

PROYECTO GESTIÓN EN BASES DE DATOS

CORTE 2

INTEGRANTES:

**JUAN DAVID ARDILA ESCOBAR
ANDRÉS FELIPE PULIDO PRIETO
JUAN DIEGO GÓMEZ FONSECA**

PROFESOR:

ANDRÉS FELIPE DUQUE MONTENEGRO

BOGOTA D.C

10 DE SEPTIEMBRE DEL 2025

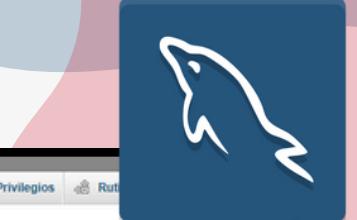
INTRODUCCIÓN

The screenshot shows a web application interface. At the top, there are two forms: 'Registro de Clientes' and 'Registro de Motos'. Both forms have input fields for Name, Document, Phone, and Address, with a 'Guardar' button. Below these forms is a table titled 'Listado de Clientes' with columns: ID, Name, Document, Phone, and Address. The table contains 18 rows of client data. To the right of the client list is another table titled 'Listado de Motos' with columns: ID, VIN, Marca, Modelo, Ciclo, Año, and Precio Base. This table also contains 18 rows of motorcycle data. At the bottom of the page, a file browser displays several PHP files: 'connection.php', 'index.php', 'cliente.php', 'process_cliente.php', and 'process.php'. The code in 'process.php' includes database connection logic and form processing logic.

php

The screenshot shows the phpMyAdmin interface. At the top, there is a search bar with the placeholder 'Que contengan la palabra:'. Below the search bar is a tree view of databases: 'information_schema', 'motoscoldb', 'motosdb', and 'mysql'. Under 'motosdb', there are tables: 'cliente', 'empleado', 'factura', 'garantia', 'orden_mantenimiento', 'proveedor', 'repuesto', 'roles', 'sucursal', 'testdrive', 'usuarios', 'vehiculo', and 'venta'. Each table has an 'Acción' column with options like 'Examinar', 'Estructura', 'Buscar', 'Insertar', 'Vaciar', and 'Eliminar'. At the bottom, it shows '13 tablas' and 'Número de filas'.

The screenshot shows the XAMPP Control Panel version 3.3.0. It lists several services: Apache (PID 19560, Port 80, 443), MySQL (PID 34048, Port 3306), FileZilla, Mercury, and Tomcat. Each service has a 'Stop' and 'Start' button, along with 'Admin', 'Config', 'Logs', 'Shell', 'Explorer', 'Services', 'Help', and 'Quit' buttons. A log window at the bottom shows MySQL errors related to port 3306 being blocked or missing dependencies.



