

TALLER UNIDAD 3 FRONTEND



Presentado por:

LUIS FELIPE SANTACRUZ CHINCHAJOA (220034097)

Docente:

VICENTE AUX

OCTUBRE 2024

UNIVERSIDAD DE NARIÑO

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

PASTO – NARIÑO

1. Introducción

En este informe se presenta el desarrollo de una aplicación frontend que hace uso del backend previamente desarrollado para gestionar una empresa de adopción de mascotas. La aplicación frontend se ha implementado utilizando Angular, junto con tecnologías como HTML5, JavaScript y Bootstrap para crear una interfaz de usuario amigable y funcional.

El objetivo principal de este proyecto es ofrecer una plataforma web donde los usuarios puedan visualizar las mascotas disponibles para adopción, enviar solicitudes de adopción y gestionar el estado de las mismas. Además, se han realizado pruebas exhaustivas para asegurar la correcta comunicación entre el frontend y el backend, garantizando una experiencia de usuario fluida.

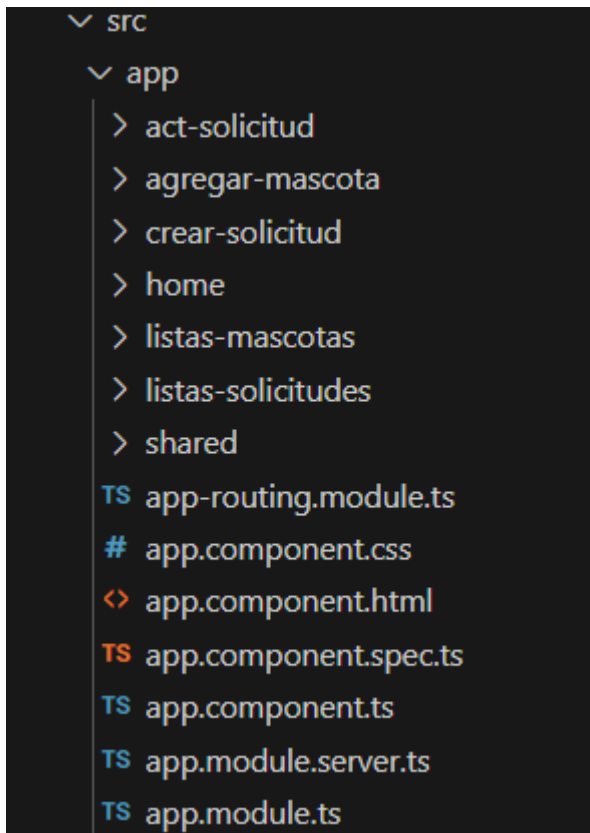
2. Desarrollo de la estructura del proyecto:

La aplicación se ha estructurado de forma organizada, siguiendo un enfoque que nos sirve para facilitar el mantenimiento y la escalabilidad del proyecto. A continuación, se describe la organización del proyecto:

2.1 Estructura de carpetas: Se ha creado la siguiente estructura de carpetas para organizar el código de manera lógica:

▼ FRONTEND-FELIPESANTACRUZ

- ▼ frontend
 - > .angular
 - > .vscode
 - > node_modules
 - > public
- ▼ src
 - > app
 - <> index.html
 - TS main.server.ts
 - TS main.ts
 - # styles.css
 - ≡ .editorconfig
 - ◆ .gitignore
 - { } angular.json
 - { } package-lock.json
 - { } package.json
 - 📘 README.md
 - TS server.ts
 - { } tsconfig.app.json
 - TS tsconfig.json
 - { } tsconfig.spec.json



2.2. Detalle de Carpetas y Archivos

2.2.1. Backend

- **/src:** Contiene el código fuente de la aplicación backend, cabe aclarar que fue el que desarrollamos previamente.
 - **/controladores:** Archivos que manejan la lógica de las rutas y procesan las solicitudes HTTP. Cada controlador está diseñado para un recurso específico, como mascotas o adoptantes.
 - **/modelos:** Contiene la definición de los modelos de datos utilizando Sequelize. Cada modelo representa una tabla en la base de datos.
 - **mascotaModelo.js:** Define el modelo de la tabla **mascota**, especificando los atributos y tipos de datos.
 - **/rutas:** Aquí se definen las rutas de la API, donde se configuran las rutas HTTP y se conectan a los controladores correspondientes.
 - **/database:** Contiene la configuración de la conexión a la base de datos (MySQL), utilizando Sequelize para manejar la comunicación entre la aplicación y la base de datos.
 - **app.js:** Archivo principal que inicializa la aplicación, define middlewares, y configura las rutas de la API.

