



Universidad de **Nariño**

TALLER FRONTEND.

ELABORADO POR:

CRISTIAN FERNANDO JACOME RECALDE

UNIVERSIDAD DE NARIÑO - SEDE IPIALES

FACULTAD DE INGENIERIA DE SISTEMAS

X SEMESTRE

**DIPLOMADO DE ACTUALIZACIÓN EN NUEVAS TECNOLOGÍAS PARA EL
DESARROLLO DE SOFTWARE.**

IPIALES – NARIÑO 2024



Universidad de **Nariño**

TALLER FRONTEND.

ELABORADO POR:

CRISTIAN FERNANDO JACOME RECALDE

PRESENTADO A:

MG. VICENTE AUX REVELO

UNIVERSIDAD DE NARIÑO - SEDE IPIALES

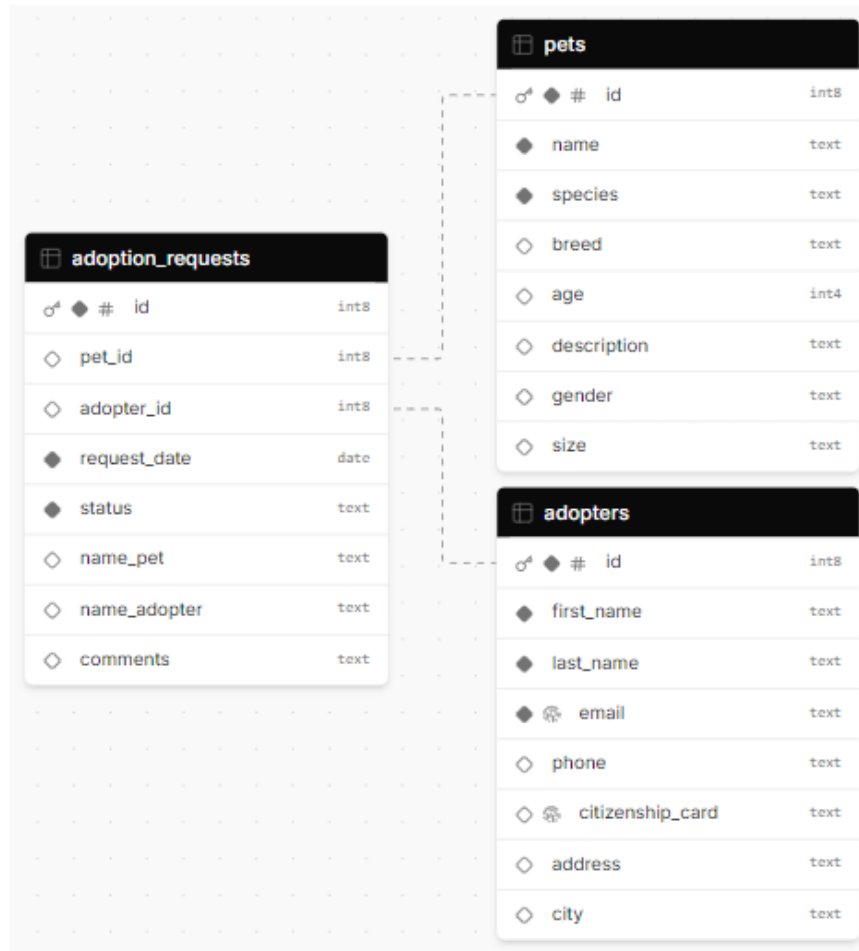
FACULTAD DE INGENIERIA DE SISTEMAS

X SEMESTRE

**DIPLOMADO DE ACTUALIZACIÓN EN NUEVAS TECNOLOGÍAS PARA EL
DESARROLLO DE SOFTWARE.**

IPIALES – NARIÑO 2024

- **Base de datos que permita llevar el registro de una empresa de adopción de mascotas:** Incluye tres tablas principales: **pets**, **adopters**, y **adoption_requests**. Cada tabla tiene su propósito y relaciones para cubrir las necesidades de este sistema.



A. Tabla pets (Mascotas): Esta tabla almacena la información de las mascotas que están disponibles para adopción.

- **id:** Es la clave primaria única de la mascota. Este valor es generado automáticamente como un número incremental.
- **name:** Nombre de la mascota. Es obligatorio.
- **species:** Especie de la mascota (perro, gato, etc.). Es obligatorio.
- **breed:** Raza de la mascota. Es opcional.
- **age:** Edad de la mascota en años. Es opcional.
- **description:** Descripción general o notas adicionales sobre la mascota. Es opcional.
- **gender:** El género de la mascota (macho, hembra). Este campo es opcional.

- **size:** El tamaño de la mascota (pequeño, mediano, grande). Este campo es opcional.

B. Tabla adopters (Adoptantes): Esta tabla almacena la información de las personas interesadas en adoptar mascotas.

- **id:** Clave primaria única del adoptante, generada automáticamente.
- **first_name:** Nombre del adoptante. Es obligatorio.
- **last_name:** Apellido del adoptante. Es obligatorio.
- **email:** Correo electrónico único del adoptante, obligatorio. No pueden existir dos registros con el mismo email.
- **phone:** Número de teléfono del adoptante. Es opcional.
- **citizenship_card:** Número de cédula o documento de identidad. Es opcional, pero debe ser único si se proporciona.
- **address:** Dirección del adoptante. Este campo es opcional.
- **city:** La ciudad donde reside el adoptante. Este campo es opcional.

C. Tabla adoption_requests (Solicitudes de adopción): Esta tabla registra las solicitudes de adopción, relacionando a los adoptantes con las mascotas que desean adoptar.

- **id:** Clave primaria única de la solicitud, generada automáticamente.
- **pet_id:** Clave foránea que hace referencia al campo id de la tabla pets, es decir, a la mascota que se desea adoptar.
- **adopter_id:** Clave foránea que hace referencia al campo id de la tabla adopters, es decir, a la persona que está realizando la solicitud de adopción.
- **request_date:** Fecha en la que se realizó la solicitud de adopción. Se genera automáticamente con la fecha actual por defecto.
- **status:** Estado de la solicitud de adopción. Solo puede ser uno de los tres valores: Pending, Approved, o Rejected (Pendiente, Aprobada, o Rechazada).
- **name_pet:** El nombre de la mascota relacionada con la solicitud de adopción. Es un campo opcional.
- **name_adopter:** El nombre del adoptante que realiza la solicitud. Es un campo opcional.
- **comments:** Comentarios adicionales sobre la solicitud de adopción. Es un campo opcional.
- **FOREIGN KEY (pet_id):** Establece una relación con la tabla pets, eliminando la solicitud de adopción si la mascota es eliminada (ON DELETE CASCADE).

- **FOREIGN KEY (adopter_id):** Establece una relación con la tabla adopters, eliminando la solicitud de adopción si el adoptante es eliminado (ON DELETE CASCADE).

Relaciones entre tablas:

- **Relación entre pets y adoption_requests:** Una mascota puede tener múltiples solicitudes de adopción (relación uno a muchos). Cada solicitud está vinculada a una mascota específica.
- **Relación entre adopters y adoption_requests:** Un adoptante puede hacer múltiples solicitudes de adopción (relación uno a muchos). Cada solicitud está vinculada a un adoptante específico.

1. Creación de componentes, uso de ngModel, RouterLink, Servicios.

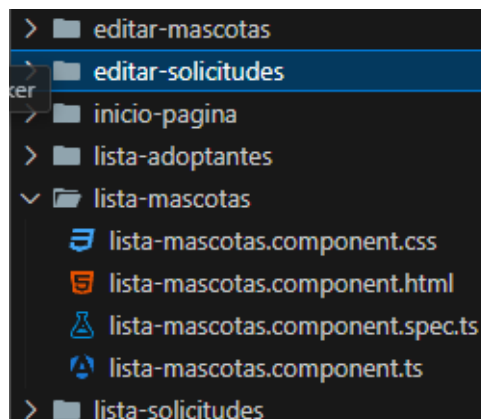
- **Creación de componentes:** Para crear un componente en Angular, se utiliza el comando **ng generate component nombre-del-componente** este comando genera automáticamente los archivos necesarios (TS, HTML, CSS y de pruebas) y añade el componente al módulo correspondiente, permitiendo así que se pueda comenzar de inmediato a trabajar en su lógica, diseño y estilos.
- **Set-ExecutionPolicy unrestricted:** Se utiliza en PowerShell para cambiar la política de ejecución de scripts, permitiendo la ejecución de scripts sin restricciones. Al establecer la política en "Unrestricted", PowerShell permitirá que se ejecuten todos los scripts, incluidos aquellos descargados de internet o de fuentes externas, aunque puede pedir confirmación antes de ejecutar archivos considerados riesgosos.

