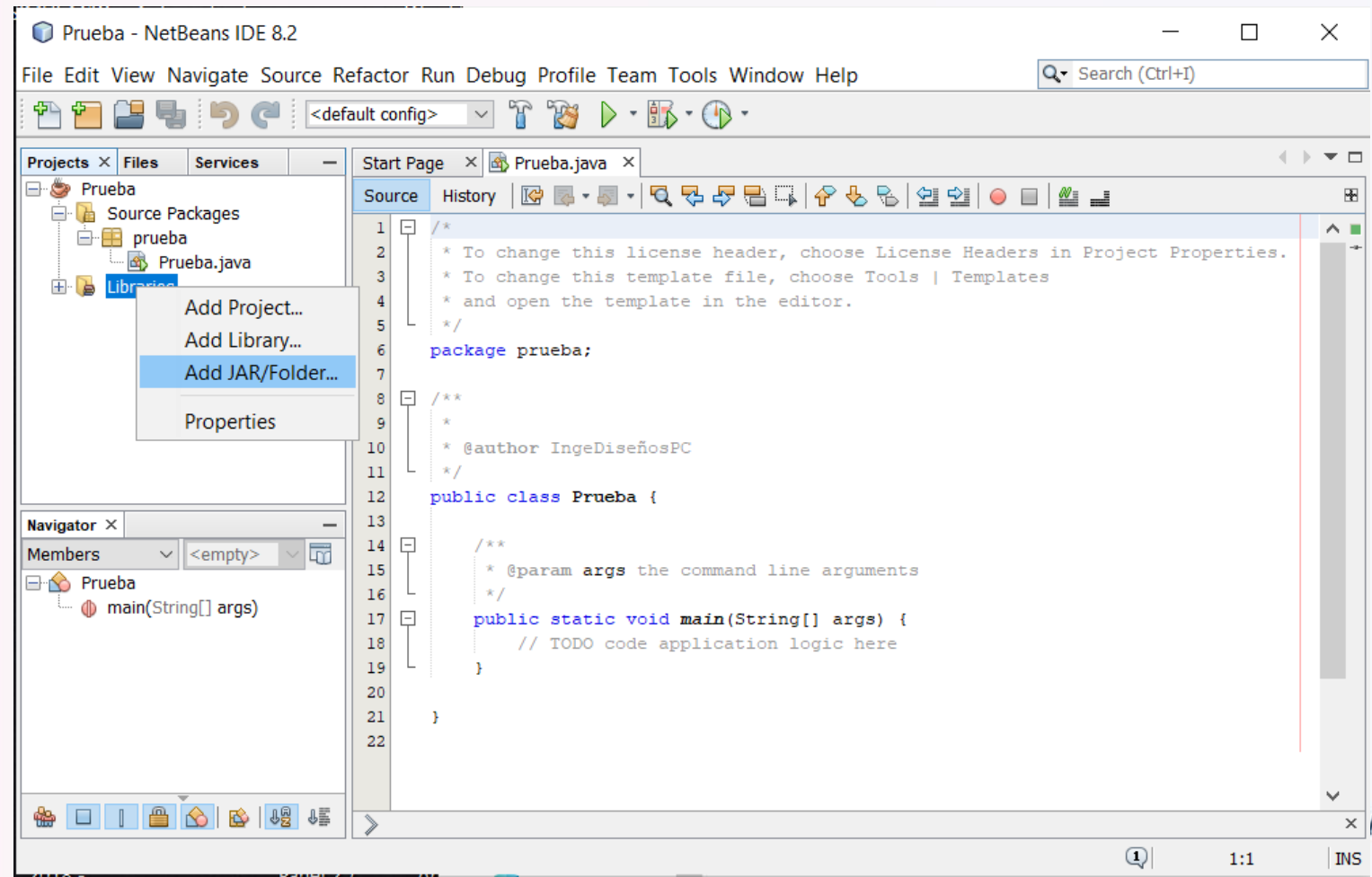
# Java DataBase Connectivity (JDBC)



El futuro digital  
es de todos

MinTIC

En el árbol del proyecto que se quiere conectar:  
Click secundario en Libraries > Add JAR/Folder



