

Universidad de Nariño.

Ingeniería de Sistemas.

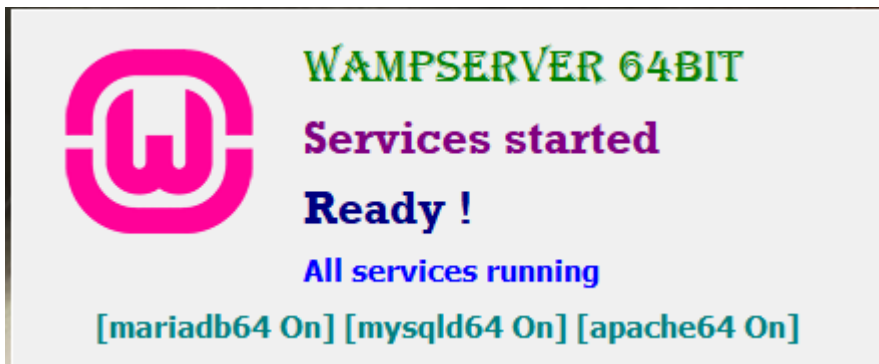
Diplomado de actualización en nuevas tecnologías para el desarrollo de Software.

Taller Unidad 2 Backend

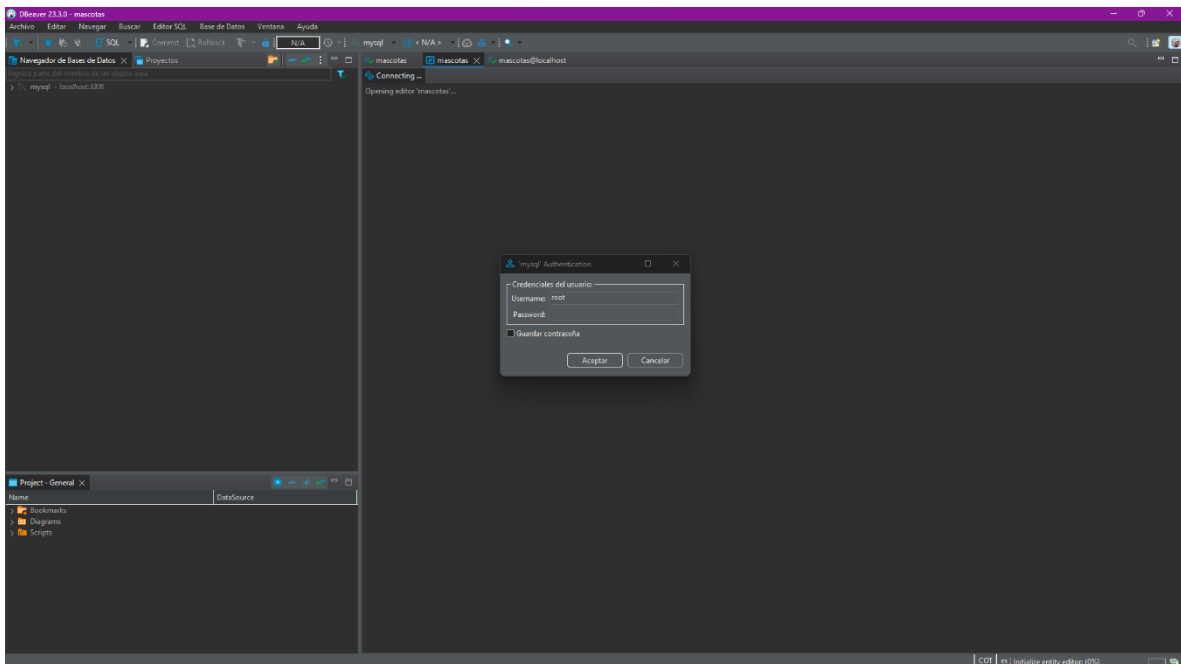
Vanessa Gustin - 219034187

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.

