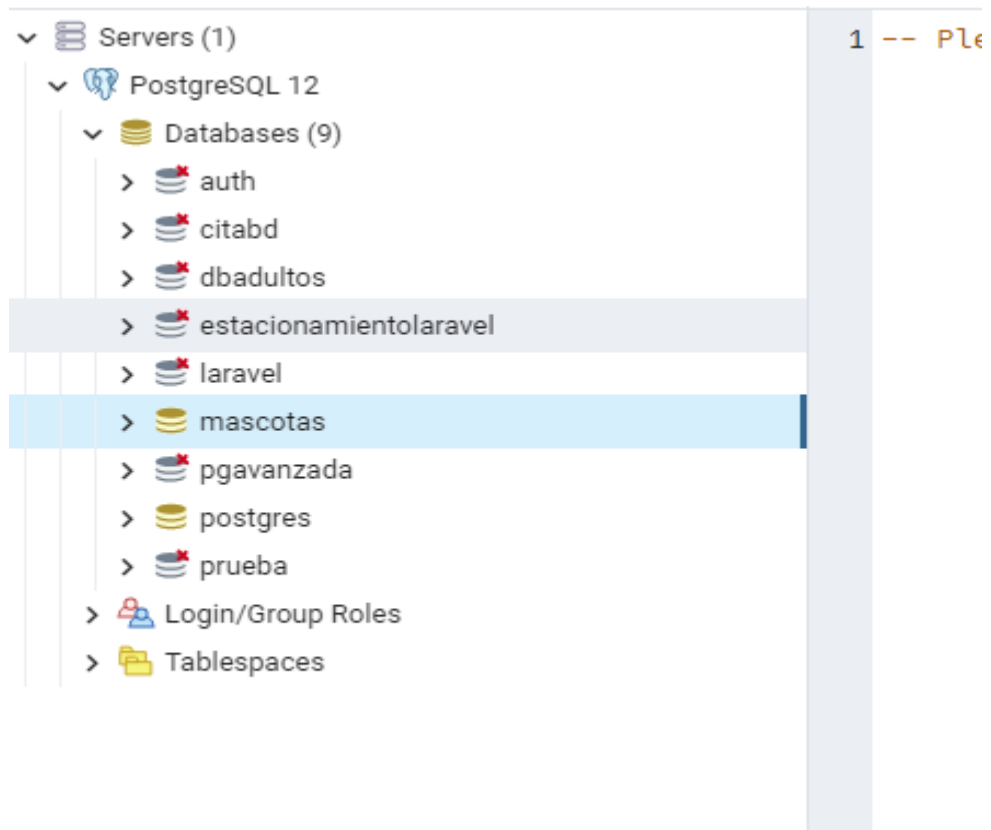


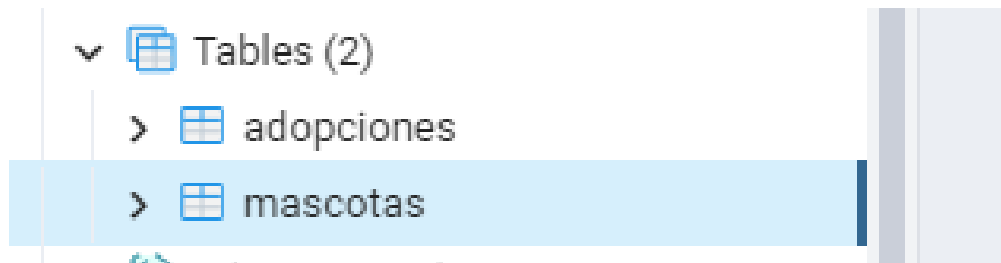
1. . Crear una base de datos POSTGRES que permita llevar el registro de mascotas (perros y gatos), así como también el proceso de solicitud de adopción de estas.