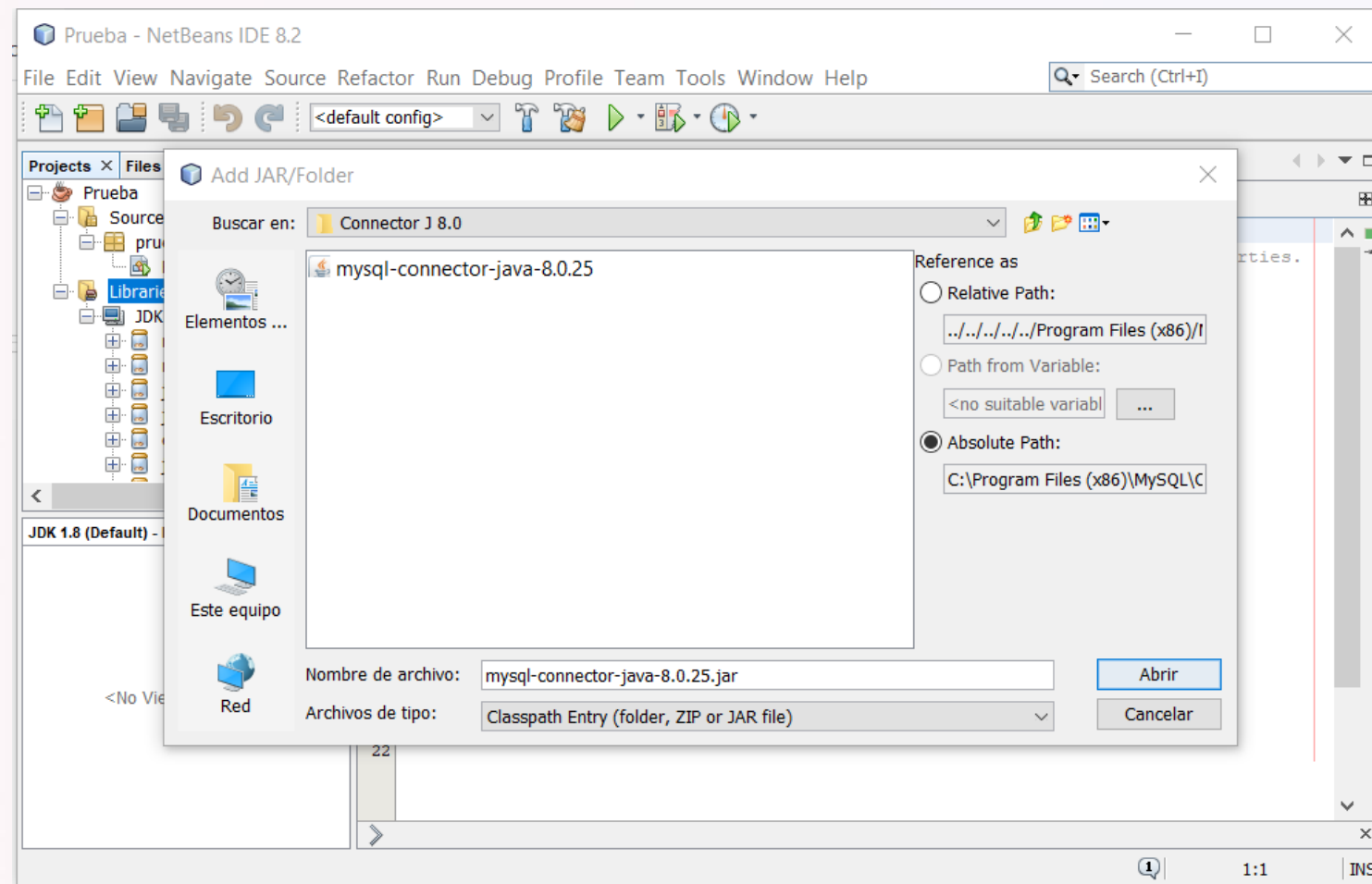
# Java DataBase Connectivity (JDBC)



El futuro digital  
es de todos

MinTIC

Buscar el archivo del conector.