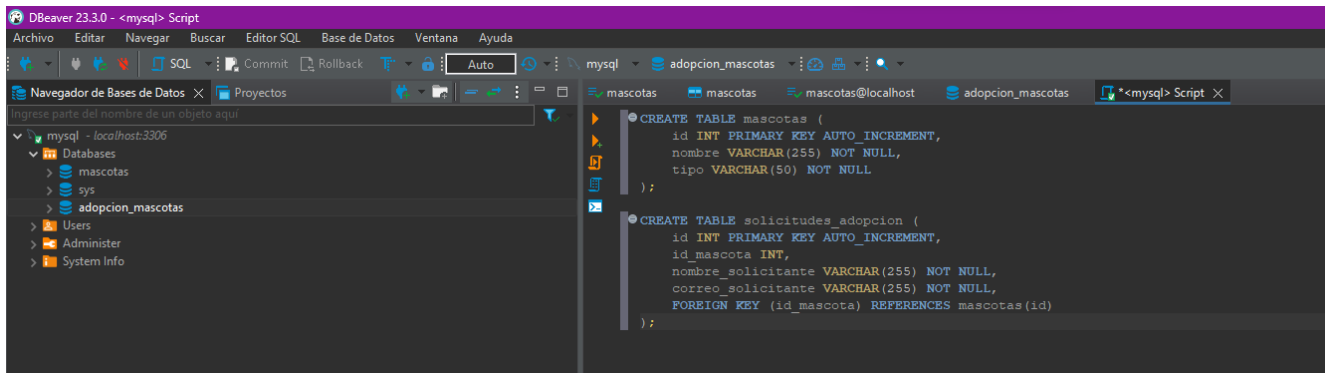
Inicialice Wamp para que levantar el servicio para la base de datos



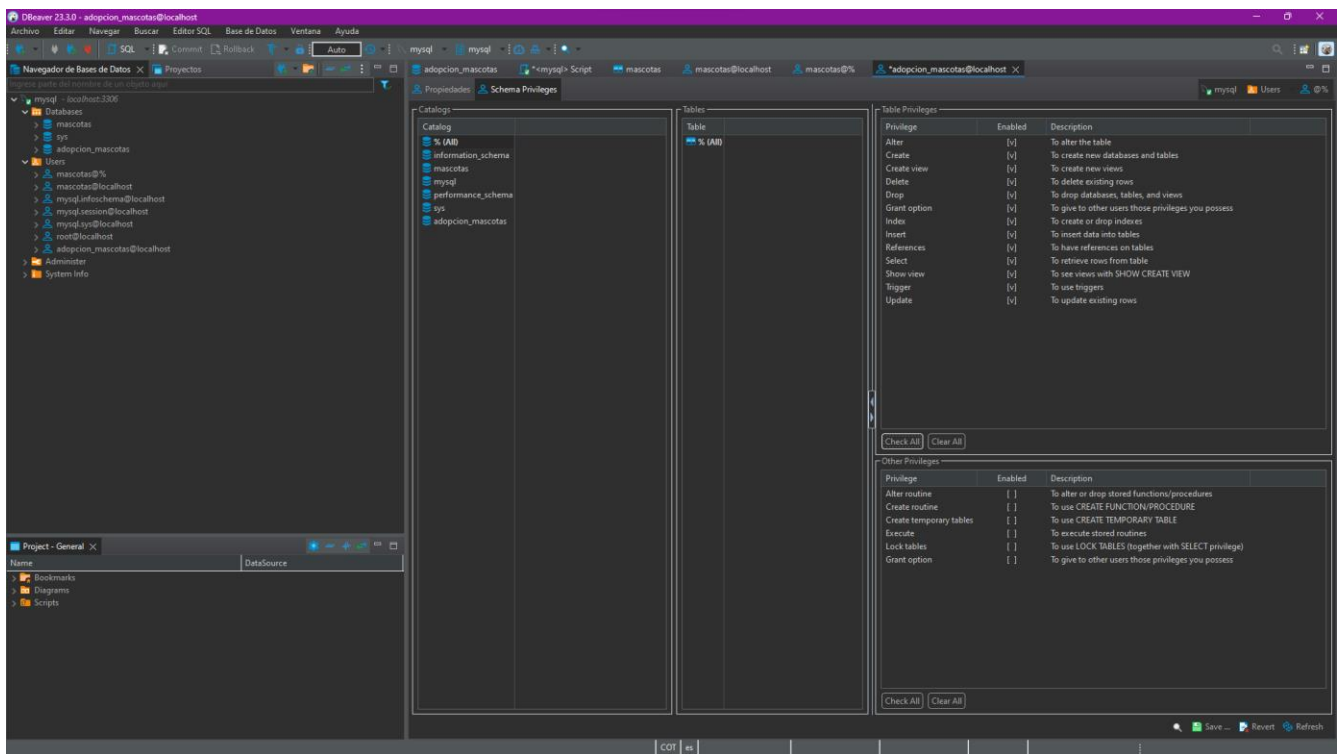
Abro el gestor de base de datos e ingreso con los datos correspondientes



