

Taller Unidad 3 Frontend

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

Ingeniería de Sistemas.

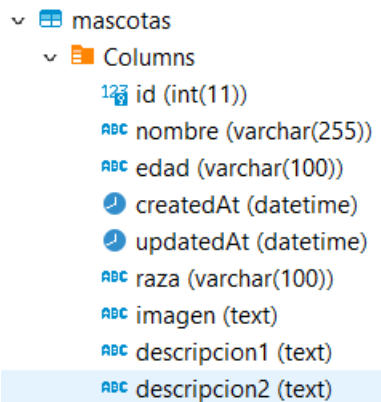
Universidad de Nariño.

Darwin Estiven Diaz Espinosa

Código: 219036069

Modificaciones en la base de datos:

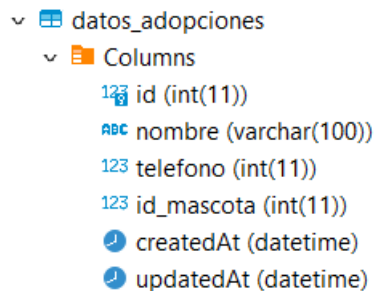
Se creó los campos que faltan en la tabla **mascotas** y quedaría así:



A screenshot of a database schema editor. The 'mascotas' table is expanded, showing its columns. The columns are: id (int(11)), nombre (varchar(255)), edad (varchar(100)), createdAt (datetime), updatedAt (datetime), raza (varchar(100)), imagen (text), descripcion1 (text), and descripcion2 (text). The 'descripcion2' column is highlighted with a blue background.

123	id (int(11))
ABC	nombre (varchar(255))
ABC	edad (varchar(100))
🕒	createdAt (datetime)
🕒	updatedAt (datetime)
ABC	raza (varchar(100))
ABC	imagen (text)
ABC	descripcion1 (text)
ABC	descripcion2 (text)

Se creó también la tabla **datos_adopcion** con sus respectivos atributos, la cual nos servirá para guardar los datos de las personas que vayan a adoptar cada mascota, ésta quedó así:



A screenshot of a database schema editor. The 'datos_adopciones' table is expanded, showing its columns. The columns are: id (int(11)), nombre (varchar(100)), telefono (int(11)), id_mascota (int(11)), createdAt (datetime), and updatedAt (datetime).

123	id (int(11))
ABC	nombre (varchar(100))
123	telefono (int(11))
123	id_mascota (int(11))
🕒	createdAt (datetime)
🕒	updatedAt (datetime)

Modificaciones en el BackEnd:

En el controlador **mascotasController.js**:

Importamos el modelo **adopcionesMascotas** en el cual se tiene la estructura de la tabla **datos_adopciones**, mas adelante del informe se verá explicado.

```
import { adopcionesMascotas } from "../modelos/adopcionesModelo.js";
```

Agregamos los campos que creamos nuevos en cada función para crear, validar y eliminar:

```
// Crear un nuevo recurso
const crear = (req, res) => {
  // Validar si el campo 'nombre' está proporcionado
  if (!req.body.nombre) {
    res.status(400).json({
      mensaje: "El nombre no puede estar vacío."
    });
    return;
  }

  // Preparar el conjunto de datos con la información del cuerpo de la solicitud
  const dataset = {
    nombre: req.body.nombre,
    edad: req.body.edad,
    imagen: req.body.imagen,
    raza: req.body.raza,
    descripcion1: req.body.descripcion1,
    descripcion2: req.body.descripcion2
  };

  // Utilizar Sequelize para crear el recurso en la tabla 'mascotas'
  mascotas.create(dataset)
    .then((resultado) => {
      res.status(200).json({
        tipo: "success",
        mensaje: "Registro creado correctamente"
      });
    })
    .catch((err) => {
      res.status(500).json({
        tipo: "error",
        mensaje: `Error al crear el registro ::: ${err}`
      });
    });
};
```

Así mismo en cada una de las funciones.

