Universidad Mariano Gálvez de Guatemala

Sede Jalapa

Facultad de Ingeniería

Ingeniería en sistemas de información y ciencias de la computación

Ing. Marco Tulio Valdez Alvisurez

Curso – Programación II

PROYECTO FINAL

Yeilin Cecilia Aragón Alvarado

0907-22-22489

Luis Gustavo Ramírez Berganza

0907-23-8082

24 de octubre del año 2025

Introducción

La aplicación de mascotas es una plataforma integral diseñada para facilitar la búsqueda, rescate y adopción de animales, promoviendo al mismo tiempo su cuidado responsable. A través de esta aplicación, los usuarios pueden acceder a diversos servicios, como consejos de cuidado, publicación de reseñas y participación en un chat general que fomenta la interacción entre amantes de las mascotas. Además, la aplicación incluye un mapa interactivo que muestra las veterinarias más cercanas según la ubicación del usuario, brindando una experiencia más práctica y completa.

Está desarrollada en **Kotlin**, utilizando **Jetpack Compose** para la creación de una interfaz moderna y dinámica, y cuenta con **Firebase** como backend, encargado de la autenticación, el almacenamiento de datos y la gestión de imágenes.

Objetivos del Proyecto

- Facilitar la adopción responsable de mascotas mediante una interfaz intuitiva que permita publicar, visualizar y gestionar perfiles de animales disponibles para adopción.
- Incorporar un sistema de búsqueda y rescate, donde los usuarios pueden reportar mascotas perdidas o encontradas, fomentando la colaboración comunitaria.
- Integrar un chat general que sirva como espacio social para la interacción entre usuarios, permitiendo el intercambio de experiencias, consejos y coordinación en rescates o adopciones.
- Proveer información y consejos de cuidado animal, a través de publicaciones o secciones dedicadas, que promuevan el bienestar de las mascotas.
- Implementar un mapa interactivo. Conectado con servicios de geolocalización, para mostrar las veterinarias más cercanas a la ubicación del usuario.
- Utilizar Firebase para la autenticación de usuarios, almacenamiento seguro de datos e imágenes, y sincronización en tiempo real entre los distinto módulos de la aplicación.
- Aplicar Jetpack Compose en el desarrollo de la interfaz de usuario, priorizando el diseño moderno, adaptable y fácil de mantener.
- Optimizar el rendimiento y la seguridad del sistema mediante el uso de buenas practicas de desarrollo en Kotlin y el manejo adecuado de dependencias en Gradle.

Requisitos

Requisitos Funcionales

- El sistema debe permitir el registro de nuevos usuarios mediante correo electrónico y contraseña.
- El sistema debe permitir el inicio y cierre de sesión utilizando Firebase Authentication.
- El sistema debe permitir que los usuarios actualicen su perfil, incluyendo nombre, foto y datos de contacto.
- El sistema debe validar los datos ingresados antes de completar el registro o el inicio de sesión.
- El sistema debe permitir a los usuarios registrar mascotas disponibles para adopción, incluyendo nombre, edad, raza, descripción y fotografía.
- El sistema debe almacenar la información de las publicaciones en Firebase Firestore y las imagnes en Firebase Storage.
- El sistema debe permitir publicar reportes de mascotas perdidas o encontradas, con descripción, ubicación y fotografía.
- El sistema debe de permitir actualizar o eliminar publicaciones creadas por el usuario.
- El sistema debe ofrecer un chat comunitario en tiempo real, accesible para todos los usuarios registrados.
- El sistema debe permitir que los usuarios publiquen reseñas o consejos relacionados con el cuidado de las mascotas.
- El sistema debe acceder a la ubicación del usuario mediante permiso de GPS.
- El sistema debe mostrar un mapa interactivo con marcadores de veterinarias cercanas.
- El sistema debe permitir navegación fluida entre pantallas mediante componentes de Compose Navigation.
- Los datos deben sincronizarse en tiempo real, reflejando los cambios de forma inmediata.

Instalación y Configuración

Clonación del Proyecto

1. https://github.com/DanielOrtiz04/ProyectoFinalPrograKotlin.git bash

Copiar código

git clone <repositorio-url>

2. Abrir el proyecto en Android Studio.

Configuración de Firebase

1. Crear un proyecto en Firebase Console.
2. Añadir tu aplicación de Android en Firebase y descargar el archivo google- services.json.
3. Colocar el archivo google-services.json en la carpeta app/ del proyecto.
4. Habilitar Firebase Authentication, Realtime Database y Firebase Storage desde Firebase Console.