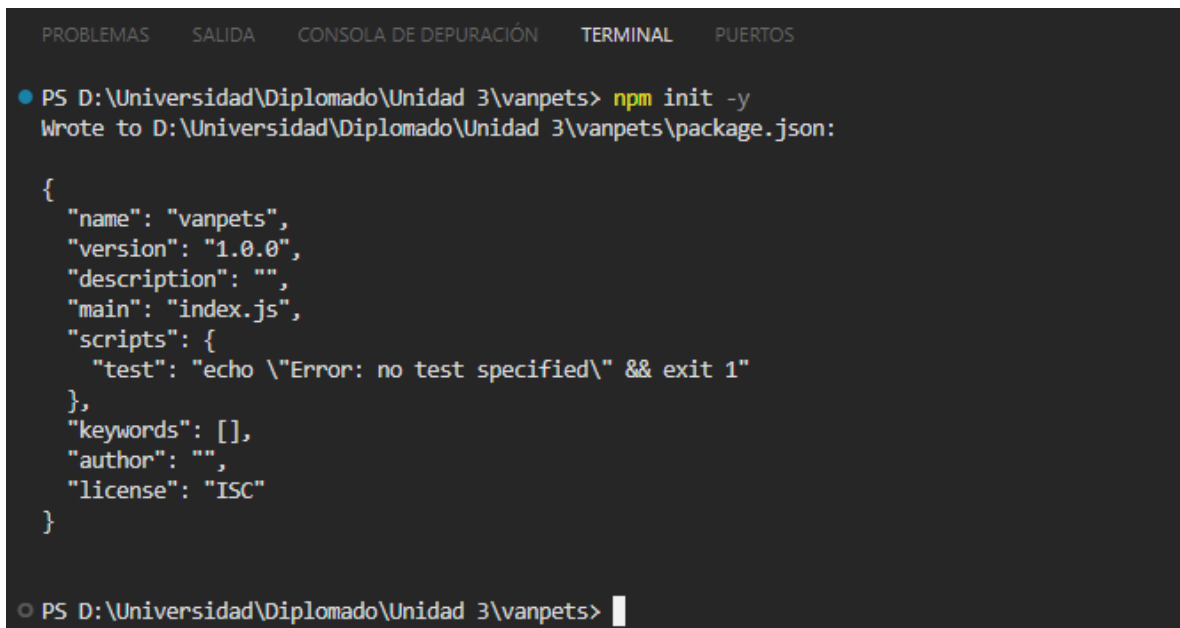
Creo las tablas con sus respectivas características



Creo el usuario para ingresar a la base de datos



Creo el proyecto con nodeJS



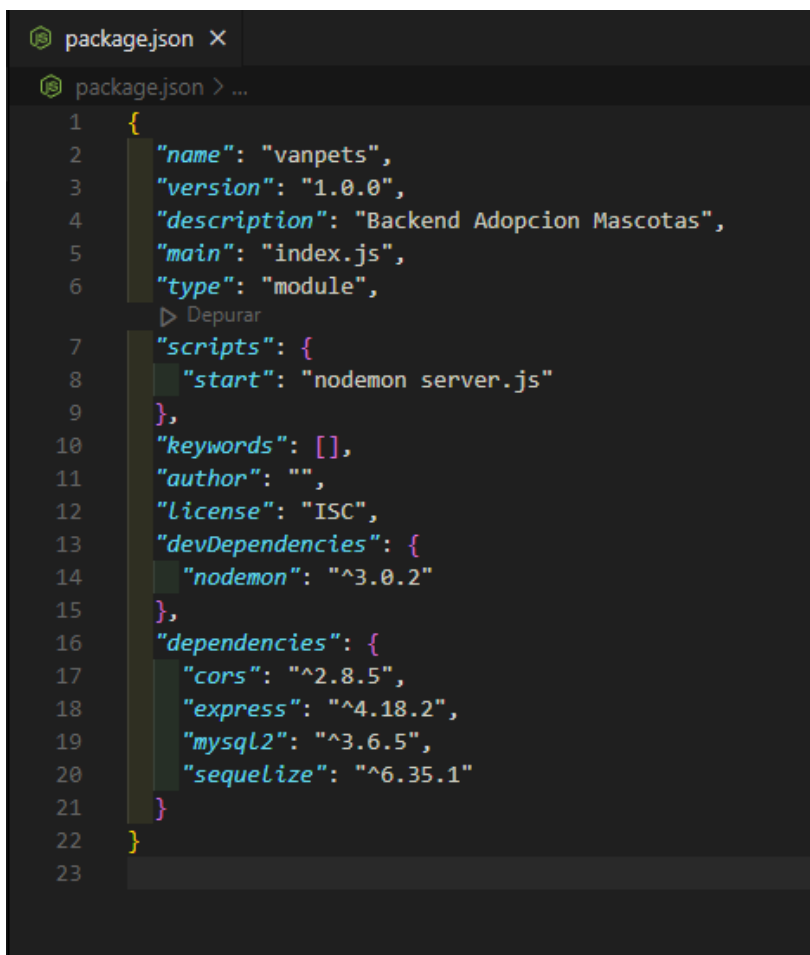
```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

● PS D:\Universidad\Diplomado\Unidad 3\vanpets> npm init -y
Wrote to D:\Universidad\Diplomado\Unidad 3\vanpets\package.json:

{
  "name": "vanpets",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}

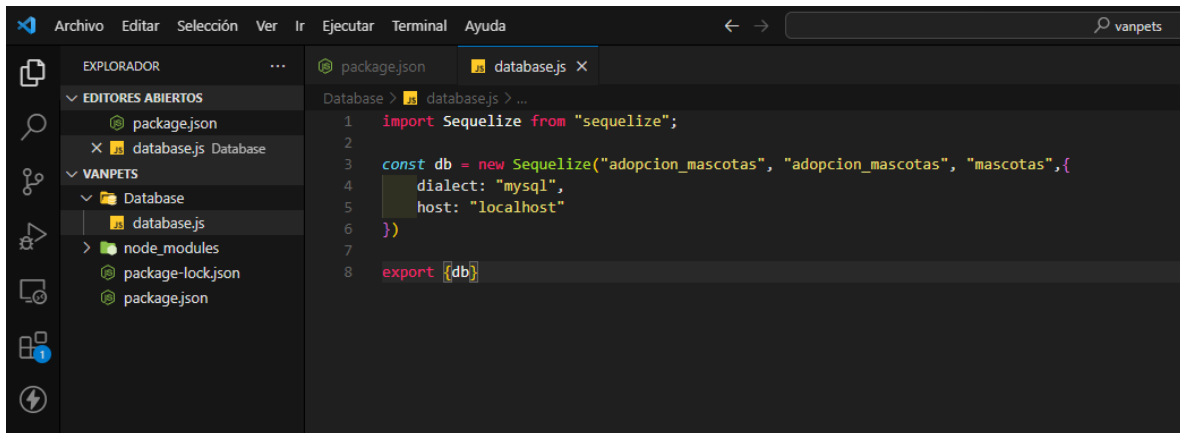
○ PS D:\Universidad\Diplomado\Unidad 3\vanpets> 
```