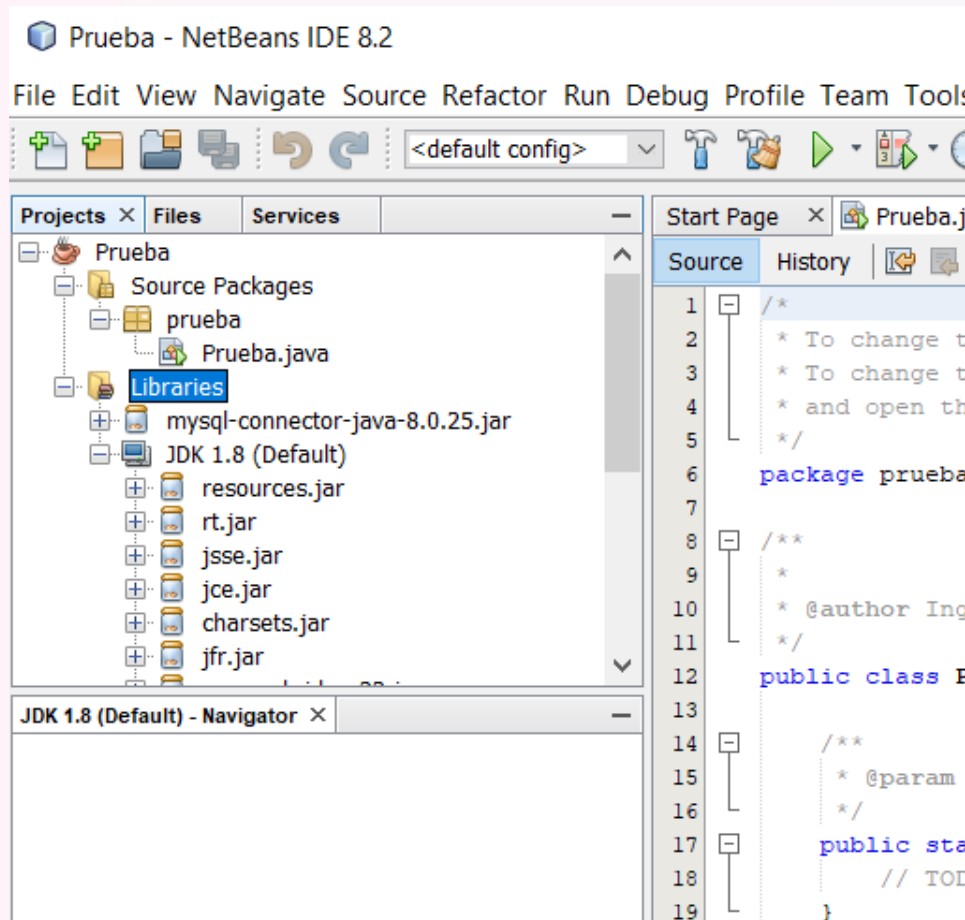
# Java DataBase Connectivity (JDBC)



El futuro digital  
es de todos

MinTIC

El conector debería quedar en las librerías.



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022





# Crear Base de Datos en MySQL WorkBench





# Creación Base de Datos / Tablas / Registros

```
ScriptDBClientes x SQL File 3*
1 • CREATE SCHEMA `erp` DEFAULT CHARACTER SET utf8mb4 ;
2 • CREATE TABLE `erp`.`clientes` (
3     `cedula` int NOT NULL,
4     `Nombre` varchar(45) DEFAULT NULL,
5     `Ciudad` varchar(45) DEFAULT NULL,
6     `Pais` varchar(45) DEFAULT NULL,
7     PRIMARY KEY (`cedula`)
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
9 • INSERT INTO `erp`.`clientes` (`cedula`, `Nombre`, `Ciudad`, `Pais`) VALUES ('76232', 'Juan Días',
    'Bogotá', 'Colombia');
10 • INSERT INTO `erp`.`clientes` (`cedula`, `Nombre`, `Ciudad`, `Pais`) VALUES ('232', 'Daniel Pardo',
    'Paris', 'Francia');
11 • INSERT INTO `erp`.`clientes` (`cedula`, `Nombre`, `Ciudad`, `Pais`) VALUES ('323', 'Stephen King',
    'Cota', 'Colombia');
```