- **Ng generate component lista-editar:** Los comandos crearán dos nuevos directorios (lista y editar), cada uno con sus propios archivos .ts, .html, .css y .spec.ts correspondientes. Angular también añadirá automáticamente estos componentes al módulo relevante, lo que te permitirá empezar a desarrollar las funcionalidades de "listar" y "editar" de inmediato.

```
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> Set-ExecutionPolicy Unrestricted
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate component lista-adoptantes
CREATE src/app/lista-adoptantes/lista-adoptantes.component.html (32 bytes)
CREATE src/app/lista-adoptantes/lista-adoptantes.component.spec.ts (684 bytes)
CREATE src/app/lista-adoptantes/lista-adoptantes.component.ts (248 bytes)
CREATE src/app/lista-adoptantes/lista-adoptantes.component.css (0 bytes)
UPDATE src/app/app.module.ts (1092 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate component editar-adoptantes
CREATE src/app/editar-adoptantes/editar-adoptantes.component.html (33 bytes)
CREATE src/app/editar-adoptantes/editar-adoptantes.component.spec.ts (691 bytes)
CREATE src/app/editar-adoptantes/editar-adoptantes.component.ts (252 bytes)
CREATE src/app/editar-adoptantes/editar-adoptantes.component.css (0 bytes)
UPDATE src/app/app.module.ts (1216 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> |
```

```
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate component lista-solicitudes
CREATE src/app/lista-solicitudes/lista-solicitudes.component.html (33 bytes)
CREATE src/app/lista-solicitudes/lista-solicitudes.component.spec.ts (691 bytes)
CREATE src/app/lista-solicitudes/lista-solicitudes.component.ts (252 bytes)
CREATE src/app/lista-solicitudes/lista-solicitudes.component.css (0 bytes)
UPDATE src/app/app.module.ts (1340 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate component editar-solicitudes
CREATE src/app/editar-solicitudes/editar-solicitudes.component.html (34 bytes)
CREATE src/app/editar-solicitudes/editar-solicitudes.component.spec.ts (698 bytes)
CREATE src/app/editar-solicitudes/editar-solicitudes.component.ts (256 bytes)
CREATE src/app/editar-solicitudes/editar-solicitudes.component.css (0 bytes)
UPDATE src/app/app.module.ts (1468 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> |
```

```
PS C:\Users\AE86> cd .\Documents\Universidad\Diplomado\frontEnd\
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate component inicio-pagina
CREATE src/app/inicio-pagina/inicio-pagina.component.html (29 bytes)
CREATE src/app/inicio-pagina/inicio-pagina.component.spec.ts (663 bytes)
CREATE src/app/inicio-pagina/inicio-pagina.component.ts (236 bytes)
CREATE src/app/inicio-pagina/inicio-pagina.component.css (0 bytes)
UPDATE src/app/app.module.ts (1796 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> |
```



- **NgModule:** En el **@NgModule**, se configura el módulo principal de la aplicación, registrando componentes como `'ListaMascotasComponent'`, `'EditarMascotasComponent'` y `'InicioPaginaComponent'` en `'declarations'` para que formen parte del módulo. En `'imports'`, se incluyen módulos esenciales como `'BrowserModule'`, `'AppRoutingModule'` y `'FormsModule'`, que habilitan funcionalidades de navegación, formularios y Bootstrap para la interfaz. La sección `'providers'` define los servicios (`'MascotaService'`, `'AdoptanteService'` y `'SolicitudService'`) y proveedores (`'provideHttpClient'` y `'provideClientHydration'`) que pueden ser inyectados en toda la aplicación, mientras que `'bootstrap'` especifica `'AppComponent'` como el componente raíz que se carga al inicio.

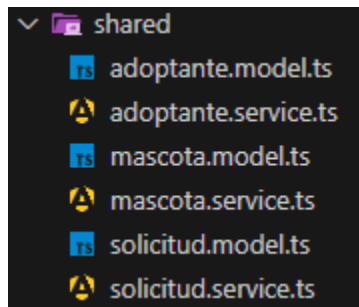
```
frontEnd > src > app > app.module.ts > AppModule
1  import { NgModule } from '@angular/core';
2  import { BrowserModule, provideClientHydration } from '@angular/platform-browser';
3  import { AppRoutingModule } from './app-routing.module';
4  import { AppComponent } from './app.component';
5  import { NgbModule } from '@ng-bootstrap/ng-bootstrap';
6  import { ListaMascotasComponent } from './lista-mascotas/lista-mascotas.component';
7  import { EditarMascotasComponent } from './editar-mascotas/editar-mascotas.component';
8  import { MascotaService } from './shared/mascota.service';
9  import { provideHttpClient } from '@angular/common/http';
10 import { FormsModule } from '@angular/forms';
11 import { ListaAdoptantesComponent } from './lista-adoptantes/lista-adoptantes.component';
12 import { EditarAdoptantesComponent } from './editar-adoptantes/editar-adoptantes.component';
13 import { ListaSolicitudesComponent } from './lista-solicitudes/lista-solicitudes.component';
14 import { EditarSolicitudesComponent } from './editar-solicitudes/editar-solicitudes.component';
15 import { AdoptanteService } from './shared/adoptante.service';
16 import { SolicitudService } from './shared/solicitud.service';
17 import { InicioPaginaComponent } from './inicio-pagina/inicio-pagina.component';
18
19 @NgModule({
20   declarations: [
21     AppComponent,
22     ListaMascotasComponent,
23     EditarMascotasComponent,
24     ListaAdoptantesComponent,
25     EditarAdoptantesComponent,
26     ListaSolicitudesComponent,
27     EditarSolicitudesComponent,
28     InicioPaginaComponent,
29   ],
30   imports: [
31     BrowserModule,
32     AppRoutingModule,
33     NgbModule,
34     FormsModule
35   ],
36   providers: [
37     MascotaService,
38     AdoptanteService,
39     SolicitudService,
40     provideHttpClient(),
41     provideClientHydration()
42   ],
43   bootstrap: [AppComponent]
44 })
45 export class AppModule { }
```

- **RouterLink:** En el archivo **AppRoutingModule**, se configura el enrutamiento de la aplicación utilizando el decorador **@NgModule**. Aquí, se definen las rutas y sus correspondientes componentes mediante un array de Routes. Cada ruta especifica un path y el componente que se debe cargar cuando se accede a esa ruta. Por ejemplo, la ruta mascotas carga el ListaMascotasComponent, mientras que la ruta mascotas/editar/:idMascota carga el EditarMascotasComponent, permitiendo la edición de una mascota específica a través del parámetro idMascota.

El módulo también incluye rutas para adoptar y gestionar solicitudes, siguiendo un patrón similar. La ruta inicio carga el InicioPaginaComponent, y se establece una ruta comodín ('**') que redirige a la página de inicio si la ruta no coincide con ninguna de las definidas. En la sección de imports, se utiliza RouterModule.forRoot(routes) para inicializar el enrutador con las rutas definidas, y se exporta RouterModule para que esté disponible en toda la aplicación. Este módulo es fundamental para permitir la navegación entre diferentes componentes de la aplicación sin recargar la página.

```
frontEnd > src > app > app-routing.module.ts > ...
1 import { NgModule } from '@angular/core';
2 import { RouterModule, Routes } from '@angular/router';
3 import { ListaMascotasComponent } from './lista-mascotas/lista-mascotas.component';
4 import { EditarMascotasComponent } from './editar-mascotas/editar-mascotas.component';
5 import { ListaAdoptantesComponent } from './lista-adoptantes/lista-adoptantes.component';
6 import { EditarAdoptantesComponent } from './editar-adoptantes/editar-adoptantes.component';
7 import { ListaSolicitudesComponent } from './lista-solicitudes/lista-solicitudes.component';
8 import { EditarSolicitudesComponent } from './editar-solicitudes/editar-solicitudes.component';
9 import { InicioPaginaComponent } from './inicio-pagina/inicio-pagina.component';
10
11 const routes: Routes = [
12   { path: 'mascotas', component: ListaMascotasComponent },
13   { path: 'mascotas/editar/:idMascota', component: EditarMascotasComponent },
14   { path: 'mascotas/agregar', component: EditarMascotasComponent },
15
16   { path: 'adoptantes', component: ListaAdoptantesComponent },
17   { path: 'adoptantes/editar/:idAdoptante', component: EditarAdoptantesComponent },
18   { path: 'adoptantes/agregar', component: EditarAdoptantesComponent },
19
20   { path: 'solicitudes', component: ListaSolicitudesComponent },
21   { path: 'solicitudes/editar/:idSolicitud', component: EditarSolicitudesComponent },
22   { path: 'solicitudes/agregar', component: EditarSolicitudesComponent },
23
24   { path: 'inicio', component: InicioPaginaComponent },
25
26   { path: '**', redirectTo: '/inicio', pathMatch: 'full' } // solo una ruta '**'
27 ];
28
29 @NgModule({
30   imports: [RouterModule.forRoot(routes)],
31   exports: [RouterModule]
32 })
33 export class AppRoutingModule { }
```