Instalo las respectivas dependencias para el funcionamiento del proyecto



```
package.json X
package.json > ...
1  {
2    "name": "vanpets",
3    "version": "1.0.0",
4    "description": "Backend Adopción Mascotas",
5    "main": "index.js",
6    "type": "module",
7    "scripts": {
8      "start": "nodemon server.js"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "devDependencies": {
14     "nodemon": "^3.0.2"
15   },
16   "dependencies": {
17     "cors": "^2.8.5",
18     "express": "^4.18.2",
19     "mysql2": "^3.6.5",
20     "sequelize": "^6.35.1"
21   }
22 }
23 
```

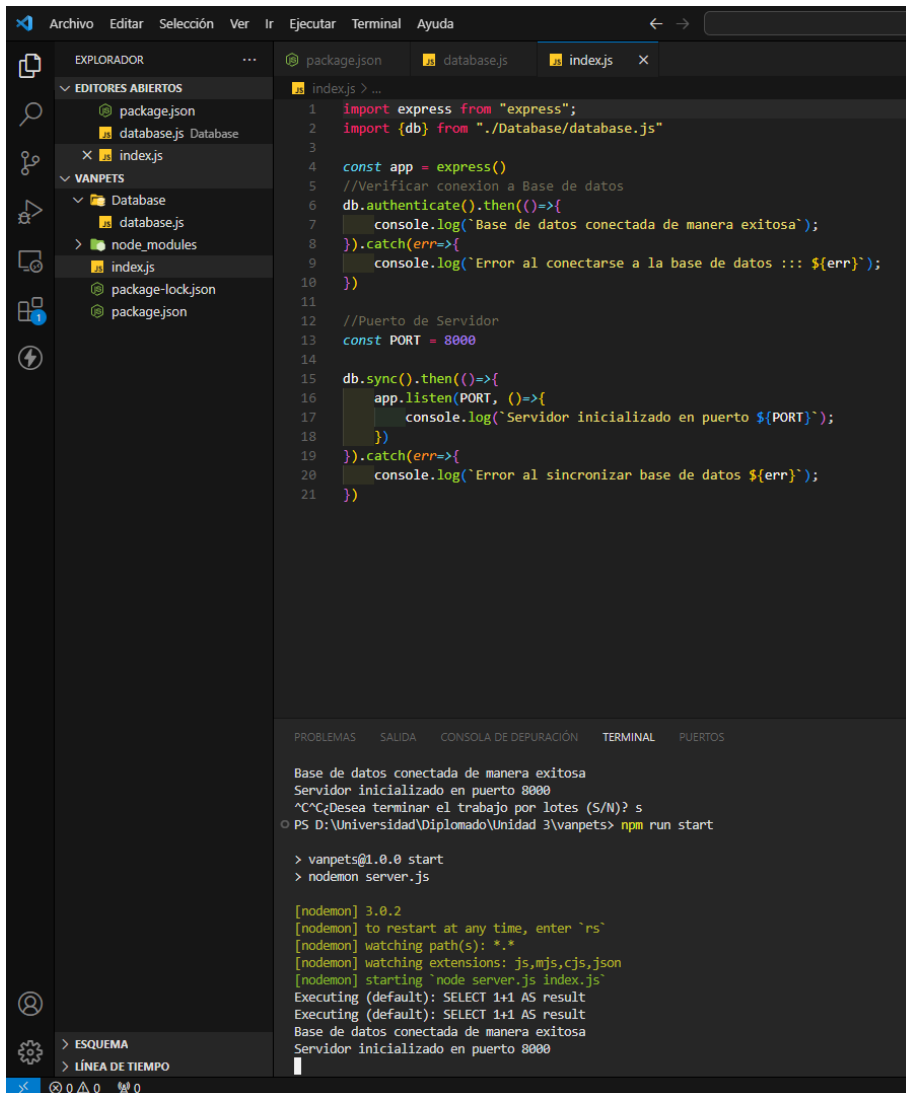
Creo una instancia de sequelize con los datos para que se conecte a mi base de datos



