

Universidad de Nariño.

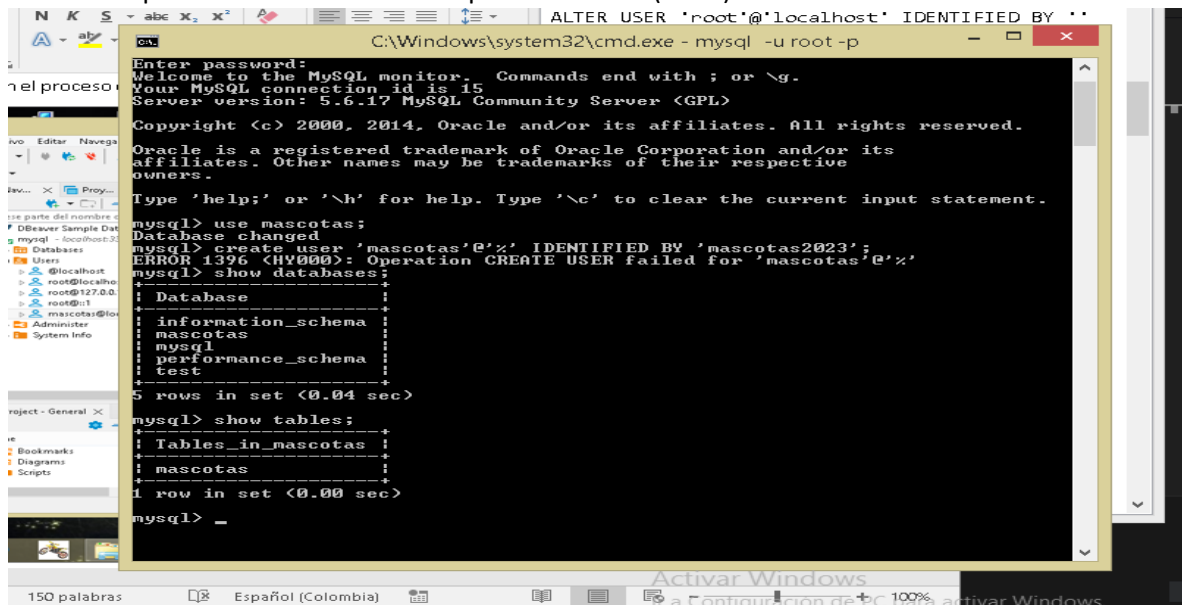
Ingeniería de Sistemas.

Diplomado de actualización en nuevas tecnologías para el desarrollo de Software.

