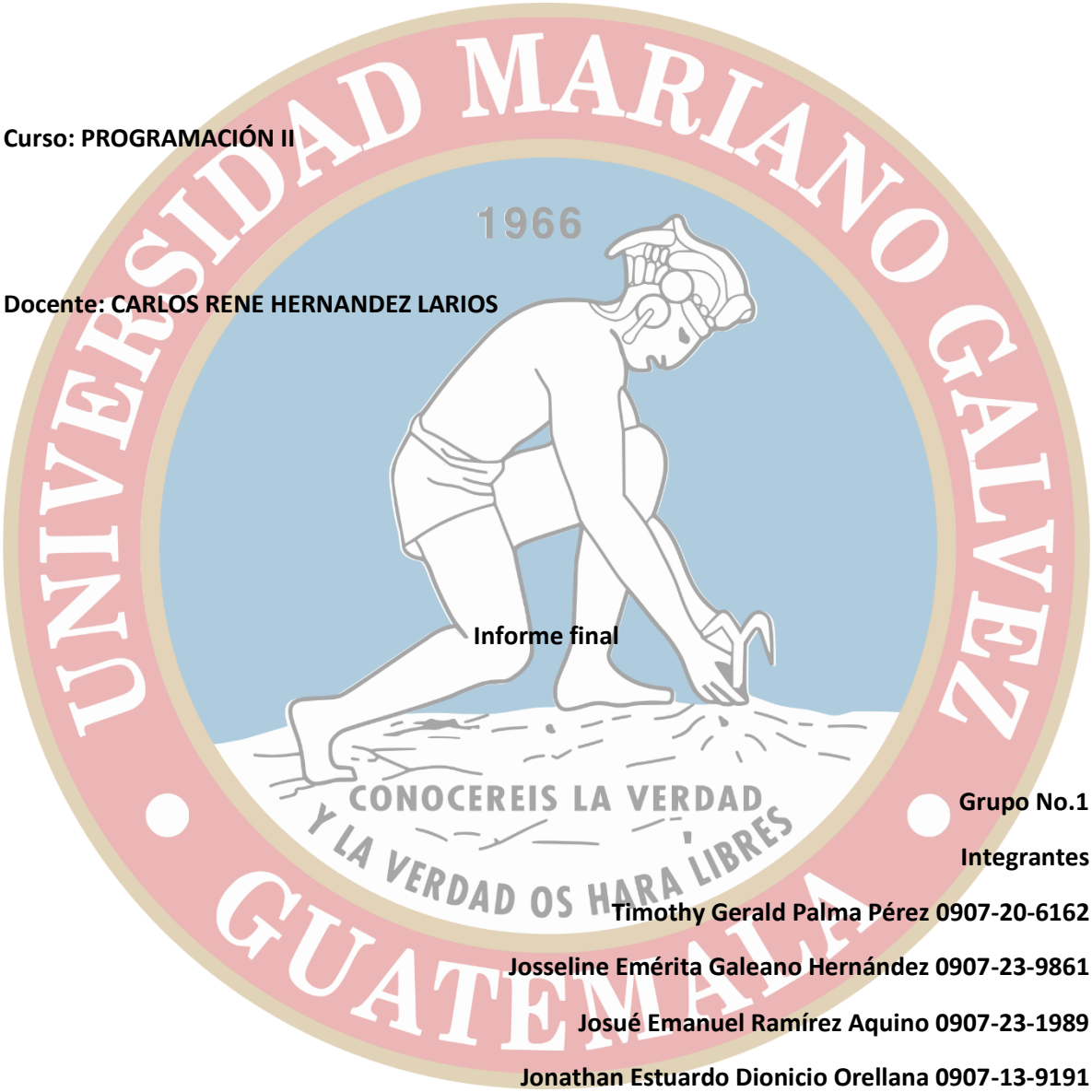
Universidad Mariano Gálvez

Curso: PROGRAMACIÓN II

Docente: CARLOS RENE HERNANDEZ LARIOS



Informe final

Grupo No.1

Integrantes

Timothy Gerald Palma Pérez 0907-20-6162

Josseline Emérita Galeano Hernández 0907-23-9861

Josué Emanuel Ramírez Aquino 0907-23-1989

Jonathan Estuardo Dionicio Orellana 0907-13-9191

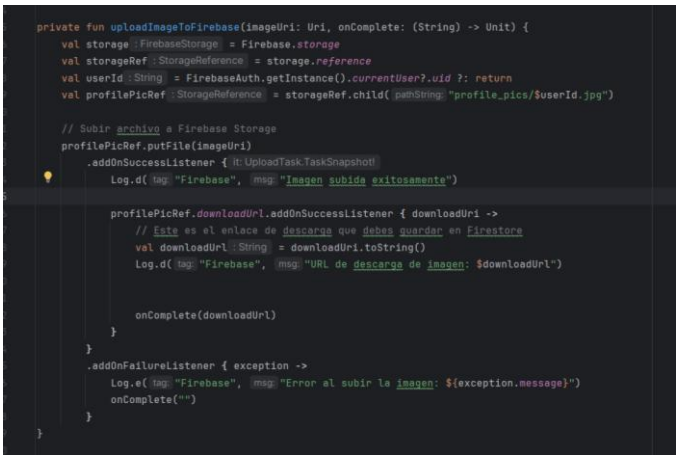
(Experiencias Timothy)

En el desarrollo de AppMascota, el primer paso fue configurar Android Studio y explorar sus herramientas y funcionamiento. Tras comprender cómo se crean los proyectos y sus componentes, comenzamos con el desarrollo de las pantallas, iniciando con las de 'Login' y 'Registro'. Nos enfocamos primero en el diseño para tener una base clara antes de conectar Firebase; todo esto se completó en la primera semana.

Durante la segunda semana, trabajamos en la integración con Firebase. La autenticación fue exitosa y la conexión con Firestore no presentó mayores problemas. Uno de los desafíos fue en la funcionalidad de 'Registro', donde la primera pantalla debía guardar datos como nombre, apellido y correo electrónico, que luego se utilizarían en la pantalla de perfil. El correo electrónico era clave para obtener los datos en la siguiente pantalla. Posteriormente, agregamos los datos adicionales necesarios en el perfil, como el nombre de la mascota, raza, edad y sexo. Para esto, creamos la función saveProfile, encargada de almacenar estos nuevos datos en los campos correspondientes de la base de datos.

```
fun saveProfileData(
    firstName: String,
    lastName: String,
    email: String,
    PetName: String,
    PetAge: String,
    PetBreed: String,
    PetGender: String,
    profilePicUri: String?,
    onComplete:   () -> Unit
) {
    val userId : String  = FirebaseAuth.getInstance().currentUser?.uid ?: "default_user_id"
    val userProfile = UserProfile(PetName, PetAge, PetBreed, PetGender, firstName, lastName, email, profilePicUri)

    Firebase.firestore.collection( collectionPath: "users").document(userId).set(userProfile)
        .addOnSuccessListener { it: Void!
            Log.d( tag: "ProfileScreen",  msg: "Perfil guardado exitosamente")
            onComplete()
        }
        .addOnFailureListener { exception ->
            Log.e( tag: "Firebase",  msg: "Error al guardar los datos: ${exception.message}")
            onComplete()
        }
}
```

Y también se agregó la función uploadImageToFirebase que se encargar de subir la imagen para el perfil esta función antes no estaba conectada a storage ya que la guardaba de forma temporal de esa manera al cerrar la app se borraba la imagen ese fue un pequeño error que se pudo solucionar descargando al url

