



Universidad de **Nariño**

TALLER UNIDAD 3 FRONTED.

ELABORADO POR:

JULIETH STEPHANIA HIDALGO NARVAEZ

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 UNIDAD 3 FRONTED.

ELABORADO POR:

JULIETH STEPHANIA HIDALGO NARVAEZ

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

TALLER UNIDAD 3 FRONTED.

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

1.1. Creación de componentes:

Los componentes son la base para la organización de la interfaz de usuario y la lógica del frontend. Dada la estructura de la base de datos y las entidades involucradas, es necesario crear componentes que reflejen las tablas y relaciones descritas:

- **Componente Pets:** Para gestionar la información de las mascotas, incluyendo su visualización, registro y edición. Este componente permitirá mostrar la lista de mascotas, así como los detalles individuales de cada una (nombre, especie, estado, tamaño, entre otras).
- **Componente Adopters:** Para gestionar a los adoptantes, donde se podrá registrar la información personal, sus motivaciones y el historial de solicitudes de adopción.
- **Componente AdoptionRequests:** Responsable de la gestión de las solicitudes de adopción, incluyendo la visualización del estado, la asignación a un empleado, y la aprobación o rechazo de las solicitudes.
- **Componente Employees:** Para la gestión de los empleados que supervisan y gestionan las solicitudes.

1.1.1. Ng generate component lista-editar:

Este comando crea un nuevo componente en Angular llamado lista-editar, generando automáticamente los archivos necesarios un archivo TypeScript para la lógica, un archivo HTML para la interfaz, un archivo CSS para los estilos y un archivo de pruebas unitarias, además, actualiza el módulo correspondiente para registrar el componente, facilitando su integración y uso dentro de la aplicación.

```
PS C:\Users\Julieth\Documents\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 (1422 bytes)
PS C:\Users\Julieth\Documents\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 (1550 bytes)
PS C:\Users\Julieth\Documents\Diplomado\frontEnd> |
```

```
PS C:\Users\Julieth\Documents\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 (1174 bytes)
PS C:\Users\Julieth\Documents\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 (1298 bytes)
PS C:\Users\Julieth\Documents\Diplomado\frontEnd> |
```

```
PS C:\Users\Julieth\Documents\Diplomado\frontEnd> ng generate component inicio
CREATE src/app/inicio/inicio.component.html (22 bytes)
CREATE src/app/inicio/inicio.component.spec.ts (620 bytes)
CREATE src/app/inicio/inicio.component.ts (209 bytes)
CREATE src/app/inicio/inicio.component.css (0 bytes)
UPDATE src/app/app.module.ts (1054 bytes)
PS C:\Users\Julieth\Documents\Diplomado\frontEnd> |
```

1.1.2. NgModules:

El ngModel será crucial en los formularios donde se recolecta información de los adoptantes, mascotas y solicitudes de adopción.

```
23 @NgModule({
24   declarations: [
25     AppComponent,
26     ListaMascotasComponent,
27     EditarMascotasComponent,
28     InicioComponent,
29     ListaAdoptantesComponent,
30     ListaSolicitudesComponent,
31     EditarAdoptantesComponent,
32     EditarSolicitudesComponent,
33     ListaEmpleadosComponent,
34     EditarEmpleadosComponent,
35   ],
36   imports: [
37     BrowserModule,
38     AppRoutingModule,
39     NgbModule,
40     FormsModule
41   ],
42   providers: [
43     MascotaService,
44     AdoptanteService,
45     SolicitudService,
46     EmpleadoService,
47     provideHttpClient(),
48     provideClientHydration()
49   ],
50   bootstrap: [AppComponent]
51 })
52 export class AppModule { }
```

Este código define el módulo principal de una aplicación Angular mediante el decorador `@NgModule`, en la sección `declarations`, se registran los componentes que forman parte de la aplicación como `ListaMascotasComponent`, `EditarMascotasComponent`, entre otros, que manejan las mascotas, adoptantes, solicitudes y empleados.

- En `imports`, se importan módulos esenciales como `BrowserModule`, `AppRoutingModule`, `NgbModule` (para Bootstrap) y `FormsModule` (para formularios).
- En `providers`, se inyectan servicios como `MascotaService`, `AdoptanteService`, `SolicitudService` y `EmpleadoService` que manejan la lógica de negocio, así como configuraciones para el cliente HTTP y la hidratación.

Finalmente, el módulo inicializa la aplicación con el componente principal `AppComponent` mediante `bootstrap`.

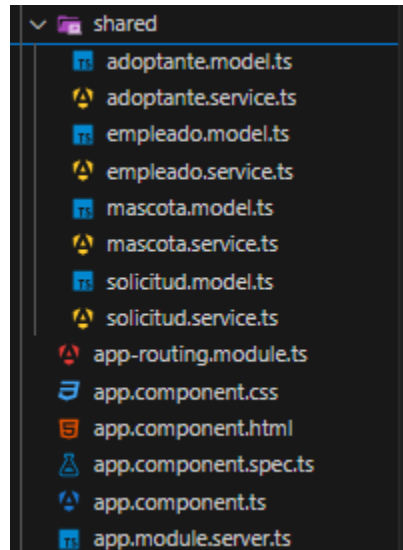
1.2. ROUTERLINK:

RouterLink permitirá la navegación entre los diferentes componentes de la aplicación, dado que se gestionan distintas entidades (mascotas, adoptantes, solicitudes, empleados), es fundamental ofrecer una experiencia de usuario fluida y sin recargas completas de página, utilizando rutas dinámicas, se lo utiliza también para redirigir desde la lista general de mascotas al componente que muestra los detalles de una mascota específica, para las solicitudes de adopción, se podría tener un enlace que permita a los empleados gestionar cada solicitud individual, accediendo a su detalle a través de la ruta correspondiente esto asegura una navegación eficiente entre los módulos del sistema, mejorando la usabilidad de la aplicación.

```
app.module.ts x
frontEnd > src > app > app.module.ts > ...
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 { InicioComponent } from './inicio/inicio.component';
12 import { ListaAdoptantesComponent } from './lista-adoptantes/lista-adoptantes.component';
13 import { ListaSolicitudesComponent } from './lista-solicitudes/lista-solicitudes.component';
14 import { EditarAdoptantesComponent } from './editar-adoptantes/editar-adoptantes.component';
15 import { EditarSolicitudesComponent } from './editar-solicitudes/editar-solicitudes.component';
16 import { AdoptanteService } from './shared/adoptante.service';
17 import { SolicitudService } from './shared/solicitud.service';
18 import { ListaEmpleadosComponent } from './lista-empleados/lista-empleados.component';
19 import { EditarEmpleadosComponent } from './editar-empleados/editar-empleados.component';
20 import { EmpleadoService } from './shared/empleado.service';
21
22
23 @NgModule({
```

1.3. SERVICIOS

Serán utilizados para abstraer la lógica de acceso a datos y comunicación con el backend. Cada tabla en la base de datos tendrá un servicio asociado que manejará las interacciones con la API.



```
PS C:\Users\Julieth\Documents\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\Julieth\Documents\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\Julieth\Documents\Diplomado\frontEnd> |
```

Los servicios listados, como adoptante.service.ts, empleado.service.ts, mascota.service.ts y solicitud.service.ts, manejan la lógica de negocio relacionada con adoptantes, empleados, mascotas y solicitudes de adopción en la aplicación, estos servicios permiten realizar operaciones CRUD (crear, leer, actualizar, eliminar) y gestionar los datos correspondientes, interactuando generalmente con una API para obtener o modificar la información en la base de datos, además centralizan el acceso a los datos, facilitando su uso en los distintos componentes de la aplicación y promoviendo una arquitectura más organizada y mantenible.