También se creó una función nueva llamada **crearAdopcion()** la cual es utilizada para guardar los registros de los datos de los usuarios que van a adoptar las mascotas en la base de datos, ésta quedó así:

```

// Crear adopción
const crearAdopcion = (req, res) => {
  // Validar si los campos 'nombre' y 'telefono' están proporcionados
  if (!req.body.nombre || !req.body.telefono) {
    res.status(400).json({
      mensaje: "El nombre o el teléfono no pueden estar vacíos."
    });
    return;
  }

  // Preparar el conjunto de datos con la información del cuerpo de la solicitud
  const dataset = {
    id_mascota: req.body.id_mascota,
    nombre: req.body.nombre,
    telefono: req.body.telefono,
  };

  // Utilizar Sequelize para crear el recurso en la tabla 'adopcionesMascotas'
  adopcionesMascotas.create(dataset)
    .then((resultado) => {
      res.status(200).json({
        tipo: "success",
        mensaje: "Registro creado correctamente"
      });
    })
    .catch((err) => {
      res.status(500).json({
        tipo: "error",
        mensaje: `Error al crear el registro ::: ${err}`
      });
    });
};

```

Esta función se encarga de crear un nuevo registro de adopción en la base de datos utilizando los datos proporcionados en la solicitud. Realiza validaciones para asegurarse de que los campos esenciales estén presentes y maneja adecuadamente los casos de éxito y error durante el proceso de creación.

Exportamos junto con las demás funciones la función **crearAdopcion**:

```

// Exportar las funciones del controlador
export { crear, buscarId, buscar, actualizar, eliminar, crearAdopcion };

```

Este es el nuevo modelo creado **adopcionesModelo**:

```
// Importar Sequelize y la instancia de la base de datos desde el archivo de conexión
import Sequelize from "sequelize";
import { db } from "../database/conexion.js";

// Definir el modelo 'adopcionesMascotas'
const adopcionesMascotas = db.define("datos_adopciones", {
  // Columna 'id'
  id: {
    type: Sequelize.INTEGER,
    primaryKey: true,
    autoIncrement: true,
    allowNull: false,
  },
  // Columna 'nombre'
  nombre: {
    type: Sequelize.STRING,
    allowNull: true
  },
  // Columna 'telefono'
  telefono: {
    type: Sequelize.INTEGER,
    allowNull: true
  },
  // Columna 'id_mascota'
  id_mascota: {
    type: Sequelize.INTEGER,
    allowNull: true
  },
});

// Exportar el modelo 'adopcionesMascotas'
export { adopcionesMascotas };
```

Este código define el modelo **adopcionesMascotas**, que representa la estructura de la tabla "datos_adopciones" en la base de datos. Cada propiedad del modelo corresponde a una columna en la tabla, especificando su tipo de datos, si permite nulos, etc. Este modelo se utilizará para interactuar con los datos de adopciones de mascotas en la base de datos.

Rutas:

Se creó una nueva ruta para lo relacionado con las adopciones como se ve a continuación:

```
// Importar express para crear el enrutador
import express from "express";
// Se importan las funciones controladoras desde el archivo mascotasController.js.
import { crear, buscarId, buscar, actualizar, eliminar, crearAdopcion } from "../controladores/mascotasController.js";

// Crear un enrutador de express
const routerMascotas = express.Router();

// Ruta de bienvenida para la API de mascotas
routerMascotas.get("/", (req, res) => {
  res.send("Bienvenido a Mascotas");
});

// Ruta para crear una nueva mascota
routerMascotas.post("/crear", (req, res) => {
  crear(req, res);
});

// Ruta para crear una nueva adopción de mascota
routerMascotas.post("/adopcion/crear", (req, res) => {
  crearAdopcion(req, res);
});

// Ruta para buscar una mascota por ID
routerMascotas.get("/buscar/:id", (req, res) => {
  buscarId(req, res);
});

// Ruta para buscar todas las mascotas
routerMascotas.get("/buscar", (req, res) => {
  buscar(req, res);
});

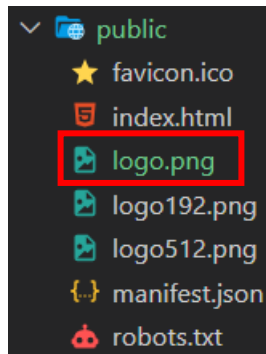
// Ruta para actualizar una mascota por ID
routerMascotas.put("/actualizar/:id", (req, res) => {
  actualizar(req, res);
});

// Ruta para eliminar una mascota por ID
routerMascotas.delete("/eliminar/:id", (req, res) => {
  eliminar(req, res);
});

// Exportar el enrutador para su uso en otros archivos
export { routerMascotas };
```