2.2.2. Frontend

- **/src**: Contiene el código fuente de la aplicación frontend desarrollada en Angular.
 - **/app**: La carpeta principal de la aplicación Angular que incluye componentes, servicios y módulos.
 - **/act-solicitud**: Componente para manejar la creación de solicitudes de adopción.
 - **/agregar-mascota**: Componente para permitir al usuario agregar nuevas mascotas a la lista.
 - **/crear-solicitud**: Componente para crear solicitudes de adopción de mascotas.
 - **/home**: Componente principal que actúa como punto de entrada de la aplicación.
 - **/listas-mascotas**: Componente que muestra la lista de mascotas disponibles para adopción.
 - **/listas-solicitudes**: Componente que muestra las solicitudes realizadas por los usuarios.
 - **/shared**: Contiene modelos y servicios compartidos, como **mascota.model.ts** y **mascota.service.ts**, que facilitan la gestión de datos y la lógica de negocio.
 - **app.module.ts**: Archivo principal del módulo de la aplicación Angular donde se declaran todos los componentes y se importan los módulos necesarios.
 - **app.component.ts**: Componente raíz de la aplicación, que controla la vista principal.
 - **app.component.html**: Archivo HTML del componente raíz, donde se define la estructura general de la interfaz de usuario.
 - **app.component.css**: Archivo CSS del componente raíz, donde se establecen los estilos generales.

2.3. Conclusiones sobre la Estructura

La estructura del proyecto está diseñada para ser modular y escalable, permitiendo un fácil mantenimiento y una clara separación de responsabilidades. Cada componente, servicio y modelo tiene su lugar específico, lo que facilita la navegación y la comprensión del código. Esta organización es clave para un desarrollo ágil y eficiente en el contexto de aplicaciones web modernas.

3. Implementación

3.1. Desarrollo del Frontend

Para iniciar el desarrollo de la aplicación de adopción de mascotas, es esencial configurar adecuadamente el entorno de desarrollo. Los pasos para realizar esta configuración son los siguientes:

El frontend se desarrolló utilizando Angular, que permite crear aplicaciones web dinámicas y escalables.

3.3.1. Flujo de Datos

- La comunicación entre los componentes y el servicio se maneja a través de inyecciones de dependencias, permitiendo que los componentes interactúen con el servicio para obtener o enviar datos.

3.3.2. Componentes

- Se crearon componentes para cada parte de la aplicación, incluyendo **listas-mascotas**, **agregar-mascota** y **home**. Por ejemplo, el componente **listas-mascotas** se encarga de mostrar la lista de mascotas y se comunica con el servicio para obtener los datos.

3.3.3. Comunicación con el Servicio

- El servicio **mascota.service.ts** se encarga de realizar las solicitudes HTTP al backend.

3.4. Conclusiones sobre la Implementación

La implementación del proyecto se llevó a cabo siguiendo buenas prácticas de desarrollo, con una clara separación entre el frontend y el backend, lo que facilita la escalabilidad y el mantenimiento. La utilización de Angular en el frontend y Node.js con Express en el backend, junto con Sequelize para la gestión de la base de datos, proporciona una base sólida para la aplicación de adopción de mascotas.

4. Resultados

4.1. Funcionamiento de la Aplicación

La aplicación de adopción de mascotas fue implementada exitosamente, logrando cumplir con los objetivos planteados en las fases de planificación y desarrollo. A continuación, se describen los principales resultados obtenidos:

4.1.1. Interfaz de Usuario

- La interfaz de usuario fue diseñada utilizando **Bootstrap**, lo que permitió crear un diseño atractivo y responsivo. Los usuarios pueden navegar fácilmente por las diferentes secciones de la aplicación, incluyendo:
 - Lista de mascotas disponibles para adopción.
 - Formulario para agregar nuevas mascotas.
 - Sección para gestionar solicitudes de adopción.