1.3.1. Adoptante.service.ts:

El código define un servicio en Angular llamado AdoptanteService que maneja las operaciones relacionadas con los adoptantes a través de la API de backend, usando HttpClient para hacer solicitudes HTTP. El servicio tiene varias funciones: obtenerAdoptantes para obtener la lista completa de adoptantes, obtenerAdoptante para buscar un adoptante por su ID, agregarAdoptante para crear un nuevo adoptante, actualizarAdoptante para modificar los datos de un adoptante existente, y borrarAdoptante para eliminar un adoptante según su ID. Todo esto se realiza conectándose a la API mediante la URL base configurada ('BASE_URL') y rutas específicas para cada operación.

```
adoptante.service.ts X
frontEnd > src > app > shared > adoptante.service.ts > ...
1 import { HttpClient } from '@angular/common/http';
2 import { Injectable } from '@angular/core';
3 import { AdoptanteModel } from '../adoptante.model';
4
5
6 @Injectable({
7   providedIn: 'root'
8 })
9 export class AdoptanteService {
10
11   BASE_URL='http://127.0.0.1:4000';
12   constructor(private http: HttpClient) {
13   }
14
15   //Lista completa de adoptantes
16   obtenerAdoptantes(){
17     | return this.http.get<AdoptanteModel[]>(`${this.BASE_URL}/adoptantes/buscar`);
18   }
19
20   //Buscar un Adoptante por ID
21   obtenerAdoptante(idAdoptante:string){
22     | return this.http.get<AdoptanteModel>(`${this.BASE_URL}/adoptantes/buscarId/${idAdoptante}`);
23   }
24
25   //Crear un Adoptante
26   agregarAdoptante(adoptante: AdoptanteModel){
27     | return this.http.post<string>(`${this.BASE_URL}/adoptantes/crear`,adoptante);
28   }
29
30   //Actualizar un Adoptante
31   actualizarAdoptante(adoptante: AdoptanteModel){
32     | return this.http.put<string>(`${this.BASE_URL}/adoptantes/actualizar/${adoptante.id}`,adoptante);
33   }
34
35   //Eliminar un Adoptante
36   borrarAdoptante(idAdoptante: string){
37     | return this.http.delete<string>(`${this.BASE_URL}/adoptantes/eliminar/${idAdoptante}`);
38   }
39
40
41 }
42
```


2. Uso de HTML 5 y JavaScript.

2.1. Adoptantes:

➤ Lista-adoptantes.component.ts:

El código define el componente ListaAdoptantesComponent, que se encarga de mostrar y gestionar una lista de adoptantes en la aplicación que implementa la interfaz OnInit para inicializar la carga de datos al momento de crear el componente, además, utiliza el servicio AdoptanteService para obtener la lista de adoptantes mediante una llamada a la API, la cual se asigna a la propiedad adoptantes como un Observable e incluye una funcionalidad para eliminar un adoptante específico a través del método borrarAdoptante, que llama al servicio para eliminar el adoptante y actualiza la lista tras la eliminación, manejando correctamente tanto el éxito como los posibles errores en la operación, este enfoque modular y basado en servicios promueve una arquitectura limpia y escalable, facilitando el mantenimiento del código.

```
lista-adoptantes.component.ts X
frontEnd > src > app > lista-adoptantes > lista-adoptantes.component.ts > ListaAdoptantesComponent
1  import { Component, OnInit } from '@angular/core';
2  import { Observable } from 'rxjs';
3  import { AdoptanteModel } from '../shared/adoptante.model';
4  import { AdoptanteService } from '../shared/adoptante.service';
5
6  @Component({
7    selector: 'app-lista-adoptantes',
8    templateUrl: './lista-adoptantes.component.html',
9    styleUrls: ['./lista-adoptantes.component.css'] // Cambia "styleUrl" por "styleUrls"
10 })
11 export class ListaAdoptantesComponent implements OnInit {
12   adoptantes: Observable<AdoptanteModel[]> | undefined; // Mantener el nombre como 'adoptantes'
13   searchText: string = ''; // Agrega esta propiedad
14
15   constructor(private adoptanteService: AdoptanteService) {}
16
17   ngOnInit() {
18     this.adoptantes = this.adoptanteService.obtenerAdoptantes();
19   }
20
21   borrarAdoptante(idAdoptante: string) {
22     this.adoptanteService.borrarAdoptante(idAdoptante).subscribe({
23       next: data => {
24         console.log(`Registro Eliminado`);
25         this.ngOnInit();
26       },
27       error: err => {
28         console.log(`Error al eliminar Registro ${err}`);
29       }
30     });
31   }
32 }
33
```

➤ Lista-adoptante.component.html:

El código proporciona la estructura HTML y los estilos CSS para el componente ListaAdoptantes en una aplicación Angular, diseñado para mostrar un registro visualmente atractivo de adoptantes, la sección de estilo establece una imagen de fondo semitransparente y define el estilo de las tarjetas de adoptantes, que incluyen un efecto de escalado al pasar el cursor sobre ellas, en el cuerpo del componente, se presenta un encabezado centrado con un botón para registrar nuevos adoptantes, se utilizan directivas Angular como ngFor para iterar sobre la lista de adoptantes, mostrar sus detalles incluyendo ID, nombre, correo electrónico, motivo de adopción, en tarjetas estilizadas y se incluyen botones para editar y eliminar información de adoptantes, que permiten al usuario gestionar de manera eficiente los registros desde la interfaz.

Este diseño intuitivo y funcional mejora la experiencia del usuario en la gestión de adoptantes.

```
lista-adoptantes.component.html M X
frontEnd > src > app > lista-adoptantes > lista-adoptantes.component.html > style > .action-buttons
1 <style>
2   .background-image {
3     background-image: url('https://img.freepik.com/vector-gratis/ilustracion-personas-mascotas_23-2148998807.jpg?st=1727841251-exp=1727844851-hmac=4e41ebac63710820b1a7a4600ad9691235858a9ff86bd7a4cc185ffb23d97ca1&w=1800');
4     background-size: cover;
5     background-position: center;
6     position: absolute;
7     top: 0;
8     left: 0;
9     right: 0;
10    bottom: 0;
11    opacity: 0.1;
12    z-index: -1;
13  }
14
15  .adoptante-card {
16    background-color: rgba(246, 202, 255, 0.74);
17    border-radius: 10px;
18    box-shadow: 0 2px 5px rgba(77, 0, 100, 0.726);
19    padding: 15px;
20    margin-bottom: 20px;
21    transition: transform 0.2s;
22  }
23
24  .adoptante-card:hover {
25    transform: scale(1.02);
26  }
27
28  .adoptante-title {
29    font-weight: bold;
30    color: #000000;
31    margin-bottom: 5px;
32  }
33
34  .action-buttons {
35    display: flex;
36    justify-content: space-between;
37    margin-top: 15px;
38  }
39
40 </style>
41
42
43
44 <!-- Imagen de fondo -->
45 <div class="background-image"></div>
46
47
48 <div class="container mt-5">
49   <div class="text-center mb-4">
50     <h2 style="font-size: 40px; text-shadow: 0 2px 3px rgba(0, 0, 0, 0.3);">Registros de Adoptantes:</h2>
51
52     <!-- Botón Registrar Nuevo Adoptante centrado -->
53     <button class="btn btn-sm mt-2" [routerLink]="['/adoptantes/agregar']">
54       Registrar Nuevo Adoptante
55     </button>
56
57   </div>
```

```

61 <div class="row mt-3">
62   <div class="col-sm-12">
63     <div class="row">
64       <ng-container *ngFor="let adoptante of (adoptantes | async)">
65         <div class="col-md-4 col-sm-6">
66           <div class="adoptante-card">
67             <div class="adoptante-info">
68               <span class="adoptante-title">ID:</span>
69               {{ adoptante.id }}
70             </div>
71             <div class="adoptante-info">
72               <span class="adoptante-title">NOMBRE:</span>
73               {{ adoptante.first_name }}
74             </div>
75             <div class="adoptante-info">
76               <span class="adoptante-title">APELLIDO:</span>
77               {{ adoptante.last_name }}
78             </div>
79             <div class="adoptante-info">
80               <span class="adoptante-title">EMAIL:</span>
81               {{ adoptante.email }}
82             </div>
83             <div class="adoptante-info">
84               <span class="adoptante-title">TELEFONO:</span>
85               {{ adoptante.phone }}
86             </div>
87             <div class="adoptante-info">
88               <span class="adoptante-title">DIRECCION:</span>
89               {{ adoptante.address }}
90             </div>
91             <div class="adoptante-info">
92               <span class="adoptante-title">FECHA DE NACIMIENTO:</span>
93               {{ adoptante.birth_date }}
94             </div>
95             <div class="adoptante-info">
96               <span class="adoptante-title">OCUPACION:</span>
97               {{ adoptante.occupation }}
98             </div>
99             <div class="adoptante-info">
100              <span class="adoptante-title">TIPO DE HOGAR:</span>
101              {{ adoptante.housing_type }}
102            </div>
103            <div class="adoptante-info">
104              <span class="adoptante-title">TIENE OTRAS MASCOTAS:</span>
105              {{ adoptante.has_other_pets }}
106            </div>
107            <div class="adoptante-info">
108              <span class="adoptante-title">QUE OTRO TIPO DE MASCOTAS:</span>
109              {{ adoptante.type_of_other_pets }}
110            </div>
111            <div class="adoptante-info">
112              <span class="adoptante-title">MOTIVO DE ADOPCION:</span>
113              {{ adoptante.adoption_reason }}
114            </div>
115            <div class="action-buttons">