- **Servicios:** En los archivos de servicios `mascota.service.ts`, `adoptante.service.ts` y `solicitud.service.ts`, se gestionan las funcionalidades necesarias para interactuar con los datos de cada entidad. El `MascotaService` se encarga de la lógica relacionada con las mascotas, permitiendo obtener, crear, editar y eliminar registros a través del `HttpClient` para realizar solicitudes HTTP a la API. De manera similar, el `AdoptanteService` maneja la información sobre los adoptantes, ofreciendo métodos para gestionar su lista y realizar operaciones de datos en el backend. Por último, el `SolicitudService` se centra en las solicitudes de adopción, permitiendo crear, editar y obtener solicitudes mediante el `HttpClient`. Juntos, estos servicios encapsulan la lógica de negocio y el acceso a datos, promoviendo la reutilización y la separación de preocupaciones en la aplicación Angular.



```
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate service shared/adoptante
CREATE src/app/shared/adoptante.service.spec.ts (388 bytes)
CREATE src/app/shared/adoptante.service.ts (147 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> ng generate service shared/solicitud
CREATE src/app/shared/solicitud.service.spec.ts (388 bytes)
CREATE src/app/shared/solicitud.service.ts (147 bytes)
PS C:\Users\AE86\Documents\Universidad\Diplomado\frontEnd> |
```

- **Ejemplo: mascota.service.ts:** El código define el servicio **MascotaService**, que está diseñado para gestionar las operaciones relacionadas con las mascotas en una aplicación Angular. Utilizando el decorador **@Injectable**, se establece que este servicio puede ser inyectado en otros componentes o servicios, y se proporciona en el ámbito raíz de la aplicación. El servicio importa **HttpClient** para realizar solicitudes HTTP a un servidor, así como el modelo **MascotaModel** que representa la estructura de los datos de las mascotas. Dentro de la clase, se define una constante **BASE_URL** que almacena la URL base del servidor. El servicio incluye métodos para obtener la lista completa de mascotas (`obtenerMascotas``), buscar una mascota por su ID (`obtenerMascota``), agregar una nueva mascota (`agregarMascota``), actualizar una mascota existente (`actualizarMascota``) y eliminar una mascota (`borrarMascota``). Cada uno de estos métodos utiliza el `HttpClient`` para realizar las operaciones correspondientes en la API, facilitando así la manipulación y gestión de los datos de las mascotas.

```
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { MascotaModel } from '../mascota.model';

@Injectable({
  providedIn: 'root'
})
export class MascotaService {

  BASE_URL='http://127.0.0.1:4000';
  constructor(private http: HttpClient) {

  }

  //Lista completa de Mascotas
  obtenerMascotas(){
    return this.http.get<MascotaModel[]>(`${this.BASE_URL}/mascotas/buscar`);
  }

  //Buscar una mascota por ID
  obtenerMascota(idMascota:string){
    return this.http.get<MascotaModel>(`${this.BASE_URL}/mascotas/buscarId/${idMascota}`);
  }

  //Crear una Mascota
  agregarMascota(mascota: MascotaModel){
    return this.http.post<string>(`${this.BASE_URL}/mascotas/crear`,mascota);
  }

  //Actualizar una Mascota
  actualizarMascota(mascota: MascotaModel){
    return this.http.put<string>(`${this.BASE_URL}/mascotas/actualizar/${mascota.id}`,mascota);
  }

  //Eliminar una Mascota
  borrarMascota(idMascota: string){
    return this.http.delete<string>(`${this.BASE_URL}/mascotas/borrar/${idMascota}`);
  }
}
```

En la aplicación, se repiten los procesos de creación de componentes, definición de modelos, implementación de RouterLink y desarrollo de servicios para los tres apartados principales: **mascotas, adoptantes y solicitudes** . Para cada entidad, se generan componentes específicos que permiten la visualización y gestión de los datos, se crean modelos que estructuran la información, y se establecen rutas a través de RouterLink para facilitar la navegación entre diferentes vistas. Además, se desarrollan servicios dedicados que gestionan las operaciones de datos mediante solicitudes HTTP, asegurando una interacción eficiente con el backend. Esta estructura modular y consistente facilita el mantenimiento y la escalabilidad de la aplicación.

2. Desarrollo con HTML 5 y JavaScript.

Modulo Mascotas

- **lista-mascotas.component.ts:**

```
import { Component, OnInit } from '@angular/core';
import { Observable } from 'rxjs';
import { MascotaModel } from '../shared/mascota.model';
import { MascotaService } from '../shared/mascota.service';

@Component({
  selector: 'app-lista-mascotas',
  templateUrl: './lista-mascotas.component.html',
  styleUrls: ['./lista-mascotas.component.css'] // Corrige "styleUrl" a "styleUrls"
})
export class ListaMascotasComponent implements OnInit {
  mascotas: Observable<MascotaModel[]> | undefined;
  searchText: string = ''; // Agrega esta propiedad

  constructor(private mascotaService: MascotaService) {}

  ngOnInit() {
    this.mascotas = this.mascotaService.obtenerMascotas();
  }

  borrarMascota(idMascota: string) {
    this.mascotaService.borrarMascota(idMascota).subscribe({
      next: data => {
        console.log(`Registro Eliminado`);
        this.ngOnInit();
      },
      error: err => {
        console.log(`Error al eliminar Registro ${err}`);
      }
    });
  }
}
```

El código define el componente `ListaMascotasComponent`, que es responsable de mostrar una lista de mascotas en la aplicación Angular. Utilizando el decorador `@Component`, se establece el selector del componente y se vincula un archivo de plantilla HTML y una hoja de estilos CSS. El componente implementa la interfaz `OnInit`, lo que permite ejecutar código adicional al momento de inicializar el componente. En el constructor, se inyecta el servicio `MascotaService`, que proporciona métodos para gestionar las mascotas. En el método `ngOnInit`, se llama a `obtenerMascotas` del servicio, asignando la lista de mascotas a la propiedad `mascotas`, que es de tipo `Observable`. Además, el componente incluye un método `borrarMascota` que se encarga de eliminar una mascota específica. Este método llama a `borrarMascota` del servicio y, al recibir una respuesta, muestra un mensaje en la consola y vuelve a cargar la lista de mascotas al invocar `ngOnInit` nuevamente. También se agrega una propiedad `searchText` para facilitar futuras implementaciones de búsqueda.

- **lista-mascotas.component.html:**

```
<!-- Imagen de fondo -->
<div class="background-image"></div>

<!-- Barra de navegación -->
<nav class="navbar navbar-expand-lg navbar custom-navbar">
  <a class="navbar-brand" href="/inicio">Tail Tails Rescue</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link custom-nav-link" href="/mascotas">Mascotas</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/adoptantes">Adoptantes</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/solicitudes">Solicitudes</a>
      </li>
    </ul>
  </div>
  <!-- Imagen de usuario -->
  <ul class="navbar-nav">
    <li class="nav-item">
      <a class="nav-link" href="#">
        
      </a>
    </li>
  </ul>
</div>
</nav>
```

Se define la estructura HTML de la página de gestión de mascotas, utilizando Bootstrap para el diseño y estilo. Comienza con un div que puede ser estilizado como una imagen de fondo y continúa con una barra de navegación que incluye un logotipo que redirige a la página de inicio y enlaces a las secciones de "Mascotas", "Adoptantes" y "Solicitudes". En la parte superior derecha de la barra, se presenta un ícono de usuario. A continuación, se define un contenedor principal con un título centrado, "Gestión de Mascotas", y un campo de búsqueda que permite a los usuarios filtrar mascotas por nombre o especie, enlazado a la propiedad `searchText` del componente Angular mediante `ngModel`. Este campo de búsqueda cuenta con estilos personalizados, como un borde redondeado y efectos de sombra, que mejoran su apariencia y brindan una experiencia de usuario más atractiva y funcional.

```
<div class="container mt-5">
  <div class="d-flex justify-content-between align-items-center mb-4">
    <h2 class="text-center">Gestión de Mascotas</h2>
    <!-- Espacio para centrar la barra de búsqueda -->
    <div style="flex-grow: 1; text-align: center;">
      <input class="form-control" type="text" placeholder="Buscar por nombre o especie..." [(ngModel)]="searchText"
        style="width: 300px; display: inline-block; border-radius: 25px; border: 2px solid #00ffaa; padding: 10px; box-shadow: 0 2px 5px #000; transition: border 0.3s; font-family: 'Arial', sans-serif; font-size: 1.2rem;">
    </div>
    <!-- Botón Registrar Nueva Mascota alineado a la derecha -->
    <button class="btn btn-sm" [routerLink]="['/mascotas/agregar']"
      style="background-color: #00ffaa; color: black; box-shadow: 0 2px 5px #000; font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      Registrar Nueva Mascota
    </button>
  </div>
</div>
```