Modificaciones en el FrontEnd:

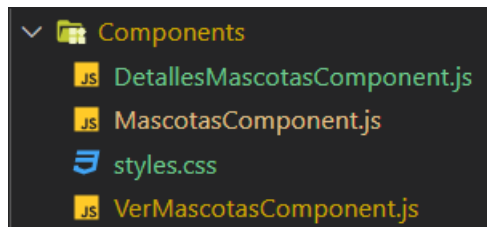
Se agrego una imagen para el logo en la carpeta public:



Esta imagen:



Se tienen 3 componentes y un archivo .css para los estilos de los botones de la paginación:



NOTA: DE LOS COMPONENTES (MascotasComponent.js y VerMascotasComponent.js) NO PEGO CAPTURA YA QUE CONTIENEN MUCHAS LINEAS DE CÓDIGO, PERO EN EL PROYECTO ESTÁN COMENTADAS Y EXPLICADAS DETALLADAMENTE LAS LINEAS DE CÓDIGO, AL IGUALMENTE QUE TODOS LOS DEMAS ARCHIVOS.

Detalles MascotasComponent.js:

```
// IMPORTACIONES
import React, { useEffect, useState } from "react";
import { useParams } from "react-router-dom"; // Hook para obtener parámetros de la URL
import axios from "axios"; // Librería para realizar solicitudes HTTP
import { Link } from "react-router-dom"; // Componente para navegar entre rutas
import './styles.css'; // Estilos del componente

// CUERPO DEL COMPONENTE
const VerMascotasComponent = () => {
  // Obtiene el parámetro de la URL usando el hook useParams de React Router
  const { id } = useParams();

  const url = "http://localhost:8000/mascotas";
  const [mascotas, setMascotas] = useState([null]); // Estado para almacenar la información de la mascota

  // Efecto de montaje: se ejecuta al cargar el componente o cuando el parámetro id cambia
  useEffect(() => {
    getMascotaById(id); // Llama a la función para obtener la información de la mascota por su ID
  }, [id]);

  // Función asincrónica para obtener la información de la mascota por su ID
  const getMascotaById = async (id) => {
    try {
      const respuesta = await axios.get(`${url}/buscar/${id}`);
      console.log(respuesta.data);
      setMascotas(respuesta.data); // Actualiza el estado con la información de la mascota
    } catch (error) {
      console.error("Error al obtener la mascota:", error);
    }
  };
};
```

```

// RENDERIZACIÓN DEL COMPONENTE
return (
  <div className="App">
    <br />
    {/} Sección para mostrar el logo o imagen de la página {/}
    <div>
      
    </div>
    <br />

    {/} Verifica si hay información de mascotas antes de renderizar {/}
    {mascotas && (
      <div className="container">
        <br />
        {/} Sección para mostrar los detalles de la mascota {/}
        <div>
          <h2>{mascotas.nombre} ({mascotas.raza})</h2>
          <p style={{ textAlign: 'justify' }}>{mascotas.descripcion2}</p>
        </div>

        {/} Botón para regresar a la página principal {/}
        <div className="d-flex justify-content-between mt-3">
          <Link to="/" className="btn btn-dark ms-auto">
            <i className="fas fa-reply"></i> Regresar
          </Link>
        </div>
        <br />
      </div>
    )}
  </div>
);
};

// EXPORTACIÓN DEL COMPONENTE
export default VerMascotasComponent;