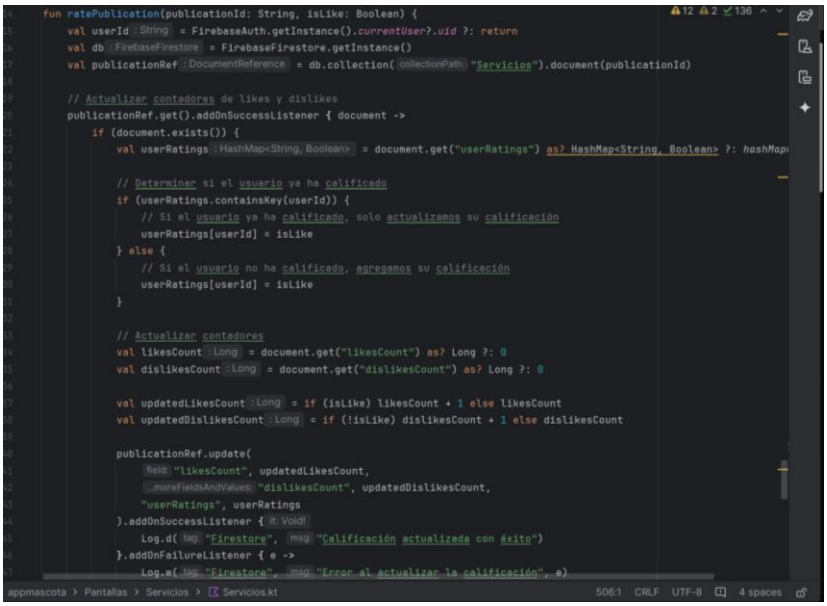


```
private fun uploadImageToFirebase(imageUri: Uri, onComplete: (String) -> Unit) {
    val storage : FirebaseStorage  = Firebase.storage
    val storageRef : StorageReference  = storage.reference
    val userId : String  = FirebaseAuth.getInstance().currentUser?.uid ?: return
    val profilePicRef : StorageReference  = storageRef.child( pathString: "profile_pics/$userId.jpg")

    // Subir archivo a Firebase Storage
    profilePicRef.putFile(imageUri)
        .addOnSuccessListener { it: UploadTask.TaskSnapshot!
            Log.d( tag: "Firebase",  msg: "Imagen subida exitosamente")

            profilePicRef.downloadUrl.addOnSuccessListener { downloadUri ->
                // Este es el enlace de descarga que debes guardar en Firestore
                val downloadUrl : String  = downloadUri.toString()
                Log.d( tag: "Firebase",  msg: "URL de descarga de imagen: $downloadUrl")


                onComplete(downloadUrl)
            }
        }
        .addOnFailureListener { exception ->
            Log.e( tag: "Firebase",  msg: "Error al subir la imagen: ${exception.message}")
            onComplete("")
        }
}
```

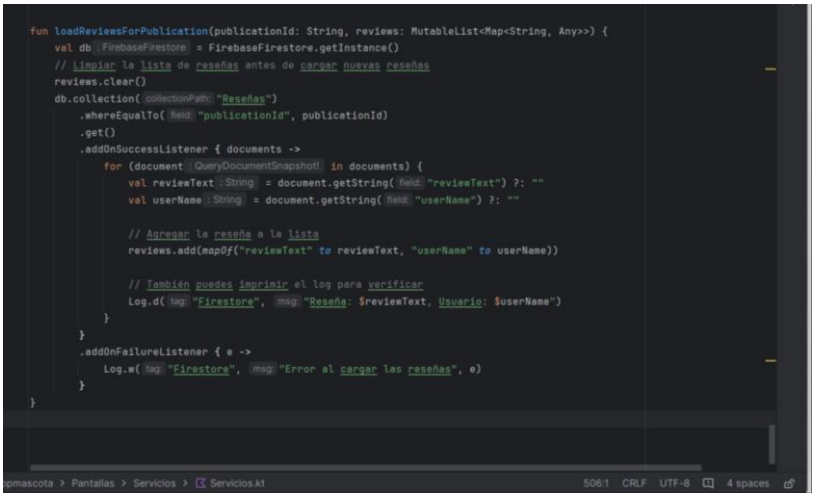
Y otras dos funciones que ayudan updateProfilePicUriInFirestore actualiza la url en la base de datos y la otra función loadProfileData recupera todos los datos que se guardaron en la base de datos para que así en la pantalla perfil mostrara los datos que registramos.