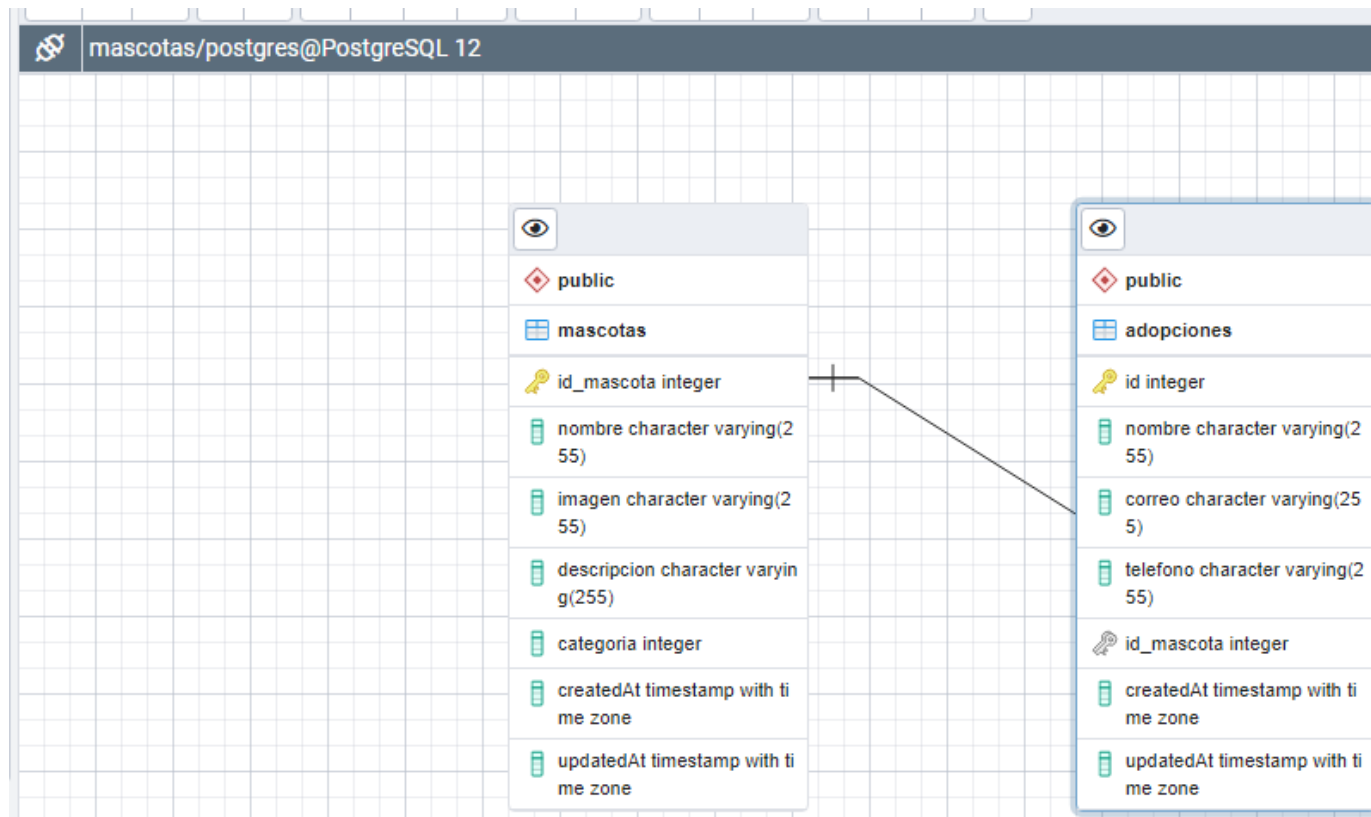
Creamos base de datos mascotas.



Creamos tablas.












- Adopciones
- mascotas





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)

Estructura de directorio de nuestro proyecto.

>  Controller	10
>  Database	104
>  Models	10
>  node_modules	10
>  Routes	10
>  storage	10
>  utils	11
 .env	11
 index.js	11
 package.json	11
 yarn.lock	11

## CONEXIÓN BASES DE DATOS

```
Database > Database.js > ...
1  import Sequelize from 'sequelize';
2
3  const sequelize = new Sequelize('mascotas', 'postgres', 'postgres', {
4    host: 'localhost',
5    dialect: 'postgres'
6  });
7
8  export {
9    sequelize
10 }
```

Crear servidor Express.

```
index.js > ...
1  import express from 'express';
2  import cors from 'cors';
3
4  import {routerCliente,routerMascota,routerAdopcion} from './Routes/index.js';
5
6  import {sequelize} from './Database/Database.js';
7
8  const testdb = async() => {
9    try {
10      await sequelize.authenticate();
11
12      // await Mascota.sync({force:true})
13      // await Cliente.sync({force:true})
14      // await Adapcion.sync({force:true})
15      console.log('Connection has been established successfully.');
```