```


App.js

```
// Importaciones necesarias para el enrutamiento
import { BrowserRouter, Routes, Route } from 'react-router-dom';

// Importaciones de componentes creados
import MascotasComponent from './Components/MascotasComponent.js';
import VerMascotasComponent from './Components/VerMascotasComponent.js';
import DetallesMascotasComponent from './Components/DetallesMascotasComponent.js';

// Estilos de la aplicación
import './App.css';

// Función principal que define la estructura de las rutas
function App() {
  return (
    <BrowserRouter>
      <Routes>
        {/* Ruta para mostrar la lista de mascotas */}
        <Route path="/" element={<VerMascotasComponent></VerMascotasComponent>}></Route>

        {/* Ruta para crear o editar mascotas */}
        <Route path="/crear" element={<MascotasComponent></MascotasComponent>}></Route>


        {/* Ruta para ver detalles de una mascota específica */}
        <Route path="/detalles/:id" element={<DetallesMascotasComponent></DetallesMascotasComponent>}></Route>
      </Routes>
    </BrowserRouter>
  );
}


// Exporta la función principal para ser utilizada en otros archivos
export default App;
```

NOTA: EN EL PROYECTO TODO EL CÓDIGO ESTA COMENTADO Y EXPLICADO DETALLADAMENTE.

Capturas de la ejecución:


Pantalla principal con la paginación funcionando correctamente:






Toby
Edad: 11 meses
Raza: Lobo Siberiano
Perro grande, juguetón y amoroso.

[i Detalles](#) [Adoptar](#)



Nina
Edad: 1 año
Raza: Pincher
Perra pequeña


[i Detalles](#) [Adoptar](#)




Luna
Edad: 3 años
Raza: Criolla
Cachorra juguetona

[i Detalles](#) [Adoptar](#)


[Anterior](#) [1](#) [2](#) [Siguiente](#)





Rambo
Edad: 10 meses
Raza: American bully
Perro de tamaño estándar.

[i Detalles](#) [Adoptar](#)



Romeo
Edad: 15 meses
Raza: Pug
Perro bajo y de aspecto cuadrado.

[i Detalles](#) [Adoptar](#)

[Anterior](#) [1](#) [2](#) [Siguiente](#)

Aquí se pueden ver los datos en la base de datos:

mascotas <small>Enter a SQL expression to filter results (use Ctrl+Space)</small>									
			createdAt	updatedAt	raza	imagen	descripcion1	descripcion2	
1	10	Toby	11	2023-12-29 20:26:23	2023-12-30 17:49:04	Lobo Siberiano	https://upload.wikimedia.org/wiki	Perro grande, juguetón y amoroso.	El perro lobo checoslovaco (tambi
2	11	Nina	1	2023-12-29 20:38:23	2023-12-30 18:00:26	Pincher	https://images.hola.com/imagen	Perra pequeña	El pinscher miniatura es una raza c
3	13	Luna	3	2023-12-30 02:19:58	2023-12-30 18:01:48	Criolla	https://www.adopta.mx/wp-conte	Cachorra juguetona	Se conoce como perro criollo o m
4	18	Rambo	10	2023-12-30 17:23:34	2024-01-03 03:08:35	American bully	https://static.wixstatic.com/media	Perro de tamaño estándar.	American Bully es una raza de per
5	22	Romeo	15	2024-01-03 03:06:49	2024-01-03 03:08:30	Pug	https://cdn.wamiz.fr/cdn-cgi/ima	Perro bajo y de aspecto cuadrado.	El pug1 (también conocido como

Captura de la pagina de detalles, esta página se abre al momento de darle clic en el botón **Detalles**.



Toby (Lobo Siberiano)

El perro lobo checoslovaco (también referido como PLC) es una raza canina relativamente nueva, y cuyo linaje original se remonta a un experimento llevado a cabo en 1955 en Checoslovaquia. Después de comenzar la gestación del linaje de los 48 ejemplares de pastor alemán con cuatro lobos europeos, se elaboró un plan para crear un híbrido que tuviera el temperamento, la mentalidad y la capacidad de entrenamiento del pastor alemán, junto con la fuerza, la constitución física y la resistencia de los lobos. Tienen una apariencia muy similar a los lobos de los Cárpatos.