```

```

116             <!-- Botón Editar -->
117             <a class="btn btn-sm mr-2" [routerLink]="['/adoptantes/editar/', adoptante.id]"
118               style="background-color: #ff0037; color: white; box-shadow: 0 2px 5px #000; font-family: 'Arial', sans-serif; font-size: 1.1rem; margin-bottom: 30px;"
119               href="#">
120               Editar Información
121             </a>
122             <!-- Botón Borrar -->
123             <button class="btn btn-sm" (click)="borrarAdoptante(adoptante.id)"
124               style="background-color: #ff7a70; color: white; box-shadow: 0 2px 5px #000; font-family: 'Arial', sans-serif; font-size: 1.1rem; margin-bottom: 30px;"
125               href="#">
126               Eliminar
127             </button>
128           </div>
129         </div>
130       </ng-container>
131     </div>
132   </div>
133 </div>
134 </div>

```

INTERFAZ

Furry Friends

Nuestras Mascotas - Conoce a los Adoptantes - Solicitudes de Adopcion - Equipo Furry Friends

Registros de Adoptantes

Registrar Nuevo Adoptante

ID: 1
NOMBRE: José
APELLIDO: Martínez
EMAIL: jose.martinez@example.com
TELEFONO: 555-123-4567
DIRECCION: Boulevard de la Libertad 456
FECHA DE NACIMIENTO: 1978-11-30
OCUPACION: Arquitecto
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: true
QUE OTRO TIPO DE MASCOTAS: Gato
MOTIVO DE ADOPCION: Deseo adoptar un
compañero para mi gato

Editar Informacion

Eliminar

ID: 2
NOMBRE: Andrés
APELLIDO: López
EMAIL: andres.lopez@example.com
TELEFONO: 555-456-7890
DIRECCION: Calle de la Amistad 202
FECHA DE NACIMIENTO: 1982-09-12
OCUPACION: Chef
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: true
QUE OTRO TIPO DE MASCOTAS: Perro
MOTIVO DE ADOPCION: Quiero un
compañero para mis paseos

Editar Informacion

Eliminar

ID: 3
NOMBRE: Lucía
APELLIDO: Rodríguez
EMAIL: lucirodriguez854@example.com
TELEFONO: 555-345-6789
DIRECCION: Calle del Mar 308
FECHA DE NACIMIENTO: 1995-03-05
OCUPACION: Estudiante
TYPO DE HOGAR: Compartido
TIENE OTRAS MASCOTAS:
QUE OTRO TIPO DE MASCOTAS:
MOTIVO DE ADOPCION: Quiero tener una
mascota que me acompañe en mis estudios

Editar Informacion

Eliminar

ID: 6
NOMBRE: Julieth
APELLIDO: Hidalgo
EMAIL: juliethhidalgo@gmail.com
TELEFONO: 3163954825
DIRECCION: las lajas
FECHA DE NACIMIENTO: 2003-01-14
OCUPACION: estudiante
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: false
QUE OTRO TIPO DE MASCOTAS: Perro
MOTIVO DE ADOPCION:

Editar Informacion

Eliminar

Furry Friends

Nuestras Mascotas - Conoce a los Adoptantes - Solicitudes de Adopcion - Equipo Furry Friends

Registros de Adoptantes

Registrar Nuevo Adoptante

ID: 1
NOMBRE: José
APELLIDO: Martínez
EMAIL: jose.martinez@example.com
TELEFONO: 555-123-4567
DIRECCION: Boulevard de la Libertad 456
FECHA DE NACIMIENTO: 1978-11-30
OCUPACION: Arquitecto
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: true
QUE OTRO TIPO DE MASCOTAS: Gato
MOTIVO DE ADOPCION: Deseo adoptar un
compañero para mi gato

Editar Informacion

Eliminar

ID: 2
NOMBRE: Andrés
APELLIDO: López
EMAIL: andres.lopez@example.com
TELEFONO: 555-456-7890
DIRECCION: Calle de la Amistad 202
FECHA DE NACIMIENTO: 1982-09-12
OCUPACION: Chef
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: true
QUE OTRO TIPO DE MASCOTAS: Perro
MOTIVO DE ADOPCION: Quiero un
compañero para mis paseos

Editar Informacion

Eliminar

ID: 3
NOMBRE: Lucía
APELLIDO: Rodríguez
EMAIL: lucirodriguez854@example.com
TELEFONO: 555-345-6789
DIRECCION: Calle del Mar 308
FECHA DE NACIMIENTO: 1995-03-05
OCUPACION: Estudiante
TYPO DE HOGAR: Compartido
TIENE OTRAS MASCOTAS:
QUE OTRO TIPO DE MASCOTAS:
MOTIVO DE ADOPCION: Quiero tener una
mascota que me acompañe en mis estudios

Editar Informacion

Eliminar

ID: 6
NOMBRE: Julieth
APELLIDO: Hidalgo
EMAIL: juliethhidalgo@gmail.com
TELEFONO: 3163954825
DIRECCION: las lajas
FECHA DE NACIMIENTO: 2003-01-14
OCUPACION: estudiante
TYPO DE HOGAR: Casa
TIENE OTRAS MASCOTAS: false
QUE OTRO TIPO DE MASCOTAS: Perro
MOTIVO DE ADOPCION:

Editar Informacion

Eliminar



➤ Editar-mascotas.component.ts:

El código corresponde al componente `EditarAdoptantesComponent`, el cual permite editar o crear un adoptante en la aplicación de gestión de adopciones. Implementa la interfaz `OnInit` para manejar la lógica de inicialización. En el constructor, se inyectan los servicios `AdoptanteService`, `ActivatedRoute`, y `Router`. Durante la inicialización (`ngOnInit`), se obtiene el ID del adoptante desde los parámetros de la ruta, y si este ID está presente, se realiza una llamada al servicio para obtener la información del adoptante correspondiente y cargarla en el formulario. Si no hay ID, se indica que se está creando un nuevo adoptante. La función `onSubmit` gestiona el envío del formulario; si el adoptante tiene un ID, se llama al método para actualizarlo, de lo contrario, se utiliza el método para agregar un nuevo adoptante. Después de cada operación, se redirige a la lista de adoptantes y se registran los posibles errores en la consola.

```
editar-adoptantes.component.ts X
frontEnd > src > app > editar-adoptantes > editar-adoptantes.component.ts > EditarAdoptantesComponent > ngOnInit
1 import { Component, OnInit } from '@angular/core';
2 import { AdoptanteModel } from '../shared/adoptante.model';
3 import { AdoptanteService } from '../shared/adoptante.service';
4 import { ActivatedRoute, Router } from '@angular/router';
5
6 @Component({
7   selector: 'app-editar-adoptantes',
8   templateUrl: './editar-adoptantes.component.html',
9   styleUrls: ['./editar-adoptantes.component.css']
10 })
11 export class EditarAdoptantesComponent implements OnInit{
12
13   idAdoptante= '';
14   adoptante = new AdoptanteModel(
15     '', '', '', '', '', '', '', '', '', '', '', ''
16   );
17
18   constructor(private adoptanteService: AdoptanteService, private route: ActivatedRoute, private router: Router){
19   }
20
21   ngOnInit(){
22     this.idAdoptante=this.route.snapshot.params['idAdoptante'];
23     console.log(`El idAdoptante es ${this.idAdoptante}`);
24
25     if(this.idAdoptante){
26       //Viene de Editar
27       console.log('La solicitud viene de Editar');
28       this.adoptanteService.obtenerAdoptante(this.idAdoptante).subscribe({
29         next: data=>{
30           console.log(data);
31           this.adoptante=data;
32           console.log(this.adoptante);
33         },
34         error: err=>{
35           console.log(`Error ${err}`);
36         }
37       });
38     }
39
40   }
41   else{
42     console.log('La solicitud viene de Nueva Adoptante');
43   }
44
45 }
```

```

47     onSubmit(){
48         console.log("On Submit");
49         //Viene de Editar
50         if(this.adoptante.id){
51             this.adoptanteService.actualizarAdoptante(this.adoptante).subscribe({
52                 next: data=>{
53                     console.log(data);
54                     this.router.navigate(['/adoptantes']);
55                 },
56                 error: err=>{
57                     console.log(`Error al actualizar ${err}`);
58                 }
59             });
60         }
61     }
62     else{
63         //Viene de Nueva Adoptante
64         this.adoptanteService.agregarAdoptante(this.adoptante).subscribe({
65             next: data=>{
66                 console.log(data);
67                 this.router.navigate(['/adoptantes']);
68             },
69             error: err=>{
70                 console.log(`Error al Agregar ${err}`);
71             }
72         });
73     }
74 }
75 }
76 }

```

➤ **Editar-mascotas.component.html:**

Este código implementa un formulario dinámico para la edición y creación de adoptantes el componente EditarAdoptantesComponent sigue un patrón estándar, facilitando la mantenibilidad y reutilización del componente en distintas partes de la aplicación.

El uso de binding bidireccional a través de `ngModel` asegura que cualquier cambio en los campos del formulario se refleje de inmediato en el modelo de datos del adoptante, y viceversa. Este enfoque es crucial para aplicaciones que dependen de datos actualizados en tiempo real, ya que permite mantener sincronizados el estado del frontend y el backend de manera eficiente, el código también incluye validación de formularios en tiempo real, bloqueando el botón de envío si los campos no cumplen con las reglas de validación. Este mecanismo refuerza la integridad de los datos desde el frontend, reduciendo errores y posibles problemas en el backend, además de mejorar la experiencia de usuario al proporcionar retroalimentación inmediata.

Finalmente, el diseño del formulario está optimizado para ofrecer una experiencia de usuario moderna y agradable, los estilos CSS personalizados, como el uso de una imagen de fondo y una tarjeta con sombras y bordes redondeados, junto con el diseño responsive basado en la grilla de Bootstrap, garantizan que la interfaz sea visualmente atractiva y accesible en distintos dispositivos, esto demuestra una atención cuidadosa tanto a la funcionalidad como a la estética, factores clave para mejorar la interacción del usuario con la aplicación.

CODIGO[illegible]


```

58 <!-- Columna para las casillas -->
59 <div class="col-md-8">
60   <div class="row">
61     <div class="col-md-6">
62       <fieldset class="form-group">
63         <label class="control-label" for="first_name"
64           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Nombre</label>
65         <input type="text" class="form-control" required [(ngModel)]="adoptante.first_name"
66           name="first_name" id="first_name"
67           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
68       </fieldset>
69     </div>
70     <div class="col-md-6">
71       <fieldset class="form-group">
72         <label class="control-label" for="last_name"
73           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Apellido</label>
74         <input type="text" class="form-control" required [(ngModel)]="adoptante.last_name"
75           name="last_name" id="last_name"
76           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
77       </fieldset>
78     </div>
79   </div>
80
81   <div class="row">
82     <div class="col-md-6">
83       <fieldset class="form-group">
84         <label class="control-label" for="email"
85           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Email</label>
86         <input type="email" class="form-control" required [(ngModel)]="adoptante.email"
87           name="email" id="email"
88           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
89       </fieldset>
90     </div>
91     <div class="col-md-6">
92       <fieldset class="form-group">
93         <label class="control-label" for="phone"
94           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Teléfono</label>
95         <input type="text" class="form-control" required [(ngModel)]="adoptante.phone"
96           name="phone" id="phone"
97           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
98       </fieldset>
99     </div>
100   </div>
101
102   <div class="row">
103     <div class="col-md-6">
104       <fieldset class="form-group">
105         <label class="control-label" for="address"
106           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Dirección</label>
107         <input type="text" class="form-control" required [(ngModel)]="adoptante.address"
108           name="address" id="address"
109           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
110       </fieldset>

```

```

112     <div class="col-md-6">
113       <fieldset class="form-group">
114         <label class="control-label" for="birth_date"
115           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Fecha
116           de Nacimiento</label>
117         <input type="date" class="form-control" required [(ngModel)]="adoptante.birth_date"
118           name="birth_date" id="birth_date"
119           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
120       </fieldset>
121     </div>
122   </div>
123
124   <div class="row">
125     <div class="col-md-6">
126       <fieldset class="form-group">
127         <label class="control-label" for="occupation"
128           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Ocupación</label>
129         <input type="text" class="form-control" required [(ngModel)]="adoptante.occupation"
130           name="occupation" id="occupation"
131           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
132       </fieldset>
133     </div>
134     <div class="col-md-6">
135       <fieldset class="form-group">
136         <label class="control-label" for="housing_type"
137           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">Tipo
138           de Vivienda</label>
139         <input type="text" class="form-control" required
140           [(ngModel)]="adoptante.housing_type" name="housing_type" id="housing_type"
141           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
142       </fieldset>
143     </div>
144   </div>
145
146   <div class="row">
147     <div class="col-md-6">
148       <fieldset class="form-group">
149         <label class="control-label" for="has_other_pets"
150           style="text-shadow: 0 3px 2px □ rgba(0, 0, 0, 0.2); font-family: 'Arial', sans-serif; font-size: 1.2rem;">¿Tiene
151           otras mascotas?</label>
152         <input type="text" class="form-control" required
153           [(ngModel)]="adoptante.has_other_pets" name="has_other_pets" id="has_other_pets"
154           style="font-family: 'Arial', sans-serif; font-size: 1.2rem;">
155       </fieldset>
156     </div>