Los mayores problemas que tenia fueron con la imagen de perfil ante había me dejado un solo botón para guardar toda la información y la imagen pero sin importar que la imagen del perfil siempre se guardaba de forma temporal la mejor solución que pude colocar fue dividir las funciones en dos una que solo guarde la información y otro que solo guarde la imagen y agregar dos botones de esa forma se evitaba ese problema con la imagen un botón se encargaba de guardar la información y los cambios del perfil sin afectar a la imagen y el otro si se cambiaba la imagen ya estaba toda la información.

Otro problema que se presento fue con la creación de las publicaciones para los servicios ya que había que implementar las reseñas y calificaciones al principio se pensó en agregar con calificación de estrella pero presentaba dos problemas el primero a la hora de implementarlo la persona podría colocar la cantidad de estrellas pero si otro usuario calificaba la primer calificiacion digamos es de una estrella si otro usuario la colocaba de 3 estrellas la de primer estrella era reemplazado con el de 3 de esa manera no se podía saber si era malo o bueno el otro problema fue agregarlo como si al calificar se hacía una operación de conteos dependiendo de la cantidad de estrellas se iba rellenando pero presentaba problemas al guardarlo en la base de datos y la función para hacer ese dibujado de estrellas cerraba la aplicación se opto por hacer un conteo de like y dislike una manera eficaz y que evita muchos problemas



El siguiente problema que tuvimos fue con las reseñas ya que no se guardaba de forma correcta o se hacia pero no se mostraban pero ese error se debía a una parte de la función

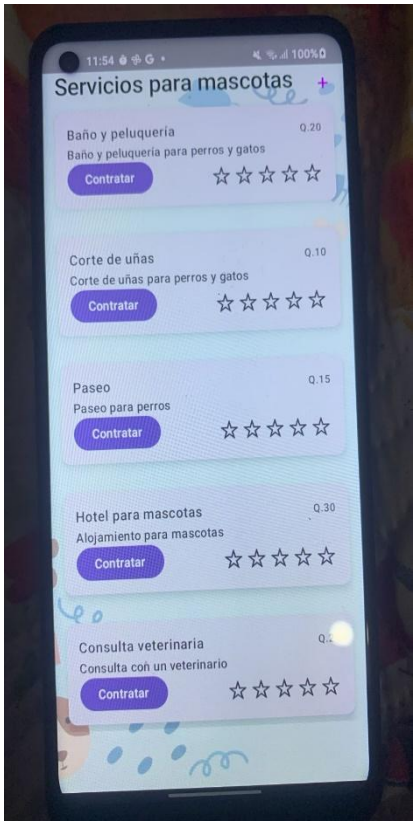


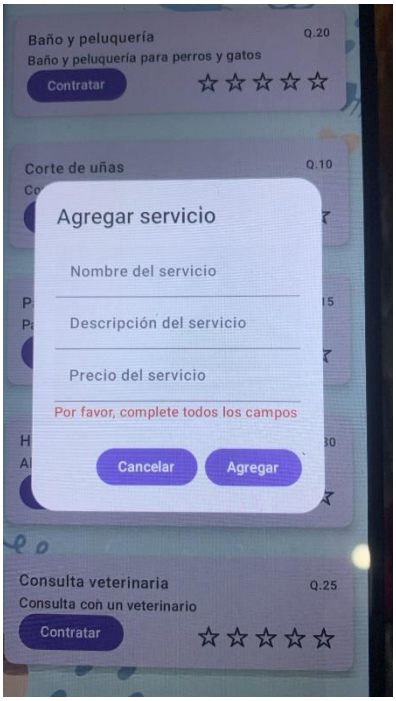
Que se traba del db.collection("Reseñas") que antes no estaba con reseñas si otro nombre lo que hacía que no encontrara los datos y no se podían mostrar de forma correcta con ese cambio se mostro todo.

