

Universidad de Nariño

Ingeniería de Sistemas

Diplomado de actualización en nuevas tecnologías para el desarrollo de  
Software

Taller Unidad 2 Backend

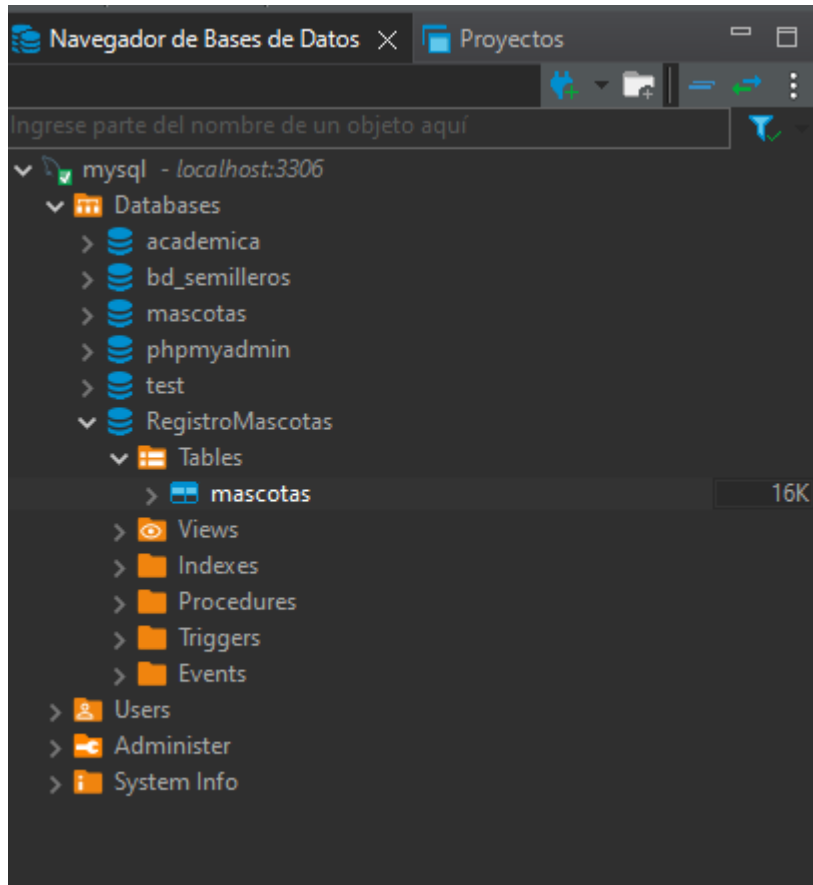
Jhonattan Gabriel Benavides Concha

Código: 219034046

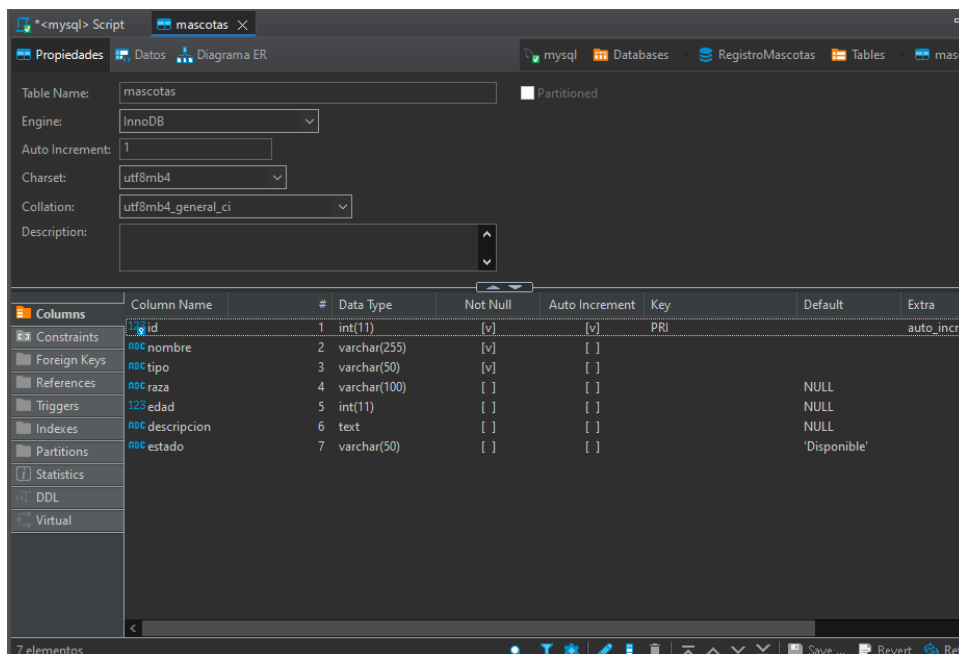
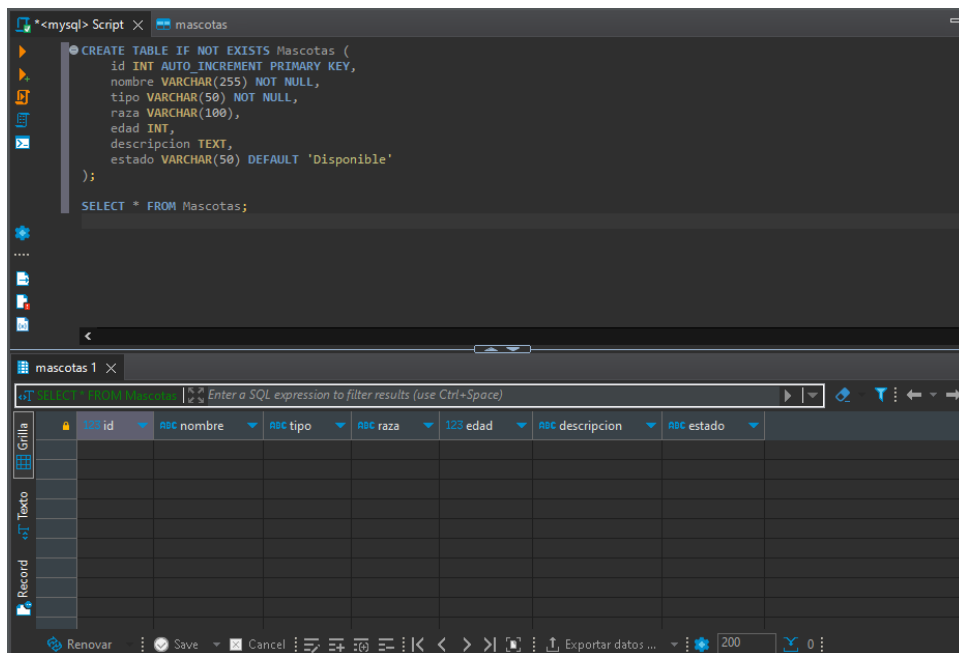
15 de diciembre de 2023