Taller Unidad 2 Backend

Marlon Tenganan

1. Crear una base de datos MYSQL que permita llevar el registro de mascotas (perros y gatos), así como también el proceso de solicitud de adopción de estas. (1 Pto).



```
ALTER USER 'root'@'localhost' IDENTIFIED BY ''
C:\Windows\system32\cmd.exe - mysql -u root -p

Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 15
Server version: 5.6.17 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

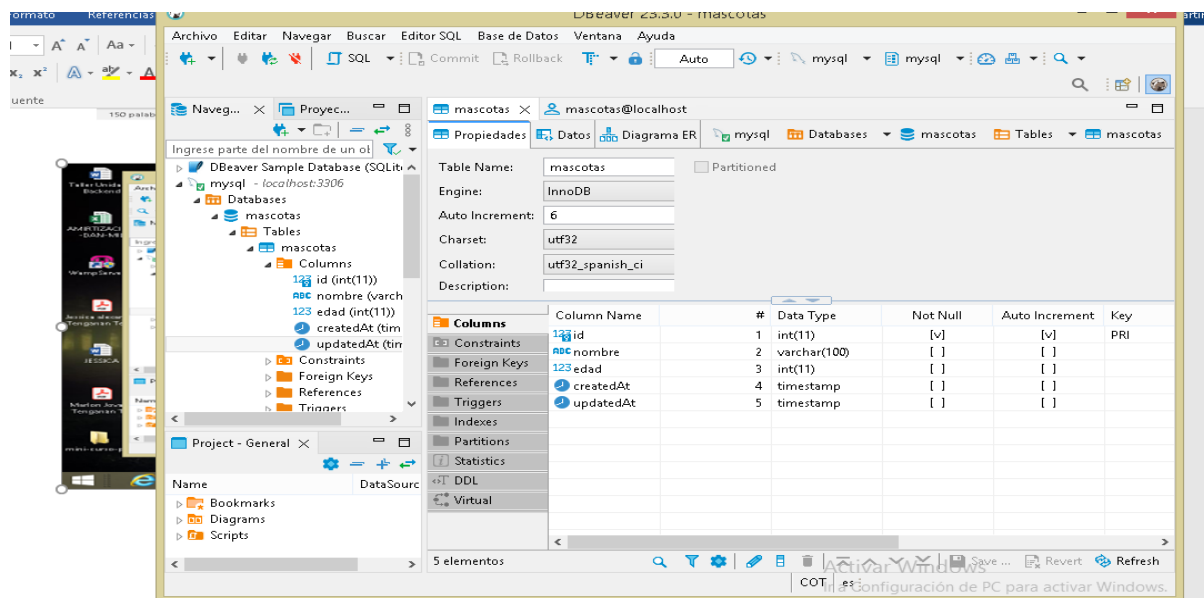
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use mascotas;
Database changed
mysql> create user 'mascotas'@'%' IDENTIFIED BY 'mascotas2023';
ERROR 1396 (HY000): Operation CREATE USER failed for 'mascotas'@'%'
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mascotas |
| mysql |
| performance_schema |
| test |
+-----+
5 rows in set (0.04 sec)

mysql> show tables;
+-----+
| Tables_in_mascotas |
+-----+
| mascotas |
+-----+
1 row in set (0.00 sec)

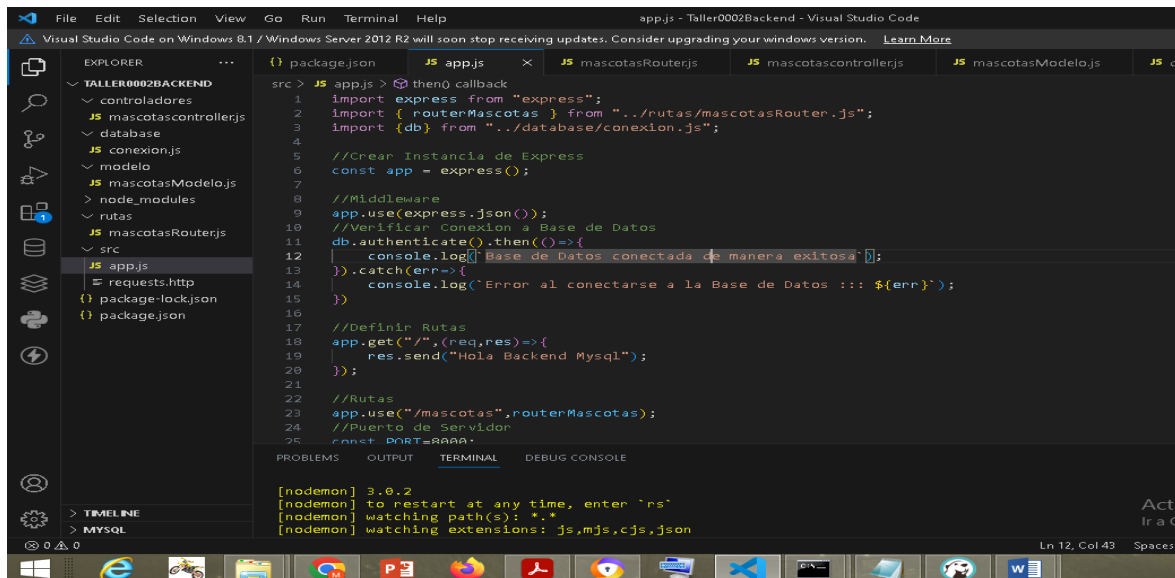
mysql>
```



The screenshot shows the DBeaver interface with the 'mascotas' table selected. The table properties are displayed on the right, and the table structure is shown in the bottom pane.

Column Name	#	Data Type	Not Null	Auto Increment	Key
id	1	int(11)	[ ]	[x]	PRI
nombre	2	varchar(100)	[ ]	[ ]	
edad	3	int(11)	[ ]	[ ]	
createdAt	4	timestamp	[ ]	[ ]	
updatedAt	5	timestamp	[ ]	[ ]	

2. Desarrollar una aplicación Backend implementada en NodeJS y ExpressJS que haga uso de la base de datos del primer punto y que permita el desarrollo de todas las tareas asociadas al registro y administración de las mascotas dadas en adopción por la empresa (La empresa debe contar con un nombre). Se debe hacer uso correcto de los verbos HTTP dependiendo de la tarea a realizar.
- (3 Ptos).