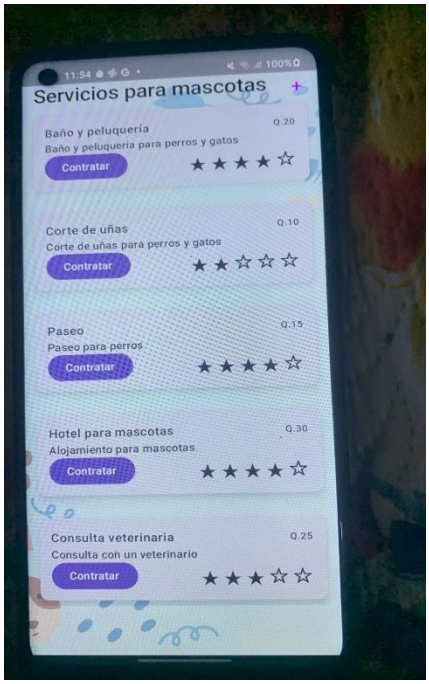
(Experiencias Josseline)

En el desarrollo de la AppMascotas lo primero que me toco realizar fue conocer Android Studios de una manera mas a fondo para poder trabajar de una manera optima para el desarrollo del proyecto lo cual vi videos para poder conocer sobre como era programas en jetpack compose, después de ver videos me dejaron la pantalla de Servicios con todas sus funcionalidades la cual con mucho esfuerzo logre realizar todo iba perfecto hasta que tuve que subirlo a git lo cual me costo ya que no tenia mucho conocimiento de como hacerlo pero junto con mis compañeros y con ayuda del ingeniero logramos entender como manejarlo y así poder subirlo, lo cual cuando se implemento al código de mi compañero Timothy nos llevo a un gran problema ya que su computadora le tiraba muchos errores con mi código, lo cual me desanimo bastante ya que pasaron muchas noches de desvelo para poder realizarlo ya que tenia la función de calificación de estrellas, contratar un servici, agregar un servicio, confirmación del servicio contratado y la ultima parte para implementar era el chat que estaba ya casi funcionando cuando hable con mi compañero y me comento la situación.

Adjunto evidencia de la pantalla que realice







Entonces tuve que iniciar desde 0 y mi compañero Timothy me ayudo con algunos elementos que yo le mandaba para ver que no sucediera de nuevo, y ya cuando el app aceptaba la pantalla pude seguir implementando las funciones necesarias en la pantalla servicios y así mismo ayudar a mis compañeros con el diseño de sus pantallas para que todas puedan ir de la misma manera con un diseño agradable a la vista y que todos estén coordinados con cada uno de las pantallas.

He adquirido experiencia con este proyecto no soy la mejor con este lenguaje pero si he aprendido tanto de jetpack compose como de github lo cual me ha dejado las experiencias que para en un próximo trabajo en grupo ya puedo realizarlo y subir el código sin ninguna confusión.

(Jonathan Dionicio)