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.

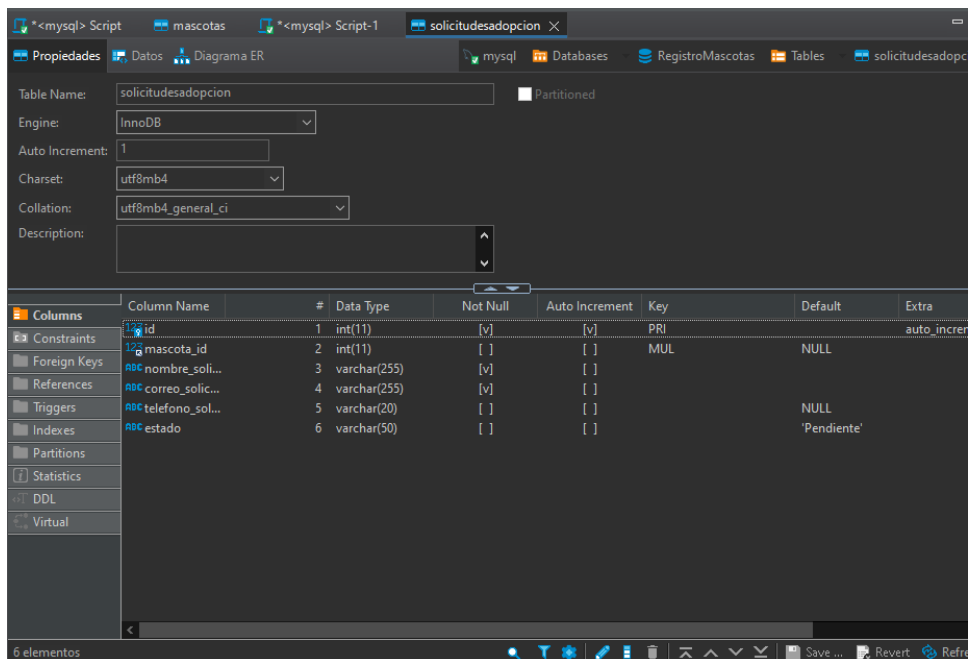
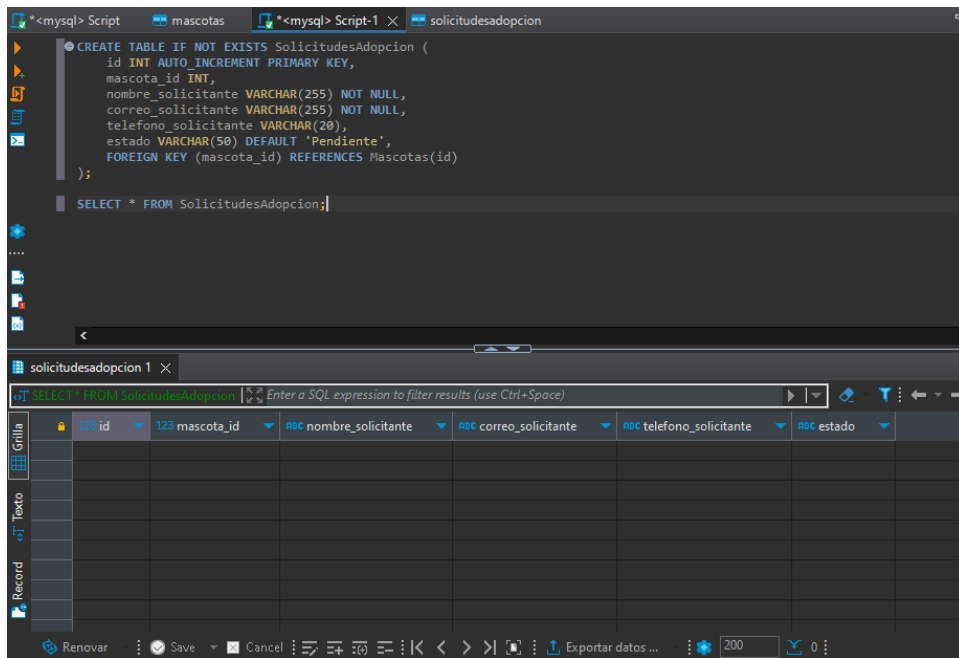
Base de datos: RegistroMascotas



## Creación de la tabla Mascotas



## Creación de la tabla SolicitudesAdopcion



- 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.