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer shows a project named 'VANPETS' with a 'Database' folder containing 'database.js'. The 'database.js' file is open in the editor, showing the following code:

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

Hago la conexión con la base de datos y levanto el servidor



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The Explorer shows a project named 'VANPETS' with a 'Database' folder containing 'database.js' and an 'index.js' file. The 'index.js' file is open in the editor, showing the following code:

```
index.js > ...
1 import express from "express";
2 import {db} from "../Database/database.js"
3
4 const app = express()
5 //Verificar conexión a Base de datos
6 db.authenticate().then(()=>{
7   console.log("Base de datos conectada de manera exitosa");
8 }).catch(err=>{
9   console.log("Error al conectarse a la base de datos ::: ${err}");
10 })
11
12 //Puerto de Servidor
13 const PORT = 8000
14
15 db.sync().then(()=>{
16   app.listen(PORT, ()=>{
17     console.log("Servidor inicializado en puerto ${PORT}");
18   })
19 }).catch(err=>{
20   console.log("Error al sincronizar base de datos ${err}");
21 })
```

The terminal output at the bottom shows the following messages:

```
Base de datos conectada de manera exitosa
Servidor inicializado en puerto 8000
^C^C^CDesear terminiar el trabajo por lotes (S/N)? s
PS D:\Universidad\Diplomado\Unidad 3\vanpets> npm run start


> vanpets@1.0.0 start
> nodemon server.js

[nodemon] 3.0.2
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node server.js index.js'
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT 1+1 AS result
Base de datos conectada de manera exitosa
Servidor inicializado en puerto 8000
```

Creo el modelo para la tabla mascotas

```
models >  mascota.js > ...  
1  import { DataTypes } from 'sequelize';  
2  import { db } from '../Database/database.js';  
3  
4  const Mascota = db.define('Mascota', {  
5    id: {  
6      type: DataTypes.INTEGER,  
7      primaryKey: true,  
8      autoIncrement: true,  
9    },  
10   nombre: {  
11     type: DataTypes.STRING,  
12     allowNull: false,  
13   },  
14   tipo: {  
15     type: DataTypes.STRING,  
16     allowNull: false,  
17   },  
18 });  
19  
20 export { Mascota };
```

Creo el modelo para la tabla de las solicitudes de adopción


```
models >  solicitudAdopcion.js > ...  
1  import { DataTypes } from 'sequelize';  
2  import { db } from '../Database/database.js';  
3  import { Mascota } from './mascota.js'; // Importa el modelo de mascotas  
4  
5  const SolicitudAdopcion = db.define('SolicitudAdopcion', {  
6    id: {  
7      type: DataTypes.INTEGER,  
8      primaryKey: true,  
9      autoIncrement: true,  
10   },  
11   nombre_solicitante: {  
12     type: DataTypes.STRING,  
13     allowNull: false,  
14   },  
15   correo_solicitante: {  
16     type: DataTypes.STRING,  
17     allowNull: false,  
18   },  
19 });  
20  
21 // Establece la relación con la tabla de mascotas  
22 SolicitudAdopcion.belongsTo(Mascota, { foreignKey: 'id_mascota' });  
23  
24 export { SolicitudAdopcion };
```

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.