Regresar



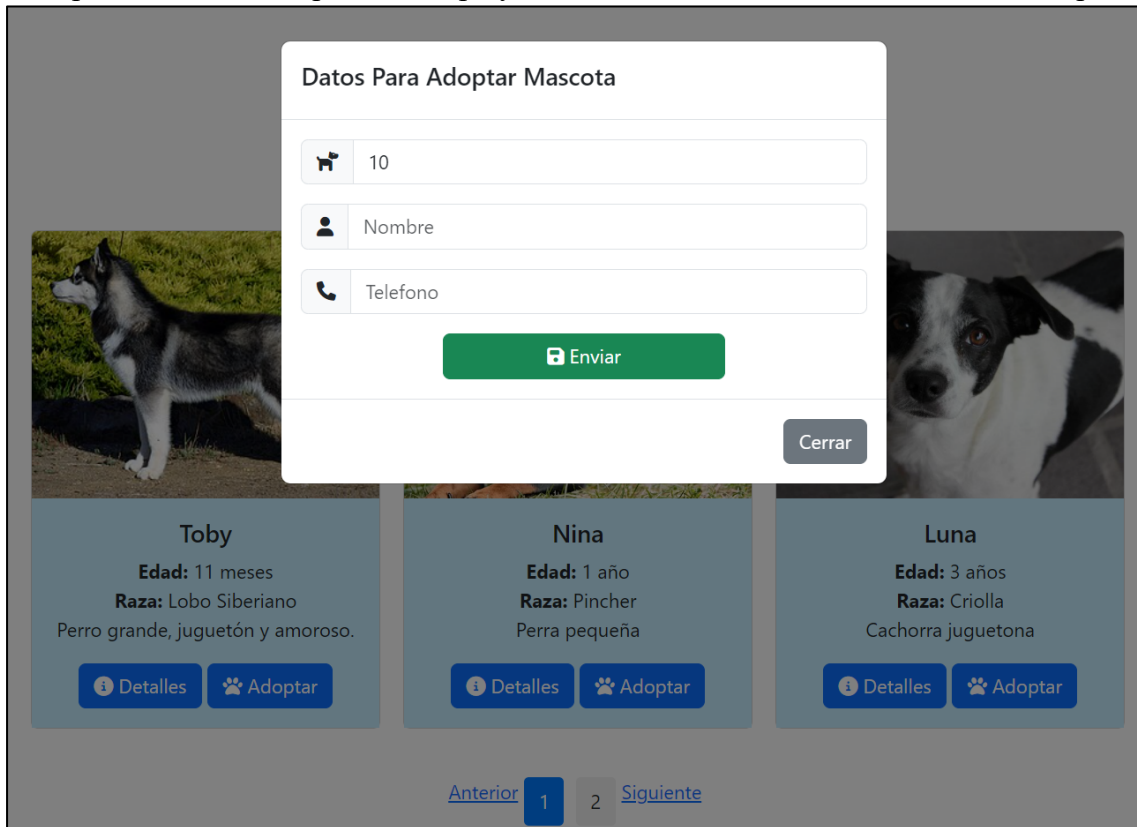
Rambo (American bully)

American Bully es una raza de perro tipo Terrier de tamaño estándar originaria de Estados Unidos.² No se utilizaban como perros de pelea aún cuando estos eventos fueron prohibidos en 1976; actualmente son criados como perros de compañía. Es reconocida principalmente por el United Kennel Club (desde 2013), Club Híbrido Canino Americano y por el ABKC (American Bully Kennel Club), en Estados Unidos, y por Alianz Canine y la CRCPE (Conservación de las razas caninas puras en España), en España, ambas entidades reconocidas oficialmente por el Ministerio de Agricultura, Ganadería y Medio Ambiente de España.

Regresar

Al darle en el botón **Regresar**, vuelve a la pantalla principal.

Al darle en el botón **Adoptar** se despliega el formulario para llenar los datos de la persona que va a adoptar la mascota, el primer campo ya trae el id de la mascota de la cual va a adoptar:



This screenshot shows a modal titled "Datos Para Adoptar Mascota" (Data for Adopting Pet) overlaid on a list of three dogs: Toby, Nina, and Luna. The modal contains three input fields: the first is pre-filled with the number "10" and has a dog icon; the second is labeled "Nombre" (Name) with a person icon; the third is labeled "Telefono" (Phone) with a phone icon. Below the fields is a green "Enviar" (Send) button and a grey "Cerrar" (Close) button. The background shows the first dog, Toby, a Siberian Husky, with details: "Edad: 11 meses", "Raza: Lobo Siberiano", and "Perro grande, juguetón y amoroso." The other two dogs, Nina and Luna, are partially visible.