Creación del proyecto

```
ttanie1@DESKTOP-NNNACB2 MINGW64 /d/UDENAR/Diplomado/Unidad 2/TallerUnidad2Backend/app_adopcion
$ npm init -y
Wrote to D:\UDENAR\Diplomado\Unidad 2\TallerUnidad2Backend\app_adopcion\package.json:

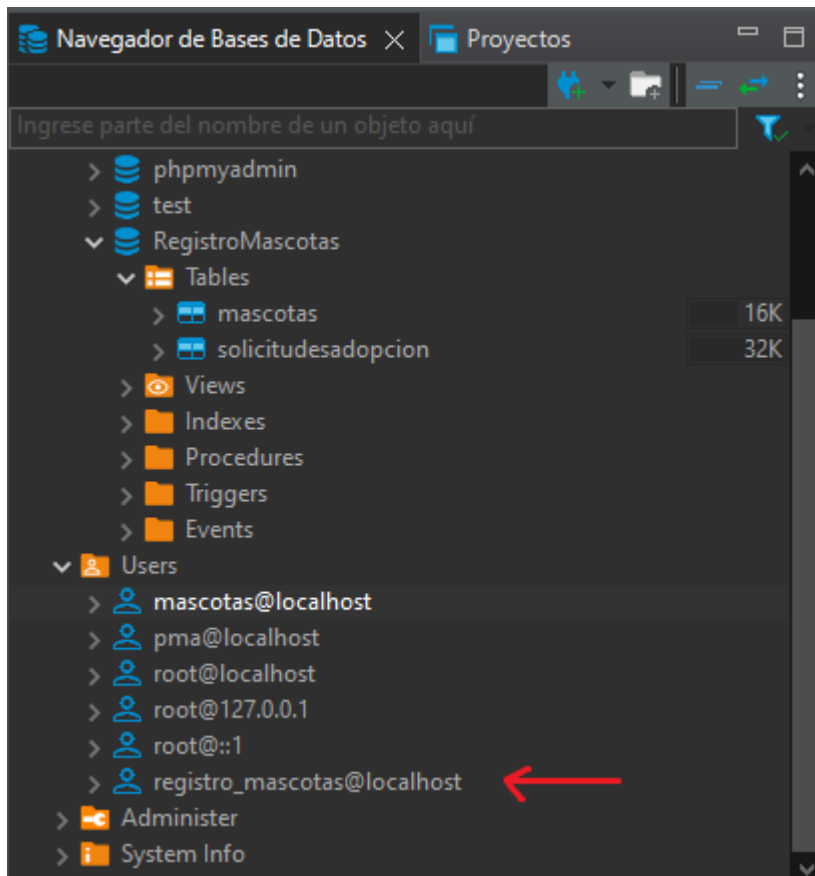
{
  "name": "app_adopcion",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```

Se realizó las instalaciones de las dependencias necesarias, express, mysql2, sequelize y nodemon

```
package.json X
1  {
2    "name": "app_adopcion",
3    "version": "1.0.0",
4    "description": "",
5    "main": "app.js",
6    "type": "module",
7    "scripts": {
8      "start": "nodemon ./src/app.js"
9    },
10   "keywords": [],
11   "author": "",
12   "license": "ISC",
13   "dependencies": {
14     "express": "^4.18.2",
15     "mysql": "^2.18.1",
16     "mysql2": "^3.6.5",
17     "sequelize": "^6.35.2"
18   },
19   "devDependencies": {
20     "nodemon": "^3.0.2"
21   }
22 }
23 |
```

Creación del archivo conexion.js

```
package.json  conexion.js X
1  import Sequelize from "sequelize";
2
3  const db = new Sequelize("RegistroMascotas", "registro_mascotas", "adopcion2023", {
4    dialect: "mysql",
5    host: "localhost"
6  });
7
8  export {db}
```



Creación de los archivos para los modelos

```

package.json  conexion.js  mascotasModel.js X  solicitudesAdopcionModel.js
1  import Sequelize from "sequelize";
2  import { db } from "../database/conexion.js";
3
4  const Mascotas = db.define("Mascotas", {
5      id: {
6          type: Sequelize.INTEGER,
7          allowNull: false,
8          autoIncrement: true,
9          primaryKey: true,
10     },
11     nombre: {
12         type: Sequelize.STRING(255),
13         allowNull: false,
14     },
15     tipo: {
16         type: Sequelize.STRING(50),
17         allowNull: false,
18     },
19     raza: {
20         type: Sequelize.STRING(100),
21     },
22     edad: {
23         type: Sequelize.INTEGER,
24     },
25     descripcion: {
26         type: Sequelize.TEXT,
27     },
28     estado: {
29         type: Sequelize.STRING(50),
30         defaultValue: 'Disponible',
31     }
32 }, {
33     timestamps: false,
34     createdAt: false,

```

```

32 }, {
33     timestamps: false,
34     createdAt: false,
35     updatedAt: false
36 });
37
38 export { Mascotas };
39

```

