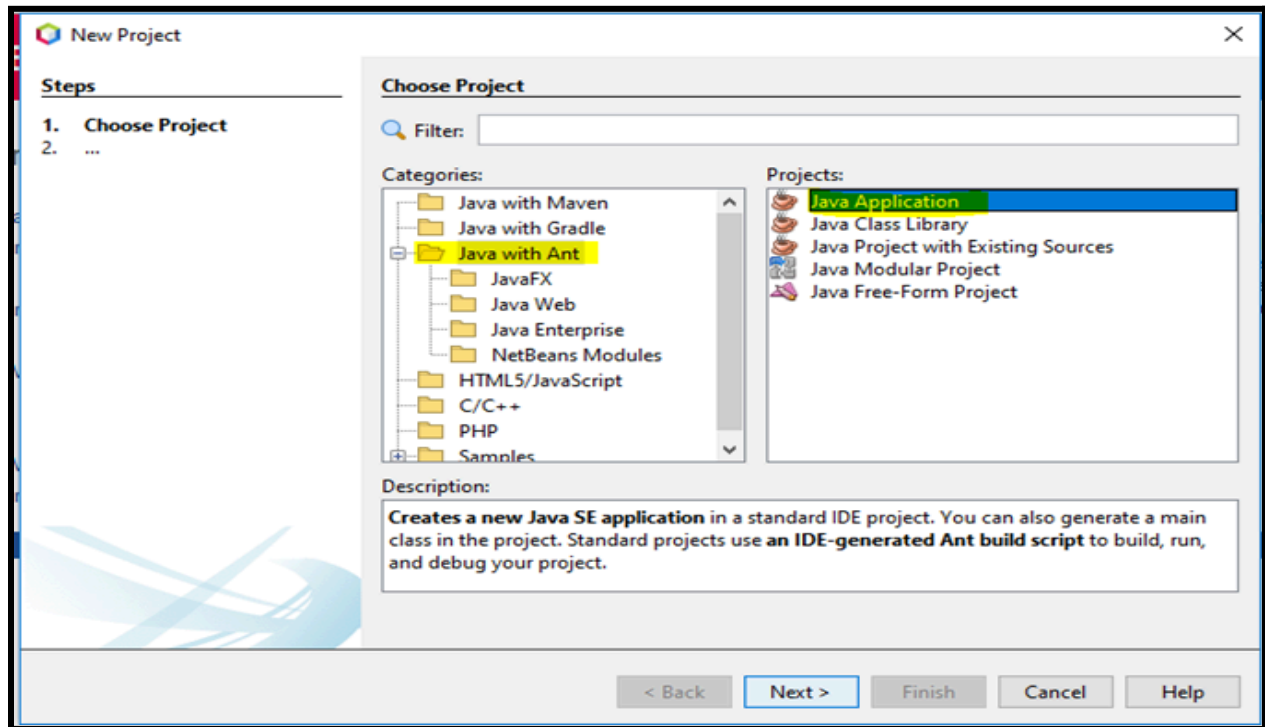


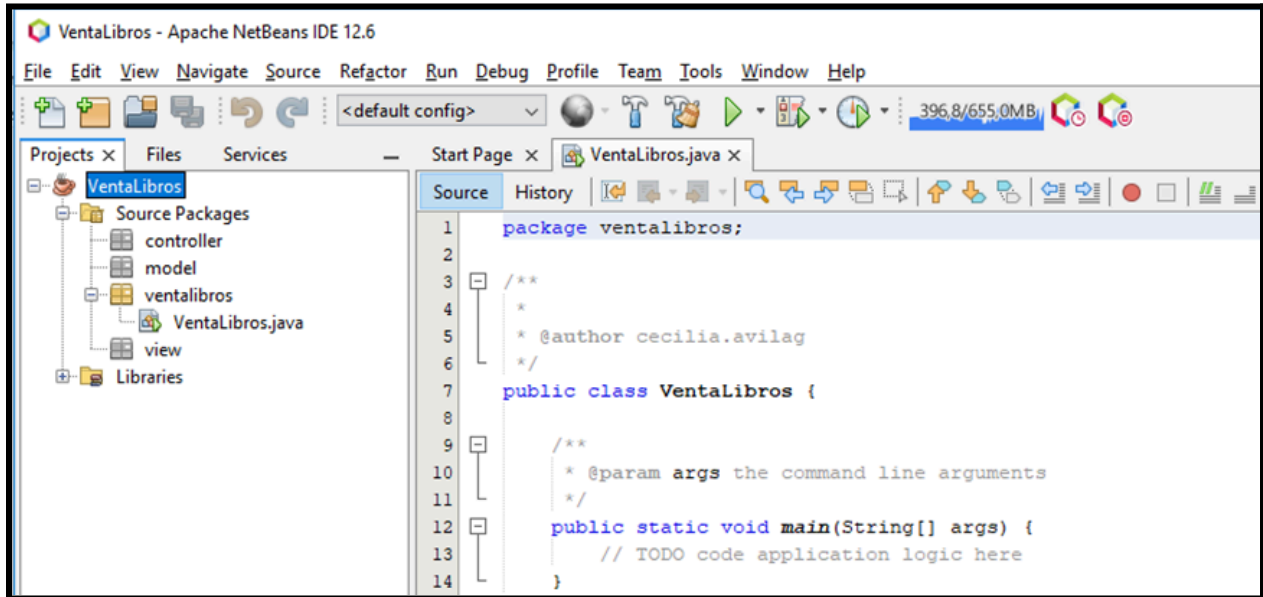
**Tutorial:** Este tutorial está creado con base en el ejemplo de “Venta Motocicletas”, por tanto se debe adaptar el nuevo proyecto al tema seleccionado previamente.

1. En MySQL crear una nueva base de datos que se llama Venta Motos.
2. En la base datos Venta Motos, crear una tabla que se llame libros con los campos: Id, Modelo, Marca, Año, Precio, Tipo, Unidad. El campo id debe quedar como auto-incremental.
3. Insertar algunos datos a la tabla Motos.
4. En el Anexo 1 de éste documento se incluye el script SQL de creación de tabla e inserción de datos.