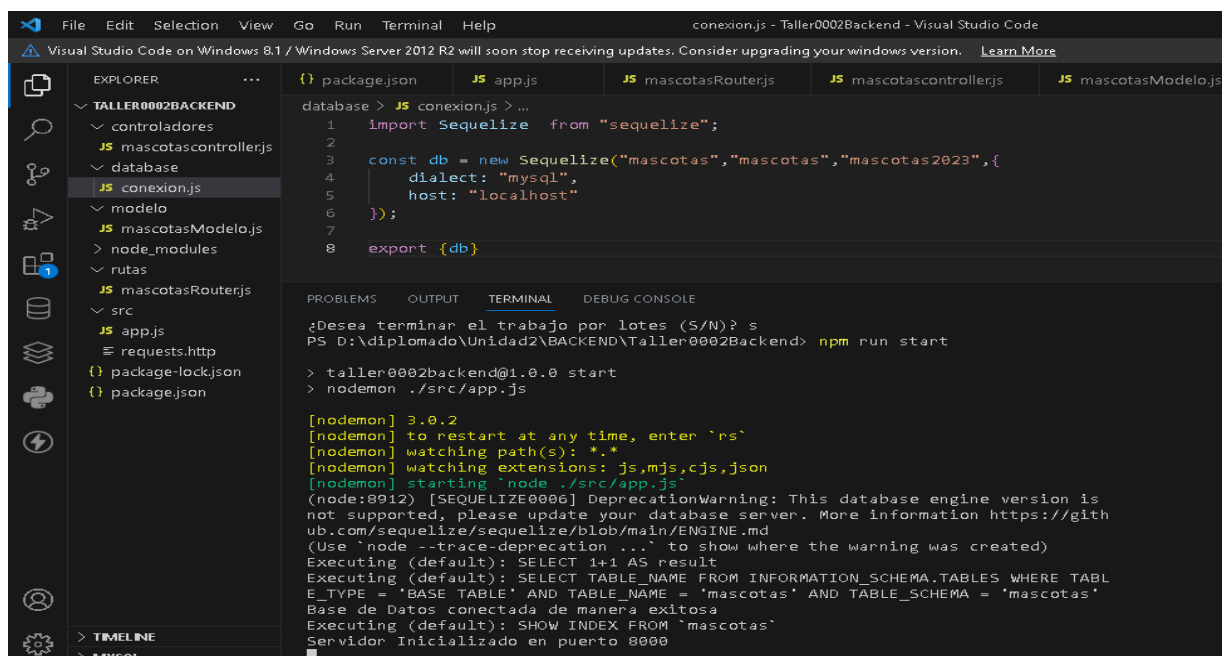


```
src > JS app.js > then() callback
1 import express from "express";
2 import { routerMascotas } from "../rutas/mascotasRouter.js";
3 import {db} from "../database/conexion.js";
4
5 //Crear Instancia de Express
6 const app = express();
7
8 //Middleware
9 app.use(express.json());
10 //Verificar Conexion a Base de Datos
11 db.authenticate().then(()=>{
12   console.log('Base de Datos conectada de manera exitosa');
13 }).catch(err=>{
14   console.log('Error al conectarse a la Base de Datos ::: ${err}');
15 })
16
17 //Definir Rutas
18 app.get("/",(req,res)=>{
19   res.send("Hola Backend Mysql");
20 });
21
22 //Rutas
23 app.use("/mascotas",routerMascotas);
24 //Puerto de Servidor
25 const PORT=8000;
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
[nodeemon] 3.0.2
[nodeemon] to restart at any time, enter 'rs'
[nodeemon] watching path(s): *.*
[nodeemon] watching extensions: js,mjs,cjs,json
```

