

# Informe FrontEnd.

## Introducción

Este proyecto consiste en el desarrollo de un sistema para la gestión de **mascotas, interesados y solicitudes** utilizando **Angular** para el frontend. El objetivo es permitir la creación, edición, y eliminación de registros, así como la visualización organizada de los datos. Además, se implementó un menú unificado para facilitar la navegación entre las diferentes funcionalidades.

## Objetivos

1. **Implementar CRUD** (Crear, Leer, Actualizar, Eliminar) para:
  - **Interesados:** Personas interesadas en adoptar mascotas.
  - **Mascotas:** Animales disponibles para adopción.
  - **Solicitudes:** Registros de solicitudes de adopción realizadas por los interesados.
2. Facilitar la **navegación** mediante un **menú unificado**.
3. Mejorar la **usabilidad** al proporcionar selectores de opciones (como estados) y permitir la selección de fechas.

## Estructura del Proyecto

- **Frontend:** Angular.
- **Librerías:** Bootstrap para los estilos del menú y la tabla.
- **Servicios:** Se consumen servicios mediante **HTTPClient** para obtener y gestionar los datos de las mascotas, interesados y solicitudes.
- **Modelos de Datos:**
  - **Mascota:** Nombre, raza, tipo, sexo y edad.
  - **Interesado:** Nombre, cédula y teléfono.
  - **Solicitud:** Estado (Aprobado, Rechazado, etc.), fecha, mascota asociada, y el interesado.

## Código de Tabla para Mascotas:

```
<table class="table table-striped">

<thead>

<tr class="row">

<th class="col-sm-2">ID</th>

<th class="col-sm-4">Nombre</th>

<th class="col-sm-4">Raza</th>

<th class="col-sm-4">Tipo</th>
```

```
<th class="col-sm-4">Sexo</th>

<th class="col-sm-4">Edad</th>

<th class="col-sm-2">Acciones</th>

</tr>

</thead>

<tbody>

<tr *ngFor="let mascota of mascotas | async" class="row">

  <td class="col-sm-2">{{ mascota.mascotasId }}</td>

  <td class="col-sm-4">{{ mascota.nombre }}</td>

  <td class="col-sm-4">{{ mascota.raza }}</td>

  <td class="col-sm-4">{{ mascota.tipo }}</td>

  <td class="col-sm-4">{{ mascota.sexo }}</td>

  <td class="col-sm-4">{{ mascota.edad }}</td>

  <td class="col-sm-2">

    <a class="btn btn-info" [routerLink]="['/mascotas/editar/', mascota.mascotasId]">Editar</a>

    <a class="btn btn-danger" (click)="borrarMascota(mascota.mascotasId)">Borrar</a>

  </td>

</tr>

</tbody>

</table>
```

## Vista Usuario...

Menú

Nuevo Interesado

Nueva Mascota

Nueva Solicitud

Nueva Mascota

ID	NOMBRE	RAZA	TIPO	SEXO	EDAD	ACCIONES
2	Yorte	potyu	Loro	M	1	<div><div>Editar</div><div>Borrar</div></div>
5	Sachat	Normal	Gato	E	4	<div><div>Editar</div><div>Borrar</div></div>

## Selección de Estados y Fechas

- En el módulo de **solicitudes**, se implementó un selector para el **estado** de la solicitud con opciones: **Aprobado**, **Rechazado**, y **Pendiente**.
- También se agregó un **selector de fecha** para elegir la fecha de la solicitud.

### Código del Selector de Estado:

```
<select [(ngModel)]="solicitud.estado" class="form-control">  
  <option value="Aprobado">Aprobado</option>  
  <option value="Rechazado">Rechazado</option>  
  <option value="Pendiente">Pendiente</option>  
</select>
```

### Código del Selector de Fecha:

```
<input type="date" [(ngModel)]="solicitud.fecha" class="form-control">
```

## Recomendaciones

1. Implementar **paginación** para manejar grandes volúmenes de datos.
2. Agregar **validaciones** en los formularios para mejorar la calidad de los datos ingresados.
3. Incorporar un sistema de **autenticación** para restringir el acceso a los módulos de administración.