5. Crear un nuevo proyecto en NetBeans.



6. Crear un nuevo paquete que se llame utils.
7. Crear un nuevo paquete que se llame model.
8. Crear un nuevo paquete que se llame controller.
9. Crear un nuevo paquete que se llame views.
10. Una vez creados los paquetes, el proyecto debe verse así:

Fundación Universitaria Konrad Lorenz.  
Desarrollo De Nuevas Tecnologías. Taller MVC Java SE.  
Estudiante: David Gutierrez Chaves Cod: 506222728.



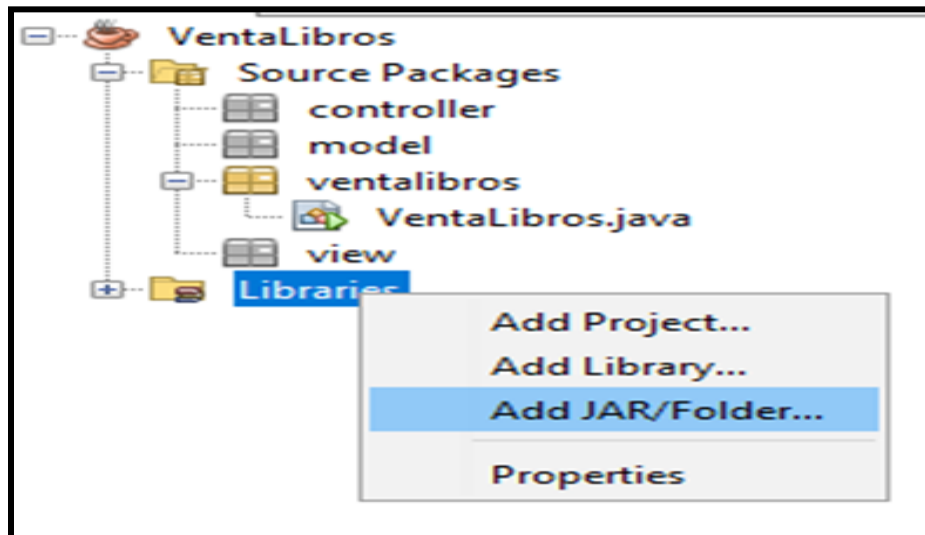
11. Descargar el driver de MySQL para Java. Ingresar a:

<https://dev.mysql.com/downloads/connector/j/>.

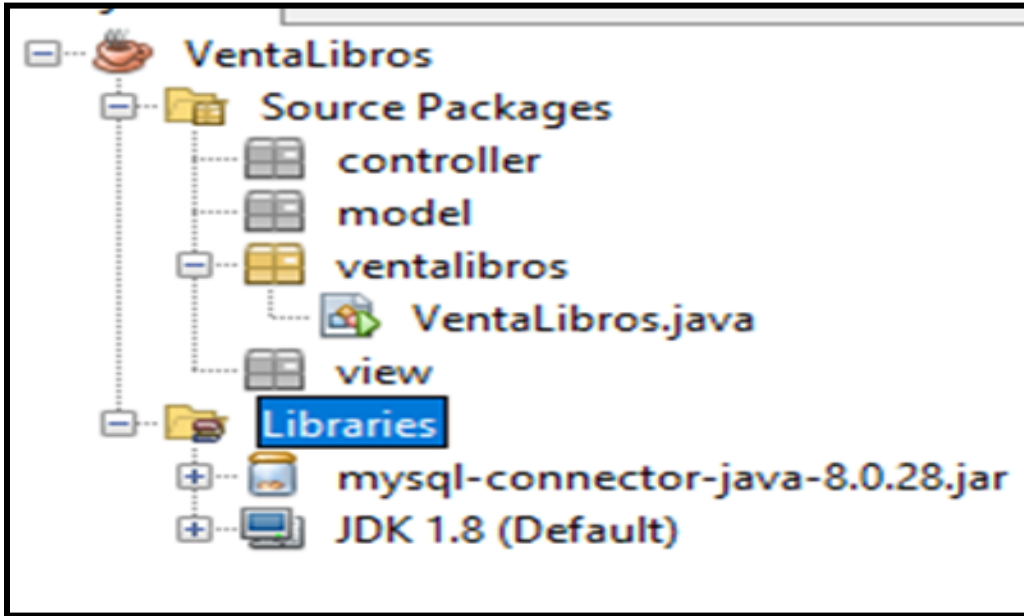
Se debe seleccionar la opción “Platform Independent”. Si es para Windows, descargar el ZIP. No es necesario iniciar sesión, usar la opción “No thanks, just start my download”.

12. Descomprimir la carpeta descargada, la cual contiene un archivo con un nombre similar a este y extensión .jar “mysql-connector-java-8.0.28.jar”.

13. En el proyecto de NetBeans agregar al proyecto el archivo jar como una nueva librería. Para ello, hacer clic derecho sobre la carpeta “Libraries” del proyecto y seleccionar la opción “Add JAR/Folder”.



14. Buscar el jar del driver de MySQL que se descargó y seleccionarlo.
15. Una vez agregado el driver, en la carpeta “Libraries” del proyecto, debe aparecer el archivo jar del driver.



16. Dentro la carpeta utils crear una clase Java que se llame Conexión y colocarle el siguiente código:

Los datos de user, password y port deben corresponder con los que se tengan en el MySQL que se tengan instalado. El nombre de la base de datos db debe corresponder a la base de datos creada previamente. Para probar la conexión, el motor o servicio de MySQL debe estar iniciado/activo.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Conexion {

    public static Connection getConnection() {
        Connection conn = null;
        //Por temas de seguridad, estos datos deberían ir en un archivo
        aparte.
        //Para efectos del ejercicio se dejan aquí.
        String host = "localhost";
        String user = "root";
        String password = "";
```

Fundación Universitaria Konrad Lorenz.  
Desarrollo De Nuevas Tecnologías. Taller MVC Java SE.  
Estudiante: David Gutierrez Chaves Cod: 506222728.