Realizo la creación de rutas

```
index.js > ...
1  import express from "express";
2  import {routerMascotas} from './routes/mascotasRoutes.js';
3  import {router} from './routes/solicitudAdopcionRoutes.js';
4  import {db} from './Database/database.js'
5  import cors from 'cors'
6  // import bodyParser from 'body-parser'
7
8  //Crear instancia de express
9  const app = express()
10 app.use(cors())
11
12 // app.use(bodyParser.json())
13 app.use(express.json())
14
15 //Verificar conexion a Base de datos
16 db.authenticate().then(()=>{
17   console.log(`Base de datos conectada de manera exitosa`);
18 }).catch(err=>{
19   console.log(`Error al conectarse a la base de datos ::: ${err}`);
20 })
21
22 // Rutas
23 app.use('/mascotas', routerMascotas);
24 app.use('/solicitudAdopcion', router);
25 //Puerto de Servidor
26 const PORT = 8000
27
28 db.sync().then(()=>{
29   app.listen(PORT, ()=>{
30     console.log(`Servidor inicializado en puerto ${PORT}`);
31   })
32 }).catch(err=>{
33   console.log(`Error al sincronizar base de datos ${err}`);
34 })
```

Hago las rutas para mascotas

```
routes >  mascotasRoutes.js > ...
1  import express from 'express';
2  import {obtenerMascotas,
3      obtenerMascotaPorId,
4      agregarMascota,
5      actualizarMascota,
6      eliminarMascota} from '../Controllers/mascotasController.js';
7  const routerMascotas = express.Router();
8
9  // Ruta para obtener todas las mascotas
10 routerMascotas.get('/obtener', obtenerMascotas);
11
12 // Ruta para obtener una mascota por ID
13 routerMascotas.get('/obtener/:id', obtenerMascotaPorId);
14
15 // Ruta para agregar una nueva mascota
16 routerMascotas.post('/agregar', agregarMascota);
17
18 // Ruta para actualizar una mascota por ID
19 routerMascotas.put('/actualizar/:id', actualizarMascota);
20
21 // Ruta para eliminar una mascota por ID
22 routerMascotas.delete('/eliminar/:id', eliminarMascota);
23
24 export {routerMascotas};
```

Hago las rutas para el proceso de adopción

```
routes >  solicitudAdopcionRoutes.js > ...
1  import express from 'express';
2  import {obtenerSolicitudesAdopcion,
3      obtenerSolicitudAdopcionPorId,
4      agregarSolicitudAdopcion,
5      actualizarSolicitudAdopcion,
6      eliminarSolicitudAdopcion} from '../Controllers/solicitudAdopcionController.js';
7  const router = express.Router();
8
9  // Ruta para obtener todas las solicitudes de adopción
10 router.get('/obtener', obtenerSolicitudesAdopcion);
11
12 // Ruta para obtener una solicitud de adopción por ID
13 router.get('/obtener/:id', obtenerSolicitudAdopcionPorId);
14
15 // Ruta para agregar una nueva solicitud de adopción
16 router.post('/agregar', agregarSolicitudAdopcion);
17
18 // Ruta para actualizar una solicitud de adopción por ID
19 router.put('/actualizar/:id', actualizarSolicitudAdopcion);
20
21 // Ruta para eliminar una solicitud de adopción por ID
22 router.delete('/eliminar/:id', eliminarSolicitudAdopcion);
23
24 export {router};
```

Hago los controladores para las rutas de mascotas

```
Controllers > JS mascotasController.js > ...
1  import { Mascota } from '../models/mascota.js';
2
3  // Controlador para obtener todas las mascotas
4  > const obtenerMascotas = async (req, res) => { ...
12 };
13
14 // Controlador para obtener una mascota por ID
15 > const obtenerMascotaPorId = async (req, res) => { ...
29 };
30
31 // Controlador para agregar una nueva mascota
32 > const agregarMascota = async (req, res) => { ...
42 };
43
44 // Controlador para actualizar una mascota por ID
45 > const actualizarMascota = async (req, res) => { ...
61 };
62
63 // Controlador para eliminar una mascota por ID
64 > const eliminarMascota = async (req, res) => { ...
79 };
80
81 export {
82     obtenerMascotas,
83     obtenerMascotaPorId,
84     agregarMascota,
85     actualizarMascota,
86     eliminarMascota,
87 };
```


Hago los controladores para las rutas de las solicitudes de adopción

```
Controllers > solicitudAdopcionController.js > ...
1  import { SolicitudAdopcion } from '../models/solicitudAdopcion.js';
2
3  // Controlador para obtener todas las solicitudes de adopción
4  > const obtenerSolicitudesAdopcion = async (req, res) => { ...
12 };
13
14 // Controlador para obtener una solicitud de adopción por ID
15 > const obtenerSolicitudAdopcionPorId = async (req, res) => { ...
29 };
30
31 // Controlador para agregar una nueva solicitud de adopción
32 > const agregarSolicitudAdopcion = async (req, res) => { ...
46 };
47
48 // Controlador para actualizar una solicitud de adopción por ID
49 > const actualizarSolicitudAdopcion = async (req, res) => { ...
69 };
70
71 // Controlador para eliminar una solicitud de adopción por ID
72 > const eliminarSolicitudAdopcion = async (req, res) => { ...
87 };
88
89 export {
90   obtenerSolicitudesAdopcion,
91   obtenerSolicitudAdopcionPorId,
92   agregarSolicitudAdopcion,
93   actualizarSolicitudAdopcion,
94   eliminarSolicitudAdopcion,
95 };
```

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.