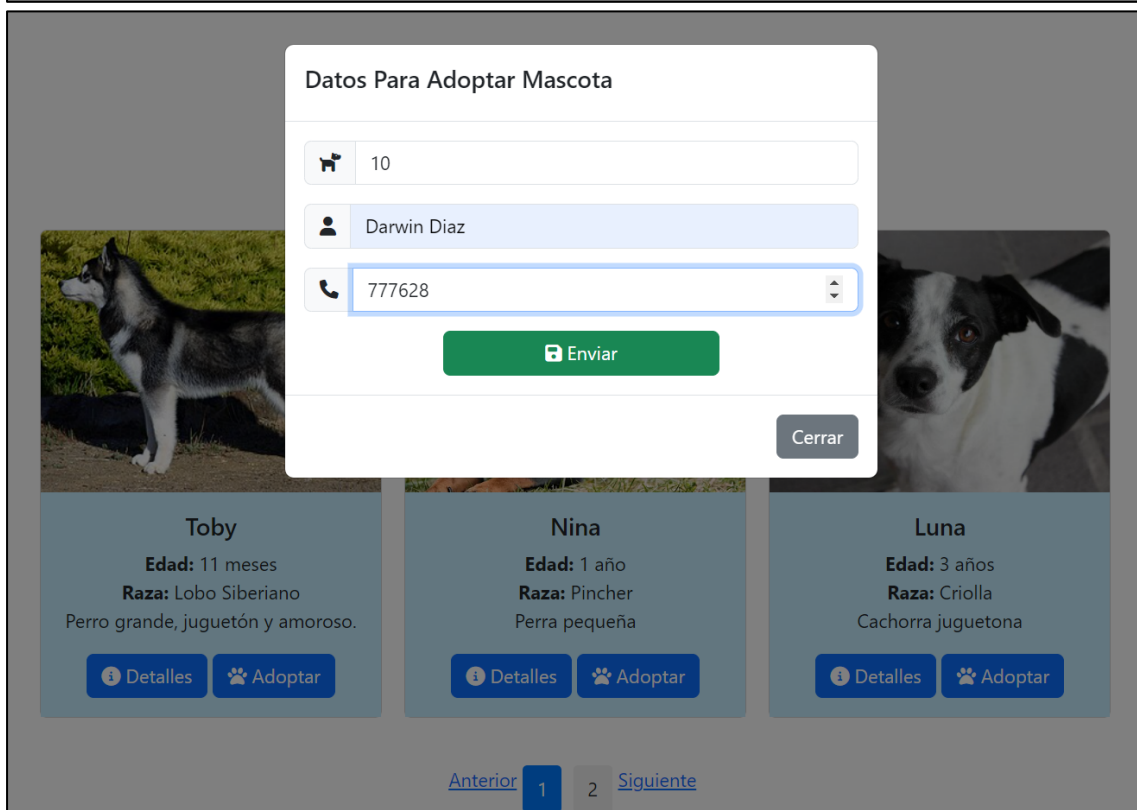
Datos Para Adoptar Mascota

Toby
Edad: 11 meses
Raza: Lobo Siberiano
Perro grande, juguetón y amoroso.

Nina
Edad: 1 año
Raza: Pincher
Perra pequeña

Luna
Edad: 3 años
Raza: Criolla
Cachorra juguetona

[Anterior](#) **1** [2](#) [Siguiete](#)



This screenshot shows the same "Datos Para Adoptar Mascota" modal, but now the fields are filled with data. The first field remains "10". The second field, "Nombre", now contains "Darwin Diaz". The third field, "Telefono", now contains "777628". The "Enviar" button is still green, and the "Cerrar" button is still grey. The background content, including the dog cards and pagination, remains the same as in the previous screenshot.