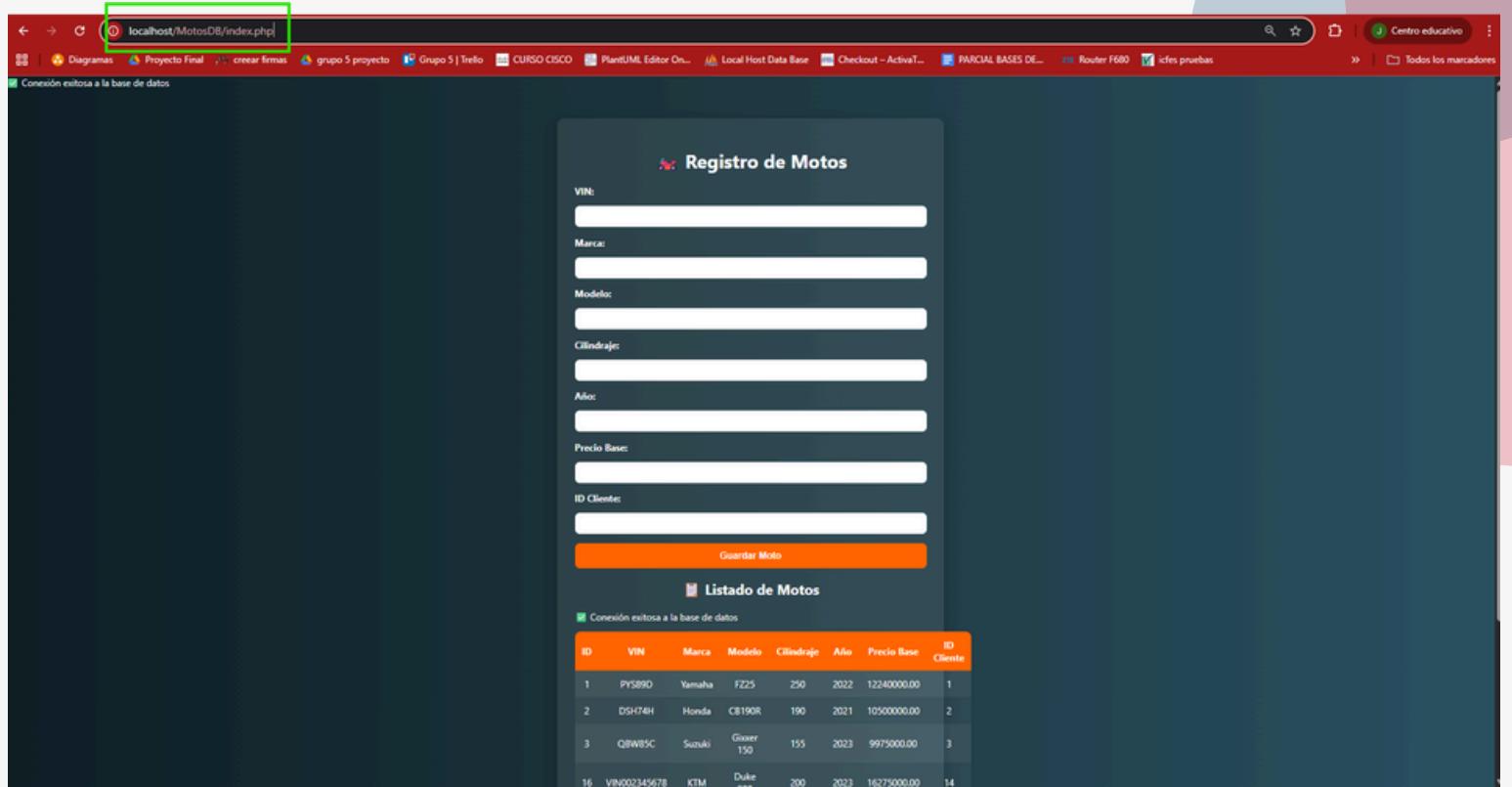
XAMPP

INTERFAZ

Esta interfaz web permite registrar y visualizar motos almacenadas en la base de datos MotosDB. La página está estructurada en dos bloques principales:

1. Formulario de Registro

- Incluye campos como VIN, marca, modelo, cilindraje, año, precio base e ID del cliente.
 - Cuando el usuario llena los datos y hace clic en "Guardar Moto", la información se envía mediante método POST al archivo process.php, que se encarga de insertar el registro en la base de datos.
 - También muestra mensajes visuales como “ Moto registrada con éxito” o “ Hubo un error”, dependiendo del resultado de la operación, lo que mejora la experiencia del usuario.



1. Tabla de Visualización

- Debajo del formulario aparece una tabla titulada “Listado de Motos”.
 - Esta tabla se genera automáticamente al incluir el archivo mostrar.php, que realiza una consulta a la base de datos y muestra los registros existentes de manera ordenada.
 - Los estilos aplicados dan una apariencia moderna con efectos de hover, colores de contraste y diseño tipo tarjeta para mayor legibilidad.



Universidad ECO

index.php

```
<?php include("conexion.php"); ?>
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Gestión de Motos</title>
    <style>
        body {
            font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
            background: linear-gradient(to right, #0f2072, #203a43, #2c5364);
            margin: 0;
            padding: 0;
            color: #fff;
        }
        h1, h2 { text-align: center; margin-bottom: 20px; }
        .container {
            max-width: 600px; margin: 50px auto;
            background: #rgba(255, 255, 255, 0.08);
            padding: 30px; border-radius: 12px;
            box-shadow: 0 8px 25px #rgba(0, 0, 0, 0.3);
        }
        form { display: flex; flex-direction: column; gap: 15px; }
        label { font-weight: bold; }
        input { padding: 10px; border-radius: 8px; border: none; outline: none; font-size: 15px; }
        input:focus { box-shadow: 0 0 8px #ff6600; border: 1px solid #ff6600; }
        button { padding: 12px; background: #ff6600; border: none; border-radius: 8px; font-size: 16px; font-weight: bold; color: white; cursor: pointer; }
        button:hover { background: #e05580; transform: scale(1.05); }
        table { width: 100%; border-collapse: collapse; margin-top: 15px; border-radius: 10px; overflow: hidden; }
        th, td { padding: 12px; text-align: center; }
        th { background: #ff6600; color: white; }
        tr:nth-child(even) { background: #rgba(255, 255, 255, 0.1); }
        tr:nth-child(odd) { background: #rgba(255, 255, 255, 0.05); }
        tr:hover { background: #rgba(255, 102, 0, 0.2); }
        .msg { text-align: center; margin: 10px; }
    </style>
</head>
<body>
    <div class="container">
        <h1>Registro de Motos</h1>
        <!-- Mensajes de éxito/error -->
        <?php if (isset($_GET['msg']) && $_GET['msg'] == 'ok') : ?>
        | <p class="msg" style="color:lightgreen;"> Moto registrada con éxito</p>
        <?php elseif (isset($_GET['msg']) && $_GET['msg'] == 'error') : ?>
        | <p class="msg" style="color:red;"> Hubo un error al registrar</p>
        <?php endif; ?>
        <form action="process.php" method="POST">
    </div>
</body>
```

INTERFAZ

Esta página web permite registrar nuevos clientes y visualizar los registros existentes en la base de datos, facilitando la administración de clientes vinculados a las motos registradas.

La interfaz se divide en dos secciones principales:

1. Formulario de Registro de Clientes

- Solicita datos clave como nombre, documento, teléfono, correo y dirección.
- Los campos están validados como requeridos para evitar registros incompletos.
- Al presionar “Guardar Cliente”, la información se envía mediante método POST al archivo process_cliente.php, que se encarga de guardar los datos en la base.
- El sistema muestra mensajes visuales de éxito o error, informando al usuario si el registro fue procesado correctamente, lo que mejora la experiencia de uso.

The screenshot shows a web browser window with a green border around the title bar. The title bar displays 'localhost/MotoDB/indexCliente.php'. The main content area shows a registration form titled 'Registro de Clientes' with fields for Nombre, Documento, Teléfono, Correo, and Dirección. Below the form is a button labeled 'Guardar Cliente'. To the right of the form is a table titled 'Listado de Clientes' showing a list of registered clients with columns: ID, Nombre, Documento, Teléfono, Correo, and Dirección. The table contains 16 rows of client data.