```



```

157 <div class="col-md-6">
158 <fieldset class="form-group">
159 <label class="control-label" for="type_of_other_pets">
160 <div class="form-control">
161 <input type="text" class="form-control" [(ngModel)]="adoptante.type_of_other_pets">
162 </div>
163 </label>
164 </div>
165 </div>
166 </div>
167 </div>
168 </div>
169 </div>
170 <!-- Botones de Guardar Cambios debajo -->
171 <div class="text-center mt-4">
172 <button type="submit" class="btn btn-sm" [disabled]="!adoptanteForm.form.valid">
173 <div class="form-control">
174 <input type="text" class="form-control" [(ngModel)]="adoptante.type_of_other_pets">
175 </div>
176 </div>
177 </div>
178 </div>
179 </div>
180 </div>
181 </div>
182 </div>
183 </div>
184 </div>

```

INTERFAZ AL EDITAR UN ADOPTANTE

ADOPTANTE ACTUAL

ID: 6

NOMBRE: Julieth

APELLIDO: Hidalgo

EMAIL: juliethhidalgo@gmail.com

TELEFONO: 3163954825

DIRECCION: las lajas

FECHA DE NACIMIENTO: 2003-01-14

OCUPACION: estudiante

TYPO DE HOGAR: Casa

TIENE OTRAS MASCOTAS:

QUE OTRO TIPO DE MASCOTAS: Perro

MOTIVO DE ADOPCION:

Editar Informacion

Eliminar

EDICION DEL ADOPTANTE

Nombre	Julieth	Apellido	Hidalgo
Email	juliethhidalgo@gmail.com	Teléfono	3163954825
Dirección	Ipiales	Fecha de Nacimiento	14/01/2003
Ocupación	estudiante	Tipo de Vivienda	Casa
¿Tiene otras mascotas?	true	Tipo de otras mascotas	Perro

Guardar Informacion

Regresar a Adoptantes

ID: 6

NOMBRE: Julieth

APELLIDO: Hidalgo

EMAIL: juliethhidalgo@gmail.com

TELEFONO: 3163954825

DIRECCION: Ipiales

FECHA DE NACIMIENTO: 2003-01-14

OCUPACION: estudiante

TYPO DE HOGAR: Casa

TIENE OTRAS MASCOTAS: true

QUE OTRO TIPO DE MASCOTAS: Perro

MOTIVO DE ADOPCION:

Editar Informacion

Eliminar

➤ **Inicio.component.ts:**

Este código crea un componente en Angular llamado InicioComponent, que representa la página de inicio de la aplicación, utiliza el decorador `@Component` para definir el selector `app-inicio`, que permite insertar el componente en el HTML, y establece los archivos de plantilla y estilos correspondientes, el componente en su estado actual no tiene lógica funcional, simplemente actúa como un contenedor para la vista y los estilos asociados a la página de inicio.

```

1  import { Component } from '@angular/core';
2
3  @Component({
4    selector: 'app-inicio',
5    templateUrl: './inicio.component.html',
6    styleUrls: ['./inicio.component.css']
7  })
8  export class InicioComponent {
9
10 }

```

➤ **Inicio.components.html:**

Este código define el estilo y la estructura de una página web de adopción de mascotas, la página incluye una imagen de fondo de huellas de animales, un "hero section" destacada con una imagen de perros y gatos, acompañada de un mensaje motivacional y un botón para ver las mascotas disponibles para adopción. También cuenta con una sección de mascotas que ya han sido adoptadas, donde se muestran tarjetas con imágenes y descripciones de las mascotas rescatadas, incluyendo detalles sobre su personalidad y el éxito de su adopción, el diseño es responsivo, lo que permite que se vea bien en diferentes tamaños de pantallas, y se aplican sombras, bordes redondeados y efectos hover en los botones para mejorar la experiencia visual y de usuario, la página utiliza una paleta de colores suaves y cálidos, lo que da un toque acogedor y amigable.

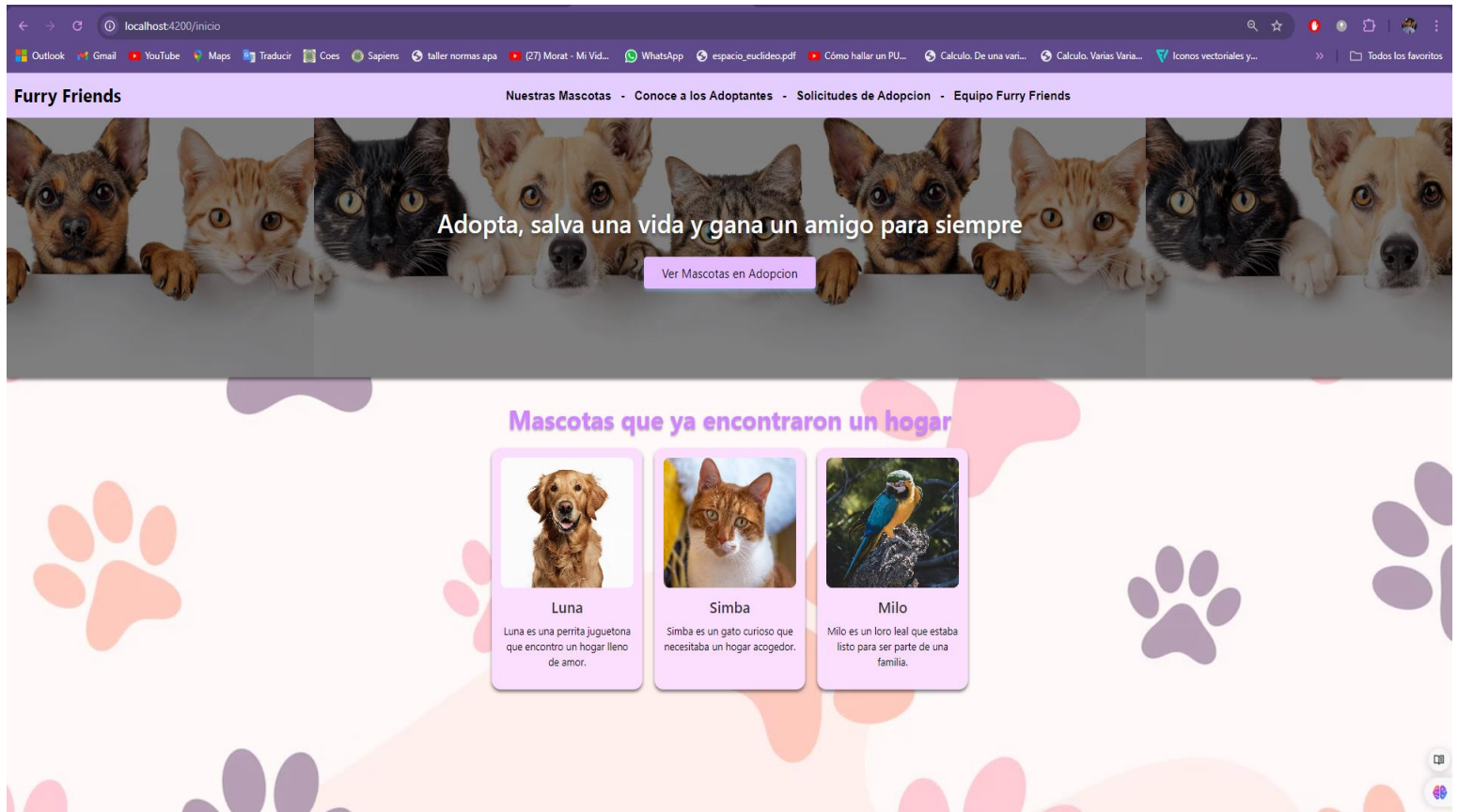
CODIGO

```

35 #!csscomponent.html X
36
37 @font-face {
38   font-family: 'serif';
39   src: url('https://www.dafont.com/serif.woff2');
40 }
41
42 .background-image {
43   background-image: url('https://www.dafont.com/serif.woff2');
44   background-size: cover;
45   position: absolute;
46   top: 0;
47   left: 0;
48   right: 0;
49   bottom: 0;
50   opacity: 0.5;
51   z-index: -1;
52 }
53
54 /* Sección Hero */
55
56 .hero-section {
57   position: relative;
58   background-image: url('https://www.dafont.com/serif.woff2');
59   background-size: cover;
60   background-position: center;
61   height: 100vh;
62   color: white;
63   text-align: center;
64   display: flex;
65   justify-content: center;
66   align-items: center;
67   box-shadow: 0 0 0 0px;
68   margin-bottom: 20px;
69 }
70
71 .hero-section:after {
72   content: '';
73   position: absolute;
74   top: 0;
75   left: 0;
76   right: 0;
77   bottom: 0;
78   background-color: rgba(0, 0, 0, 0.5);
79   z-index: 1;
80 }