```
String port = "3306";
String db = "ventamotos";
String dbURL = "jdbc:mysql://" + host + ":" + port + "/" + db;

try {
    conn = DriverManager.getConnection(dbURL, user, password);
} catch (SQLException ex) {

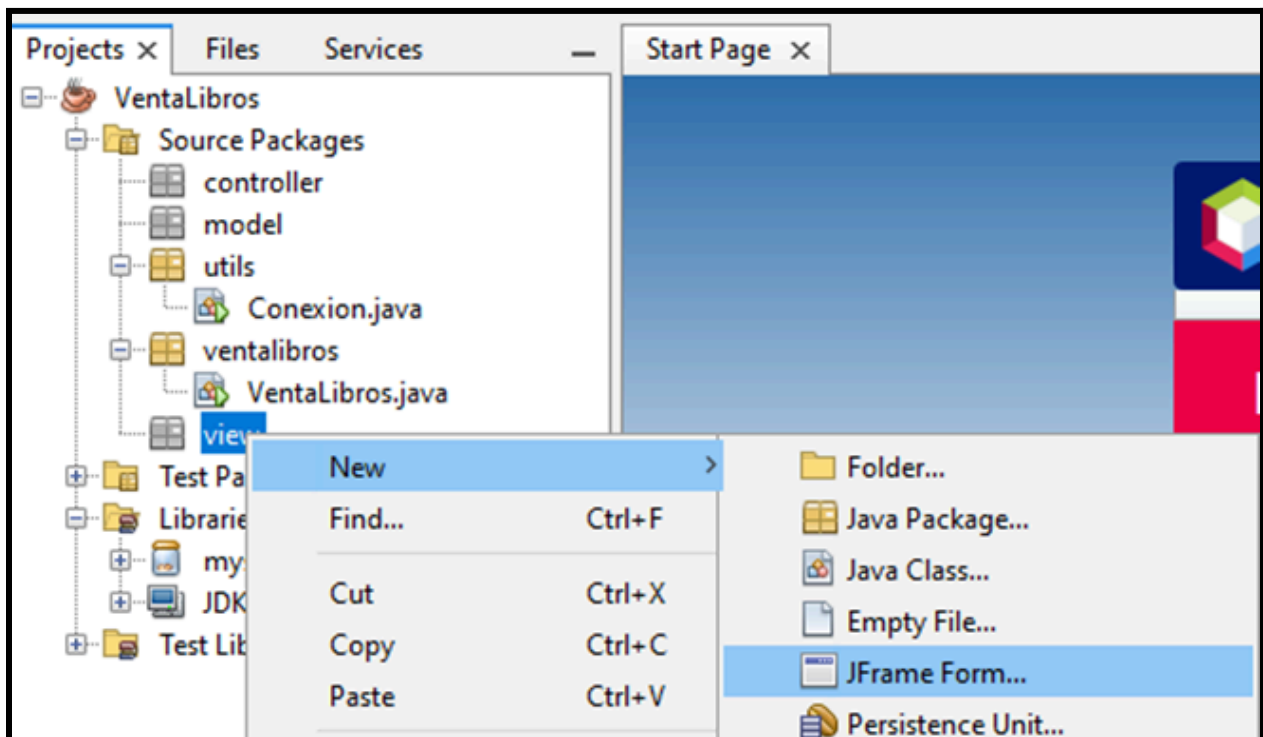
    Logger.getLogger(Conexion.class.getName()).log(Level.SEVERE, null, ex);
}