## RUTAS UTILIZADAS

### Rutas Mascotas:

```
Routes >  mascotas.js > ...
1  import {Router} from 'express';
2
3  import {getMascotas,getMascotasById,getMascotasByCategoria,postMascota,deleteMascotas,putMascotas} from '../Controller/Macotas.js';
4  import { cargarImagenMiddleware } from '../utils/cargarImagen.js';
5
6  export const routerMascota = Router();
7
8  routerMascota.post('/mascotas',cargarImagenMiddleware.single("imagen"),postMascota)
9
10 routerMascota.get('/mascotas',getMascotas)
11
12 routerMascota.get('/mascotas/:id',getMascotasById)
13 routerMascota.get('/mascotasCategoria/:id',getMascotasByCategoria)
14
15 routerMascota.put('/mascotas/:id',putMascotas)
16
17 routerMascota.delete('/mascotas/:id',deleteMascotas)
18
19
20
21
```

### Rutas Adopción:

```
Routes > adopciones.js > ...
1  import {Router} from 'express';
2
3
4  import {getAdopciones,postAdopcion,putAdopciones,deleteAdopciones,getAll} from '../Controller/Adopciones.js';
5  import { cargarImagenMiddleware } from '../utils/cargarImagen.js';
6
7
8  export const routerAdopcion = Router();
9
10
11 routerAdopcion.post('/adopciones',cargarImagenMiddleware.single("imagen"),postAdopcion)
12
13 routerAdopcion.get('/adopcionesAll',getAll)
14
15 routerAdopcion.get('/adopciones',getAdopciones)
16
17 routerAdopcion.put('/adopciones/:id',putAdopciones)
18
19 routerAdopcion.delete('/adopciones/:id',deleteAdopciones)
20
21
```




## MODELS



Modelo Mascota.

```
Models >  Mascotas.js >  Mascota
1  import { DataTypes } from 'sequelize';
2  import { sequelize } from '../Database/Database.js';
3
4  export const Mascota = sequelize.define('mascotas', {
5
6      id_mascota: {
7          type: DataTypes.INTEGER,
8          primaryKey: true
9      },
10     nombre: {
11         type: DataTypes.STRING
12     },
13 },
14     {
15         imagen: {
16             type: DataTypes.STRING
17         },
18         descripcion: {
19             type: DataTypes.STRING
20         },
21         categoria: {
22             type: DataTypes.INTEGER
23         }
24     })
25
26
27
```


## Modelo Adopción

```
Models >  Adapciones.js >  Adapcion >  correo
1  import { DataTypes } from 'sequelize';
2  import { sequelize } from '../Database/Database.js';
3
4
5  export const Adapcion = sequelize.define('adopciones', {
6    id: {
7      type: DataTypes.INTEGER,
8      primaryKey: true,
9      autoIncrement: true
10   },
11   nombre: {
12     type: DataTypes.STRING,
13   },
14   correo: {
15     type: DataTypes.STRING,
16   },
17 },
18   telefono: {
19     type: DataTypes.STRING,
20   },
21 },
22
23   id_mascota: {
24     type: DataTypes.INTEGER,
25     references: {
26       key: "id_mascota",
27       model: "mascotas"
28     }
29   }
30 },
31 {});
32
33
34
```

## Controller Mascota

```
Controller >  Macotas.js > ...
  1  import { Mascota } from '../Models/Mascotas.js';
  2  const publicUrl = process.env.PUBLIC_URL;
  3
  4  > const getMascotas = async (req, res) => { ...
17  }
18
19  > const getMascotasByCategoria = async (req, res) => { ...
38  }
39
40  > const getMascotasById = async (req, res) => { ...
59  }
60
61  > const postMascota = async (req, res) => { ...
84  }
85
86  > const deleteMascotas = async (req, res) => { ...
108  }
109  > const putMascotas = async (req, res) => { ...
133  }
134  export {
135      getMascotas,
136      getMascotasById,
137      getMascotasByCategoria,
138      postMascota,
139      deleteMascotas,
140      putMascotas
141  }
```

getMascotas

Controller >  Macotas.js > ...

```
1  import { Mascota } from '../Models/Mascotas.js';
2  const publicUrl = process.env.PUBLIC_URL;
3
4  const getMascotas = async (req, res) => {
5    try {
6
7      const response = await Mascota.findAll();
8
9      res.status(200).json(response)
10
11    } catch (error) {
12
13      res.status(404).json(`mensaje: ${error}`)
14      console.log('error')
15
16    }
17  }
18
```