# Pasos fundamentales para conectarse a una Base de Datos



# Pasos



El futuro digital  
es de todos

MinTIC

1. Abra una conexión (connection) a la base de datos.
2. Cree un objeto de declaración (Statement ó PreparedStatement) para ser ejecutado.
3. Ejecute el objeto de declaración (Statement ó PreparedStatement), procese los resultados.
4. Procese un resulset si el Statement ó PreparedStatement fue una consulta (query)



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC2022’



# Abrir Conexión a Base de datos







# API – JDBC Clases para crear la Conexión

No.	Clase	Descripción
1	java.sql.Connection	Representa una conexión a la base de datos. Abstrae el detalle de cómo comunicarse con el servidor de base de datos.
2	java.sql.DriverManager	Administra los drivers JDBC usados por la aplicación. En conjunto con el URL y la autenticación apropiada, puede proveer aplicaciones con objetos Connection válidos.





# java.sql.DriverManager /java.sql.Connection

Permite al desarrollador conseguir un objeto Connection la cual se puede usar para ejecutar actividades en la base de datos.

Para establecer la conexión a la base de datos, se debe usar el método `getConnection()` dándole el URL JDBC, también con el usuario y contraseña para acceder a la base de datos si aplica.

El URL debe seguir la sintaxis requerida por la implementación particular de la base de datos.

```
String url = "jdbc:<SGBD>:<database>";
String user = "username";
String password = "password";
Connection conn = null;

try {
    conn = DriverManager.getConnection(url, user, password);
    ...
} catch (SQLException e) {
    // Manejo de los errores
}
```





# Cadenas de Conexión JDBC

DBMS	URL
IBM DB2	<code>jdbc:db2://&lt;HOST&gt;:&lt;PORT&gt;/&lt;DB&gt;</code>
MySQL	<code>jdbc:mysql://&lt;HOST&gt;:&lt;PORT&gt;/&lt;DB&gt;</code>
Postgresql	<code>jdbc:postgresql://&lt;HOST&gt;:&lt;PORT&gt;/&lt;DB&gt;</code>
Ms SQL Server	<code>jdbc:sqlserver://&lt;HOST&gt;\&lt;DB&gt;:&lt;PORT&gt;</code>
Oracle	<code>jdbc:oracle:thin:@&lt;HOST&gt;:&lt;PORT&gt;:&lt;SID&gt;</code>
JDBC-ODBC Bridge	<code>jdbc:odbc:&lt;DB&gt;</code>
SQLite	<code>jdbc:sqlite:&lt;FILE_PATH&gt;</code> <code>jdbc:sqlite::memory:</code>



# Java DataBase Connectivity (JDBC) – Conexión - MySQL



El futuro digital  
es de todos

MinTIC

Hay algunas propiedades que deben estar claras al momento de iniciar una conexión Java-MySQL:

- **URL:** Se debería ver similar a esto:  
`jdbc:mysql://localhost:3306/nombre_esquema?serverTimezone=UTC`
- **Usuario y contraseña:** Puede ser el root con la clave que se creo al instalar MySQL o se puede crear una cuenta nueva.



UNIVERSIDAD  
SERGIO ARBOLEDA



# Java DataBase Connectivity (JDBC) - Conexión

Para crear la conexión se necesita el siguiente código:



El futuro digital  
es de todos

MinTIC

```
public static void main(String[] args) {  
  
    String url = "jdbc:mysql://localhost:3306/erp?serverTimezone=UTC";  
    String user = "root";  
    String password = "admin1234";  
    try {  
        Connection conn = DriverManager.getConnection(url, user, password);  
        if (conn != null) {  
            System.out.println("Conectado");  
        }  
    } catch (SQLException ex) {  
        ex.printStackTrace();  
    }  
}
```



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC 2022





# Usando JDBC para INSERT, UPDATE, o DELETE

No.	Clase	Descripción
1	java.sql.Connection	Representa una conexión a la base de datos. Abstrae el detalle de cómo comunicarse con el servidor de base de datos.
2	java.sql.DriverManager	Administra los drivers JDBC usados por la aplicación. En conjunto con el URL y la autenticación apropiada, puede proveer aplicaciones con objetos Connection válidos.
3	java.sql.Statement	Provee métodos para que los desarrolladores puedan ejecutar sentencias SQL.
	java.sql.PreparedStatement	Subclase de java.sql.Statement, la cual va a contener una consulta a la cual se le pueden enviar parámetros.
4	java.sql.ResultSet	Representa los resultados de una sentencia SQL. Estos objetos son retornados usualmente por métodos del objeto Statement o del objeto PreparedStatement.