return conn;
} //cierra getConnection

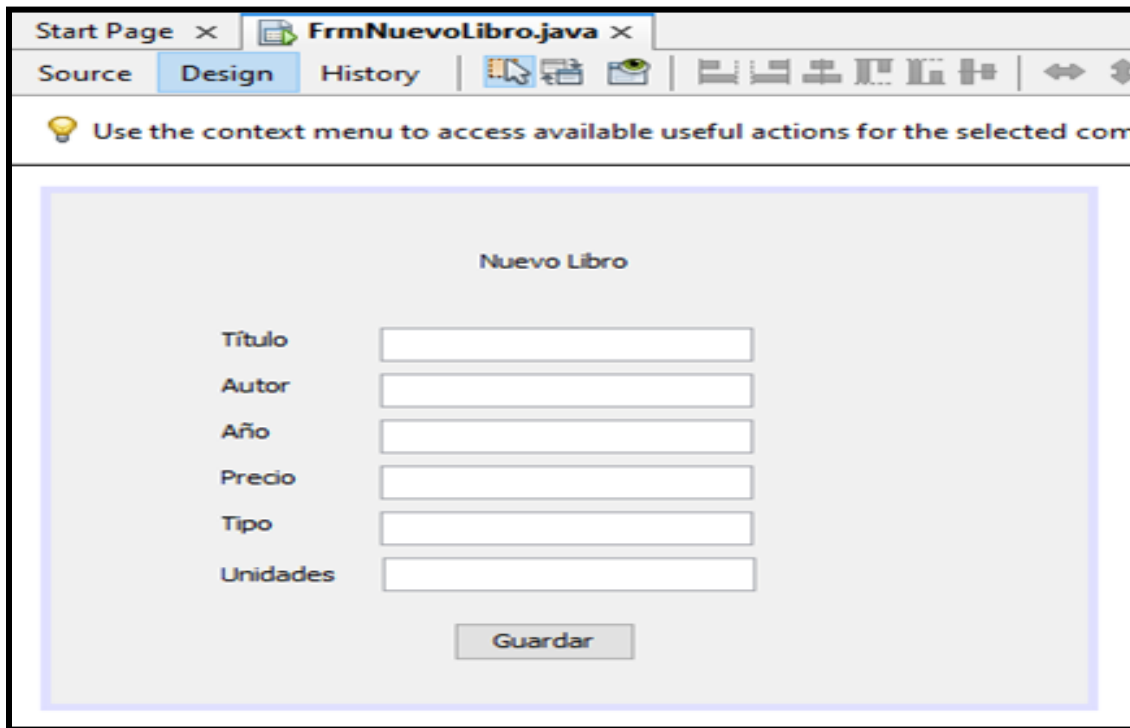
//Para probar la conexión: Si se hace clic derecho >> Run file
//sobre este archivo , no debería aparecer errores
//Si aparecen errores, verificar que estén bien los datos de
conexión
public static void main(String[] args){
    getConnection();
}

} //cierra class
```

17. En la carpeta view, crear un nuevo JFrame Form que se llame FrmNuevoLibro.



18. En ese nuevo formulario, agregar los campos que se definieron en la tabla motos. Utilizando para ello los componentes Label, Text Field y Button. Debe quedar así:



The screenshot shows a Java Swing window titled 'FrmNuevoLibro.java' in the Design view. The window contains a form titled 'Nuevo Libro'. The form has six text input fields arranged vertically, each preceded by a label: 'Título', 'Autor', 'Año', 'Precio', 'Tipo', and 'Unidades'. At the bottom center of the form is a button labeled 'Guardar'.

En el panel de Properties >> properties se pueden cambiar los textos usando la propiedad text.

19. En el paquete model agregar una clase llamada Moto con los siguientes campos.