```
package.json  conexion.js  mascotasModel.js  solicitudesAdopcionModel.js X
1  import Sequelize from "sequelize";
2  import { db } from "../database/conexion.js";
3  import { Mascotas } from "../mascotasModel.js";
4
5  const SolicitudesAdopcion = db.define("SolicitudesAdopcion", {
6    id: {
7      type: Sequelize.INTEGER,
8      allowNull: false,
9      autoIncrement: true,
10     primaryKey: true,
11   },
12   mascota_id: {
13     type: Sequelize.INTEGER,
14     allowNull: false,
15   },
16   nombre_solicitante: {
17     type: Sequelize.STRING(255),
18     allowNull: false,
19   },
20   correo_solicitante: {
21     type: Sequelize.STRING(255),
22     allowNull: false,
23   },
24   telefono_solicitante: {
25     type: Sequelize.STRING(20),
26   },
27   estado: {
28     type: Sequelize.STRING(50),
29     defaultValue: 'Pendiente',
30   }
31 }, {
32   timestamps: false,
33   createdAt: false,
34   updatedAt: false
35 });
```

```
31 }, {
32   timestamps: false,
33   createdAt: false,
34   updatedAt: false
35 });
36
37 SolicitudesAdopcion.belongsTo(Mascotas, { foreignKey: 'mascota_id' });
38
39 export { SolicitudesAdopcion };
40
```

Creación de los archivos para los controladores

```
conexion.js  mascotasModel.js  solicitudesAdopcionModel.js  mascotasController.js X
1  import { Mascotas } from "../modelos/mascotasModel";
2
3  export {
4
5  }
```

```
conexion.js  mascotasModel.js  solicitudesAdopcionModel.js  mascotasController.js  solicitudesAdopcionController.js X
1  import { SolicitudesAdopcion } from "../modelos/solicitudesAdopcionModel";
2
3  export {
4
5  }
```



## Creación de los archivos para las rutas

```
mascotasRouter.js X solicitudesAdopcionRouter.js
1 import express from "express";
2 import {} from "../controladores/mascotasController.js";
3
4 const routerMascotas = express.Router();
5
6 routerMascotas.get("/", (req, res) => {
7   res.send("Bienvenido a Mascotas");
8 });
9
10 export { routerMascotas }
```

```
mascotasRouter.js solicitudesAdopcionRouter.js X
1 import express from "express";
2 import {} from "../controladores/solicitudesAdopcionController.js";
3
4 const routerSolicitudesAdopcion = express.Router();
5
6 routerSolicitudesAdopcion.get("/", (req, res) => {
7   res.send("Bienvenido a Solicitudes Adopción");
8 });
9
10 export { routerSolicitudesAdopcion }
```

## Creación del archivo app.js

```
app.js X
1 import express from "express";
2 import bodyParser from "body-parser";
3
4 import { routerMascotas } from "../rutas/mascotasRouter.js";
5 import { routerSolicitudesAdopcion } from "../rutas/solicitudesAdopcionRouter.js";
6 import { db } from "../database/conexion.js";
7
8 const app = express();
9 app.use(bodyParser.json());
10
11 //
12 db.authenticate().then(() => {
13   console.log(`Base de datos conectada de manera exitosa.`);
14 }).catch(err => {
15   console.log(`Error al conectarse a la base de datos ::: ${err}`);
16 })
17 //
18 app.get("/", (req, res) => {
19   res.send(`Hola, este es el Taller Unidad 2 Backend.`);
20 });
21
22 app.use("/mascotas", routerMascotas);
23 app.use("/solicitudesAdopcion", routerSolicitudesAdopcion);
24
25 //
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 })
```

Desarrollo de todas las tareas asociadas al registro y administración de las mascotas dadas en adopción por la empresa.

Nombre de la empresa: LASCAN

Controlador de mascotas

```
app.js  mascotasController.js X  mascotasRouter.js
1  import { Mascotas } from "../modelos/mascotasModel.js"
2
3  // Obtener todas las mascotas
4  const obtenerTodasLasMascotas = (req, res) => {
5      Mascotas.findAll()
6          .then((resultados) => {
7              if (resultados.length === 0) {
8                  res.status(404).json({
9                      mensaje: "No se encontraron mascotas."
10                 });
11             } else {
12                 res.status(200).json({
13                     mensaje: "Búsqueda exitosa",
14                     mascotas: resultados
15                 });
16             }
17         })
18         .catch((err) => {
19             res.status(500).json({
20                 mensaje: `Error al realizar la búsqueda: ${err}`
21             });
22         });
23     };
24
25     // Obtener información de una mascota por ID
26     const obtenerMascotaPorId = (req, res) => {
27         const mascotaId = req.params.id;
28
29         if (!mascotaId || isNaN(mascotaId)) {
30             res.status(400).json({
31                 mensaje: "No es válido el Id de mascota."
32             });
33             return;
```

```
app.js  mascotasController.js X  mascotasRouter.js
32      },
33      return;
34  }
35
36  Mascotas.findByPk(mascotaId)
37  .then((resultado) => {
38      if (!resultado) {
39          res.status(404).json({
40              mensaje: "No se encontró la mascota."
41          });
42      } else {
43          res.status(200).json({
44              mensaje: "Búsqueda exitosa",
45              mascota: resultado
46          });
47      }
48  })
49  .catch((err) => {
50      res.status(500).json({
51          mensaje: `Error al realizar la búsqueda: ${err}`
52      });
53  });
54  };
55
56  // Crear una nueva mascota
57  const agregarNuevaMascota = (req, res) => {
58      if (!req.body.nombre) {
59          res.status(400).json({
60              mensaje: "El nombre no puede estar vacío."
61          });
62          return;
63      }
64
65      if (!req.body.tipo) {
```

