



UTT

UNIVERSIDAD TECNOLÓGICA DE TIJUANA

GOBIERNO DE BAJA CALIFORNIA

Autenticación con Firebase

PRESENTADO POR:

García Arreola Howard Isaí

GRUPO: 9B

MATERIA:

Desarrollo para Dispositivos inteligentes

PROFESOR:

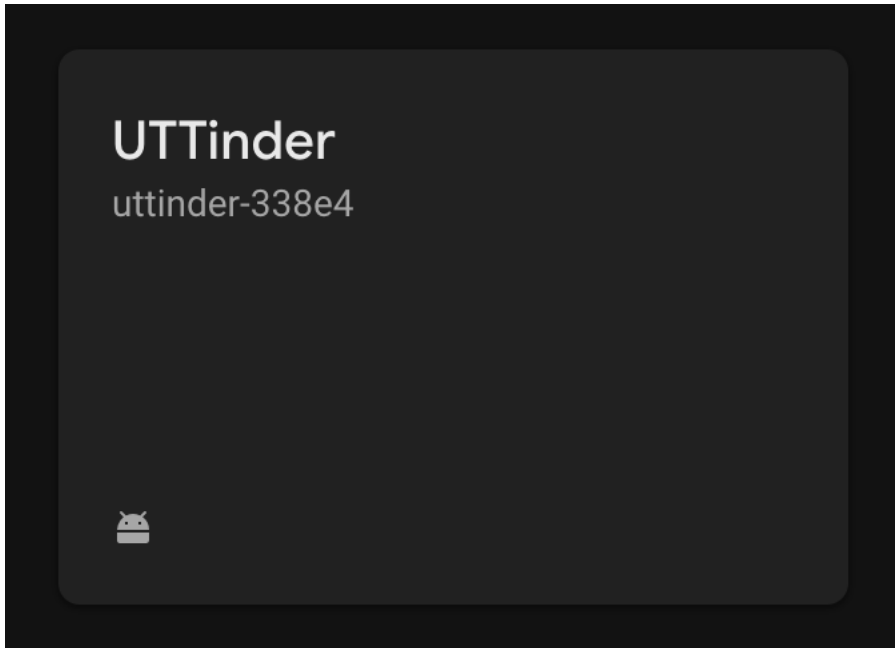
Ray Brunett Parra Galaviz

FECHA:

30 de octubre de 2024

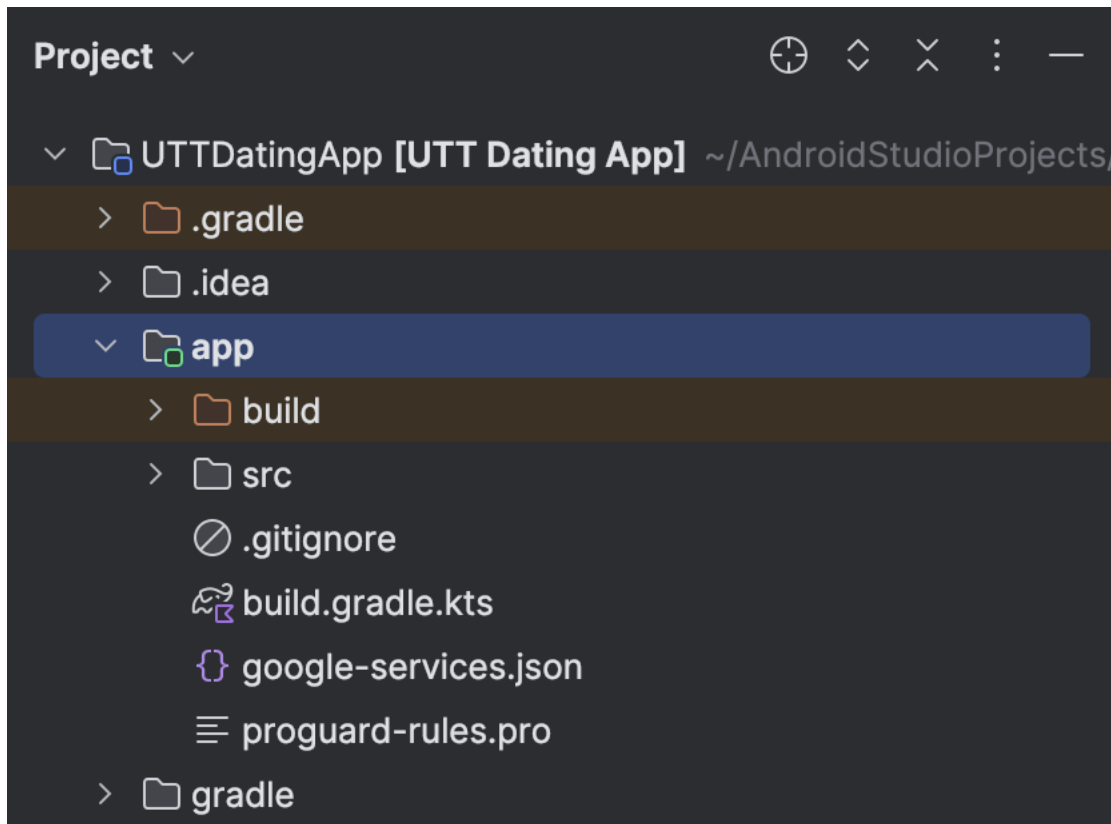
Para utilizar la autenticación de Firebase dentro de nuestro proyecto se deben de realizar los siguientes pasos:

- **Crear un proyecto en Firebase:** Ve a la consola de Firebase y crea un nuevo proyecto.

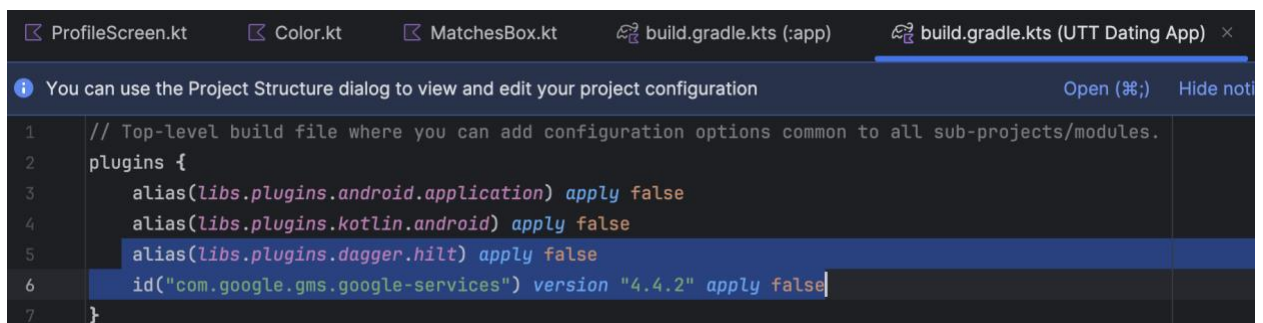
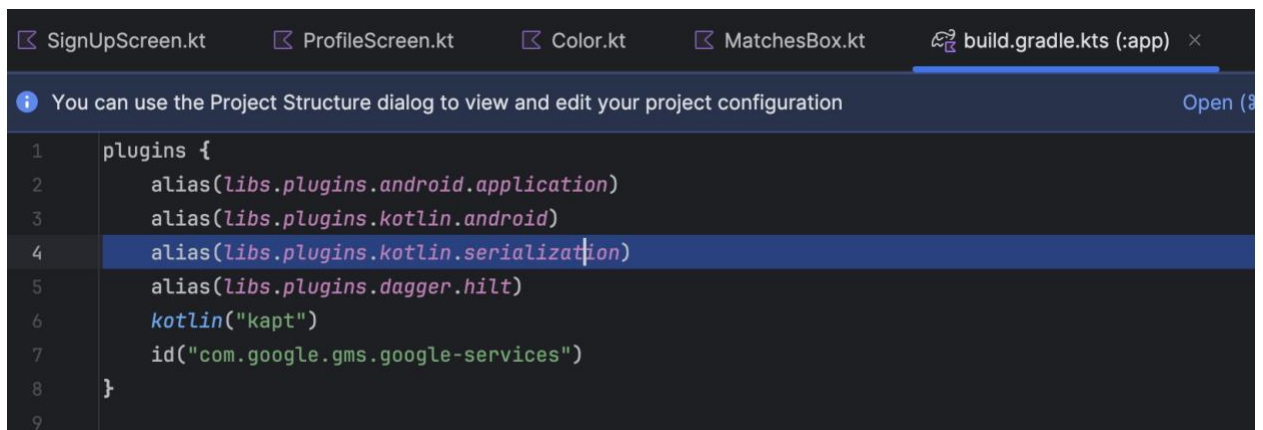