## Conexión a la base de datos



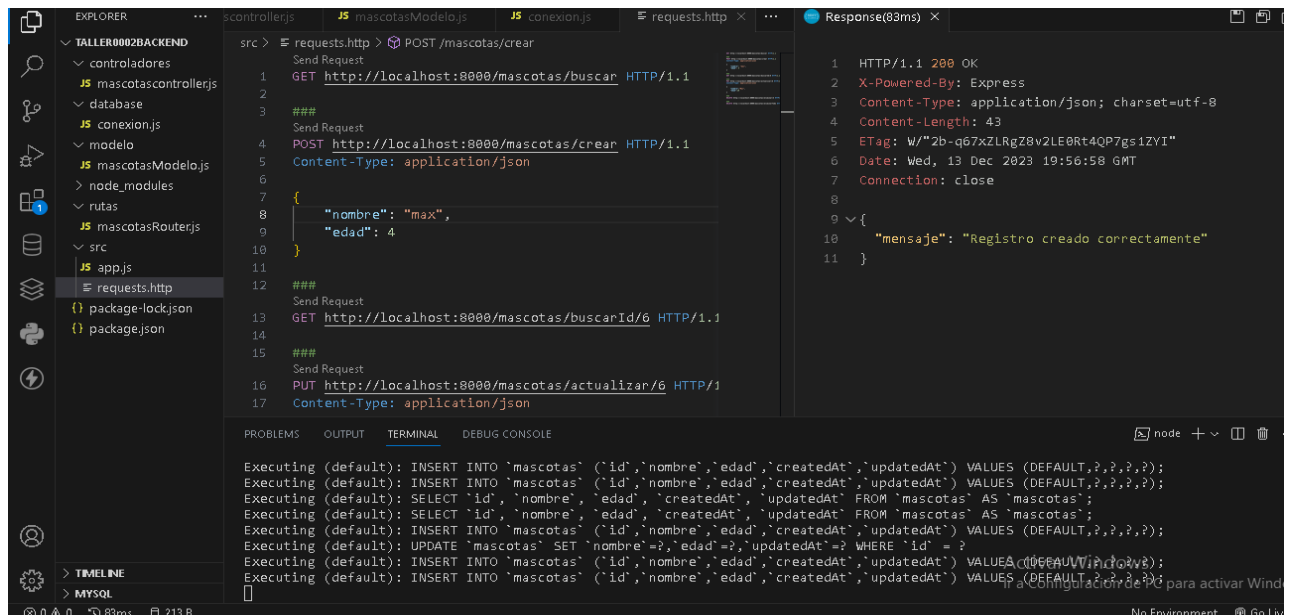
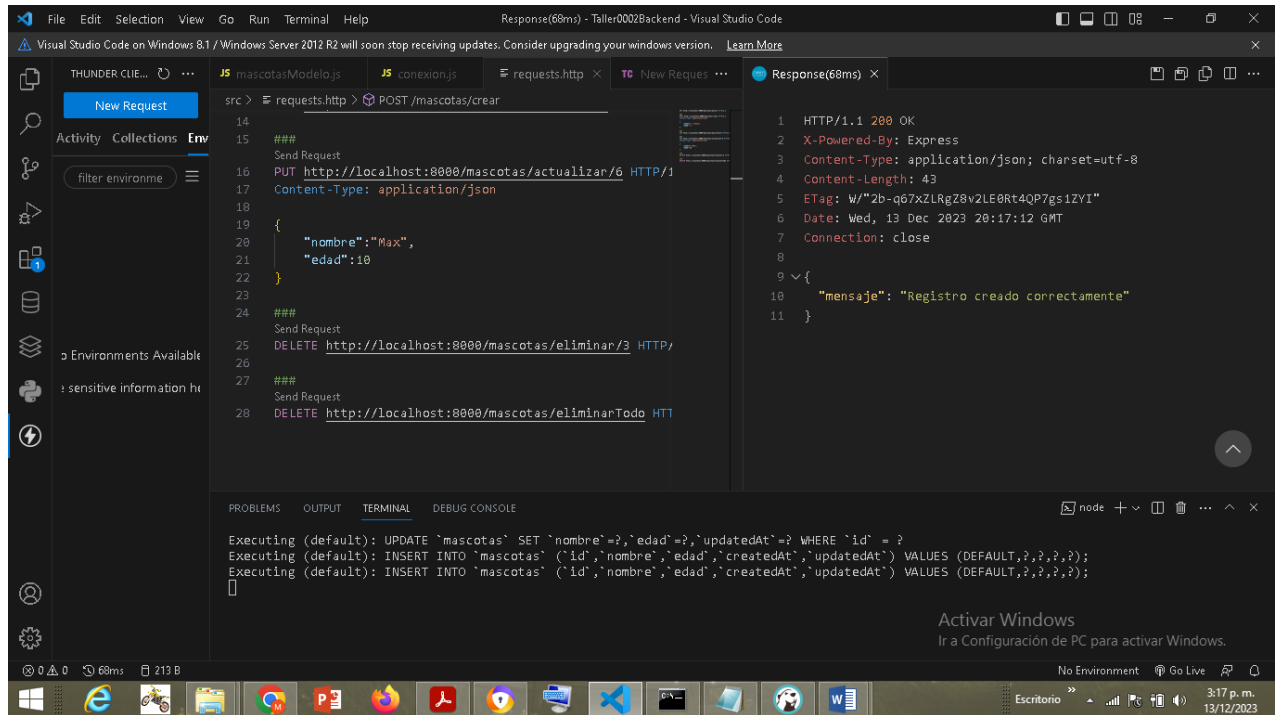
```
database > JS conexion.js > ...
1 import Sequelize from "sequelize";
2
3 const db = new Sequelize("mascotas","mascotas","mascotas2023",{
4   dialect: "mysql",
5   host: "localhost"
6 });
7
8 export {db}
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