Gestión de Mascotas

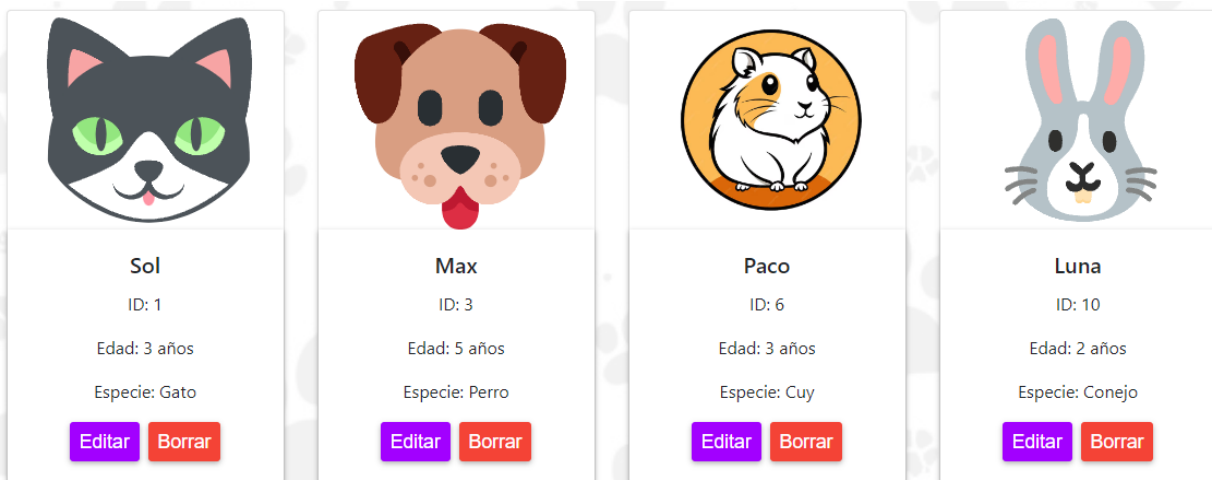
[Registrar Nueva Mascota](#)

Una sección de la página de gestión de mascotas, ubicada dentro de un contenedor Bootstrap que agrega un margen superior. En esta sección, se utiliza una clase de flexbox (`d-flex`) para alinear horizontalmente un título y un campo de búsqueda, junto con un botón para registrar nuevas mascotas. El título, "Gestión de Mascotas", está centrado y se encuentra a la izquierda. A la derecha, se presenta un campo de entrada diseñado para buscar mascotas por nombre o especie, que utiliza `ngModel` para enlazarlo a la propiedad `searchText` del componente Angular, lo que permite filtrar los resultados en tiempo real. Este campo tiene un estilo atractivo, con un borde redondeado y una sombra que le da profundidad. Finalmente, hay un botón de "Registrar Nueva Mascota" alineado a la derecha, que utiliza `[routerLink]` para redirigir a los usuarios a la página de registro de nuevas mascotas, con un color de fondo personalizado y un estilo que complementa el resto del diseño.

```

<div class="container mt-5">
  <div class="row">
    <ng-container *ngFor="let mascota of (mascotas | async)">
      <div class="col-lg-3 col-md-6 col-sm-12">
        <div class="card mb-4 shadow-sm">
          <img class="card-img-top" [src]="(mascota.species.toLowerCase() === 'perro') ? 'https://www.svgrepo.com/show/405231/dog-face.svg' :
            (mascota.species.toLowerCase() === 'gato') ? 'https://www.svgrepo.com/show/401273/cat-face.svg' :
            (mascota.species.toLowerCase() === 'hamster') ? 'https://www.svgrepo.com/show/405777/hamster.svg' :
            (mascota.species.toLowerCase() === 'conejo') ? 'https://www.svgrepo.com/show/398149/rabbit-face.svg' :
            (mascota.species.toLowerCase() === 'cuy' || mascota.species.toLowerCase() === 'conejillo de indias') ? 'https://img.freepik.com/premium-vector/guinea-
            (mascota.species.toLowerCase() === 'ratón') ? 'https://www.svgrepo.com/show/402148/rat.svg' :
            (mascota.species.toLowerCase() === 'rata') ? 'https://www.svgrepo.com/show/402177/mouse.svg' :
            (mascota.species.toLowerCase() === 'loro') ? 'https://www.svgrepo.com/show/397712/parrot.svg' :
            (mascota.species.toLowerCase() === 'canario') ? 'https://img.freepik.com/free-vector/yellow-bird-illustration_1142-206621.jpg?t=st=1727738059-exp=1727
            (mascota.species.toLowerCase() === 'periquito') ? 'https://www.svgrepo.com/show/417456/bird.svg' :
            (mascota.species.toLowerCase() === 'gallo') ? 'https://www.svgrepo.com/show/402629/rooster.svg' :
            (mascota.species.toLowerCase() === 'aguila') ? 'https://www.svgrepo.com/show/401438/eagle.svg' :
            (mascota.species.toLowerCase() === 'tortuga') ? 'https://www.svgrepo.com/show/402833/turtle.svg' :
            (mascota.species.toLowerCase() === 'serpiente') ? 'https://www.svgrepo.com/show/289236/snake.svg' :
            (mascota.species.toLowerCase() === 'iguana') ? 'https://img.freepik.com/free-vector/hand-drawn-cartoon-chameleon-illustration_23-2150387676.jpg?t=st=1
            (mascota.species.toLowerCase() === 'cangrejo') ? 'https://www.svgrepo.com/show/530182/crab.svg' :
            (mascota.species.toLowerCase() === 'tarantula') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
            (mascota.species.toLowerCase() === 'araña') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
            (mascota.species.toLowerCase() === 'hurón') ? 'https://img.freepik.com/free-vector/cute-otter-wild-animal-cartoon-sticker_1308-78999.jpg?t=st=17277384
            (mascota.species.toLowerCase() === 'cacatúa') ? 'https://img.freepik.com/premium-vector/cute-vector-illustration-parrot-kids_925324-31237.jpg?w=740' :
            (mascota.species.toLowerCase() === 'ave') ? 'https://www.svgrepo.com/show/484500/bird.svg' :
            (mascota.species.toLowerCase() === 'pez') ? 'https://www.svgrepo.com/show/317413/fish.svg' :
            (mascota.species.toLowerCase() === 'murcielago') ? 'https://www.svgrepo.com/show/425730/bat-scary.svg' :
            'https://www.svgrepo.com/show/366549/application-pet.svg'" alt="Imagen de {{mascota.name}}"/>
          <div class="card-body">
            <h5 class="card-title">{{ mascota.name }}</h5>
            <p class="card-text">ID: {{ mascota.id }}</p>
            <p class="card-text">Edad: {{ mascota.age }} años</p>
            <p class="card-text">Especie: {{ mascota.species }}</p>
            <!-- Botón Editar -->
            <a class="btn btn-sm mr-2" [routerLink]="['/mascotas/editar/', mascota.id]"
              style="background-color: #a200ff; color: white; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.1rem;">
              Editar
            </a>
            <!-- Botón Borrar -->
            <button class="btn btn-sm" (click)="borrarMascota(mascota.id)"
              style="background-color: #f44336; color: white; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.1rem;">
              Borrar
            </button>
          </div>
        </div>
      </div>
    </ng-container>
  </div>
</div>

```



Se implementa una lista de tarjetas para mostrar información sobre las mascotas disponibles, utilizando un contenedor ``ng-container`` que recorre cada mascota obtenida a través de un observable. Cada tarjeta se presenta en una columna que se adapta a diferentes tamaños de pantalla gracias a las clases de Bootstrap (``col-lg-3``, ``col-md-6``, ``col-sm-12``). Se muestra una imagen que varía según la especie de la mascota, utilizando un condicional en el atributo ``src`` para asignar una URL específica a cada especie. Además, cada tarjeta incluye detalles como el nombre, ID, y edad de la mascota, además de dos botones: uno para editar la información de la mascota, que redirige a la página de edición correspondiente, y otro para eliminar la mascota, que ejecuta el método ``borrarMascota()`` al hacer clic. La funcionalidad de búsqueda está implementada a través de un filtro que muestra las tarjetas de mascotas solo si el nombre o especie coinciden con el texto ingresado, utilizando ``searchText`` para hacer la comparación. El diseño es responsivo y visualmente atractivo, proporcionando una buena experiencia de usuario.