```


INTERFAZ



3. Estilos CSS, interface ordenada estructurada y agradable para el usuario final.

3.1. Inicio de la página.

- El diseño es visualmente atractivo, con un enfoque centrado en la adopción y el rescate de animales.
- **Encabezado:**
 - El encabezado superior es de color púrpura y contiene el logo o nombre "Furry Friends" en la esquina izquierda, a la derecha se despliega un menú de navegación con varias opciones: "Nuestras Mascotas", "Conoce a los Adoptantes", "Solicitudes de Adopción", y "Equipo Furry Friends". estas pestañas permiten explorar diferentes secciones del sitio relacionadas con la adopción de mascotas.

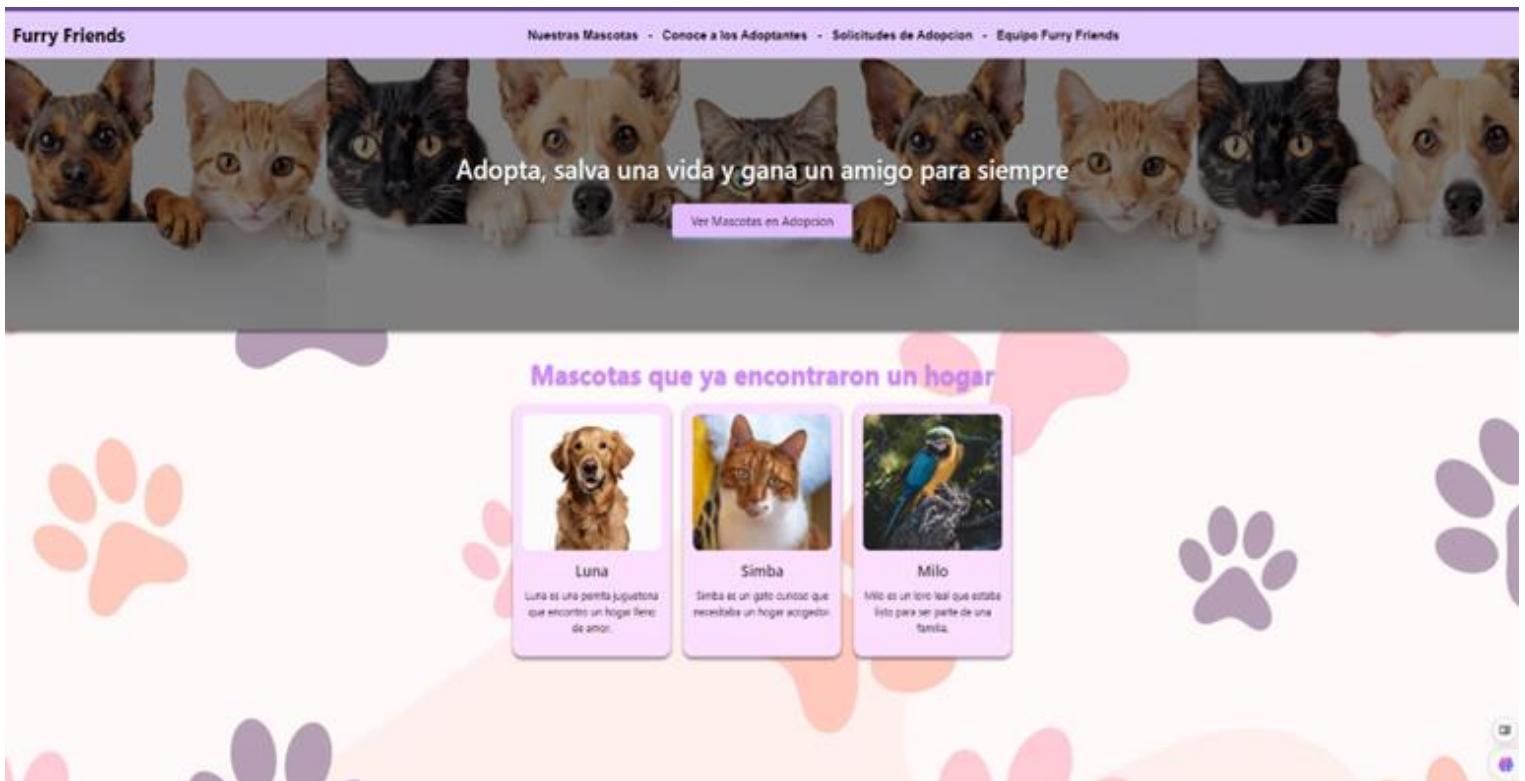
➤ Sección de Mascotas Adoptadas:

- Debajo de la sección héroe, se encuentra otra sección con el título "Mascotas que ya encontraron un hogar", en un color púrpura brillante que destaca del fondo.
- La sección muestra tres tarjetas de mascotas que han sido adoptadas, con imágenes de cada una: "Luna" (un perro), "Simba" (un gato), y "Milo" (un loro). Cada tarjeta incluye una pequeña descripción de la mascota y cómo encontraron su nuevo hogar.

➤ Estética y Estilos:

- El fondo de la página presenta un patrón de huellas de animales en colores pastel (rosado, púrpura, gris) que añade un toque divertido y coherente con la temática de adopción de animales.
- Los colores predominantes incluyen varios tonos de púrpura y rosa, creando una atmósfera acogedora y amigable, ideal para un sitio web que busca promover la adopción de mascotas.
- Las tarjetas de mascotas están bien estructuradas, con bordes redondeados y sombras que las hacen resaltar visualmente.