Utilidad con librería multer para imágenes

```
utils > JS cargarImagen.js > ...
1
2 import multer from 'multer';
3
4 const storage = multer.diskStorage({
5
6
7   destination: (req, file, cb) => {
8     console.log(req)
9     const pathStorage = `${process.cwd()}/storage`
10    console.log('multer')
11    cb(null, pathStorage)
12  },
13  filename: (req, file, cb) => {
14    const {id_mascota} = req.body
15    const ext = file.originalname.split(".").pop();
16    const filename = `file-${Date.now()}.${ext}`
17    cb(null, filename)
18  }
19 })
20
21
22 export const cargarImagenMiddleware = multer({storage})
23
24
```

Implementación de la utilidad cargarImagenMiddleware

```
routerMascota.post('/mascotas', cargarImagenMiddleware.single("imagen"), postMascota)
```

Insertar url imagen en el método POST

```
const postMascota = async (req, res) => {
  try {
    const {file} = req

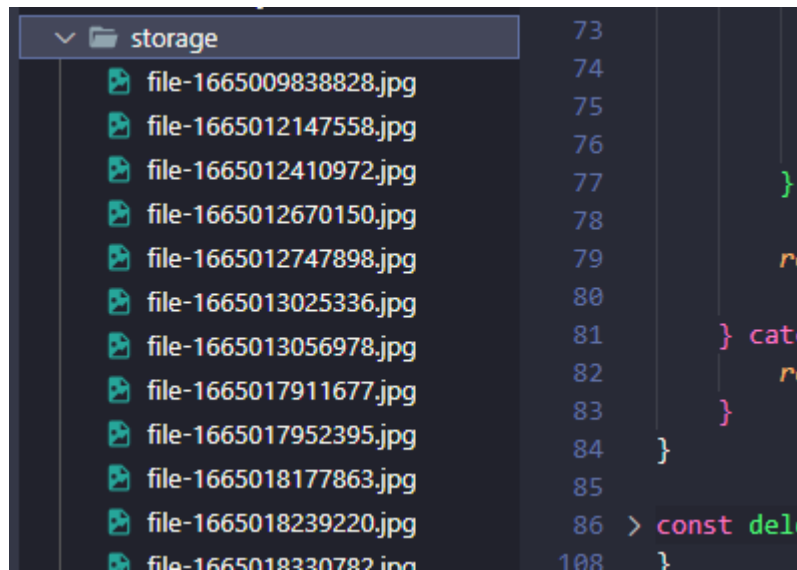
    const imagen = `http://localhost:5000/${file.filename}`

    const { id_mascota, nombre, descripcion, categoria } = req.body


    const newMascota = await Mascota.create({
      id_mascota,
      nombre,
      descripcion,
      categoria,
      imagen
    })

    res.status(200).json(newMascota)
  } catch (error) {
    res.status(404).json(`mensaje: ${error}`)
  }
}
```

Directorio para guardar las imágenes.



CONTROLLERS ADOPCIÓN

```
Controller >  Adopciones.js > ...
1  import {Adapcion,Mascota} from '../Models/index.js';
2
3
4  > const getAdapciones = async(req,res) => { ...
17  }
18
19  > const getAll = async(req,res) => { ...
40  }
41
42  |
43  > const postAdapcion = async( req,res) => { ...
58  }
59
60  > const deleteAdapciones = async(req,res) => { ...
82  }
83  > const putAdapciones = async(req,res) => { ...
105  }
106  export {
107      getAdapciones,
108      postAdapcion,
109      deleteAdapciones,
110      putAdapciones,
111      getAll
112  }
```

getAdopciones



Controller > JS Adopciones.js > ...

```
4  const getAdapciones = async(req,res) => {
5      try {
6
7          const response =await Adapcion.findAll();
8
9          res.status(200).json(response)
10
11      } catch (error) {
12
13          res.status(404).json(`mensaje:${error}`)
14          console.log('error')
15
16      }
17  }
18
19  const getAll = async(req,res) => {
20      try {
21
22          const response =await Adapcion.findAll(
23              {
24                  include: [{
25                      model: Mascota,
26
27                  }
28              ]
29          }
30      );
31
32      res.status(200).json(response)
33
34      } catch (error) {
35
36          res.status(404).json(`mensaje:${error}`)
37          console.log('error')
```