- **editar-mascotas.component.ts:** El componente `EditarMascotasComponent` es una parte esencial de la aplicación que permite a los usuarios editar o agregar información sobre mascotas. En su constructor, se inyectan los servicios ``MascotaService``, ``ActivatedRoute``, y ``Router``, facilitando la obtención de datos de mascotas y la navegación entre las diferentes vistas. Durante la inicialización (``ngOnInit``), se captura el ``idMascota`` de la ruta activa; si se proporciona, el componente solicita la información correspondiente a esa mascota a través del servicio y la asigna a la propiedad ``mascota``. Esto permite que los usuarios vean los detalles actuales de la mascota al editarla. En el método ``onSubmit``, el componente determina si se está editando una mascota existente o creando una nueva, y llama al método adecuado del ``MascotaService``: ``actualizarMascota`` para editar o ``agregarMascota`` para crear. Después de una operación exitosa, el usuario es redirigido a la lista de mascotas. Si ocurre algún error durante el proceso, se muestra un mensaje de error en la consola para facilitar la depuración. Este componente proporciona una funcionalidad esencial para gestionar la información de las mascotas dentro de la aplicación.

```

import { Component, OnInit } from '@angular/core';
import { MascotaModel } from '../shared/mascota.model';
import { MascotaService } from '../shared/mascota.service';
import { ActivatedRoute, Router } from '@angular/router';

@Component({
  selector: 'app-editar-mascotas',
  templateUrl: './editar-mascotas.component.html',
  styleUrls: ['./editar-mascotas.component.css']
})
export class EditarMascotasComponent implements OnInit {

  idMascota= '';
  mascota = new MascotaModel(
    '', // Convertir a number (aunque aquí la cadena vacía será NaN)
    '', // nombre
    '', // species
    '', // breed
    '', // Convertir a number para edad
    '', // gender
    '', // size
    '' // description
  );

  constructor(private mascotaService: MascotaService, private route: ActivatedRoute, private router: Router){
  }

  ngOnInit(){
    this.idMascota=this.route.snapshot.params['idMascota'];
    console.log('El idMascota es ${this.idMascota}');

    if(this.idMascota){
      //Viene de Editar
      console.log('La solicitud viene de Editar');
      this.mascotaService.obtenerMascota(this.idMascota).subscribe({
        next: data=>{
          console.log(data);
          this.mascota=data;
          console.log(this.mascota);
        },
        error: err=>{
          console.log('Error ${err}');
        }
      });
    }
    else{
      console.log('La solicitud viene de Nueva Mascota');
    }
  }
}

```

```

onSubmit(){
  console.log("On Submit");
  //Viene de Editar
  if(this.mascota.id){
    this.mascotaService.actualizarMascota(this.mascota).subscribe({
      next: data=>{
        console.log(data);
        this.router.navigate(['/mascotas']);
      },
      error: err=>{
        console.log('Error al actualizar ${err}');
      }
    });
  }
  else{
    //Viene de Nueva Mascota
    this.mascotaService.agregarMascota(this.mascota).subscribe({
      next: data=>{
        console.log(data);
        this.router.navigate(['/mascotas']);
      },
      error: err=>{
        console.log('Error al Agregar ${err}');
      }
    });
  }
}
}

```


- **editar-mascotas.component.html:**

Es un fragmento del template del componente ``EditarMascotasComponent`` en Angular, que permite editar la información de una mascota. Utiliza una estructura de contenedor de Bootstrap con un margen superior significativo y un título centrado que dice "Editar Información de la Mascota", estilizado con sombra y fuente personalizada para un mejor aspecto visual. Dentro de una tarjeta (``card``) con una clase personalizada, se incluye un formulario vinculado a la directiva ``ngForm`` de Angular, que maneja el evento de envío a través de ``(ngSubmit)``. El formulario contiene una fila que alberga una imagen representativa de la especie de la mascota, la cual se selecciona dinámicamente según el valor de la propiedad ``species`` de la mascota.

```

<!-- Columna para las casillas -->
<div class="col-md-8">
  <div class="row">
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="name" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Nombre</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.name" name="name" id="name" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="species" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Especie</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.species" name="species" id="species" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="breed" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Raza</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.breed" name="breed" id="breed" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="age" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Edad</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.age" name="age" id="age" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
  </div>
  <div class="row">
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="gender" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Género</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.gender" name="gender" id="gender" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
    <div class="col-md-6">
      <fieldset class="form-group">
        <label class="control-label" for="size" style="text-shadow: 0 3px 2px rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Tamaño</label>
        <input type="text" class="form-control" required [(ngModel)]="mascota.size" name="size" id="size" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
      </fieldset>
    </div>
  </div>
</div>

```

Incluye un formulario para editar la información de una mascota. Dentro de una columna de tamaño mediano (`col-md-8`), se organizan varias filas (`row`), cada una conteniendo campos de entrada (`input`) para diferentes atributos de la mascota, como "Nombre", "Especie", "Raza", "Edad", "Género", "Tamaño" y "Descripción". Cada campo está diseñado con una etiqueta (`label`) que presenta un estilo personalizado, incluyendo sombra de texto y una fuente de tamaño adecuado, lo que mejora la legibilidad. Los campos de entrada utilizan `ngModel` para la vinculación de datos bidireccional, asegurando que la información se mantenga actualizada en el modelo de la mascota. Todos los campos están marcados como obligatorios (`required`), lo que garantiza que el formulario no se pueda enviar sin completarlos.

```

<!-- Botones de Guardar Cambios debajo -->
<div class="text-center mt-4">
  <button type="submit" class="btn btn-sm" [disabled]="!mascotaForm.form.valid"
    style="background-color: #80ffb9; color: rgba(0, 0, 0); box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.2rem;" >
    Guardar Cambios
  </button>
  <button class="btn btn-sm" [routerLink]="['/mascotas']"
    style="background-color: #6faeff; color: rgba(0, 0, 0); box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); margin-left: 10px; font-family: 'Arial', sans-serif; font-size: 1.2rem;">
    Volver
  </button>
</div>
</form>
</div>
</div>
</div>

```

Se definen los botones para guardar los cambios y volver en el formulario del componente `EditarMascotasComponent`. Los botones están centrados en la parte

inferior del formulario y tienen un margen superior (`mt-4`) para separarlos del contenido anterior.

El botón "Guardar Cambios" es de tipo `submit` y está habilitado solo si el formulario es válido (`!mascotaForm.form.valid`). Este botón tiene un estilo personalizado con un fondo color verde claro (`#80ffb9`), texto en negro, y un efecto de sombra para darle profundidad. Además, se utiliza una fuente de estilo Arial y un tamaño de 1.2rem.

El botón "Volver" utiliza la directiva `[routerLink]` para redirigir a la ruta `/mascotas` cuando se hace clic en él. Este botón tiene un fondo azul (`#6faeff`) y un estilo similar al botón de guardar, incluyendo un margen a la izquierda para separarlo visualmente del botón anterior. Ambos botones están diseñados para ser visualmente atractivos y funcionales dentro de la interfaz de usuario del formulario.

Editar Información de la Mascota



Nombre	Especie
<input type="text" value="Sol"/>	<input type="text" value="Gato"/>
Raza	Edad
<input type="text" value="Siames"/>	<input type="text" value="3"/>
Género	Tamaño
<input type="text" value="Hembra"/>	<input type="text" value="Pequeño"/>
Descripción	
<input type="text" value="Color gris con manchas blancas"/>	

Gestionar Información de la Mascota



Nombre	Especie
<input type="text"/>	<input type="text"/>
Raza	Edad
<input type="text"/>	<input type="text"/>
Género	Tamaño
<input type="text"/>	<input type="text"/>
Descripción	
<input type="text"/>	

En el desarrollo de la aplicación, se ha implementado una estructura similar para los módulos de Adoptantes y Solicitudes, aunque con un diseño diferenciado que se adapta a las características y necesidades específicas de cada uno. Esta coherencia en la estructura permite mantener una experiencia de usuario fluida y familiar, facilitando la navegación y la interacción con las distintas funcionalidades de la aplicación. Al mismo tiempo, el uso de diseños únicos en cada módulo ayuda a resaltar la identidad y el propósito particular de cada sección, garantizando que cada parte de la aplicación se sienta distintiva y alineada con su función.

Ejemplos visuales de Adoptantes:

Gestión de Adoptantes

Buscar por nombre o identificación...

Registrar Nuevo Adoptante

Lista de Adoptantes

ID:	NOMBRE:	APELLIDO:	EMAIL:	TELÉFONO:	IDENTIFICACIÓN:	DIRECCIÓN:	CIUDAD:		
1	Andrés	Ríos	andres.rios@gmail.com	3137894560	192837465	Calle 123 #9-10	Cartagena	Editar	Borrar
3	Anita	García	ana.garcia@gmail.com	3104567890	123456789	Calle 45 #20-30	Bogotá	Editar	Borrar
4	Carlos	Pérez	carlos.perez@gmail.com	3117894561	987654321	Carrera 7 #15-67	Cali	Editar	Borrar
5	Lucía	Martínez	lucia.martinez@gmail.com	3124567891	564738291	Avenida Siempre Viva #101	Medellín	Editar	Borrar

Editar Información del Adoptante



Nombre

Andrés

Apellido

Ríos

Email

andres.rios@gmail.com

Teléfono

3137894560

Identificación

192837465

Dirección

Calle 123 #9-10

Ciudad

Cartagena

Guardar Cambios

Volver

Ejemplos visuales de Solicitudes:

Gestión de Solicitudes


Buscar por nombre o identificación...