```
private int id;  
private String modelo;  
private String marca;  
private int anio;  
private double precio;  
private String tipo;  
private int unidades;
```

20. En esa misma clase agregar el constructor sin el campo id.

21. Agregar los get y set.

22.En el paquete controller agregar una clase llamada MotoDAO (DAO significa Data Access Object)

23.En LibroDAO agregar el siguiente código:

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.swing.JOptionPane;
import model.Moto;
import utils.Conexion;

public class LibroDAO {

    private Connection conecta = null;

    public void insertLibro(Moto moto) {
        try {
            if (conecta == null) {
                conecta = Conexion.getConnection();
            }

            String sql = "INSERT INTO libros(modelo, marca, anio, precio,
            tipo, unidades) VALUES (?, ?, ?, ?, ?, ?)";

            PreparedStatement statement = conecta.prepareStatement(sql);

            statement.setString(1, moto.getModelo());

            statement.setString(2, moto.getMarca());

            statement.setInt(3, moto.getAnio());

            statement.setDouble(4, moto.getPrecio());
```



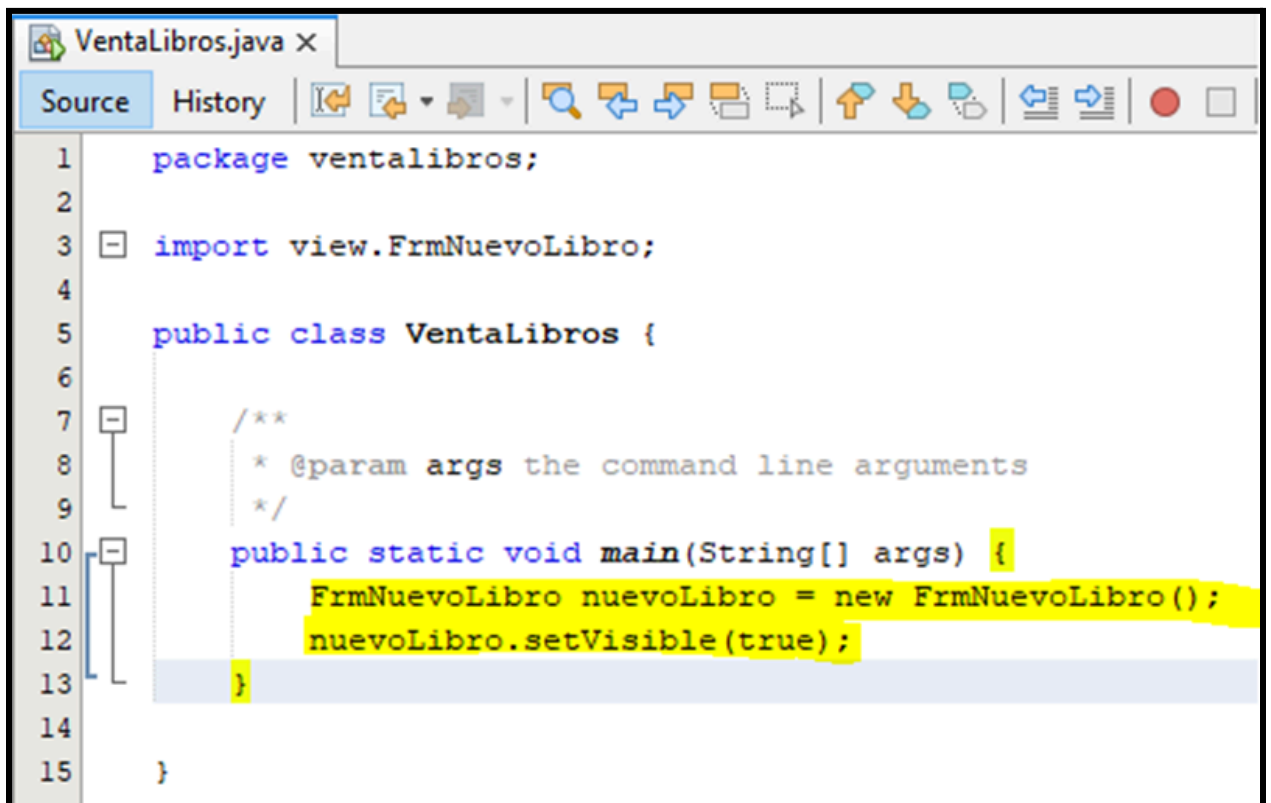
```
int unidad = Integer.parseInt(this.jTextField7.getText());

//Se crea el objeto libro
Moto moto = new Moto(modelo, marca, anio, precio, tipo, unidad);

//Se crea el objeto motoDAO
MotoDAO motoDAO = new MotoDAO();

//Se llama el método para insertar pasando el objeto moto
motoDAO.insertMoto(moto);
```