Ha sido un proyecto desafiante, Nunca había tenido la experiencia de poder ejecutar ningún proyecto de esta manera y magnitud. En un principio me costo poder entender el lenguaje, conforme a las clases impartidas, tutoriales e investigando, haciendo uso de todos los recursos a mi alcance pude obtener cierto grado de experiencia. Aunque no lo domino en su totalidad, he aprendido bastante. Uno de los primeros desafíos fue implementar el uso de los commits para trabajar en equipo, porque no quería estropear el trabajo de mis compañeros hasta que nos proporcionó ayuda el Ingeniero Carlos Hernandez para dar uso a la aplicación **Fork.** Al principio fue difícil poder comprender haciendo uso de Android Studio y no sabía por donde empezar. Nos reunimos y empezamos a pensar en la estructura de nuestra aplicación. Y de esta manera empezamos a trabajar me encargue de la parte del menú de adopción. Inicie implementando el menú inicial de adopción. El hacer uso de jetpack compose fue un poco complicado al principio. Habían cosas que no comprendía en su totalidad conforme iba aprendiendo con los videos que posteaba el Ingeniero Carlos Hernandez, pude comprender en parte la interfaz a implementar. Aprendí sobre Scaffold, LazyColumn, Alert Dialog y botones para poder implementar en el menú adopción. Y con muchas pruebas iba buscando la forma al estilo del menú que sea amigable con el usuario. Luego tuve que entender como hacer uso de los botones y sus acciones al presionarlos. Aprendí a la importancia de las funciones para luego llamarlas o posicionarlas en el menú. En lo que se me dificulto bastante fue entender sobre como hacer uso de las funciones y conectarlas a la firebase, mi compañero Timothy me ayudó mucho a comprender la forma en que se podía implementar el uso y otros detalles sobre navegación e interacción con botones. En el menú inicial tuve dificultades para poder manejar la manera en que aparezcan las publicaciones de solicitudes en pantalla ya sean las que publico y las que publican otros usuarios. Por lo que me guíe por un id único que se crea para cada usuario registrado en la aplicación y por medio de esta pude implementar funciones que aplique a una base de datos llamada PublicationsAdopcion que me serviría como punto de partida por medio del cual aceptaría campos como: Nombre de la mascota, descripción, entre otros. Y de esta manera poder manejar estos posts para ser interactuados en otras pantallas. Otro detalle en el cual me tomo tiempo entender es la navegación para interactuar con lo distintos botones, pero luego de comentarlo a mis compañeros pude entender la manera de como hacer uso. Por medio de navController y hacer uso de un archivo navegación y en otro archivo colocar las rutas. Aprendí sobre el uso de las líbrerias y lo importante que es importarlas para fluya el correr la aplicación. Aprendí sobre el uso de las dependencias y como conectar a firebase y permisos que deben colocarse en el archivo AndroidManifest.xml para conectarse a internet. De lo que más pude tener experiencia es sobre jetpack compose comprendiendo la forma en que se

puede dar estilo a una aplicación por medio de columnas, filas, navegación, estilos, modificadores y otros.

(Experiencia en el Proyecto Josué Ramírez)

Fue un proyecto desafiante, poniendo a prueba prácticamente la totalidad de los conocimientos impartidos no solo de este curso, si no de los recibidos con anterioridad, aprender este nuevo lenguaje desde 0, con las clases impartidas, ejemplos dados y demás material, fue complicado entender al principio entonces el inicio de nuestro proyecto fue lento, pero se fue avanzando de manera segura, evitando cometer errores entre todos, mi aportación fue la pantalla de Mascotas Pérdidas, dada la similitud con el funcionamiento de mascotas adoptadas, con mi compañero fuimos compartiendo ideas sobre como implementar las funcionalidades, aplicando de buena manera Jetpack compose para optimizar la codificación de nuestro proyecto y el uso de firebase como nuestra base de datos, la utilización de estas herramientas fue de gran utilidad.

La primera problemática que tuve fue la representación de la publicación de una mascota perdida al presentarse en la pantalla, dado que no se presentaba pero los datos si eran guardados en la base de datos, entonces tras revisar las funciones que ocupan el guardado de datos en la base de datos y la escritura de los datos en pantalla, me di cuenta sobre que había hecho una mala escritura en el código, entonces al corregirlo y poner el nombre que correspondía se corrigió ese error, la segunda problemática fueron las imágenes, fue una problemática que tuvimos en común con los compañeros del proyecto, después de buscar soluciones se encontró la manera y fue implementada en la funcionalidad de mascotas perdidas y fue solucionado. Además de problemas que el mismo programa había mostrado, como no reconocer el sdk o necesitar la modificación de local properties, a pesar de eso la implementación de esta pantalla fue de buena manera y la comunicación entre el grupo fue necesaria y de mucha ayuda.