Registrar Nueva Solicitud

Lista de Solicitudes de Adopción

ID:	ID Mascota:	ID Adopante:	Fecha de Solicitud:	Estado:	Mascota:	Adoptante:	Comentarios:	Editar	Borrar
3	6	3	2024-09-25	Approved	Paco	Anita García	Ninguno		
4	3	5	2024-09-25	Approved	Max	Lucía Martínez	Ninguno		

Editar Información de la Solicitud



ID Mascota

6

ID Adoptante

3

Fecha de Solicitud

2024-09-25

Estado

Approved

Nombre Mascota

Paco

Nombre Adoptante

Anita García

Comentarios

Ninguno

Guardar Cambios

Volver

Modulo Página de Inicio.

- inicio-pagina.component.html:

```
<!-- Imagen de fondo -->
<div class="background-image"></div>

<!-- Barra de navegación -->
<nav class="navbar navbar-expand-lg navbar custom-navbar">
  <a class="navbar-brand" href="/inicio">Tail Tails Rescue</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link custom-nav-link" href="/mascotas">Mascotas</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/adoptantes">Adoptantes</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/solicitudes">Solicitudes</a>
      </li>
    </ul>
    <!-- Imagen de usuario -->
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">
          
        </a>
      </li>
    </ul>
  </div>
</nav>

<!-- Sección de bienvenida centrada -->
<div class="welcome-section">
  <!-- Caja con logo -->
  <div class="logo-card">
    
  </div>

  <!-- Texto de bienvenida -->
  <div>
    <h1>¡Bienvenidos a Tail Tails Rescue!</h1>
    <p>Nos dedicamos a rescatar y encontrar hogares para nuestras queridas mascotas. ¡Gracias por unirte a nuestra misión de brindarles una segunda oportunidad!</p>
  </div>
</div>

<!-- Sección acerca de la adopción -->
<div class="about-adoption">
  <h2>¿Por qué adoptar una mascota?</h2>
  <p>Adoptar una mascota no solo les brinda un hogar a estos maravillosos animales, sino que también te da la oportunidad de recibir amor incondicional. Cada mascota que adoptas tiene una historia y merece un hogar donde pueda ser amada y cuidada para siempre.</p>
  <p>El proceso de adopción es simple, pero responsable. Queremos asegurarnos de que cada animal encuentre el lugar perfecto donde pueda ser amado y cuidado para siempre.</p>
</div>
```

Configura la página de inicio de la aplicación **"Tail Tails Rescue"**, comenzando con una imagen de fondo que crea un ambiente atractivo y acogedor. La barra de navegación, diseñada con Bootstrap, proporciona acceso rápido a las secciones principales de la aplicación: Mascotas, Adoptantes y Solicitudes. Incluye un botón de colapso para dispositivos móviles y un ícono de usuario, mejorando la accesibilidad y la experiencia del usuario. La sección de bienvenida, centrada en la pantalla, destaca un logo y un mensaje que invita a los visitantes a unirse a la misión de rescatar y encontrar hogares para mascotas. Además, se incluye una sección que aborda la importancia de adoptar, enfatizando los beneficios tanto para los animales como para las familias adoptantes, así como un proceso de adopción responsable, lo que refuerza el compromiso de la organización con el bienestar animal.

3. Estilos CSS (Uso de Bootstrap), se debe generar una interface ordenada estructurada y agradable para el usuario final.

Página de Inicio.

- **Barra de navegación:** Se implementa la clase navbar de Bootstrap, que proporciona un diseño responsivo y estilizado para la barra de navegación. El uso de navbar-expand-lg permite que la barra se expanda en pantallas grandes y se colapse en pantallas más pequeñas. Los enlaces de navegación (nav-link) y el botón para colapsar la barra también utilizan clases de Bootstrap para mantener un diseño uniforme.
- **Estructura de navegación:** La lista de navegación (navbar-nav) y los elementos de la lista (nav-item) se organizan usando clases de Bootstrap, lo que facilita la creación de un menú que se adapta automáticamente a diferentes tamaños de pantalla.
- **Estilos responsivos:** El uso de clases como collapse y navbar-toggler permite que la barra de navegación funcione correctamente en dispositivos móviles, ofreciendo una mejor experiencia de usuario en diferentes dispositivos.
- **Imágenes redondeadas:** Se utiliza la clase rounded-circle en la imagen del usuario para darle un estilo de círculo, lo que es una característica de Bootstrap que ayuda a mantener un diseño moderno y limpio.
- **Sistema de rejillas (Grid system):** Aunque no se usa explícitamente en este fragmento, el uso de clases como col-md-* en otras secciones de la aplicación podría integrarse con Bootstrap, permitiendo una disposición fluida de los elementos en la página.
- **Scripts de Bootstrap:** Al final del código, se incluyen los scripts de Bootstrap, junto con jQuery y Popper.js, que son necesarios para que algunas funcionalidades de Bootstrap, como los menús desplegables y el colapso, funcionen correctamente.


```

<!-- Barra de navegación -->
<nav class="navbar navbar-expand-lg navbar custom-navbar">
  <a class="navbar-brand" href="/inicio">Tail Tails Rescue</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
    aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarNav">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link custom-nav-link" href="/mascotas">Mascotas</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/adoptantes">Adoptantes</a>
      </li>
      <li class="nav-item">
        <a class="nav-link custom-nav-link" href="/solicitudes">Solicitudes</a>
      </li>
    </ul>
    <!-- Imagen de usuario -->
    <ul class="navbar-nav">
      <li class="nav-item">
        <a class="nav-link" href="#">
          
        </a>
      </li>
    </ul>
  </div>
</nav>

```

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.2/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

```

Modulo Mascotas

➤ Listar Mascotas:

- **Inclusión de Bootstrap CSS:** Se carga el archivo CSS de Bootstrap para acceder a sus estilos.
- **Barra de Navegación:** Se utiliza la clase navbar para crear una barra de navegación responsiva. La clase navbar-expand-lg permite que la barra se expanda en pantallas grandes y se colapse en pantallas más pequeñas.
- **Estructura de la Barra de Navegación:** Se utilizan las clases navbar-nav, nav-item, y nav-link para estructurar y estilizar los elementos de la barra de navegación.
- **Contenedor y Diseño de Tarjetas:** La clase container se usa para crear un contenedor con márgenes automáticos y ancho fijo. row se utiliza para crear filas en el sistema de grid de Bootstrap. Las clases de columnas (col-lg-3, col-md-6, col-sm-12) definen el diseño responsivo.
- card se usa para crear tarjetas que contienen información sobre las mascotas.

- ```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
```

```

<div class="row">
 <div container="ngFor="let mascota of mascotas | async">
 <div class="col-lg-3 col-md-6 col-sm-12">
 *ngIf="searchText || mascota.name.toLowerCase().includes(searchText.toLowerCase()) || mascota.species.toLowerCase().includes(searchText.toLowerCase())"
 <div class="card mb-4 shadow-sm">
 <!-- Asignar imagen según la especie usando un condicional en el src -->
 <img class="card-img-top" [src]="(mascota.species.toLowerCase() === 'perro') ? 'https://www.svgrepo.com/show/405231/dog-face.svg' :
 (mascota.species.toLowerCase() === 'gato') ? 'https://www.svgrepo.com/show/401273/cat-face.svg' :
 (mascota.species.toLowerCase() === 'hamster') ? 'https://www.svgrepo.com/show/405771/hamster.svg' :
 (mascota.species.toLowerCase() === 'conejo') ? 'https://www.svgrepo.com/show/398149/rabbit-face.svg' :
 (mascota.species.toLowerCase() === 'cuy') || mascota.species.toLowerCase() === 'conejo de indias' ? 'https://img.freepik.com/premium-vector/guinea-pig-vector-illustration_1142-206621.jpg?st=1727738859-exp' :
 (mascota.species.toLowerCase() === 'ratón') ? 'https://www.svgrepo.com/show/402148/rat.svg' :
 (mascota.species.toLowerCase() === 'rata') ? 'https://www.svgrepo.com/show/402177/mouse.svg' :
 (mascota.species.toLowerCase() === 'loro') ? 'https://www.svgrepo.com/show/397712/parrot.svg' :
 (mascota.species.toLowerCase() === 'canario') ? 'https://img.freepik.com/free-vector/yellow-bird-illustration_1142-206621.jpg?st=1727738859-exp' :
 (mascota.species.toLowerCase() === 'periquito') ? 'https://www.svgrepo.com/show/417456/bird.svg' :
 (mascota.species.toLowerCase() === 'gallo') ? 'https://www.svgrepo.com/show/402629/rooster.svg' :
 (mascota.species.toLowerCase() === 'anguila') ? 'https://www.svgrepo.com/show/401438/eagle.svg' :
 (mascota.species.toLowerCase() === 'tortuga') ? 'https://www.svgrepo.com/show/402833/turtle.svg' :
 (mascota.species.toLowerCase() === 'serpiente') ? 'https://www.svgrepo.com/show/289236/snake.svg' :
 (mascota.species.toLowerCase() === 'iguana') ? 'https://img.freepik.com/free-vector/hand-drawn-cartoon-chameleon-illustration_23-2150387676.jpg?st=1727738859-exp' :
 (mascota.species.toLowerCase() === 'cangrejo') ? 'https://www.svgrepo.com/show/530182/crab.svg' :
 (mascota.species.toLowerCase() === 'tarantula') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
 (mascota.species.toLowerCase() === 'araña') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
 (mascota.species.toLowerCase() === 'hurón') ? 'https://img.freepik.com/free-vector/cute-otter-wild-animal-cartoon-sticker_1308-78999.jpg?st=1727738859-exp' :
 (mascota.species.toLowerCase() === 'cacatúa') ? 'https://img.freepik.com/premium-vector/cute-vector-illustration-parrot-kids_925324-31237.jpg?w=768' :
 (mascota.species.toLowerCase() === 'ave') ? 'https://www.svgrepo.com/show/484500/bird.svg' :
 (mascota.species.toLowerCase() === 'pez') ? 'https://www.svgrepo.com/show/317413/fish.svg' :
 (mascota.species.toLowerCase() === 'murcielago') ? 'https://www.svgrepo.com/show/425730/bat-scary.svg' :
 'https://www.svgrepo.com/show/366549/application-pet.svg'" alt="Imagen de {{mascota.name}}"/>
 <div class="card-body">
 <h5 class="card-title">{{mascota.name}}</h5>
 <p class="card-text">ID: {{mascota.id}}</p>
 <p class="card-text">Edad: {{mascota.age}} años</p>
 <p class="card-text">Especie: {{mascota.species}}</p>
 </div>
 </div>
 </div>
 </div>

```