2. Listado Dinámico de Clientes

- Despues del formulario, se muestra una tabla titulada “Listado de Clientes”.
- Esta tabla se carga automáticamente desde el archivo mostrarClientes.php, que consulta la base de datos y muestra los registros de manera organizada.
- Se aplican estilos visuales para hacer la tabla más legible y profesional, con efectos hover y alternancia de colores entre filas.

Diseño y experiencia de usuario

- Se utiliza un estilo moderno con degradados y sombras, para simular una interfaz profesional tipo sistema administrativo.
- Los campos tienen efecto de foco luminoso, lo que guía la interacción del usuario.
- El botón tiene animación al pasar el cursor, mejorando la percepción visual del sistema.

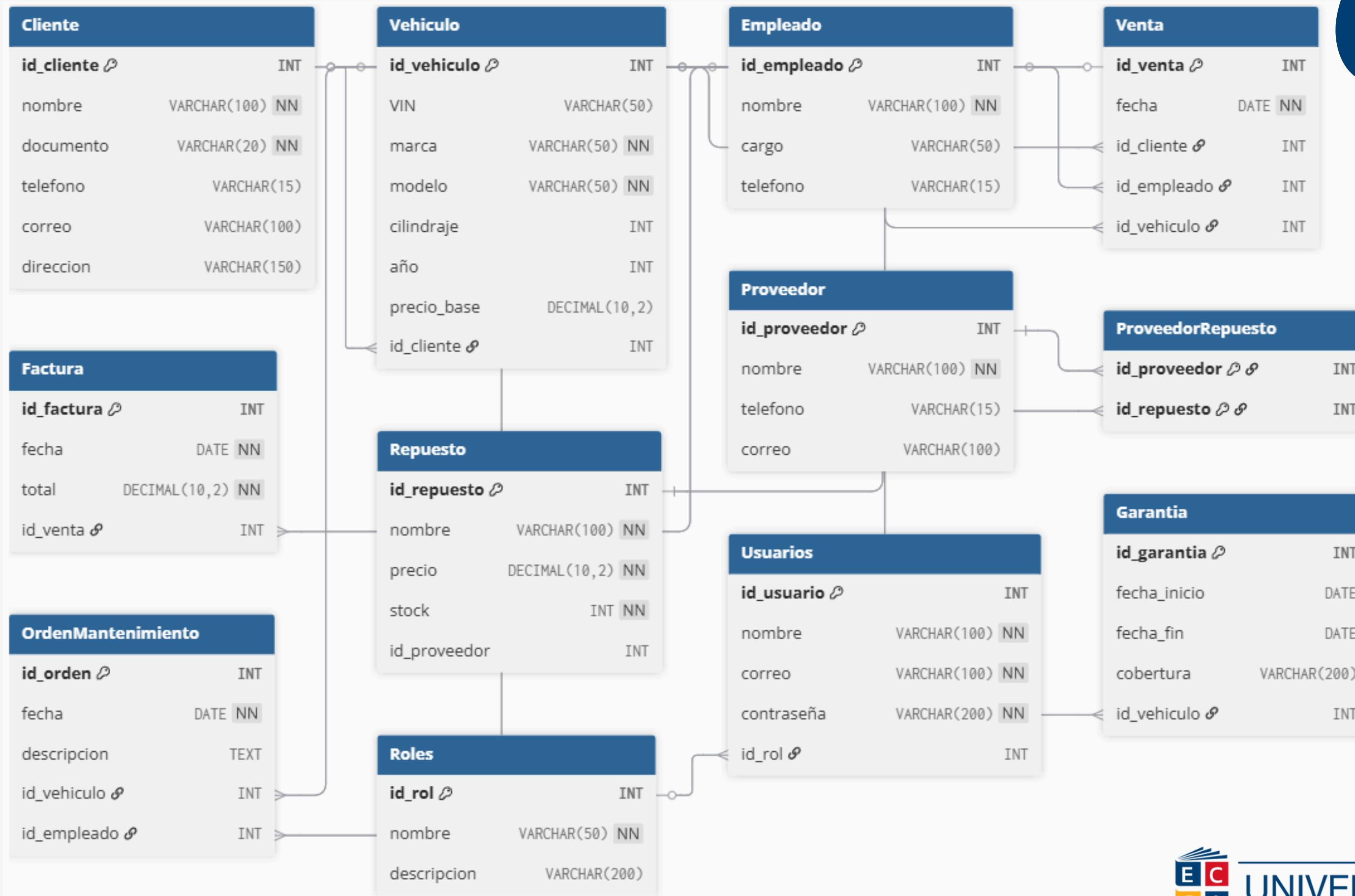
The screenshot shows a code editor with multiple tabs open. The active tab is 'indexCliente.php', which contains PHP code. To the right of the code is a large block of CSS styles. The CSS includes various selectors for HTML elements like h1, h2, form, label, input, and button, applying styles such as gradients, borders, and font weights. The code editor interface shows a sidebar with file navigation and a bottom status bar with file information.

SCRIPTS DE LAS CONEXION DB - INTERFAZ

Como equipo, hemos publicado los scripts de nuestra base de datos en un repositorio de GitHub, esto con el fin de mantener evidencia del trabajo realizado y asegurar una gestión organizada y accesible de los avances del proyecto.



