



Universidad Mariano Gálvez de Guatemala

Facultad de Ingeniería en Sistemas de Información y Ciencias de la
Computación

Programación II

Ing. Carlos René Hernández

Proyecto final: "Pet Garden"

Daniel Alexander Ortiz Cabrera 0907-23-14054

Edgar Roberto Quiñones Herrera 0907-22-545

Diego Alexander Contreras Duarte 0907-23-8207

Luis Gustavo Ramírez Berganza 0907-23-8082

Sábado, 02 de noviembre del 2024

Introducción

La aplicación de mascotas es una plataforma que facilita la búsqueda y rescate de mascotas, provee consejos de cuidado, permite a los usuarios publicar reseñas, y ofrece un espacio para la interacción social mediante un chat general. La aplicación está construida utilizando Kotlin, con Jetpack Compose para la interfaz de usuario y Firebase como backend para autenticación, almacenamiento de datos e imágenes.

Objetivos del Proyecto

- Facilitar la búsqueda de mascotas perdidas y el rescate de mascotas encontradas.
- Proveer una plataforma para compartir información sobre el cuidado de mascotas.
- Permitir la interacción social entre los usuarios a través de un chat y reseñas.

Requisitos

Requisitos Funcionales

- **Búsqueda y Rescate:** Los usuarios pueden reportar mascotas perdidas o encontradas.
- **Cuidado de Mascotas:** La aplicación ofrece consejos y guías sobre el cuidado de diferentes tipos de mascotas.
- **Reseñas:** Los usuarios pueden escribir y leer reseñas de otros usuarios relacionadas con el cuidado de mascotas y servicios relacionados.
- **Chat General:** La aplicación cuenta con un espacio de chat en tiempo real para la interacción entre usuarios.
- **Inicio de Sesión con Firebase:** Autenticación de usuarios a través de Firebase.
- **Almacenamiento en Firebase:** Uso de Firebase para el almacenamiento de datos e imágenes (por ejemplo, imágenes de mascotas perdidas o encontradas).

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.

Pruebas

Pruebas Unitarias

Se han implementado pruebas unitarias utilizando JUnit para verificar la funcionalidad del ViewModel, asegurando que los datos se gestionen correctamente y que las interacciones con Firebase se realicen según lo esperado.

Pruebas de Interfaz

Se utilizan las herramientas de testing de Compose para realizar pruebas sobre la interfaz de usuario. Las principales funcionalidades probadas incluyen la interacción con los elementos de la pantalla y la correcta navegación entre actividades.

kotlin

Copiar código

```
androidTestImplementation("androidx.compose.ui:ui-test-junit4")
```

```
debugImplementation("androidx.compose.ui:ui-tooling")
```

Conclusiones

La aplicación de mascotas se presenta como una solución integral para quienes buscan un entorno en el que puedan conectar con otros dueños de mascotas, buscar y rescatar mascotas, compartir consejos sobre cuidados y participar en una comunidad activa mediante reseñas y un chat general. El uso de tecnologías modernas como Jetpack Compose y Firebase permite una experiencia rápida, segura y escalable.

Autores

- Edgar Roberto Quiñones Herrera
- Diego Alexander Contreras Duarte
- Daniel Alexander Ortiz Cabrera
- Luis Gustavo Ramírez Berganza