# JDBC Clases para INSERT, UPDATE, o DELETE



El futuro digital  
es de todos

MinTIC

Una vez tenemos la conexión, podemos crear una instancia de `java.sql.PreparedStatement`, la cual nos permitirá ejecutar sentencias SQL.

Los objetos `PreparedStatement` provee varios métodos para ejecutar SQL. El más usado para sentencias INSERT, UPDATE, o DELETE es:

- **`executeUpdate`**: Toma una sentencia INSERT, UPDATE, o DELETE y retorna el numero de filas afectadas por la operacion.



UNIVERSIDAD  
SERGIO ARBOLEDA



# CRUD usando JDBC



El futuro digital  
es de todos

MinTIC

CRUD: Las operaciones básicas de consulta: Crear, Consultar, Actualizar y Borrar.

La sintaxis para conectarse a la base de datos MySQL se puede usar para cualquier otro manejador SQL ya que es estándar.

```
public static void main(String[] args) {  
  
    String url = "jdbc:mysql://localhost:3306/erp?serverTimezone=UTC";  
    String user = "root";  
    String password = "admin1234";  
    try {  
        Connection conn = DriverManager.getConnection(url, user, password);  
        if (conn != null) {  
            System.out.println("Conectado");  
        }  
        //Aquí va el código  
    } catch (SQLException ex) {  
        ex.printStackTrace();  
    }  
}
```



UNIVERSIDAD  
SERGIO ARBOLEDA

**Mision**  
**TIC2022**

# CRUD usando JDBC – Create (Insertar)



El futuro digital  
es de todos

MinTIC

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
232	Daniel Pardo	Paris	Francia
323	Stephen King	Cota	Colombia

```
String sql= "INSERT INTO Clientes (Cedula, Nombre, Ciudad, Pais) VALUES (?, ?, ?, ?)";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1,32563);
pstmt.setString(2,"Luis Diaz");
pstmt.setString(3,"Barrancas");
pstmt.setString(4,"Colombia");
int filasAgregadas = pstmt.executeUpdate();
if (filasAgregadas > 0){
    System.out.println("¡Inserción exitosa!");
}
```



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC2022’

# CRUD usando JDBC – Create (Insertar)



El futuro digital  
es de todos

MinTIC

“conn” es el nombre de la instancia de la conexión creada previamente.

```
String sql= "INSERT INTO Clientes (Cedula, Nombre, Ciudad, Pais) VALUES (?, ?, ?, ?)";
PreparedStatement pstmt = conn.prepareStatement(sql);
pstmt.setInt(1,32563);
pstmt.setString(2,"Luis Diaz");
pstmt.setString(3,"Barrancas");
pstmt.setString(4,"Colombia");
int filasAgregadas = pstmt.executeUpdate();
if (filasAgregadas > 0){
    System.out.println("¡Inserción exitosa!");
}
```

Es necesario que los valores sean signos de interrogación (placeholders) para luego ser introducidos.

El comando es ejecutado en este momento.

Agregar los datos de esta manera le permite al driver verificar el tipo de dato que se está pasando. Ej: `setString("")`;



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022



# CRUD usando JDBC - UPDATE

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
232	Daniel Pardo	Paris	Francia
323	Stephen King	Cota	Colombia



El futuro digital  
es de todos

MinTIC

```
String sql = "UPDATE Clientes SET Ciudad=?, Pais=? WHERE Nombre=";
```

```
PreparedStatement pstmt = conn.prepareStatement(sql);
```

```
pstmt.setString(1,"New York");
```

```
pstmt.setString(2,"Estados Unidos");
```

```
pstmt.setString(3,"Stephen King");
```

```
int filasAgregadas = pstmt.executeUpdate();
```

```
if (filasAgregadas > 0){
```

```
    System.out.println("¡Actualización exitosa!");
```

```
}
```

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
232	Daniel Pardo	Paris	Francia
323	Stephen King	New York	Estados Unidos