DIAGRAMA ENTIDAD RELACIÓN



TABLAS DE MOTOS_DB

EMPLEADO

```
3 • CREATE TABLE `empleado` (
4     `id_empleado` int(11) NOT NULL,
5     `nombre` varchar(100) DEFAULT NULL,
6     `rol` varchar(50) DEFAULT NULL,
7     `correo` varchar(100) DEFAULT NULL,
8     `salario` decimal(10,2) DEFAULT NULL,
9     `id_sucursal` int(11) DEFAULT NULL
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
11
12
```

CLIENTE

```
1
2
3 • CREATE TABLE `cliente` (
4     `id_cliente` int(11) NOT NULL,
5     `nombre` varchar(100) NOT NULL,
6     `documento` varchar(50) DEFAULT NULL,
7     `telefono` varchar(20) DEFAULT NULL,
8     `correo` varchar(100) DEFAULT NULL,
9     `direccion` varchar(150) DEFAULT NULL
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
11
12
13
14
15
16
```

TABLAS DE MOTOS_DB

FACTURA

```
3 • ⊖ CREATE TABLE `factura` (
4     `id_factura` int(11) NOT NULL,
5     `monto_total` decimal(10,2) DEFAULT NULL,
6     `fecha_emision` date DEFAULT NULL,
7     `id_venta` int(11) DEFAULT NULL
8 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
9
```

VEHICULO

```
5 • ⊖ CREATE TABLE `vehiculo` (
6     `id_vehiculo` int(11) NOT NULL,
7     `VIN` varchar(50) DEFAULT NULL,
8     `marca` varchar(50) DEFAULT NULL,
9     `modelo` varchar(50) DEFAULT NULL,
10    `cilindraje` int(11) DEFAULT NULL,
11    `anio` int(11) DEFAULT NULL,
12    `precio_base` decimal(10,2) DEFAULT NULL,
13    `id_cliente` int(11) DEFAULT NULL
14 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
15
```

TABLAS DE MOTOS_DB

GARANTIA

```
3 • Ⓜ CREATE TABLE `garantia` (
4     `id_garantia` int(11) NOT NULL,
5     `fecha_inicio` date DEFAULT NULL,
6     `fecha_fin` date DEFAULT NULL,
7     `cobertura` varchar(100) DEFAULT NULL,
8     `id_vehiculo` int(11) DEFAULT NULL
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
10
```

ORDEN_MANTENIMIENTO

```
5 • Ⓜ CREATE TABLE `orden_mantenimiento` (
6     `id_orden` int(11) NOT NULL,
7     `fecha` date DEFAULT NULL,
8     `tipo_servicio` varchar(100) DEFAULT NULL,
9     `estado` varchar(50) DEFAULT NULL,
10    `id_vehiculo` int(11) DEFAULT NULL,
11    `id_empleado` int(11) DEFAULT NULL
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

TABLAS DE MOTOS_DB

PROVEEDOR

```
● CREATE TABLE `proveedor` (
    `id_proveedor` int(11) NOT NULL,
    `nombre` varchar(100) DEFAULT NULL,
    `contacto` varchar(100) DEFAULT NULL,
    `direccion` varchar(150) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

REPUESTO

```
● CREATE TABLE `repuesto` (
    `id_repuesto` int(11) NOT NULL,
    `nombre` varchar(100) DEFAULT NULL,
    `categoria` varchar(50) DEFAULT NULL,
    `precio` decimal(10,2) DEFAULT NULL,
    `stock` int(11) DEFAULT NULL,
    `id_proveedor` int(11) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

TABLAS DE MOTOS_DB

SUCURSAL

```
5 • ⓧ CREATE TABLE `sucursal` (
6   `id_sucursal` int(11) NOT NULL,
7   `nombre` varchar(100) DEFAULT NULL,
8   `direccion` varchar(150) DEFAULT NULL,
9   `ciudad` varchar(100) DEFAULT NULL
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
11
```

VENTA

```
5 • ⓧ CREATE TABLE `venta` (
6   `id_venta` int(11) NOT NULL,
7   `fecha` date DEFAULT NULL,
8   `forma_pago` varchar(50) DEFAULT NULL,
9   `estado` varchar(50) DEFAULT NULL,
10  `id_cliente` int(11) DEFAULT NULL,
11  `id_vehiculo` int(11) DEFAULT NULL
12 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
13
```

TABLAS DE MOTOS_DB

ROLES

```
5 • CREATE TABLE `roles` (
6   `rol_id` int(11) NOT NULL,
7   `nombre` varchar(50) NOT NULL,
8   `descripcion` text DEFAULT NULL
9 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

USUARIOS

```
5 • CREATE TABLE `usuarios` (
6   `usuario_id` int(11) NOT NULL,
7   `nombre` varchar(100) DEFAULT NULL,
8   `correo` varchar(100) DEFAULT NULL,
9   `rol_id` int(11) DEFAULT NULL,
10  `contraseña` text DEFAULT NULL
11 ) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci;
```

DDL – DATA DEFINITION LANGUAGE (LENGUAJE DE DEFINICIÓN DE DATOS)

TIPOS DE EJEMPLOS:

```
1  -- ====== --  
2  -- ===== Definición de Datos ===== --  
3  
4 • CREATE DATABASE MotosDB;           -- Crear una base de datos  
5 • USE MotosDB;                     -- Seleccionarla  
6 ✘ CREATE TABLE Cliente (...);       -- Crear tabla  
7 • ALTER TABLE Cliente ADD COLUMN edad INT;  -- Modificar una tabla  
8 • DROP TABLE Vehiculo;             -- Eliminar una tabla  
9  
10
```