```
app.js  mascotasController.js X  mascotasRouter.js
64
65   if (!req.body.tipo) {
66     res.status(400).json({
67       mensaje: "El tipo no puede estar vacío."
68     });
69     return;
70   }
71
72   const dataset = {
73     nombre: req.body.nombre,
74     tipo: req.body.tipo,
75     raza: req.body.raza,
76     edad: req.body.edad,
77     descripcion: req.body.descripcion,
78     estado: req.body.estado
79   };
80
81   Mascotas.create(dataset)
82     .then((resultado) => {
83       res.status(200).json({
84         mensaje: "Registro creado correctamente"
85       });
86     })
87     .catch((err) => {
88       res.status(500).json({
89         mensaje: `Error al crear el registro ::: ${err}`
90       });
91     });
92   };
93
94   // Actualizar información de una mascota por ID
95   const actualizarMascotaPorId = (req, res) => {
96     const mascotaId = req.params.id;
97   }
```

```
app.js  mascotasController.js X  mascotasRouter.js

96     const mascotaId = req.params.id;
97
98     if (!mascotaId || isNaN(mascotaId)) {
99         res.status(400).json({
100             mensaje: "No es válido el Id de mascota."
101         });
102         return;
103     }
104
105     Mascotas.findById(mascotaId)
106         .then((mascota) => {
107             if (!mascota) {
108                 res.status(404).json({
109                     mensaje: "No se encontró la mascota."
110                 });
111                 return;
112             } else {
113                 res.status(200).json({
114                     mensaje: "Actualización exitosa."
115                 });
116
117                 mascota.nombre = req.body.nombre || mascota.nombre;
118                 mascota.tipo = req.body.tipo || mascota.tipo;
119                 mascota.raza = req.body.raza || mascota.raza;
120                 mascota.edad = req.body.edad || mascota.edad;
121                 mascota.descripcion = req.body.descripcion || mascota.descripcion;
122                 mascota.estado = req.body.estado || mascota.estado;
123
124                 return mascota.save();
125             }
126         })
127         .catch((err) => {
128             res.status(500).json({
```

```

app.js      mascotasController.js X  mascotasRouter.js
127
128     .catch((err) => {
129         res.status(500).json({
130             mensaje: `Error al realizar la actualización: ${err}`
131         });
132     });
133
134 // Eliminar una mascota por ID
135 const eliminarMascotaPorId = (req, res) => {
136     const mascotaId = req.params.id;
137
138     if (!mascotaId || isNaN(mascotaId)) {
139         res.status(400).json({
140             mensaje: "No es válido el Id de mascota."
141         });
142         return;
143     }
144
145     Mascotas.destroy({
146         where: {
147             id: mascotaId
148         }
149     })
150     .then((resultado) => {
151         if (resultado === 0) {
152             res.status(404).json({
153                 mensaje: "No se encontró la mascota."
154             });
155         } else {
156             res.status(200).json({
157                 mensaje: "Eliminación exitosa."
158             });
159         }
160     })

```

```

app.js      mascotasController.js X  mascotasRouter.js
159
160     })
161     .catch((err) => {
162         res.status(500).json({
163             mensaje: `Error al realizar la eliminación: ${err}`
164         });
165     });
166 };
167
168 export {
169     obtenerTodasLasMascotas,
170     obtenerMascotaPorId,
171     agregarNuevaMascota,
172     actualizarMascotaPorId,
173     eliminarMascotaPorId
174 }
175

```

Rutas de mascotas

```
app.js mascotasController.js mascotasRouter.js X
1 import express from "express";
2 import {
3   obtenerTodasLasMascotas,
4   obtenerMascotaPorId,
5   agregarNuevaMascota,
6   actualizarMascotaPorId,
7   eliminarMascotaPorId
8 } from "../controladores/mascotasController.js";
9
10 const routerMascotas = express.Router();
11
12 routerMascotas.get("/", (req, res) => {
13   res.send("Bienvenido a Mascotas");
14 });
15
16 // Obtener todas las mascotas
17 routerMascotas.get("/buscar", (req, res) => {
18   obtenerTodasLasMascotas(req, res);
19 });
20
21 // Obtener mascota por ID
22 routerMascotas.get("/buscar/:id", (req, res) => {
23   obtenerMascotaPorId(req, res);
24 });
25
26 // Agregar nueva mascota
27 routerMascotas.post("/crear", (req, res) => {
28   agregarNuevaMascota(req, res);
29 });
30
31 // Actualizar mascota por ID
32 routerMascotas.put("/actualizar/:id", (req, res) => {
33   actualizarMascotaPorId(req, res);
34 });
```

```
app.js mascotasController.js mascotasRouter.js X
35 // Eliminar mascota por ID
36 routerMascotas.delete("/eliminar/:id", (req, res) => {
37   eliminarMascotaPorId(req, res);
38 });
39
40 export { routerMascotas }
```

Controlador de SolicitudesAdopcion