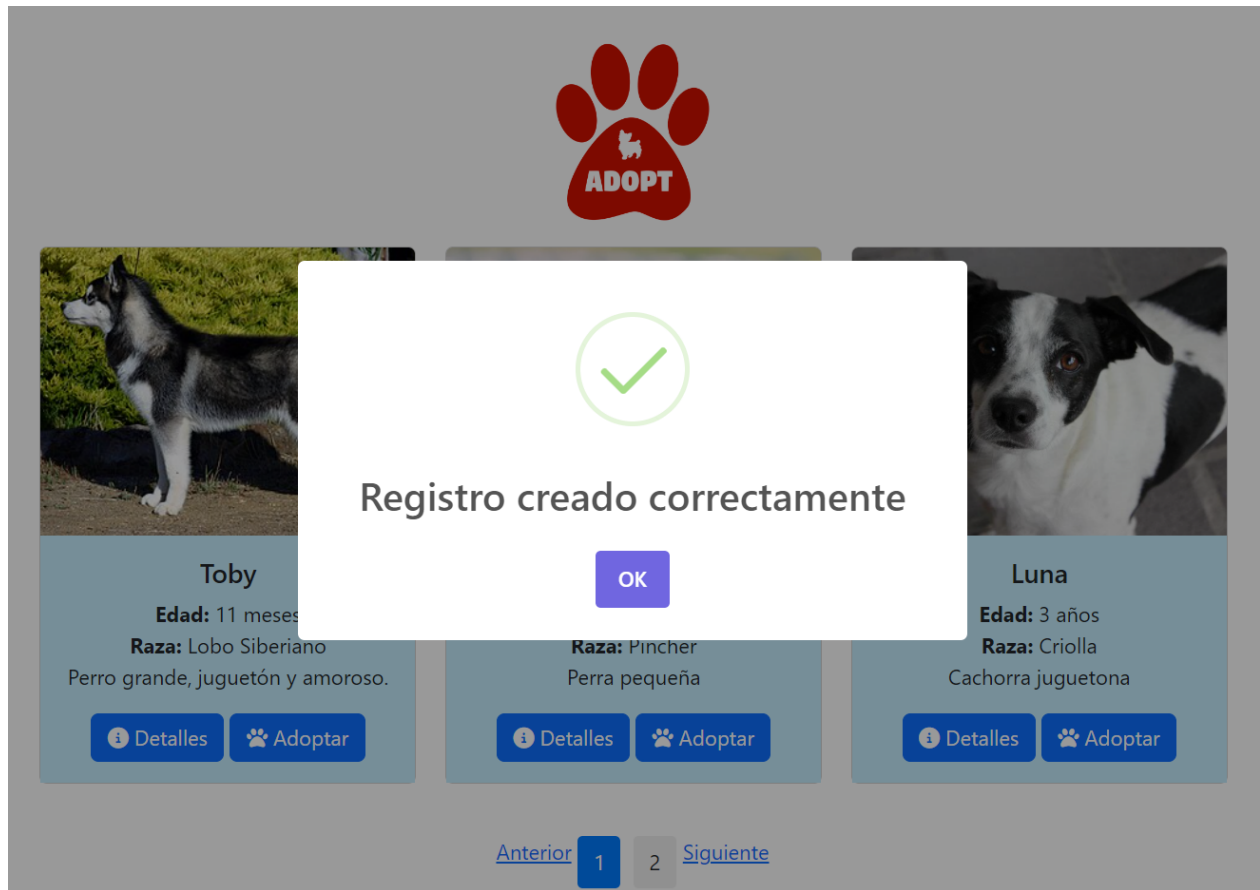
Datos Para Adoptar Mascota

Toby
Edad: 11 meses
Raza: Lobo Siberiano
Perro grande, juguetón y amoroso.

Nina
Edad: 1 año
Raza: Pincher
Perra pequeña

Luna
Edad: 3 años
Raza: Criolla
Cachorra juguetona

[Anterior](#) **1** [2](#) [Siguiete](#)



Aquí se puede ver los registros en la tabla de adopciones:

datos_adopciones <small>Enter a SQL expression to filter results (use Ctrl+Space)</small>						
Grilla	id	nombre	telefono	id_mascota	createdAt	updatedAt
1	7	Sandra	7.256.036	13	2024-01-02 19:55:13	2024-01-02 19:55:13
2	9	Darwin Diaz	777.628	10	2024-01-03 03:32:28	2024-01-03 03:32:28