



## Descripción del Proyecto

PawFriends es una aplicación diseñada para conectar a dueños de mascotas y amantes de los animales en una comunidad de apoyo mutuo. Los usuarios pueden registrarse para reportar mascotas perdidas, ofrecer mascotas en adopción, solicitar y ofrecer servicios de cuidado temporal, y acceder a recursos para el bienestar animal. La aplicación está desarrollada en Android Studio utilizando Jetpack Compose para la interfaz y Firebase para la autenticación y el almacenamiento de datos.

## Estructura de Directorios

- `ui/`: Contiene las interfaces de usuario, organizadas en diferentes pantallas y componentes de diseño.
- `data/`: Almacena los repositorios, modelos de datos, y clases para acceder a Firebase.
- `navigation/`: Define las rutas de navegación y la configuración de NavHost.

## Rutas de Navegación

- `elementos`: Pantalla de inicio.
- `black`: Pantalla de prueba en blanco.
- `login`: Pantalla de inicio de sesión.
- `home`: Pantalla principal para la vista general de la aplicación.
- `register_pet`: Permite a los usuarios registrar una nueva mascota.
- `pet_reports`: Pantalla para reportar mascotas perdidas.
- `adopt_pet`: Pantalla para ver mascotas en adopción.

## Firebase

*Firebase se utiliza para:*

- Autenticación de usuarios con `firebase-auth-ktx`.
- Almacenamiento de datos como reportes de mascotas y solicitudes de adopción en Firestore, utilizando `firebase-firestore-ktx`.

## Componentes Personalizados

- **Imágenes:** Se incluyen iconos personalizados para el menú lateral y fotos de perfil de mascotas.
- **Paleta de colores:** Se ha definido una paleta en colores pastel para una apariencia amigable y coherente.
- **Fuentes:** Fuentes redondeadas y legibles para mejorar la accesibilidad visual.

## Dependencias

- **Firestore:**
  - firebase-auth-ktx para autenticación.
  - firebase-firestore-ktx para almacenamiento de datos.
- **Jetpack Compose:**
  - compose.ui y material3 para el diseño de la interfaz de usuario.
- **Navegación:**
  - navigation-compose para gestionar la navegación.

Ruta de navegación: NavManager

```

31 @Composable
32 fun NavManager(loginViewModel: LoginViewModel,
33               notesViewModel: NotesViewModel){
34     val navController = rememberNavController()
35     val petReportRepository = PetReportRepository()
36
37     NavHost(navController = navController, startDestination = "elementos"){
38         composable(route: "elementos") {
39             Elementos(navController)
40         }
41         composable(route: "black") {
42             BlankView(navController = navController)
43         }
44         composable(route: "login") {
45             TabsView(navController = navController, loginViewModel = loginViewModel)
46         }
47         composable(route: "home") {
48             HomeView(navController = navController, viewModel = notesViewModel)
49         }
50         composable(route: "register_pet"){
51             RegisterPetScreen(navController = navController)
52         }
53         composable(route: "pet_reports") {
54             ReportPetScreen(
55                 onReportSubmitted = { /* Acción a realizar después de reportar */ },
56                 repository = petReportRepository,
57                 onViewReports = { navController.navigate(route: "petReports") }
58             )
59         }
60     }
61 }

```

Union de los Items