4.1.2. API REST

- Se creó una **API REST** funcional que permite realizar las operaciones CRUD para las entidades **mascota**, **adoptante** y **solicitud**. Esto incluye:
 - Obtener la lista de mascotas disponibles.
 - Agregar nuevas mascotas a la base de datos.
 - Actualizar los detalles de una mascota existente.
 - Eliminar mascotas que ya no están disponibles.

4.2. Rendimiento

- La aplicación mostró un rendimiento óptimo en diferentes dispositivos y navegadores, manteniendo tiempos de carga bajos y respuestas rápidas a las interacciones del usuario.
- La conexión entre el frontend y el backend se realizó de manera eficiente, utilizando solicitudes HTTP para obtener y enviar datos de forma fluida.

4.3. Datos de Ejemplo

- Para realizar las pruebas, se utilizaron datos de ejemplo que permitieron verificar el correcto funcionamiento de la aplicación.

5. Interfaces Principales

A continuación, se detallan las principales interfaces de la aplicación, cada una diseñada para facilitar la interacción del usuario con el sistema de adopción de mascotas:

5.1. Interfaz de Lista de Mascotas

- **Descripción:** Muestra una tarjeta por cada mascota disponible para adopción, mostrando su imagen, nombre, especie, raza, edad y estado de adopción.
- **Características:**
 - Tarjetas de mascotas con un diseño consistente.
 - Botones para filtrar o buscar mascotas según preferencias del usuario.

Lista de Mascotas

Aquí puedes ver todas las mascotas disponibles para adopción.



Max

Especie: Perro

Raza: Golden Retriever

Edad: 3 años

Estado de Adopción: Disponible



Luna

Especie: Gato

Raza: Siamés

Edad: 2 años

Estado de Adopción: Disponible



Rocky

Especie: Perro

Raza: Bulldog

Edad: 4 años

Estado de Adopción: Disponible

5.2. Interfaz de Agregar Mascota

- **Descripción:** Permite a los administradores o usuarios autorizados agregar nuevas mascotas al sistema.
- **Características:**
 - Formulario para introducir detalles como nombre, especie, raza, edad e imagen.
 - Validación de campos para asegurar que la información ingresada es correcta y completa.



El formulario 'Agregar Mascota' se presenta sobre un fondo gris oscuro. El título 'Agregar Mascota' está en la parte superior en un color naranja brillante. Debajo del título, hay cinco campos de entrada blancos, cada uno con un label a su izquierda: 'Nombre:', 'Especie:', 'Raza:', 'Edad:' y 'URL de la Imagen:'. El campo de 'URL de la Imagen:' contiene el texto 'https://example.com/fido.jpg'. En la parte inferior del formulario, hay un botón rectangular de color verde con el texto 'Agregar Mascota' en blanco.

5.3. Interfaz de Solicitudes de Adopción

- **Descripción:** Muestra el historial de solicitudes de adopción y permite a los usuarios realizar nuevas solicitudes.
- **Características:**
 - Lista de solicitudes pasadas y su estado actual (aprobada, pendiente, rechazada).
 - Formulario para crear nuevas solicitudes de adopción, vinculando la mascota seleccionada.

Crear Solicitud

Completa el siguiente formulario para crear una nueva solicitud de adopción.

Mascota:

Crear Solicitud

6. Conclusión

El desarrollo de la aplicación de adopción de mascotas ha demostrado ser una valiosa experiencia de aprendizaje y aplicación práctica de tecnologías como Node.js, Express, Angular y MySQL. La implementación de una API REST bien estructurada y el uso de Sequelize como ORM facilitaron la gestión eficiente de datos, mientras que un diseño intuitivo mejoró la experiencia del usuario. Este proyecto no solo aporta un valor técnico, sino que también tiene un impacto social positivo al promover la adopción responsable de mascotas. A medida que avanzamos, se identifican oportunidades de mejora, como la incorporación de un sistema de autenticación, lo que permitirá continuar enriqueciendo la aplicación y su utilidad en la comunidad.