EJEMPLO PRACTICO:

```
1 • use MotosDB;  
2   -- Crear una nueva tabla para registrar test drives  
3 • ⌂ CREATE TABLE TestDrive (  
4     id_testdrive INT AUTO_INCREMENT PRIMARY KEY,  
5     fecha DATE,  
6     resultado VARCHAR(50),  
7     id_cliente INT,  
8     id_vehiculo INT,  
9     FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente),  
10    FOREIGN KEY (id_vehiculo) REFERENCES Vehiculo(id_vehiculo)  
11 );  
12  
13  
14  
15
```

RESULTADO:

```
1 • use motosDB;  
2  
3 • SHOW COLUMNS FROM testdrive;  
4  
5  
6
```

Result Grid | Filter Rows: Export: Wrap Cell Content: □

Field	Type	Null	Key	Default	Extra
id_testdrive	int(11)	NO	PRI	NULL	auto_increment
fecha	date	YES		NULL	
resultado	varchar(50)	YES		NULL	
id_cliente	int(11)	YES	MUL	NULL	
id_vehiculo	int(11)	YES	MUL	NULL	

Output

Action Output	Time	Action	Message
1	22:05:53	use MotosDB	0 row(s) affected
2	22:05:53	CREATE TABLE TestDrive (id_testdrive INT AUTO_INCREMENT PRIMARY KEY, fecha DATE, resultado VARCHAR(50), id_cliente INT, id_vehiculo INT, FOREIGN KEY (id_cliente) REFERENCES Cliente(id_cliente), FOREIGN KEY (id_vehiculo) REFERENCES Vehiculo(id_vehiculo));	0 row(s) affected

DML – DATA MANIPULATION LANGUAGE (LENGUAJE DE MANIPULACIÓN DE DATOS)

TIPOS DE EJEMPLOS:

```
3 -- =====
4 -- ===== Manipulación de Datos =====
5
6 • INSERT INTO Cliente (nombre, documento, telefono, correo, direccion)
7   VALUES ('Juan Pérez', '12345678', 'juanperez@gmail.com', 'Cra 10 #12-34');
8
9 • UPDATE Cliente SET telefono = '3009876543' WHERE id_cliente = 1;
10
11 • DELETE FROM Cliente WHERE id_cliente = 3;
12
```

RESULTADO:

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
id_cliente	nombre	documento	telefono	correo	direccion
1	Juan Pérez	12345678	3001112233	juanperez@gmail.com	Cra 10 #12-34
2	Maria Gómez	98765432	3109876543	mariagomez@hotmail.com	Calle 5 #45-22
3	Carlos Ramírez	45678912	3204567890	carlosramirez@yahoo.com	Calle 32 B sur #29-21
14	Andrés Torres	11223344	3019876543	andrestorres@gmail.com	Carrera 8 #14-25
15	Laura Méndez	22334455	3201122334	lauramendez@hotmail.com	Avenida 9 #23-41
16	Felipe Gómez	33445566	3112233445	felipegomez@yahoo.com	Calle 12 #9-18
17	Sofía Ramírez	44556677	3023344556	sofiaramirez@gmail.com	Transversal 5 #67-20
18	Camilo Herrera	55667788	3004455667	camilo.herrera@outlook.com	Diagonal 21 #45-60
21	Juliana Pérez	66778899	3125566778	julianaperez@gmail.com	Calle 25 #45-30

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
id_cliente	nombre	documento	telefono	correo	direccion
1	Juan Pérez	12345678	3001112233	juanperez@gmail.com	Cra 10 #12-34
2	Maria Gómez	98765432	3109876543	mariagomez@hotmail.com	Calle 5 #45-22
3	Carlos Ramírez	45678912	3204567890	carlosramirez@yahoo.com	Calle 32 B sur #29-21
14	Andrés Torres	11223344	3019876543	andrestorres@gmail.com	Carrera 8 #14-25
15	Laura Méndez	22334455	3201122334	lauramendez@hotmail.com	Avenida 9 #23-41
16	Felipe Gómez	33445566	3112233445	felipegomez@yahoo.com	Calle 12 #9-18
17	Sofía Ramírez	44556677	3023344556	sofiaramirez@gmail.com	Transversal 5 #67-20
18	Camilo Herrera	55667788	3004455667	camilo.herrera@outlook.com	Diagonal 21 #45-60
*	NULL	NULL	NULL	NULL	NULL

ANTES:

Result Grid Filter Rows: Edit: Export/Import: Wrap Cell Content:					
id_cliente	nombre	documento	telefono	correo	direccion
1	Juan Pérez	12345678	3001234567	juanperez@gmail.com	Cra 10 #12-34
2	Maria Gómez	98765432	3109876543	mariagomez@hotmail.com	Calle 5 #45-22
3	Carlos Ramírez	45678912	3204567890	carlosramirez@yahoo.com	Calle 32 B sur #29-21
14	Andrés Torres	11223344	3019876543	andrestorres@gmail.com	Carrera 8 #14-25
15	Laura Méndez	22334455	3201122334	lauramendez@hotmail.com	Avenida 9 #23-41
16	Felipe Gómez	33445566	3112233445	felipegomez@yahoo.com	Calle 12 #9-18
17	Sofía Ramírez	44556677	3023344556	sofiaramirez@gmail.com	Transversal 5 #67-20
18	Camilo Herrera	55667788	3004455667	camilo.herrera@outlook.com	Diagonal 21 #45-60