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022

# CRUD usando JDBC - UPDATE



El futuro digital  
es de todos

MinTIC

“conn” es el nombre de la instancia de la conexión creada previamente.

```
String sql = "UPDATE Clientes SET Ciudad=?, Pais=? WHERE Nombre=?";
```

```
PreparedStatement pstm = conn.prepareStatement(sql);  
pstm.setString(1,"New York");  
pstm.setString(2,"Estados Unidos");  
pstm.setString(3,"Stephen King");
```

```
int filasAgregadas = pstm.executeUpdate();  
if (filasAgregadas > 0){  
    System.out.println("¡Actualización exitosa!");  
}
```

Los tipos de datos deben coincidir con los de la tabla en MySQL.

Se agregan los elementos sobre los signos de interrogación dándole el numero que le corresponda de acuerdo a su aparición en la sentencia.



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022

# CRUD usando JDBC - DELETE



El futuro digital  
es de todos

MinTIC

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
232	Daniel Pardo	Paris	Francia
323	Stephen King	New York	Estados Unidos

```
String sql = "DELETE FROM Clientes WHERE Nombre=?";
```

```
PreparedStatement pstmt = conn.prepareStatement(sql);  
pstmt.setString(1,"Daniel Pardo");
```

```
int filasEliminadas = pstmt.executeUpdate();  
if (filasEliminadas > 0){  
    System.out.println("¡Eliminación exitosa!");  
}
```

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
323	Stephen King	New York	Estados Unidos



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022

# CRUD usando JDBC - DELETE



El futuro digital  
es de todos

MinTIC

“conn” es el nombre de la instancia de la conexión creada previamente.

```
String sql = "DELETE FROM Clientes WHERE Nombre=?";
```

```
PreparedStatement pstmt = conn.prepareStatement(sql);  
pstmt.setString(1, "Daniel Pardo");
```

```
int filasEliminadas = pstmt.executeUpdate();  
if (filasEliminadas > 0){  
    System.out.println("¡Eliminación exitosa!");  
}
```

Se recomienda usar atributos únicos para eliminar. Si por ejemplo ponemos eliminar con País = “Colombia” se van a borrar TODOS los registro de Colombia.



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC2022’



# CRUD usando JDBC - Consultas

No.	Clase	Descripción
1	java.sql.Connection	Representa una conexión a la base de datos. Abstrae el detalle de cómo comunicarse con el servidor de base de datos.
2	java.sql.DriverManager	Administra los drivers JDBC usados por la aplicación. En conjunto con el URL y la autenticación apropiada, puede proveer aplicaciones con objetos Connection válidos.
3	java.sql.Statement	Provee métodos para que los desarrolladores puedan ejecutar sentencias SQL.
	java.sql.PreparedStatement	Subclase de java.sql.Statement, la cual va a contener una consulta a la cual se le pueden enviar parámetros.
4	java.sql.ResultSet	Representa los resultados de una sentencia SQL. Estos objetos son retornados usualmente por métodos del objeto Statement o del objeto PreparedStatement.







# java.sql.Statement/java.sql. PreparedStatement

- **executeQuery**: Toma una sentencia SELECT y retorna el resultado de la operación como un objeto ResultSet.

## java.sql.ResultSet

Encapsula los resultados de una consulta a la base de datos.

Los datos dentro de un objeto ResultSet pueden ser mejor visualizados como una tabla. La información puede ser recuperada de a una fila a la vez, y el objeto ResultSet mantendrá el control de la fila actual.

Para iterar por las filas, usamos el método **next()**, el cual mueve el apuntador de fila a la siguiente. Este método retorna true si existe una siguiente fila.

```
while (rs.next()) {  
    //Lee los datos de la fila actual  
}
```

Para obtener los datos de cada fila, el objeto ResultSet no da varios métodos get los cuales toman como parámetro el número de columna o el nombre de la columna.

- El método getString() es para obtener un dato String
- El método getInt() es para obtener un dato int
- El método getBoolean() es para obtener un dato boolean.

Es recomendado, sin embargo, usar los nombres de los campos en lugar de la posición en la fila.