```
¿Desea terminar el trabajo por lotes (S/N)? s
PS D:\diplomado\Unidad2\BACKEND\Taller0002Backend> npm run start
> taller0002backend@1.0.0 start
> nodeemon ./src/app.js

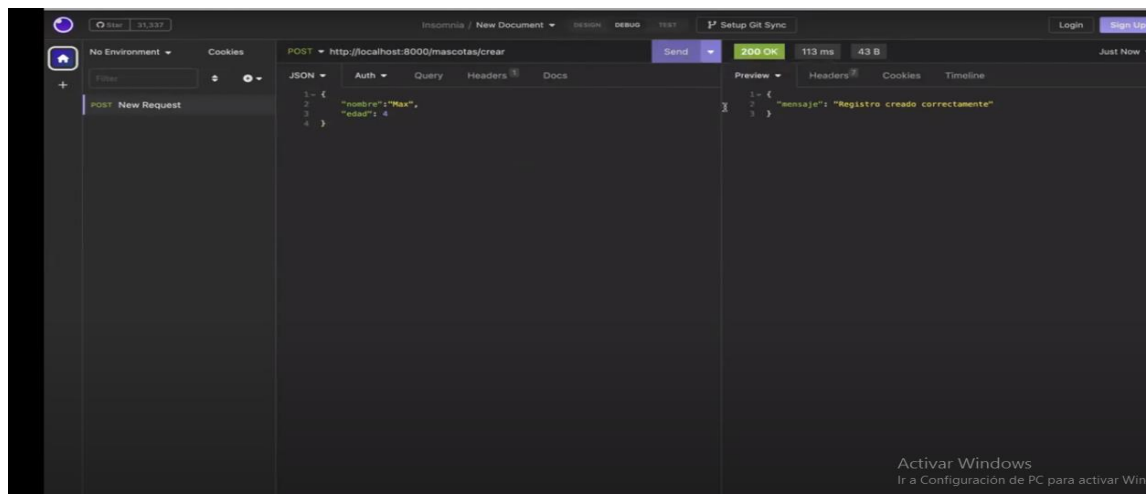
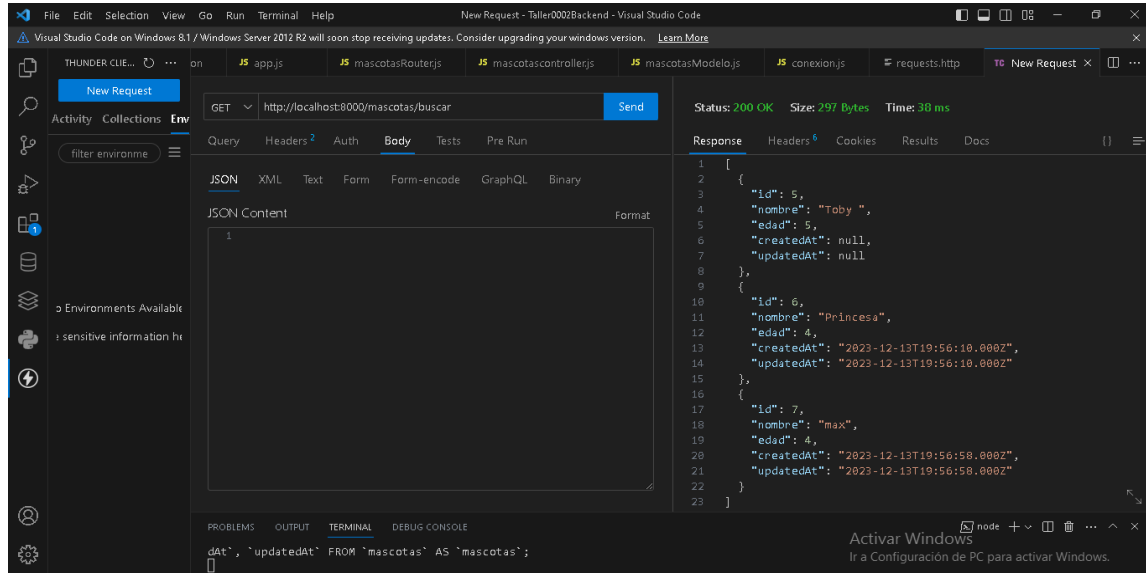
[nodeemon] 3.0.2
[nodeemon] to restart at any time, enter 'rs'
[nodeemon] watching path(s): *.*
[nodeemon] watching extensions: js,mjs,cjs,json
[nodeemon] starting 'node ./src/app.js'
(node:8912) [SEQUELIZE0006] DeprecationWarning: This database engine version is
not supported, please update your database server. More information https://gith
ub.com/sequelize/sequelize/blob/main/ENGINE.md
(Use 'node --trace-deprecation ...' to show where the warning was created)
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABL
E_TYPE = 'BASE TABLE' AND TABLE_NAME = 'mascotas' AND TABLE_SCHEMA = 'mascotas'
Base de Datos conectada de manera exitosa