EJEMPLO PRACTICO:

```
1 -- Insertar un nuevo cliente
2 • INSERT INTO Cliente (nombre, documento, telefono, correo, direccion)
3   VALUES ('Juliana Pérez', '66778899', '3125566778', 'julianaperez@gmail.com', 'Calle 25 #45-30');
4
5 -- Actualizar el teléfono de un cliente
6 • UPDATE Cliente
7   SET telefono = '3001112233'
8   WHERE id_cliente = 1 ;
9
10 -- Eliminar un cliente
11 • DELETE FROM Cliente
12   WHERE id_cliente = 21;
13
```

DQL – DATA QUERY LANGUAGE (LENGUAJE DE CONSULTA DE DATOS)

TIPOS DE EJEMPLOS:

```
1
2  -- =====
3  -- ===== Consulta de Datos===== --
4
5 • SELECT * FROM Cliente;          -- Mostrar todo
6 • SELECT nombre, correo FROM Cliente; -- Mostrar columnas específicas
7 • SELECT c.nombre, v.marca, v.modelo
8   FROM Cliente c
9   JOIN Vehiculo v ON c.id_cliente = v.id_cliente; -- Consultar datos relacionados
10
```

EJEMPLOS PRACTICOS:

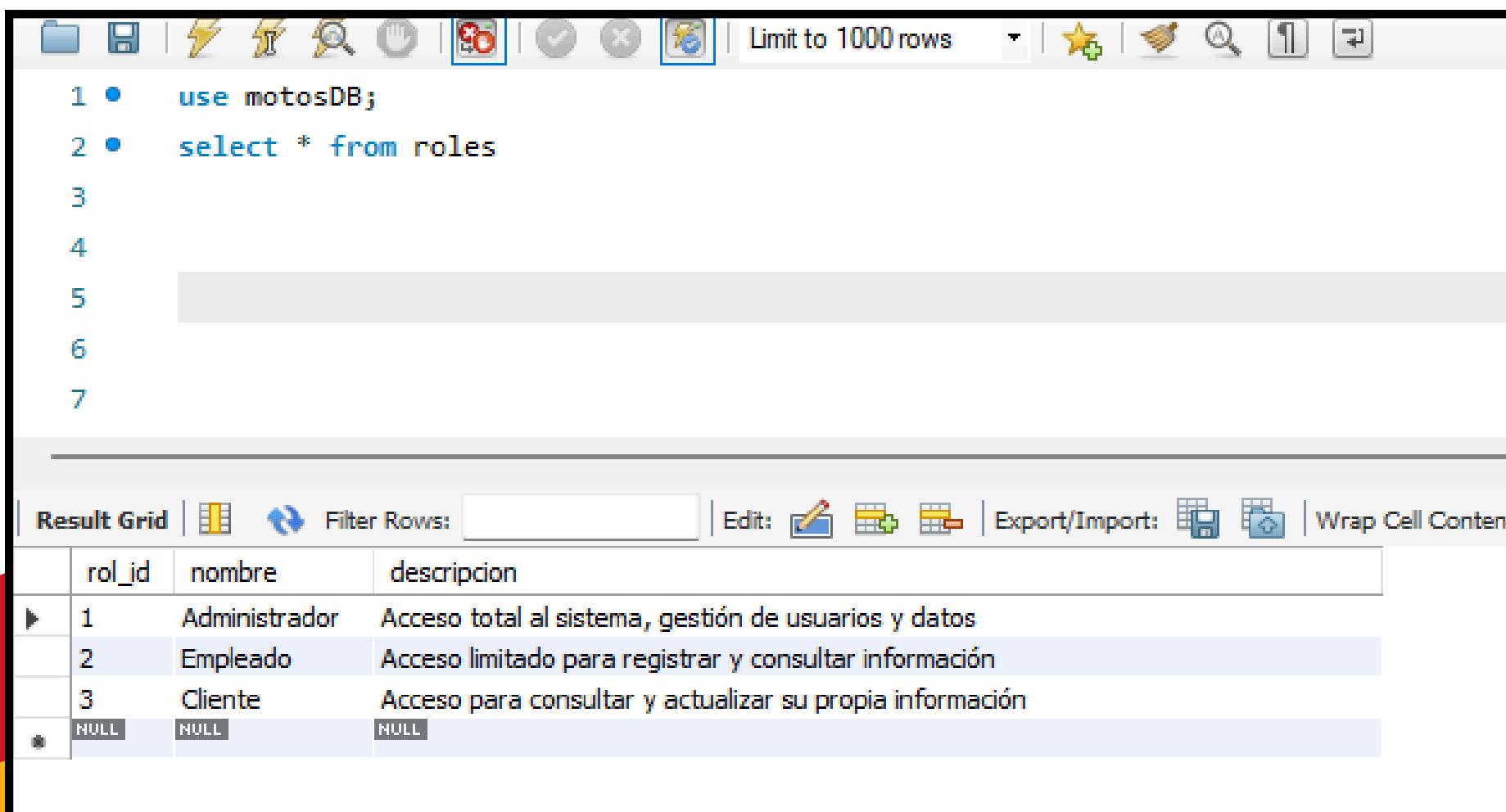
```
1  -- Mostrar todos los vehículos con su respectivo dueño
2 • SELECT v.marca, v.modelo, c.nombre AS dueño
3   FROM Vehiculo v
4   JOIN Cliente c ON v.id_cliente = c.id_cliente;
5
6
7
```