Agregar una mascota

POST localhost:8000/mascotas/agregar Send

Status: 201 Created Size: 117 Bytes Time: 26 ms

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "nombre": "Toby",
3   "tipo": "perro"
4 }
```

Response Headers⁷ Cookies Results Docs

```
1 {
2   "id": 1,
3   "nombre": "Toby",
4   "tipo": "perro",
5   "updatedAt": "2023-12-19T03:53:20.033Z",
6   "createdAt": "2023-12-19T03:53:20.033Z"
7 }
```

Obtener todas las mascotas

GETlocalhost:8000/mascotas/obtenerSend

QueryHeaders 2AuthBody 1TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "nombre": "Juana",
3   "tipo": "gato"
4 }
```

Status: 200 OKSize: 237 BytesTime: 12 ms

ResponseHeaders 7CookiesResultsDocs

```
1 [
2   {
3     "id": 1,
4     "nombre": "Toby",
5     "tipo": "perro",
6     "createdAt": "2023-12-19T03:53:20.000Z",
7     "updatedAt": "2023-12-19T03:53:20.000Z"
8   },
9   {
10    "id": 2,
11    "nombre": "Juana",
12    "tipo": "gato",
13    "createdAt": "2023-12-19T03:54:28.000Z",
14    "updatedAt": "2023-12-19T03:54:28.000Z"
15  }
16 ]
```

Obtener una mascota por ID

GETlocalhost:8000/mascotas/obtener/1Send

QueryHeaders 2AuthBody 1TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "nombre": "Juana",
3   "tipo": "gato"
4 }
```

Status: 200 OKSize: 117 BytesTime: 17 ms

ResponseHeaders 7CookiesResultsDocs

```
1 {
2   "id": 1,
3   "nombre": "Toby",
4   "tipo": "perro",
5   "createdAt": "2023-12-19T03:53:20.000Z",
6   "updatedAt": "2023-12-19T03:53:20.000Z"
7 }
```

Actualizar una mascota

PUTlocalhost:8000/mascotas/actualizar/2Send

QueryHeaders 2AuthBody 1TestsPre Run

JSONXMLTextFormForm-encodeGraphQLBinary

JSON ContentFormat

```
1 {
2   "nombre": "Juanita",
3   "tipo": "gato"
4 }
```

Status: 200 OKSize: 119 BytesTime: 24 ms

ResponseHeaders 7CookiesResultsDocs

```
1 {
2   "id": 2,
3   "nombre": "Juanita",
4   "tipo": "gato",
5   "createdAt": "2023-12-19T03:54:28.000Z",
6   "updatedAt": "2023-12-19T03:55:46.212Z"
7 }
```

Eliminar una mascota

DELETElocalhost:8000/mascotas/eliminar/2Send

QueryHeaders 2AuthBodyTestsPre Run

Query Parameters

☐ parameter

value

Status: 200 OKSize: 45 BytesTime: 16 ms

ResponseHeaders 7CookiesResultsDocs

```
1 {
2   "mensaje": "Mascota eliminada correctamente"
3 }
```

GETlocalhost:8000/mascotas/obtenerSend

QueryHeaders 2AuthBodyTestsPre Run

Query Parameters

☐ parameter

value

Status: 200 OKSize: 119 BytesTime: 10 ms

ResponseHeaders 7CookiesResultsDocs

```
1 [
2   {
3     "id": 1,
4     "nombre": "Toby",
5     "tipo": "perro",
6     "createdAt": "2023-12-19T03:53:20.000Z",
7     "updatedAt": "2023-12-19T03:53:20.000Z"
8   }
9 ]
```

Agregar una solicitud de adopción

POST localhost:8000/solicitudAdopcion/agregar **Send**

Status: 201 Created Size: 182 Bytes Time: 35 ms

Query Headers 2 Auth **Body 1** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "id_mascota": 1,
3   "nombre_solicitante": "Juan Pérez",
4   "correo_solicitante": "juan.perez@example.com"
5 }
6
```

Response Headers 7 Cookies Results Docs

```
1 {
2   "id": 1,
3   "id_mascota": 1,
4   "nombre_solicitante": "Juan Pérez",
5   "correo_solicitante": "juan.perez@example.com",
6   "updatedAt": "2023-12-19T04:09:16.237Z",
7   "createdAt": "2023-12-19T04:09:16.237Z"
8 }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN **TERMINAL** PUERTOS