```

solicitudesAdopcionController.js x solicitudesAdopcionRouter.js
1 import { SolicitudesAdopcion } from "../modelos/solicitudesAdopcionModel.js"
2
3 // Obtener todas las solicitudes de adopción
4 const obtenerTodasLasSolicitudes = (req, res) => {
5     SolicitudesAdopcion.findAll()
6         .then((resultados) => {
7             if (resultados.length === 0) {
8                 res.status(404).json({
9                     mensaje: "No se encontraron solicitudes de adopción."
10                });
11             } else {
12                 res.status(200).json({
13                     mensaje: "Búsqueda exitosa",
14                     solicitudes: resultados
15                });
16             }
17         })
18         .catch((err) => {
19             res.status(500).json({
20                 mensaje: `Error al realizar la búsqueda: ${err}`
21             });
22         });
23 };
24
25 // Obtener información de una solicitud de adopción por ID
26 const obtenerSolicitudPorId = (req, res) => {
27     const solicitudId = req.params.id;
28
29     if (!solicitudId || isNaN(solicitudId)) {
30         res.status(400).json({
31             mensaje: "No es válido el Id de solicitud de adopción."
32         });
33         return;
34     }

```

```

solicitudesAdopcionController.js x solicitudesAdopcionRouter.js
33     return;
34 }
35
36 SolicitudesAdopcion.findByPk(solicitudId)
37     .then((resultado) => {
38         if (!resultado) {
39             res.status(404).json({
40                 mensaje: "No se encontró la solicitud de adopción."
41             });
42         } else {
43             res.status(200).json({
44                 mensaje: "Búsqueda exitosa",
45                 solicitud: resultado
46             });
47         }
48     })
49     .catch((err) => {
50         res.status(500).json({
51             mensaje: `Error al realizar la búsqueda: ${err}`
52         });
53     });
54 };
55
56 // Crear una nueva solicitud de adopción
57 const agregarNuevaSolicitud = (req, res) => {
58     if (!req.body.mascota_id) {
59         res.status(400).json({
60             mensaje: "El id de la mascota la cual se pide adoptar no puede estar vacío."
61         });
62         return;
63     }
64
65     if (!req.body.nombre_solicitante || !req.body.correo_solicitante) {

```



```
solicitudesAdopcionController.js X solicitudesAdopcionRouter.js
64
65     if (!req.body.nombre_solicitante || !req.body.correo_solicitante) {
66         res.status(400).json({
67             mensaje: "El nombre o el correo del solicitante no puede estar vacío."
68         });
69         return;
70     }
71
72     const dataset = {
73         mascota_id: req.body.mascota_id,
74         nombre_solicitante: req.body.nombre_solicitante,
75         correo_solicitante: req.body.correo_solicitante,
76         telefono_solicitante: req.body.telefono_solicitante,
77         estado: req.body.estado
78     };
79
80     SolicitudesAdopcion.create(dataset)
81         .then((resultado) => {
82             res.status(200).json({
83                 mensaje: "Solicitud de adopción creada correctamente"
84             });
85         })
86         .catch((err) => {
87             res.status(500).json({
88                 mensaje: `Error al crear la solicitud de adopción: ${err}`
89             });
90         });
91 };
92
93 // Actualizar información de una solicitud de adopción por ID
94 const actualizarSolicitudPorId = (req, res) => {
95     const solicitudId = req.params.id;
96
```

```
solicitudesAdopcionController.js X solicitudesAdopcionRouter.js
96
97     if (!solicitudId || isNaN(solicitudId)) {
98         res.status(400).json({
99             mensaje: "No es válido el Id de solicitud de adopción."
100         });
101         return;
102     }
103
104     SolicitudesAdopcion.findById(solicitudId)
105         .then((solicitud) => {
106             if (!solicitud) {
107                 res.status(404).json({
108                     mensaje: "No se encontró la solicitud de adopción."
109                 });
110                 return;
111             } else {
112                 res.status(200).json({
113                     mensaje: "Actualización exitosa."
114                 });
115
116                 solicitud.mascota_id = req.body.mascota_id || solicitud.mascota_id;
117                 solicitud.nombre_solicitante = req.body.nombre_solicitante || solicitud.nombre_solicitante;
118                 solicitud.correo_solicitante = req.body.correo_solicitante || solicitud.correo_solicitante;
119                 solicitud.telefono_solicitante = req.body.telefono_solicitante || solicitud.telefono_solicitante;
120                 solicitud.estado = req.body.estado || solicitud.estado;
121
122                 return solicitud.save();
123             }
124         })
125         .catch((err) => {
126             res.status(500).json({
127                 mensaje: `Error al realizar la actualización: ${err}`
128             });
129
```

```

127         mensaje: 'Error al realizar la actualizacion: ${err}'
128     });
129 });
130 };
131
132 // Eliminar una solicitud de adopción por ID
133 const eliminarSolicitudPorId = (req, res) => {
134     const solicitudId = req.params.id;
135
136     if (!solicitudId || isNaN(solicitudId)) {
137         res.status(400).json({
138             mensaje: "No es válido el Id de solicitud de adopción."
139         });
140         return;
141     }
142
143     SolicitudesAdopcion.destroy({
144         where: {
145             id: solicitudId
146         }
147     })
148     .then((resultado) => {
149         if (resultado === 0) {
150             res.status(404).json({
151                 mensaje: "No se encontró la solicitud de adopción."
152             });
153         } else {
154             res.status(200).json({
155                 mensaje: "Eliminación exitosa."
156             });
157         }
158     })
159     .catch((err) => {
160         res.status(500).json({

```