```

<!-- Botón Editar -->
<a class="btn btn-sm mr-2" [routerLink]="['/mascotas/editar/', mascota.id]"
 style="background-color: #a200ff; color: white; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.1rem;"
 Editar

<!-- Botón Borrar -->
<button class="btn btn-sm" (click)="borrarMascota(mascota.id)"
 style="background-color: #f44336; color: white; box-shadow: 0 2px 5px rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.1rem;"
 Borrar
</button>
</div>

```

### ➤ Editar Mascotas:

- **Inclusión de Bootstrap CSS:** Se carga el archivo CSS de Bootstrap desde un CDN para aplicar estilos predefinidos.
- **Meta Etiquetas:** Se incluyen meta etiquetas para definir el conjunto de caracteres y la configuración de la vista responsiva.
- **Barra de Navegación:** Se utiliza una barra de navegación (navbar) responsiva que se expande en pantallas grandes.
- Se aplican estilos personalizados a la barra de navegación, incluyendo un color de fondo y estilo de texto.
- **Estructura de la Barra de Navegación:** Se utilizan clases de Bootstrap para crear una lista de navegación (navbar-nav).
- Se incluyen elementos de la lista (nav-item y nav-link) con estilos personalizados.
- **Contenedor Principal:** Se utiliza un contenedor (container) para organizar el contenido con márgenes y alineación.
- **Tarjeta de Contenido:** Se aplica la clase card para crear una tarjeta que contiene la información de la mascota, con un fondo y sombra personalizados.
- **Uso de Filas y Columnas:** Se utiliza el sistema de grid de Bootstrap (row y col-md) para organizar el contenido en columnas responsivas.
- **Formularios:** Se utilizan clases de Bootstrap para los campos de formulario (form-group y form-control) para asegurar una presentación coherente.
- **Botones:** Se aplican clases de Bootstrap para crear botones con estilos predefinidos, además de estilos personalizados para los colores y sombras.
- **Inclusión de Scripts de Bootstrap:** Se incluyen scripts de jQuery, Popper.js y Bootstrap para habilitar componentes interactivos y la funcionalidad responsiva.

```
<title>Gestión de Mascotas - Editar</title>
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
<style>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<!-- Barra de navegación -->
<nav class="navbar navbar-expand-lg navbar custom-navbar">
 Tail Tails Rescue
 <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarNav" aria-controls="navbarNav"
 aria-expanded="false" aria-label="Toggle navigation">

 </button>
 <div class="collapse navbar-collapse" id="navbarNav">
 <ul class="navbar-nav mr-auto">
 <li class="nav-item active">
 Mascotas

 <li class="nav-item">
 Adoptantes

 <li class="nav-item">
 Solicitudes

 <!-- Imagen de usuario -->
 <ul class="navbar-nav">
 <li class="nav-item">

 </div>
</nav>
```

```
<div class="container mt-5">
 <div class="d-flex justify-content-center align-items-center mb-4">
 <h2 class="text-center" style="text-shadow: 0 3px 2px #000, 0 0, 0.2px; font-family: 'Arial', sans-serif; font-size: 3rem;">Gestionar Información de la Mascota</h2>
 </div>

 <!-- Tarjeta para contener la imagen y las columnas -->
 <div class="card custom-card"> <!-- Aplica la clase personalizada aquí -->
 <div class="card-body">
 <form #mascotaForm="ngForm" (ngSubmit)="onSubmit()">
 <div class="row">
 <!-- Columna para la imagen -->
 <div class="col-md-4 text-center">
 <img class="card-img-top" [src]="(mascota.species.toLowerCase() === 'perro') ? 'https://www.svgrepo.com/show/405231/dog-face.svg' :
 (mascota.species.toLowerCase() === 'gato') ? 'https://www.svgrepo.com/show/401273/cat-face.svg' :
 (mascota.species.toLowerCase() === 'hamster') ? 'https://www.svgrepo.com/show/405777/hamster.svg' :
 (mascota.species.toLowerCase() === 'conejo') ? 'https://www.svgrepo.com/show/398149/rabbit-face.svg' :
 (mascota.species.toLowerCase() === 'cuy' || mascota.species.toLowerCase() === 'conejillo de indias') ? 'https://img.freepik.com/premium-vector/guinea-pig-illustration_1142-206621.jpg?st=1727738059-exp=1727741659' :
 (mascota.species.toLowerCase() === 'ratón') ? 'https://www.svgrepo.com/show/402148/rat.svg' :
 (mascota.species.toLowerCase() === 'rata') ? 'https://www.svgrepo.com/show/402177/mouse.svg' :
 (mascota.species.toLowerCase() === 'loro') ? 'https://www.svgrepo.com/show/397712/parrot.svg' :
 (mascota.species.toLowerCase() === 'canario') ? 'https://img.freepik.com/free-vector/yellow-bird-illustration_1142-206621.jpg?st=1727738059-exp=1727741659' :
 (mascota.species.toLowerCase() === 'periquito') ? 'https://www.svgrepo.com/show/417456/bird.svg' :
 (mascota.species.toLowerCase() === 'gallo') ? 'https://www.svgrepo.com/show/402629/rooster.svg' :
 (mascota.species.toLowerCase() === 'anguila') ? 'https://www.svgrepo.com/show/401438/eagle.svg' :
 (mascota.species.toLowerCase() === 'tortuga') ? 'https://www.svgrepo.com/show/402833/turtle.svg' :
 (mascota.species.toLowerCase() === 'serpiente') ? 'https://www.svgrepo.com/show/289236/snake.svg' :
 (mascota.species.toLowerCase() === 'iguana') ? 'https://img.freepik.com/free-vector/hand-drawn-cartoon-chameleon-illustration_23-2150387676.jpg?st=1727738059-exp=1727741659' :
 (mascota.species.toLowerCase() === 'cangrejo') ? 'https://www.svgrepo.com/show/530182/crab.svg' :
 (mascota.species.toLowerCase() === 'tarántula') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
 (mascota.species.toLowerCase() === 'araña') ? 'https://www.svgrepo.com/show/317393/tarantula.svg' :
 (mascota.species.toLowerCase() === 'hurón') ? 'https://img.freepik.com/free-vector/cute-otter-wild-animal-cartoon-sticker_1308-78999.jpg?st=1727738436-exp=1727741659' :
 (mascota.species.toLowerCase() === 'cacatúa') ? 'https://img.freepik.com/free-vector/cute-vector-illustration-parrot-kids_925324-31237.jpg?w=740' :
 (mascota.species.toLowerCase() === 'ave') ? 'https://www.svgrepo.com/show/484500/bird.svg' :
 (mascota.species.toLowerCase() === 'pez') ? 'https://www.svgrepo.com/show/317413/fish.svg' :
 (mascota.species.toLowerCase() === 'murciélago') ? 'https://www.svgrepo.com/show/425730/bat-scary.svg' :
 'https://www.svgrepo.com/show/366549/application-pet.svg'" alt="Imagen de {(mascota.name)}"
 style="height: 250px; width: 250px;">
 </div>
 </div>
```