```
Executing (default): SELECT `id`, `nombre`, `tipo`, `createdAt`, `updatedAt` FROM `Mascota` AS `Mascota`;
[nodemon] restarting due to changes...
[nodemon] starting `node server.js index.js`
Error al conectarse a la base de datos ::: SequelizeConnectionError: Unknown database 'adopcion_mascota'
Error al sincronizar base de datos SequelizeConnectionError: Unknown database 'adopcion_mascota'
[nodemon] clean exit - waiting for changes before restart
[nodemon] restarting due to changes...
[nodemon] starting `node server.js index.js`
Executing (default): SELECT 1+1 AS result
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = 'Mascota' AND TABLE_SCHEMA = 'adopcion_mascotas'
Base de datos conectada de manera exitosa
Executing (default): SHOW INDEX FROM `Mascota`
Executing (default): SELECT TABLE_NAME FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_TYPE = 'BASE TABLE' AND TABLE_NAME = 'SolicitudAdopcions' AND TABLE_SCHEMA = 'adopcion_mascotas'
Executing (default): SHOW INDEX FROM `SolicitudAdopcions`
Servidor inicializado en puerto 8000
Executing (default): INSERT INTO `Mascota` (`id`,`nombre`,`tipo`,`createdAt`,`updatedAt`) VALUES (DEFAULT,?,?,?,?);
Executing (default): INSERT INTO `SolicitudAdopcions` (`id`,`nombre_solicitante`,`correo_solicitante`,`createdAt`,`updatedAt`,`id_mascota`) VALUES (DEFAULT,?,?,?,?);
[]
```

Obtener todas las solicitudes de adopción

GET localhost:8000/solicitudAdopcion/obtener **Send**

Status: 200 OK Size: 374 Bytes Time: 15 ms

Query Headers 2 Auth **Body 1** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "id_mascota": 1,
3   "nombre_solicitante": "Vanessa Gustin",
4   "correo_solicitante": "vanesagustin98@hotmail.com"
5 }
6
```

Response Headers 7 Cookies Results Docs

```
1 [
2   {
3     "id": 1,
4     "nombre_solicitante": "Juan Pérez",
5     "correo_solicitante": "juan.perez@example.com",
6     "createdAt": "2023-12-19T04:09:16.000Z",
7     "updatedAt": "2023-12-19T04:09:16.000Z",
8     "id_mascota": 1
9   },
10  {
11    "id": 3,
12    "nombre_solicitante": "Vanessa Gustin",
13    "correo_solicitante": "vanesagustin98@hotmail.com",
14    "createdAt": "2023-12-19T04:12:39.000Z",
15    "updatedAt": "2023-12-19T04:12:39.000Z",
16    "id_mascota": 1
17  }
18 ]
```

Obtener una solicitud de adopción por ID

GET localhost:8000/solicitudAdopcion/obtener/3 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

```
1 {
2   "id_mascota": 1,
3   "nombre_solicitante": "Vanessa Gustin",
4   "correo_solicitante": "vanesagustin98@hotmail.com"
5 }
6
```

Format

Status: 200 OK Size: 189 Bytes Time: 14 ms

Response Headers 7 Cookies Results Docs

```
1 {
2   "id": 3,
3   "nombre_solicitante": "Vanessa Gustin",
4   "correo_solicitante": "vanesagustin98@hotmail.com",
5   "createdAt": "2023-12-19T04:12:39.000Z",
6   "updatedAt": "2023-12-19T04:12:39.000Z",
7   "id_mascota": 1
8 }
```

Actualizar una solicitud de adopción

PUT localhost:8000/solicitudAdopcion/actualizar/3 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

```
1 {
2   "id_mascota": 1,
3   "nombre_solicitante": "Vanessa Gustin",
4   "correo_solicitante": "vanesagustin1998@hotmail.com"
5 }
6
```

Format

Status: 200 OK Size: 191 Bytes Time: 20 ms

Response Headers 7 Cookies Results Docs

```
1 {
2   "id": 3,
3   "nombre_solicitante": "Vanessa Gustin",
4   "correo_solicitante": "vanesagustin1998@hotmail.com",
5   "createdAt": "2023-12-19T04:12:39.000Z",
6   "updatedAt": "2023-12-19T04:14:13.706Z",
7   "id_mascota": 1
8 }
```

Eliminar una solicitud de adopción

DELETE localhost:8000/solicitudAdopcion/eliminar/1 Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content

```
1
```

Format

Status: 200 OK Size: 60 Bytes Time: 17 ms

Response Headers 7 Cookies Results Docs

```
1 {
2   "mensaje": "Solicitud de adopción eliminada correctamente"
3 }
```