Dependencias

El archivo build.gradle contiene todas las dependencias necesarias para Firebase, Jetpack Compose y otras librerías:

kotlin

Copiar código dependencies {

```
implementation(platform("com.google.firebase:firebase-bom:33.5.0"))

implementation("com.google.firebase:firebase-auth-ktx:23.1.0")

implementation("com.google.firebase:firebase-database-ktx")
```

```
    implementation("com.google.firebase:firebase-storage-ktx:20.5.1")
    implementation("io.coil-kt:coil-compose:2.1.0")
    implementation("androidx.compose.ui:ui:1.5.4")
    implementation("androidx.compose.material3:material3:1.1.2")
    implementation("androidx.navigation:navigation-compose:2.7.5")
    // Más dependencias...
}
```

Configuración de la Aplicación

En el archivo build.gradle del módulo app, se deben especificar las versiones mínimas y objetivo de SDK, junto con las opciones de compatibilidad de Kotlin:

kotlin

Copiar código

```
android {
    compileSdk = 34

    defaultConfig {
        applicationId = "com.example.tiendamascotas" minSdk = 24
        targetSdk = 33
        versionCode = 1
        versionName = "1.0"
    }
```

```
buildFeatures { compose =
    true viewBinding = true
}

kotlinOptions { jvmTarget =
    "1.8"
}

composeOptions { kotlinCompilerExtensionVersion = "1.4.3"
}
```

Estructura del Código

Arquitectura del Proyecto

El proyecto sigue el patrón MVVM (Model-View-ViewModel) para mantener la separación de responsabilidades, lo que permite una mejor organización y mantenibilidad del código.

- Modelos (Model): Clases que representan los datos, por ejemplo, los objetos de mascotas o usuarios.
- Vistas (View): Compuestas por las pantallas desarrolladas con Jetpack Compose.
- ViewModels: Controlan la lógica de negocio y gestionan el estado de las vistas.

Principales Componentes

- ViewMascotas: Muestra una lista de mascotas disponibles, ya sea para búsqueda, rescate o adopción.
- Búsqueda y Rescate: Módulo para reportar y buscar mascotas perdidas o encontradas.
- Cuidado de Mascotas: Ofrece una colección de artículos y consejos sobre cómo cuidar diferentes tipos de mascotas.
- Reseñas: Permite a los usuarios publicar y leer reseñas relacionadas con cuidados y servicios.
- Chat: Implementación de un chat general para que los usuarios se comuniquen.

Interfaces de Usuario (UI)

Pantallas Clave

1. Pantalla de Inicio de Sesión:
    - Permite que los usuarios inicien sesión utilizando Firebase Authentication.
    - Diseño simple con Jetpack Compose y fácil navegación.
2. Pantalla Principal:
    - Muestra opciones como búsqueda y rescate, cuidado de mascotas, reseñas y chat.
    - Organizada con un diseño moderno y de fácil uso.
3. Pantalla de Búsqueda y Rescate:
    - Permite a los usuarios reportar mascotas perdidas o encontradas.
    - Incluye la funcionalidad de carga de imágenes con Firebase Storage.
4. Pantalla de Reseñas:
    - Los usuarios pueden leer y escribir reseñas sobre servicios o consejos para el cuidado de mascotas.
5. Pantalla de Chat:
    - Implementa un chat en tiempo real donde los usuarios pueden comunicarse.

Conclusiones

El desarrollo de la aplicación **PetShop** representa una solución tecnológica moderna e integral para promover la **adopción responsable, el rescate y el cuidado de las mascotas**, integrando en una sola plataforma herramientas de comunicación, geolocalización y gestión de datos en tiempo real.

Gracias al uso de **Kotlin** y **Jetpack Compose**, se logró una interfaz de usuario moderna, intuitiva y adaptable, mientras que **Firebase** proporcionó un backend robusto para la autenticación, el almacenamiento de información y la sincronización instantánea entre usuarios. Esto permitió crear una aplicación eficiente, segura y escalable, capaz de ofrecer una experiencia fluida y confiable.

El proyecto no solo cumple con su propósito funcional de conectar a personas con mascotas que buscan un hogar, sino que también fomenta la **colaboración y conciencia social** en torno al bienestar animal. Asimismo, su arquitectura basada en el patrón **MVVM** y el uso de tecnologías de última generación garantizan la mantenibilidad y la posibilidad de futuras mejoras, como notificaciones, seguimiento de adopciones o integración con servicios externos.

En conclusión, **PetShop** demuestra cómo la tecnología puede contribuir de manera significativa al bienestar animal y a la construcción de comunidades más empáticas, responsables y conectadas.