Result Grid			
	marca	modelo	dueño
▶	Yamaha	FZ25	Juan Pérez
	Honda	CB190R	Maria Gómez
	Suzuki	Gixxer 150	Carlos Ramírez
	KTM	Duke 200	Andrés Torres
	TVS	Apache 160 4V	Laura Méndez
	Yamaha	MT-15	Felipe Gómez
	Bajaj	Dominar 400	Sofia Ramírez
	Hero	Hunk 160R	Camilo Herrera

```
1  -- Mostrar clientes que compraron motos con cilindraje mayor a 200
2 • SELECT c.nombre, v.marca, v.cilindraje
3   FROM Cliente c
4   JOIN Vehiculo v ON c.id_cliente = v.id_cliente
5   WHERE v.cilindraje > 200;
6
7
```

Result Grid			
	nombre	marca	cilindraje
▶	Juan Pérez	Yamaha	250
	Sofia Ramírez	Bajaj	400

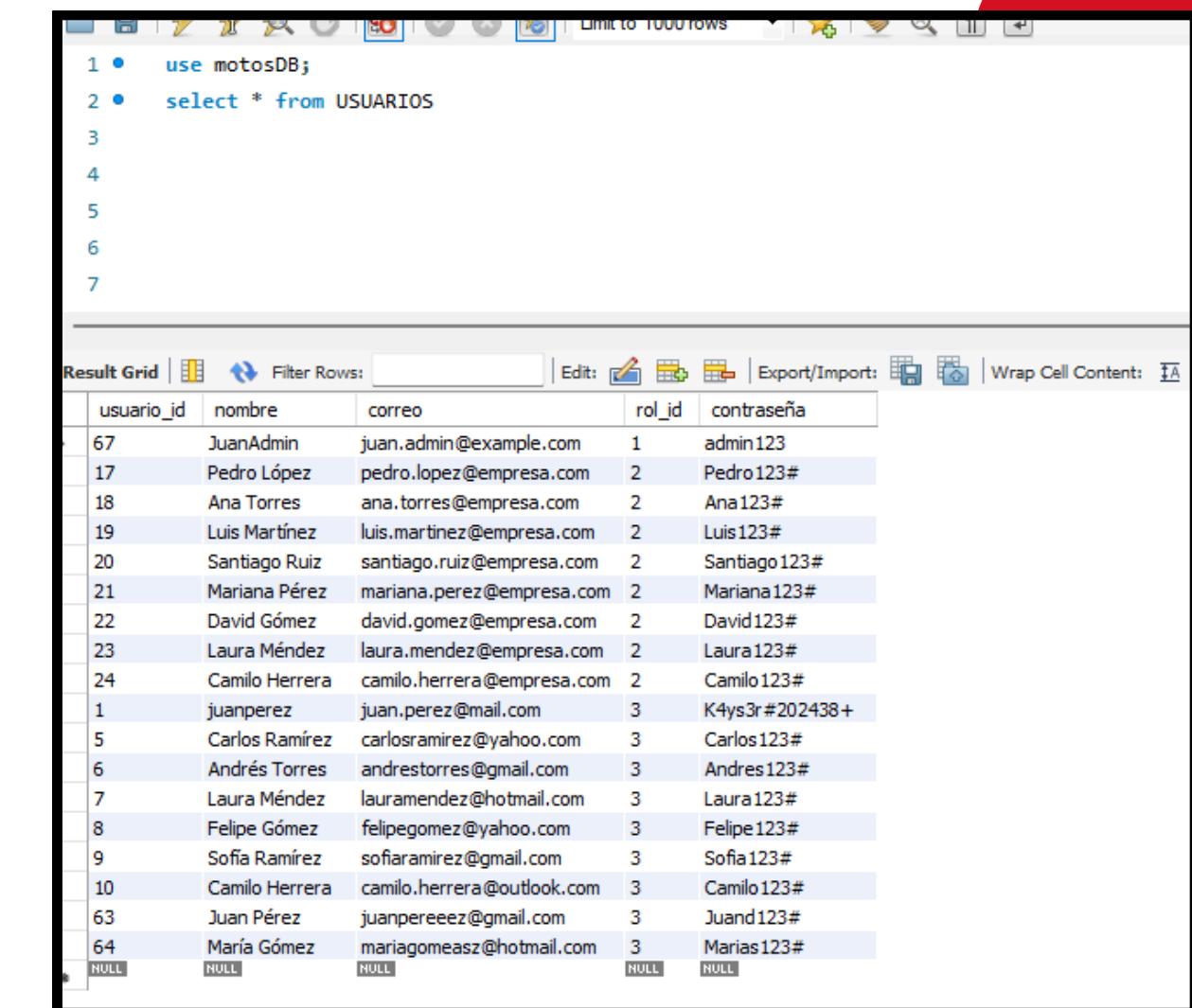
DCL – DATA CONTROL LANGUAGE (LENGUAJE DE CONTROL DE DATOS)



```
1 • use motosDB;
2 • select * from roles
```

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a text area containing the SQL code. At the bottom, there's a result grid table with columns: rol_id, nombre, and descripción.

rol_id	nombre	descripción
1	Administrador	Acceso total al sistema, gestión de usuarios y datos
2	Empleado	Acceso limitado para registrar y consultar información
3	Cliente	Acceso para consultar y actualizar su propia información



```
1 • use motosDB;
2 • select * from USUARIOS
```