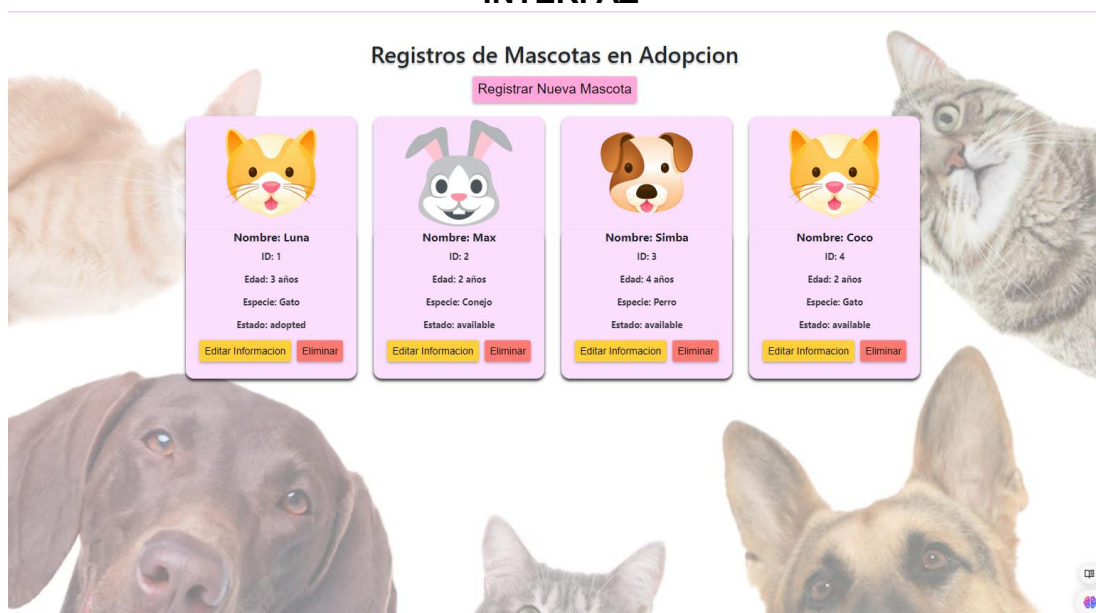
En conjunto, este diseño presenta un enfoque amigable y accesible para invitar a los visitantes a participar en la adopción de mascotas, con una navegación clara y elementos visuales que refuerzan la misión del sitio.



3.2. Mascotas

- Cada mascota está representada en un formato de tarjeta que incluye la siguiente información.
 - **Nombre:** Identificación del animal.
 - **ID:** Un número único asignado a cada mascota, crucial para la gestión de datos y evitar duplicados.
 - **Edad:** Indica la edad del animal, lo que puede influir en la decisión de adopción.
 - **Especie:** Especifica el tipo de animal (gato, perro, conejo), lo que ayuda a los adoptantes a filtrar según sus preferencias.
 - **Estado:** Indica si la mascota está disponible para adopción o ya ha sido adoptada, lo que es fundamental para la gestión del inventario de mascotas.
- **Funcionalidades**
 - **Editar Información:** Permite modificar los datos de cada mascota. Esta función es importante para mantener la precisión de la información.
 - **Eliminar:** Opción para eliminar un registro, lo que es útil para gestionar mascotas que ya no están disponibles o han sido adoptadas.
- **Diseño Visual**
 - **Colores y Gráficos:** La interfaz utiliza un esquema de colores suaves (fondo rosa) y gráficos de mascotas que hacen la aplicación visualmente atractiva y amigable. Esto puede influir positivamente en la experiencia del usuario.
 - **Imágenes de Fondo:** Las imágenes de mascotas en el fondo refuerzan el tema de la adopción y crean un ambiente acogedor.

INTERFAZ



REGISTRO DE MASCOTAS

Información de la Mascota




Nombre	Especie
Raza	Edad
Genero	Tamano
Peso	Fecha de Ingreso
Vacunado	Descripcion

[Guardar Informacion](#) [Regresar a Mascotas](#)

EDICION DE LA MASCOTA

Información de la Mascota



Nombre	Especie
Luna	Gato
Raza	Edad
Siamés	3
Genero	Tamano
femenino	pequeño
Peso	Fecha de Ingreso
4.20	2024-08-20
Vacunado	Descripcion
true	Gata tranquila y cariñosa, ideal para interiores

[Guardar Informacion](#) [Regresar a Mascotas](#)

3.3. Adoptantes

- **Registros de Adoptantes:** Indica claramente que la aplicación se centra en la gestión de información de adoptantes.

➤ Botón de Registro

- **Registrar Nuevo Adoptante:** Este botón permite al usuario añadir un nuevo registro de adoptante, lo que implica una funcionalidad de entrada de datos para mantener la base de datos actualizada.

➤ Entradas de Adoptantes

Cada adoptante está representado en un formato de tarjeta que incluye la siguiente información:

- **ID:** Un número único asignado a cada adoptante, esencial para la gestión de datos y evitar duplicados.
- **Nombre y Apellido:** Identificación del adoptante, lo que es fundamental para la comunicación y seguimiento.
- **Email:** Proporciona un medio de contacto directo.
- **Teléfono:** Información adicional de contacto, importante para la comunicación rápida.
- **Dirección:** Ubicación del adoptante, relevante para la logística de la adopción.
- **Fecha de Nacimiento:** Permite verificar la edad del adoptante, lo que puede ser un criterio en el proceso de adopción.
- **Ocupación:** Información que puede ser relevante para evaluar la capacidad del adoptante para cuidar de una mascota.
- **Tipo de Hogar:** Indica si el adoptante vive en una casa o en un lugar compartido, lo que puede influir en la elección de la mascota.
- **Tiene Otras Mascotas:** Indica si el adoptante ya tiene mascotas, lo que es importante para la integración de nuevos animales.
- **Qué Otro Tipo de Mascotas:** Especifica el tipo de mascota que el adoptante ya posee o desea tener.
- **Motivo de Adopción:** Proporciona contexto sobre las intenciones del adoptante, lo que puede ser útil para la selección de la mascota adecuada.