```

159         .catch((err) => {
160             res.status(500).json({
161                 mensaje: `Error al realizar la eliminación: ${err}`
162             });
163         });
164     });
165
166     export {
167         obtenerTodasLasSolicitudes,
168         obtenerSolicitudPorId,
169         agregarNuevaSolicitud,
170         actualizarSolicitudPorId,
171         eliminarSolicitudPorId
172     }

```

Rutas de SolicitudesAdopcion

```
solicitudesAdopcionController.js  solicitudesAdopcionRouter.js X
1  import express from "express";
2  import { obtenerTodasLasSolicitudes,
3          obtenerSolicitudPorId,
4          agregarNuevaSolicitud,
5          actualizarSolicitudPorId,
6          eliminarSolicitudPorId
7          } from "../controladores/solicitudesAdopcionController.js";
8
9  const routerSolicitudesAdopcion = express.Router();
10
11  routerSolicitudesAdopcion.get("/", (req, res) => {
12      res.send("Bienvenido a Solicitudes Adopción");
13  });
14
15  // Obtener todas las solicitudes
16  routerSolicitudesAdopcion.get("/buscar", (req, res) => {
17      obtenerTodasLasSolicitudes(req, res);
18  });
19
20  // Obtener solicitud por ID
21  routerSolicitudesAdopcion.get("/buscar/:id", (req, res) => {
22      obtenerSolicitudPorId(req, res);
23  });
24
25  // Agregar nueva solicitud
26  routerSolicitudesAdopcion.post("/crear", (req, res) => {
27      agregarNuevaSolicitud(req, res);
28  });
29
30  // Actualizar solicitud por ID
31  routerSolicitudesAdopcion.put("/actualizar/:id", (req, res) => {
32      actualizarSolicitudPorId(req, res);
33  });
```

```
solicitudesAdopcionController.js  solicitudesAdopcionRouter.js X
33  });
34
35  // Eliminar solicitud por ID
36  routerSolicitudesAdopcion.delete("/eliminar/:id", (req, res) => {
37      eliminarSolicitudPorId(req, res);
38  });
39
40  export { routerSolicitudesAdopcion }
```

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

Utilizando thunder client.

Tareas para manejar la tabla mascotas.

## Obtener todas las mascotas

GET

http://localhost:8000/mascotas/buscar

Send

Query

Headers<sup>2</sup>

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK

Size: 295 Bytes

Time: 9 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "mascotas": [
4     {
5       "id": 1,
6       "nombre": "Firulais",
7       "tipo": "Perro",
8       "raza": "Labrador",
9       "edad": 3,
10      "descripcion": "Juguetero y amigable",
11      "estado": "Disponible"
12    },
13    {
14      "id": 2,
15      "nombre": "Whiskers",
16      "tipo": "Gato",
17      "raza": "Siamés",
18      "edad": 2,
19      "descripcion": "Muy cariñoso",
20      "estado": "Disponible"
21    }
22  ]
23 }
```

## Obtener mascota por ID

GET

http://localhost:8000/mascotas/buscar/2

Send

Query

Headers<sup>2</sup>

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK

Size: 162 Bytes

Time: 50 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "mascota": {
4     "id": 2,
5     "nombre": "Whiskers",
6     "tipo": "Gato",
7     "raza": "Siamés",
8     "edad": 2,
9     "descripcion": "Muy cariñoso",
10    "estado": "Disponible"
11  }
12 }
```

## Agregar nueva mascota

POST

http://localhost:8000/mascotas/crear

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1 {
2   "nombre": "Kira",
3   "tipo": "Perro",
4   "raza": "Husky",
5   "edad": 7,
6   "descripcion": "Hiperactivo",
7   "estado": "Adoptado"
8 }
```

Status: 200 OK

Size: 43 Bytes

Time: 489 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

```
1 {
2   "mensaje": "Registro creado correctamente"
3 }
```

## Actualizar mascota por ID

PUT

http://localhost:8000/mascotas/actualizar/2

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

JSON

XML

Text

Form

Form-encode

GraphQL

Binary

JSON Content

Format

```
1 {
2   "estado": "Adoptado"
3 }
```

Status: 200 OK

Size: 37 Bytes

Time: 15 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

```
1 {
2   "mensaje": "Actualización exitosa."
3 }
```

Resultados antes de realizar la eliminación

GET http://localhost:8000/mascotas/buscar

Send

Query

Headers<sup>2</sup>

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK   Size: 405 Bytes   Time: 12 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

"mascotas": [

{

"id": 1,

"nombre": "Firulais",

"tipo": "Perro",

"raza": "Labrador",

"edad": 3,

"descripcion": "Juguetón y amigable",

"estado": "Disponible"

},

{

"id": 2,

"nombre": "Whiskers",

"tipo": "Gato",

"raza": "Siamés",

"edad": 2,

"descripcion": "Muy cariñoso",

"estado": "Adoptado"

},

{

"id": 3,

"nombre": "Kira",

"tipo": "Perro",

"raza": "Husky",

"edad": 7,

"descripcion": "Hiperactivo",

"estado": "Adoptado"

}

]

Eliminar mascota por ID

DELETE http://localhost:8000/mascotas/eliminar/3

Send

Query

Headers<sup>2</sup>

Auth

Body<sup>1</sup>

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK   Size: 35 Bytes   Time: 172 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

1

2

3

{

"mensaje": "Eliminación exitosa."

