**Universidad de Nariño.   
Ingeniería de Sistemas.   
Diplomado de actualización en nuevas tecnologías para el desarrollo de Software.**

**Taller Unidad 3 Frontend**

**Presentando por: Tania Trejo - 220036029**

**Estructura del Proyecto Angular**

* **Componentes:** El proyecto "Rincón Peludo" está estructurado en varios componentes de Angular, cada uno con su respectivo archivo HTML, TypeScript (TS) y CSS.
  + about: Página de información acerca del proyecto.
  + home: Página de inicio de la aplicación.
  + crear-mascotas, crear-usuario, crear-solicitud: Componentes para la creación de mascotas, usuarios y solicitudes respectivamente.
  + detalle-mascota, detalle-usuario, detalle-solicitud: Componentes para mostrar detalles específicos de mascotas, usuarios y solicitudes.
  + editar-mascotas, editar-usuario, editar-solicitud: Componentes para la edición de mascotas, usuarios y solicitudes.
  + lista-mascotas, lista-solicitudes, lista-usuarios: Componentes para listar mascotas, solicitudes y usuarios respectivamente.
* **Routing:**
  + Configuración de rutas en app-routing.module.ts para todos los componentes mencionados.
* **Carpetas adicionales:**

**Modelos (.model.ts):**

* + **Definición de Estructuras de Datos:** Los modelos definen las estructuras de datos que se utilizan en la aplicación. Por ejemplo:
    - mascota.model.ts: Define las propiedades y tipos de datos de una mascota (por ejemplo, id, nombre, especie, raza, edad, etc.).
    - usuario.model.ts: Describe la estructura de un usuario, incluyendo atributos como id, nombres, usuario, password, etc.
    - solicitud.model.ts: Define la estructura de una solicitud de adopción.
  + **Facilita la Tipificación:** Al utilizar TypeScript, tener modelos permite que los datos sean tipados correctamente, lo que ayuda a prevenir errores y mejora la autocompletación en el editor.

1. **Servicios (.service.ts):**
   * **Lógica de Negocio y Acceso a Datos:** Los servicios son responsables de manejar la lógica de negocio y la comunicación con el backend. Por ejemplo:
     + mascota.service.ts: Contiene métodos para realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las mascotas, interactuando con el API del backend.
     + usuario.service.ts: Maneja las operaciones relacionadas con usuarios, como la autenticación y la gestión de datos.
     + solicitud.service.ts: Administra las solicitudes de adopción, facilitando su creación, consulta y modificación.
   * **Inyección de Dependencias:** Los servicios se pueden inyectar en cualquier componente que los necesite, promoviendo la reutilización de código y la separación de preocupaciones.

**Modificaciones y Consideraciones**

* Ajustes en app.module.ts para la configuración general del proyecto Angular.
* Modificaciones en app.component.html y .ts para la estructura general de la aplicación y lógica de inicio.