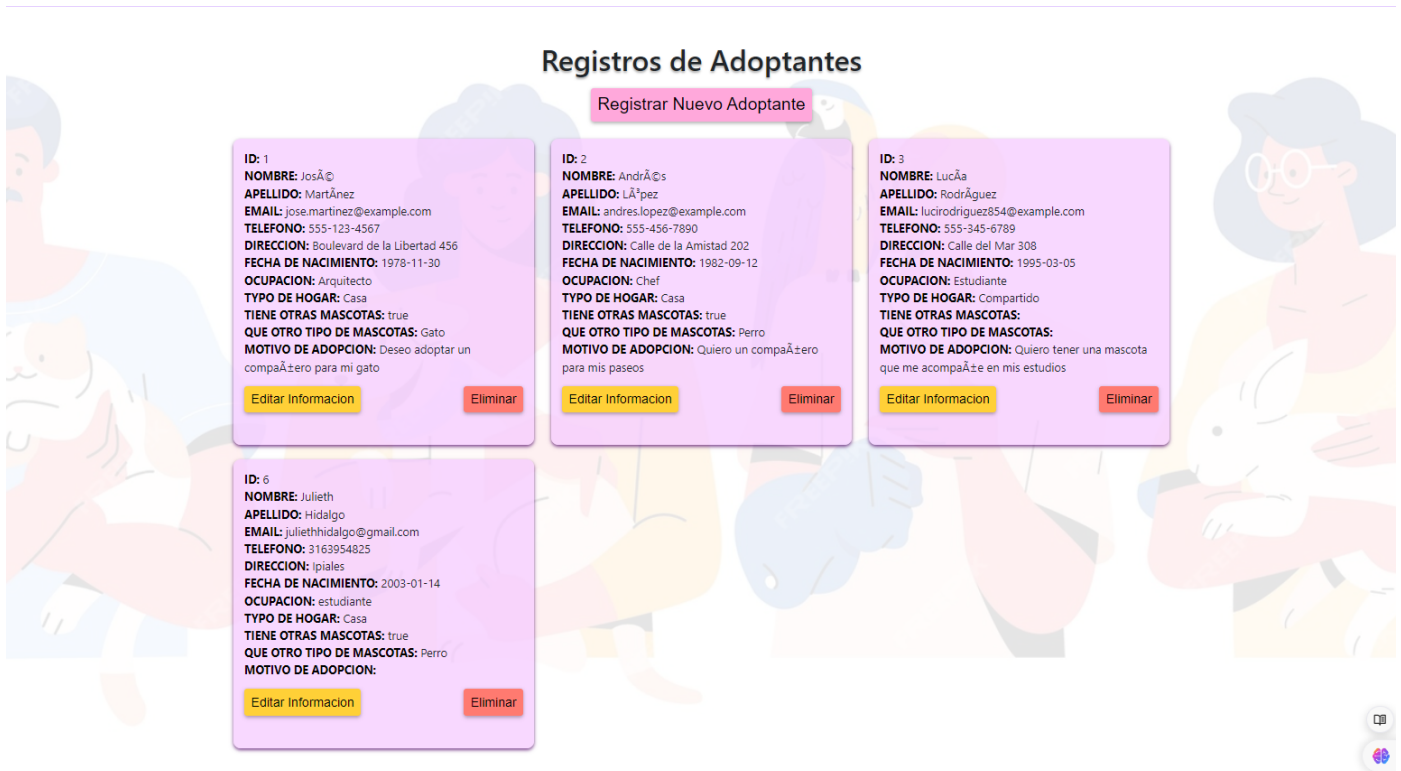
➤ Funcionalidades

- **Editar Información:** Permite modificar los datos de cada adoptante. Esta función es crucial para mantener la precisión de la información y adaptarse a cambios en la situación del adoptante.
- **Eliminar:** Opción para eliminar un registro, útil para gestionar adoptantes que ya no están interesados o que han completado el proceso de adopción.

➤ Diseño Visual

- **Colores y Gráficos:** La interfaz utiliza un esquema de colores suaves (fondo rosa) y gráficos que representan a personas con mascotas, creando un ambiente acogedor y amigable.
- **Imágenes de Fondo:** Las imágenes refuerzan el tema de la adopción y hacen la aplicación más atractiva visualmente.

INTERFAZ



REGISTRO DE ADAPTANTES

Informaci3n del Adoptante

Nombre	Apellido
Email	Tel3fono
Direcci3n	Fecha de Nacimiento
Ocupaci3n	Tipo de Vivienda
¿Tiene otras mascotas?	Tipo de otras mascotas

Guardar Informacion Regresar a Adoptantes

3.4. Solicitud de adoptantes

➤ Título de la Página:

- **Registros de Solicitudes:** Indica claramente que la aplicación se centra en la gestión de solicitudes de adopción.

➤ Botón de Registro:

- **Registrar Nueva Solicitud:** Este botón permite al usuario añadir un nuevo registro de solicitud, lo que implica una funcionalidad de entrada de datos para mantener la base de datos actualizada.

➤ Entradas de Solicitudes:

Cada solicitud está representada en un formato de tarjeta que incluye la siguiente información:

- **ID:** Un número único asignado a cada solicitud, esencial para la gestión de datos y evitar duplicados.
- **ID Mascota:** Identificación de la mascota para la cual se está realizando la solicitud, lo que permite vincular la solicitud con el registro de mascotas.
- **ID Adoptante:** Identificación del adoptante que realiza la solicitud, lo que es fundamental para la comunicación y seguimiento.
- **Fecha Solicitud:** Indica la fecha en que se realizó la solicitud, lo que es importante para el seguimiento del proceso.
- **Estado:** Muestra el estado actual de la solicitud (en este caso, "pending"), lo que permite a los administradores gestionar el flujo de solicitudes.
- **Fecha Aprobación:** Campo que puede ser utilizado para registrar la fecha en que se aprueba la solicitud, aunque actualmente está vacío.
- **ID Empleado:** Identificación del empleado que gestiona la solicitud, lo que es útil para la asignación de responsabilidades.

➤ Funcionalidades

- **Editar Información:** Permite modificar los datos de cada solicitud, esta función es crucial para mantener la precisión de la información y adaptarse a cambios en el estado de la solicitud.
- **Eliminar:** Opción para eliminar un registro, útil para gestionar solicitudes que ya no son relevantes o que han sido canceladas.

➤ Diseño Visual

- **Colores y Gráficos:** La interfaz utiliza un esquema de colores suaves (fondo rosa) y gráficos que representan a personas, creando un ambiente acogedor y amigable.
- **Imágenes de Fondo:** Las imágenes refuerzan el tema de la adopción y hacen la aplicación más atractiva visualmente.

INTERFAZ



REGISTRO DE SOLICITUD



3.5. Empleados

➤ **Título de la Página:**

- **Registros de Empleados:** Indica claramente que la aplicación se centra en la gestión de información de los empleados.

➤ **Botón de Registro:**

- **Registrar Nuevo Empleado:** Este botón permite al usuario añadir un nuevo registro de empleado, lo que implica una funcionalidad de entrada de datos para mantener la base de datos actualizada.

➤ **Entradas de Empleados**

Cada empleado está representado en un formato de tarjeta que incluye la siguiente información:

- **ID:** Un número único asignado a cada empleado, esencial para la gestión de datos y evitar duplicados.
- **Nombre Completo:** Identificación del empleado, fundamental para la comunicación y gestión interna.
- **Cargo:** Especifica el puesto que ocupa el empleado dentro de la organización.
- **Teléfono:** Proporciona un medio de contacto directo.
- **Email:** Información adicional de contacto, importante para la comunicación rápida.
- **Fecha de Contrato:** Indica la fecha en que el empleado fue contratado, lo que es relevante para la gestión de recursos humanos.

➤ **Funcionalidades**

- **Editar Información:** Permite modificar los datos de cada empleado, esta función es crucial para mantener la precisión de la información y adaptarse a cambios en la situación laboral del empleado.
- **Eliminar:** Opción para eliminar un registro, útil para gestionar empleados que ya no forman parte de la organización.

➤ **Diseño Visual**

- **Colores y Gráficos:** La interfaz utiliza un esquema de colores suaves (fondo rosa) y gráficos que representan a personas en un entorno laboral, creando un ambiente acogedor y profesional.
- **Imágenes de Fondo:** Las imágenes refuerzan el tema de la gestión de empleados y hacen la aplicación más atractiva visualmente.

INTERFAZ

Registros de Empleados

Registrar Nuevo Empleado

ID: 1

NOMBRE COMPLETO: Carlos Rodríguez

CARGO: Gerente de Marketing

TELEFONO: 555-987-6543

EMAIL: carlos.rodriguez@example.com

FECHA CONTRATO: 2022-03-15

Editar Informacion Eliminar

ID: 2

NOMBRE COMPLETO: Ana GÃmez

CARGO: Desarrolladora de Software

TELEFONO: 555-321-6789

EMAIL: ana.gomez@example.com

FECHA CONTRATO: 2020-05-20

Editar Informacion Eliminar

ID: 3

NOMBRE COMPLETO: Pedro SÃnchez

CARGO: Asistente Administrativo

TELEFONO: 555-123-4567

EMAIL: pedro.sanchez@example.com

FECHA CONTRATO: 2019-08-30

Editar Informacion Eliminar

ID: 5

NOMBRE COMPLETO: Javier PÃrez

CARGO: Analista de Datos

TELEFONO: 555-456-7890

EMAIL: javier.perez@example.com

FECHA CONTRATO: 2022-09-01

Editar Informacion Eliminar

REGISTRO DE EMPLEADOS

Informaci3n del Empleado

Nombre Completo

Cargo

Tel3fono

Email

Fecha Contrato

Guardar Informaci3n Regresar a Empleados