}

Resultados después de realizar la eliminación

GET http://localhost:8000/mascotas/buscar

Send

Query

Headers<sup>2</sup>

Auth

Body

Tests

Pre Run

Query Parameters

☐

parameter

value

Status: 200 OK   Size: 293 Bytes   Time: 9 ms

Response

Headers<sup>6</sup>

Cookies

Results

Docs

{}

≡

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

{

"mensaje": "Búsqueda exitosa",

"mascotas": [

{

"id": 1,

"nombre": "Firulais",

"tipo": "Perro",

"raza": "Labrador",

"edad": 3,

"descripcion": "Juguetón y amigable",

"estado": "Disponible"

},

{

"id": 2,

"nombre": "Whiskers",

"tipo": "Gato",

"raza": "Siamés",

"edad": 2,

"descripcion": "Muy cariñoso",

"estado": "Adoptado"

}

]

}

Tareas para manejar la tabla SolicitudesAdopcion

Obtener todas las solicitudes

GET  Send

Query Headers<sup>2</sup> Auth Body Tests Pre Run

Query Parameters

☐ parameter value

Status: 200 OK Size: 361 Bytes Time: 15 ms

Response Headers<sup>6</sup> Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "solicitudes": [
4     {
5       "id": 1,
6       "mascota_id": 1,
7       "nombre_solicitante": "Juan Pérez",
8       "correo_solicitante": "juan@gmail.com",
9       "telefono_solicitante": "123-456-7890",
10      "estado": "Pendiente"
11    },
12    {
13      "id": 2,
14      "mascota_id": 2,
15      "nombre_solicitante": "María Gómez",
16      "correo_solicitante": "maria@gmail.com",
17      "telefono_solicitante": "987-654-3210",
18      "estado": "Aprobada"
19    }
20  ]
21 }
```

## Obtener solicitud por ID

GET  Send

Query Headers<sup>2</sup> Auth Body Tests Pre Run

Query Parameters

☐ parameter value

Status: 200 OK Size: 199 Bytes Time: 15 ms

Response Headers<sup>6</sup> Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "solicitud": {
4     "id": 1,
5     "mascota_id": 1,
6     "nombre_solicitante": "Juan Pérez",
7     "correo_solicitante": "juan@gmail.com",
8     "telefono_solicitante": "123-456-7890",
9     "estado": "Pendiente"
10  }
11 }
```

## Agregar nueva solicitud

POST  Send

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "mascota_id": 4,
3   "nombre_solicitante": "Gabriel Benavides",
4   "correo_solicitante": "jhonattangbc@udenar.edu.co",
5   "telefono_solicitante": "3166815760",
6   "estado": "Aprobada"
7 }
```

Status: 200 OK Size: 57 Bytes Time: 134 ms

Response Headers<sup>6</sup> Cookies Results Docs {} ≡

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

## Actualizar solicitud por ID

PUT  Send

Query Headers<sup>2</sup> Auth **Body<sup>1</sup>** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "estado": "Rechazada"
3 }
```

Status: 200 OK Size: 37 Bytes Time: 16 ms

Response Headers<sup>6</sup> Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Actualización exitosa."
3 }
```

## Resultados antes de realizar la eliminación

GET  Send

Status: 200 OK Size: 532 Bytes Time: 10 ms

Query Headers 2 Auth Body Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value
------------------------------------	-------

Response Headers 6 Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "solicitudes": [
4     {
5       "id": 1,
6       "mascota_id": 1,
7       "nombre_solicitante": "Juan Pérez",
8       "correo_solicitante": "juan@gmail.com",
9       "telefono_solicitante": "123-456-7890",
10      "estado": "Rechazada"
11    },
12    {
13      "id": 2,
14      "mascota_id": 2,
15      "nombre_solicitante": "María Gómez",
16      "correo_solicitante": "maria@gmail.com",
17      "telefono_solicitante": "987-654-3210",
18      "estado": "Aprobada"
19    },
20    {
21      "id": 3,
22      "mascota_id": 4,
23      "nombre_solicitante": "Gabriel Benavides",
24      "correo_solicitante": "jhonattangbc@udenar.edu.co",
25      "telefono_solicitante": "3166815760",
26      "estado": "Aprobada"
27    }
28  ]
29 }
```

## Eliminar solicitud por ID

DELETE  Send

Status: 200 OK Size: 35 Bytes Time: 139 ms

Query Headers 2 Auth Body 1 Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value
------------------------------------	-------

Response Headers 6 Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Eliminación exitosa."
3 }
```

## Resultados después de realizar la eliminación

GET  Send

Status: 200 OK Size: 361 Bytes Time: 10 ms

Query Headers 2 Auth Body Tests Pre Run

Query Parameters

<input type="checkbox"/> parameter	value
------------------------------------	-------

Response Headers 6 Cookies Results Docs {} ≡

```
1 {
2   "mensaje": "Búsqueda exitosa",
3   "solicitudes": [
4     {
5       "id": 1,
6       "mascota_id": 1,
7       "nombre_solicitante": "Juan Pérez",
8       "correo_solicitante": "juan@gmail.com",
9       "telefono_solicitante": "123-456-7890",
10      "estado": "Rechazada"
11    },
12    {
13      "id": 2,
14      "mascota_id": 2,
15      "nombre_solicitante": "María Gómez",
16      "correo_solicitante": "maria@gmail.com",
17      "telefono_solicitante": "987-654-3210",
18      "estado": "Aprobada"
19    }
20  ]
21 }
```