# CRUD usando JDBC - RETRIEVE

Cedula	Nombre	Ciudad	País
762332	Juan Días	Bogotá	Colombia
232	Daniel Pardo	Paris	Francia
323	Stephen King	Cota	Colombia



El futuro digital  
es de todos

MinTIC

```
String sql = "SELECT * FROM Clientes";
```

```
Statement stm = conn.createStatement();
```

```
ResultSet rs = stm.executeQuery(sql);
```

```
while (rs.next()){
```

```
    int cedula = rs.getInt("cedula");
```

```
    String nombre = rs.getString("nombre");
```

```
    String ciudad = rs.getString("ciudad");
```

```
    String pais = rs.getString("pais");
```

```
    System.out.println(cedula+", "+ nombre + ", " + ciudad + ", "+ pais);
```

```
}
```



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022

# CRUD usando JDBC - RETRIEVE



El futuro digital  
es de todos

MinTIC

“conn” es el nombre de la instancia de la conexión creada previamente.

```
String sql = "SELECT * FROM Clientes";
```

```
Statement stm = conn.createStatement();
```

```
ResultSet rs = stm.executeQuery(sql);
```

```
while (rs.next()){
```

```
    int cedula = rs.getInt("cedula");
```

```
    String nombre = rs.getString("nombre");
```

```
    String ciudad = rs.getString("ciudad");
```

```
    String pais = rs.getString("pais");
```

```
    System.out.println(cedula+","+ nombre + "," + ciudad
```

```
+ ","+ pais);
```

```
}
```

Para manipular los datos resultantes de la consulta, se pasan el nombre de los campos a extraer,



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022

# CRUD usando JDBC - RETRIEVE



El futuro digital  
es de todos

MinTIC

“conn” es el nombre de la instancia de la conexión creada previamente.

```
String sql = "SELECT * FROM Clientes WHERE cedula=?";
```

```
PreparedStatement pstmt = conn.prepareStatement(sql);  
pstmt.setInt(1,323);
```

```
ResultSet rs = pstmt.executeQuery();
```

```
while (rs.next()){
```

```
    int cedula = rs.getInt("cedula");
```

```
    String nombre = rs.getString("nombre");
```

```
    String ciudad = rs.getString("ciudad");
```

```
    String pais = rs.getString("pais");
```

```
    System.out.println(cedula+","+ nombre + "," + ciudad  
+ ","+ pais);  
}
```

AL usar el WHERE se debe usar un prepareStatement con los signos de interrogación.

Para manipular los datos resultantes de la consulta, se pasan el nombre de los campos a extraer,



UNIVERSIDAD  
SERGIO ARBOLEDA

Mision  
TIC2022



El futuro digital  
es de todos

MinTIC

# Ejercicios



UNIVERSIDAD  
SERGIO ARBOLEDA

‘Mision  
TIC 2022’



# Ejercicios



El futuro digital  
es de todos

MinTIC

- Construya un esquema en MySQL con una tabla llamada Productos. La tabla debe tener los atributos: código, nombre, precio e inventario (inicialmente la tabla debe estar vacía)
- Implemente las operaciones CRUD para la tabla utilizando JDBC
- Utilizando los métodos implementados en el punto anterior:
  - Ingrese los siguientes registros a la tabla

código	nombre	precio	inventario
1	Manzanas	5000.0	25
2	Limonos	2300.0	15
3	Peras	2700.0	33
4	Arandanos	9300.0	5
5	Tomates	2100.0	42

- Actualice el inventario de las peras a 50
- Elimine todos los productos que comienzan con la letra “A” y con un inventario menor a 30
- Verifique que los cambios en la base de datos



UNIVERSIDAD  
SERGIO ARBOLEDA





# Referencias

- [1] Sierra, K., & Bates, B. (2013). Head first java. " O'Reilly Media, Inc."
- [2] Martin, R. C. (2009). Clean code: a handbook of agile software craftsmanship. Pearson Education. (Chap 1, 2)
- [3] Sommerville, I. (2016). Software Engineering GE. Pearson Australia Pty Limited. (Chap 6, 6.3)
- [4] Coronel, C., Morris, S., & Rob, P. (2011). Bases de datos: diseño, implementación y administración. Cengage Learning Editores. (Chap 1)
- [5] Seidl, M., Scholz, M., Huemer, C., & Kappel, G. (2015). UML@ classroom: An introduction to object-oriented modeling. Springer. (Chap 1, 4)