Executing (default): SHOW INDEX FROM `mascotas`
Servidor Inicializado en puerto 8000
```

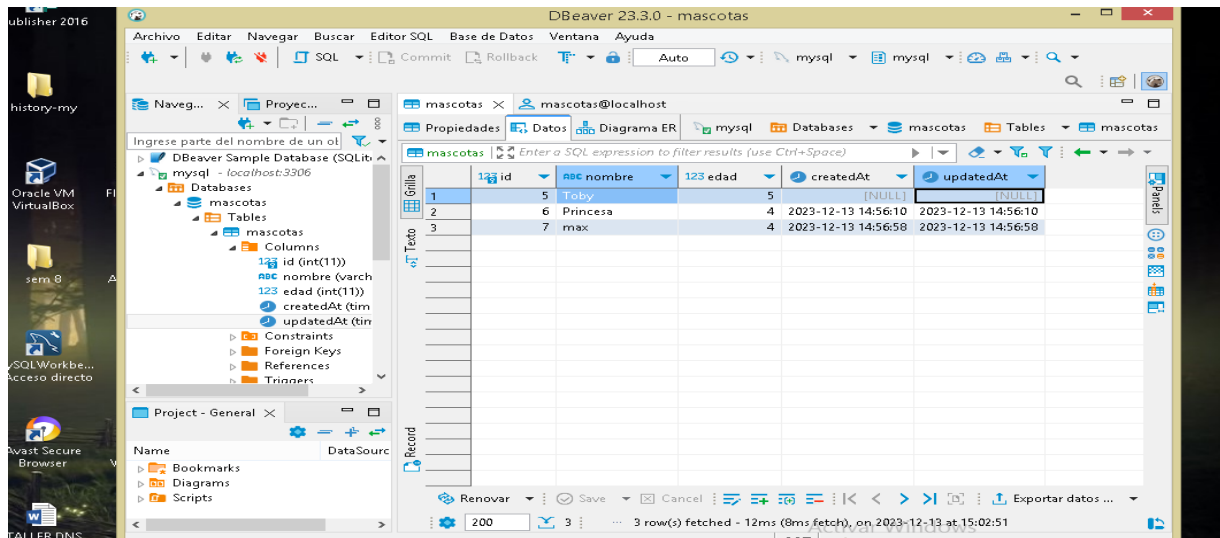
## Registro



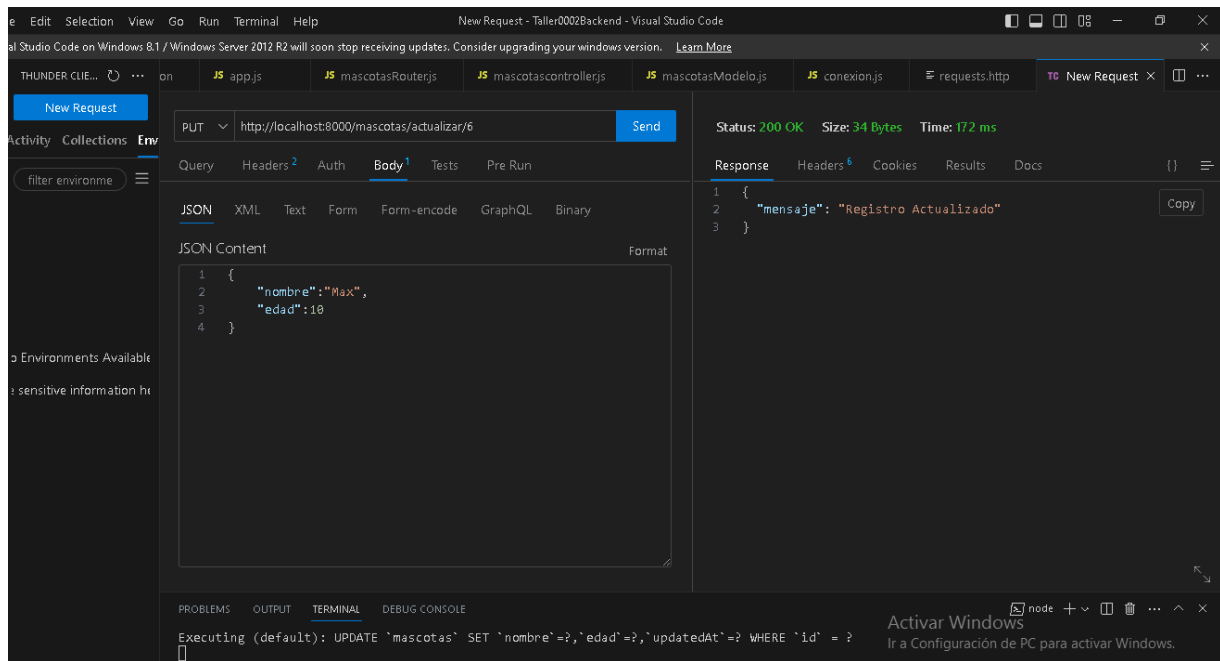
3. Realizar verificación de las diferentes operaciones a través de un cliente grafico (Postman, Imnsomnia, etc.), tomar capturas de pantalla que evidencien el resultado de las solicitudes realizadas. (1 Pto).

## Buscar





## Actualización



## Eliminar

The screenshot displays the Visual Studio Code interface with a REST client request configured for a DELETE operation. The request is sent to the URL `http://localhost:8000/mascotas/eliminar/3`. The response is a 200 OK status with a JSON body containing the message `"Registro Eliminado"`. The terminal at the bottom shows the execution of SQL commands to update, insert, and delete records from the `mascotas` table.

**Request Details:**

- Method: DELETE