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a text area containing the SQL code. At the bottom, there's a result grid table with columns: usuario_id, nombre, correo, rol_id, and contraseña.

usuario_id	nombre	correo	rol_id	contraseña
67	JuanAdmin	juan.admin@example.com	1	admin123
17	Pedro López	pedro.lopez@empresa.com	2	Pedro123#
18	Ana Torres	ana.torres@empresa.com	2	Ana123#
19	Luis Martínez	luis.martinez@empresa.com	2	Luis123#
20	Santiago Ruiz	santiago.ruiz@empresa.com	2	Santiago123#
21	Mariana Pérez	mariana.perez@empresa.com	2	Mariana123#
22	David Gómez	david.gomez@empresa.com	2	David123#
23	Laura Méndez	laura.mendez@empresa.com	2	Laura123#
24	Camilo Herrera	camilo.herrera@empresa.com	2	Camilo123#
1	juanperez	juan.perez@mail.com	3	K4ys3r#202438+
5	Carlos Ramírez	carlosramirez@yahoo.com	3	Carlos123#
6	Andrés Torres	andrestorres@gmail.com	3	Andres123#
7	Laura Méndez	lauramendez@hotmail.com	3	Laura123#
8	Felipe Gómez	felipegomez@yahoo.com	3	Felipe123#
9	Sofía Ramírez	sofiaramirez@gmail.com	3	Sofia123#
10	Camilo Herrera	camilo.herrera@outlook.com	3	Camilo123#
63	Juan Pérez	juanpereeez@gmail.com	3	Juand123#
64	María Gómez	mariagomeasz@hotmail.com	3	Marias123#

TCL – TRANSACTION CONTROL LANGUAGE (LENGUAJE DE CONTROL DE TRANSACCIONES)

TIPOS DE EJEMPLOS:

```
1
2      -- =====
3      -- ===== Control de transacciones =====
4
5 • START TRANSACTION;
6 • UPDATE Vehiculo SET precio_base = precio_base * 1.05;
7 • COMMIT;    -- Confirma los cambios
8 • ROLLBACK; -- Revierte los cambios si hubo error
9
```

EJEMPLO PRACTICO:

```
1   -- Iniciar una transacción
2 • START TRANSACTION;
3
4   -- Intentar actualizar varios precios de vehículos
5 • UPDATE Vehiculo SET precio_base = precio_base * 1.05 WHERE anio = 2023;
6 • UPDATE Vehiculo SET precio_base = precio_base * 1.02 WHERE anio = 2022;
7
8   -- Confirmar los cambios
9 • COMMIT;
10
11  -- Si quisieras cancelar en lugar de confirmar:
12  -- ROLLBACK;
13
```

ANTES:

use motosDB;

Select * from vehiculo;

	id_vehiculo	VIN	marca	modelo	cilindraje	anio	precio_base	id_cliente
1	1	PYS89D	Yamaha	FZ25	250	2022	12000000.00	1
2	2	DSH74H	Honda	CB190R	190	2021	10500000.00	2
3	3	QBW85C	Suzuki	Gixxer 150	155	2023	9500000.00	3
16	16	VIN002345678	KTM	Duke 200	200	2023	15500000.00	14
17	17	VIN003456789	TVS	Apache 160 4V	160	2022	11500000.00	15
18	18	VIN005678901	Yamaha	MT-15	155	2024	17500000.00	16
19	19	VIN006789012	Bajaj	Dominar 400	400	2023	18500000.00	17
20	20	VIN007890123	Hero	Hunk 160R	160	2021	9500000.00	18
*		HULL	HULL	HULL	HULL	HULL	HULL	HULL

RESULTADO:

use motosDB;

Select * from vehiculo;

	id_vehiculo	VIN	marca	modelo	cilindraje	anio	precio_base	id_cliente
1	1	PYS89D	Yamaha	FZ25	250	2022	12240000.00	1
2	2	DSH74H	Honda	CB190R	190	2021	10500000.00	2
3	3	QBW85C	Suzuki	Gixxer 150	155	2023	9975000.00	3
16	16	VIN002345678	KTM	Duke 200	200	2023	16275000.00	14
17	17	VIN003456789	TVS	Apache 160 4V	160	2022	11730000.00	15
18	18	VIN005678901	Yamaha	MT-15	155	2024	17500000.00	16
19	19	VIN006789012	Bajaj	Dominar 400	400	2023	19425000.00	17
20	20	VIN007890123	Hero	Hunk 160R	160	2021	9500000.00	18
*		HULL	HULL	HULL	HULL	HULL	HULL	HULL

EN RESUMEN...

Tipo	Nombre	Qué hace	Ejemplo
DDL	<i>Definición de datos</i>	<i>Crea y modifica tablas</i>	<i>CREATE TABLE, ALTER, DROP</i>
DML	<i>Manipulación de datos</i>	<i>Inserta, actualiza o borra registros</i>	<i>INSERT, UPDATE, DELETE</i>
DQL	<i>Consulta de datos</i>	<i>Lee o muestra datos</i>	<i>SELECT</i>
DCL	<i>Control de datos</i>	<i>Permisos y roles</i>	<i>GRANT, REVOKE</i>
TCL	<i>Control de transacciones</i>	<i>Confirma o revierte cambios</i>	<i>COMMIT, ROLLBACK</i>

**GRACIAS POR SU
ATENCION:**