```

<!-- Columna para las casillas -->
<div class="col-md-8">
 <div class="row">
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="name" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Nombre</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.name" name="name" id="name" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="species" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Especie</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.species" name="species" id="species" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 </div>
 <div class="row">
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="breed" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Raza</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.breed" name="breed" id="breed" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="age" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Edad</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.age" name="age" id="age" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 </div>
</div>
<div class="row">
 <div class="col-md-8">
 <div class="row">
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="name" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Nombre</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.name" name="name" id="name" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="species" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Especie</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.species" name="species" id="species" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 </div>
 <div class="row">
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="breed" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Raza</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.breed" name="breed" id="breed" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 <div class="col-md-6">
 <fieldset class="form-group">
 <label class="control-label" for="age" style="text-shadow: 0 3px 2px □rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Edad</label>
 <input type="text" class="form-control" required [(ngModel)]="mascota.age" name="age" id="age" style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 </fieldset>
 </div>
 </div>
 </div>
</div>
</div>

```

```

<!-- Botones de Guardar Cambios debajo -->
<div class="text-center mt-4">
 <button type="submit" class="btn btn-sm" [disabled]="!mascotaForm.form.valid"
 style="background-color: #80ffb9; color: □rgb(0, 0, 0); box-shadow: 0 2px 5px □rgba(0, 0, 0, 0.3); font-family: 'Arial', sans-serif; font-size: 1.2rem;" >
 Guardar Cambios
 </button>
 <button class="btn btn-sm" [routerLink]="['/mascotas']"
 style="background-color: #6faeff; color: □rgb(0, 0, 0); box-shadow: 0 2px 5px □rgba(0, 0, 0, 0.3); margin-left: 10px; font-family: 'Arial', sans-serif; font-size: 1.2rem;">
 Volver
 </button>
</div>

```

```

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.11.6/dist/umd/popper.min.js"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

```

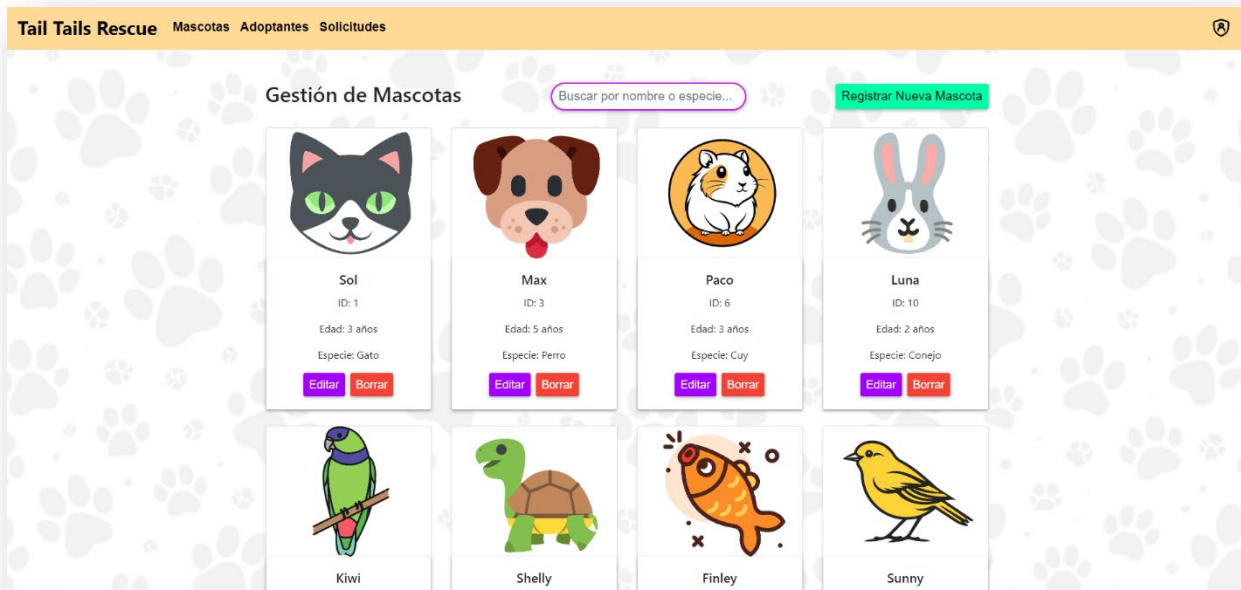
La implementación de Bootstrap ha facilitado de manera significativa el desarrollo del frontend de este proyecto, permitiendo una creación rápida y eficiente de interfaces responsivas y visualmente atractivas. Su uso se extiende a todos los módulos del sistema, incluidos el inicio, mascotas, adoptantes y solicitudes, asegurando una experiencia de usuario coherente y optimizada en cada sección. Gracias a las herramientas y componentes que ofrece Bootstrap, hemos podido concentrarnos en la funcionalidad y el diseño, mejorando así la usabilidad y la accesibilidad de la aplicación.

#### 4. Resultados de interfaz del proyecto Frontend Tail Tails Rescue:

- **Modulo Inicio.**
  - **Página de Inicio:**



- **Modulo Mascotas**
  - **Listar Mascotas:**



- **Crear - Editar Mascotas:**

**Tail Tails Rescue** Mascotas Adoptantes Solicitudes

### Gestionar Información de la Mascota



Nombre	<input type="text" value="Luna"/>	Especie	<input type="text" value="Conejo"/>
Raza	<input type="text" value="Holland Lop"/>	Edad	<input type="text" value="2"/>
Género	<input type="text" value="Hembra"/>	Tamaño	<input type="text" value="Pequeño"/>
Descripción	<input type="text" value="Color Blanco y Marrón"/>		

Guardar Cambios Volver

**Tail Tails Rescue** Mascotas Adoptantes Solicitudes

### Gestionar Información de la Mascota



Nombre	<input type="text"/>	Especie	<input type="text"/>
Raza	<input type="text"/>	Edad	<input type="text"/>
Género	<input type="text"/>	Tamaño	<input type="text"/>
Descripción	<input type="text"/>		

Guardar Cambios Volver



- **Modulo Adoptantes**
  - **Listar Adoptantes:**

Tail Tails Rescue Mascotas Adoptantes Solicitudes

### Gestión de Adoptantes

Buscar por nombre o identificación...

Registrar Nuevo Adoptante

#### Lista de Adoptantes

ID:	NOMBRE:	APELLIDO:	EMAIL:	TÉLEFONO:	IDENTIFICACIÓN:	DIRECCIÓN:	CIUDAD:		
1	Andrés	Ríos	andres.ríos@gmail.com	3137894560	192837465	Calle 123 #9-10	Cartagena	Editar	Borrar
3	Anita	García	ana.garcía@gmail.com	3104567890	123456789	Calle 45 #20-30	Bogotá	Editar	Borrar
4	Carlos	Pérez	carlos.perez@gmail.com	3117894561	987654321	Carrera 7 #15-67	Cali	Editar	Borrar
5	Lucía	Martínez	lucia.martínez@gmail.com	3124567891	564738291	Avenida Siempre Viva #101	Medellín	Editar	Borrar

- **Crear - Editar Adoptantes:**

Tail Tails Rescue Mascotas Adoptantes Solicitudes

### Editar Información del Adoptante



**Nombre**  
Andrés

**Apellido**  
Ríos

**Email**  
andres.ríos@gmail.com

**Teléfono**  
3137894560

**Identificación**  
192837465

**Dirección**  
Calle 123 #9-10

**Ciudad**  
Cartagena

Guardar Cambios

Volver

Tail Tails Rescue

Mascotas

Adoptantes

Solicitudes

### Editar Información del Adoptante



Nombre

Apellido

Email

Teléfono

Identificación

Dirección

Ciudad

Guardar Cambios

Volver

- **Modulo Solicitudes**
  - **Listar Solicitudes:**

Tail Tails Rescue

Mascotas

Adoptantes

Solicitudes

Gestión de Solicitudes

Buscar por nombre o identificación...

Registrar Nueva Solicitud

### Lista de Solicitudes de Adopción

ID:	ID Mascota:	ID Adoptante:	Fecha de Solicitud:	Estado:	Mascota:	Adoptante:	Comentarios:		
3	6	3	2024-09-25	Approved	Paco	Anita García	Ninguno	Editar	Borrar

ID:	ID Mascota:	ID Adoptante:	Fecha de Solicitud:	Estado:	Mascota:	Adoptante:	Comentarios:		
4	3	5	2024-09-25	Approved	Max	Lucía Martínez	Ninguno	Editar	Borrar



- **Crear - Editar Adoptantes:**

Tail Tails Rescue

Mascotas

Adoptantes

Solicitudes

Editar Información de la Solicitud



ID Mascota

6

ID Adoptante

3

Fecha de Solicitud

2024-09-25

Estado

Approved

Nombre Mascota

Paco

Nombra Adoptante

Anita García

Comentarios

Ninguno

Guardar Cambios

Volver

Tail Tails Rescue

Mascotas

Adoptantes

Solicitudes

Editar Información de la Solicitud



ID Mascota

ID Adoptante

Fecha de Solicitud

Estado

Nombre Mascota

Nombra Adoptante

Comentarios

Guardar Cambios

Volver