- URL: `http://localhost:8000/mascotas/eliminar/3`
- Body: `1`
- Status: 200 OK
- Size: 32 Bytes
- Time: 21 ms

**Response (71ms):**

```
1 HTTP/1.1 200 OK
2 X-Powered-By: Express
3 Content-Type: application/json; charset=utf-8
4 Content-Length: 32
5 ETag: W/"20-hkH0JCv/EfydTXvDeEcSo30b58A"
6 Date: Wed, 13 Dec 2023 20:28:13 GMT
7 Connection: close
8
9 {
10   "mensaje": "Registro Eliminado"
11 }
```

**Terminal Output:**

```
Executing (default): UPDATE `mascotas` SET `nombre`=?,`edad`=?,`updatedAt`=? WHERE `id` = ?
Executing (default): INSERT INTO `mascotas` (`id`,`nombre`,`edad`,`createdAt`,`updatedAt`) VALUES (DEFAULT,?,?,?,?);
Executing (default): INSERT INTO `mascotas` (`id`,`nombre`,`edad`,`createdAt`,`updatedAt`) VALUES (DEFAULT,?,?,?,?);
Executing (default): DELETE FROM `mascotas` WHERE `id` = '3'
Executing (default): DELETE FROM `mascotas` WHERE `id` = '3'
```