25. En el main de la clase principal agregar el siguiente código:



```
1 package ventallibros;
2
3 import view.FrmNuevoLibro;
4
5 public class VentaLibros {
6
7     /**
8      * @param args the command line arguments
9      */
10    public static void main(String[] args) {
11        FrmNuevoLibro nuevoLibro = new FrmNuevoLibro();
12        nuevoLibro.setVisible(true);
13    }
14
15 }
```

26. Ejecutar el proyecto. Debería aparecer una pequeña ventana con el formulario creado e ingresar los datos. Si todo funciona correctamente, al hacer clic en el botón guardar debería mostrarse el mensaje “¡El registro fue agregado exitosamente!”.



27. Para verificar, ir a MySQL y verificar que en la base de datos se haya agregado el registro.

#### **ANEXO 1 – SCRIPT BD**

```
-- La base de datos debe estar creada, este script contiene la creación de la  
tabla y la inserción de datos.
```

```
--
```

```
-- Base de datos: `ventamotos`
```

```
--
```

```
--
```

```
-- Estructura de tabla para la tabla `motos`
```

```
--
```

```
CREATE TABLE `motos` (
```

```
  `id` int(11) NOT NULL,
```

```
  `modelo` varchar(200) NOT NULL,
```

```
  `marca` varchar(50) NOT NULL,
```

```
  `anio` int(11) NOT NULL,
```

```
  `precio` double NOT NULL,
```

```
  `tipo` varchar(20) NOT NULL,
```

```
  `unidad` int(11) NOT NULL
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
--
```

```
-- Volcado de datos para la tabla `motos`
```

```
--
```

Fundación Universitaria Konrad Lorenz.  
Desarrollo De Nuevas Tecnologías. Taller MVC Java SE.  
Estudiante: David Gutierrez Chaves Cod: 506222728.

```
INSERT INTO `libros` (`id`, `modelo`, `marca`, `anio`, `precio`, `tipo`,  
`unidades`) VALUES  
  
(1, '450-XEC', 'KTM', 57990000, 2024, 'Enduro', 4);  
  
(2, 'Ninja zx4rr', 'Kawasaki', 36990000, 2024, 'Deportiva', 6);  
  
(3, 'Hypermotard 950 RVE', 'Ducati', 110000000, 2023, 'Hypermotard', 3);  
  
(4, 'Diavel V4 Carbon', 'Ducati', 99500000, 2023, 'Deportiva', 2);  
  
(5, 'FS-450', 'Husqvarna', 56000000, 2020, 'Supermoto', 3);  
  
(6, 'Ninja H2R', 'Kawasaki', 190000000, 2024, 'Deportiva', 6);  
  
(7, 'Multistrada V4 Rally', 'Ducati', 303788268, 2025, 'Deportiva', 4);  
  
(8, 'Six Days', 'KTM', 53990000, 2022, 'Enduro', 3);  
  
(9, 'YZ250X', 'Yamaha', 38500000, 2025, 'Motocross', 2);  
  
(10, '300-EXC', 'KTM', 44905000, 2025, 'Enduro', 3);  
  
--  
  
-- Índices para tablas volcadas  
  
--  
  
--  
  
-- Índices de la tabla `motos`  
  
--  
  
ALTER TABLE `motos`  
  
    ADD PRIMARY KEY (`id`);  
  
--
```

Fundación Universitaria Konrad Lorenz.  
Desarrollo De Nuevas Tecnologías. Taller MVC Java SE.  
Estudiante: David Gutierrez Chaves Cod: 506222728.