- **Agregar tu aplicación Android:** Sigue las instrucciones para conectar tu aplicación Android a tu proyecto de Firebase. Esto implicará descargar un archivo `google-services.json` y agregarlo a tu proyecto.

Como nota, el JSON debe de estar guardado dentro de la Project, en la carpeta de App:



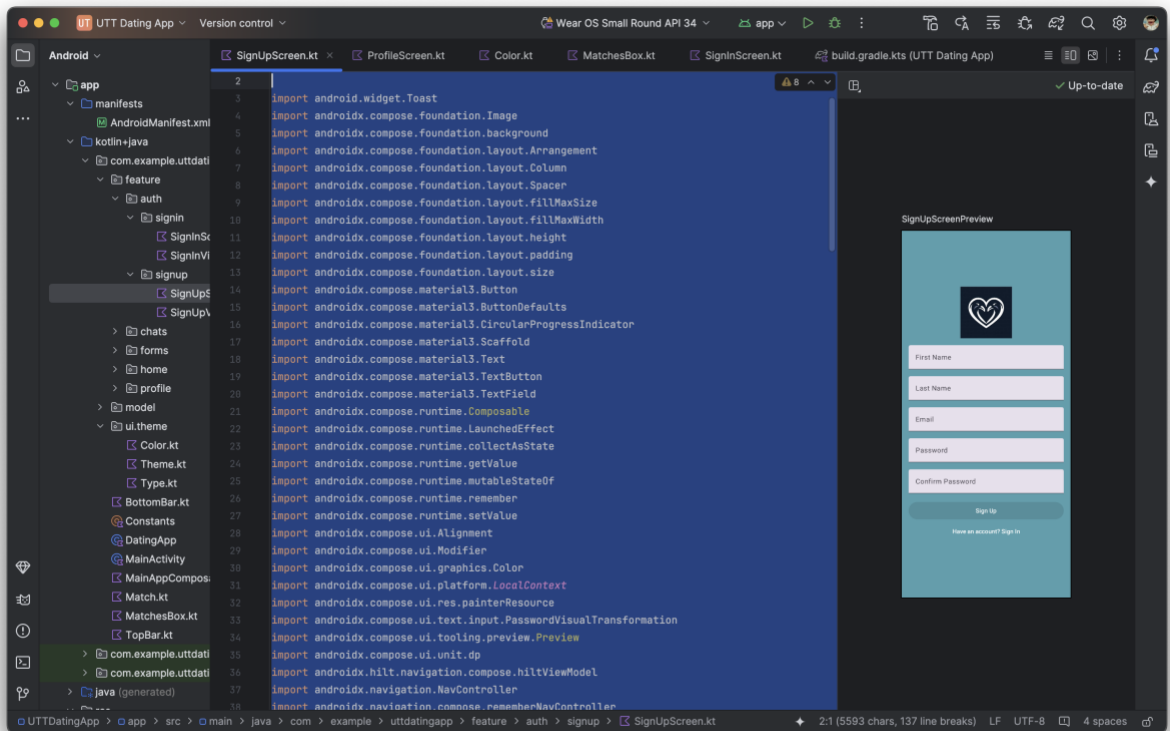
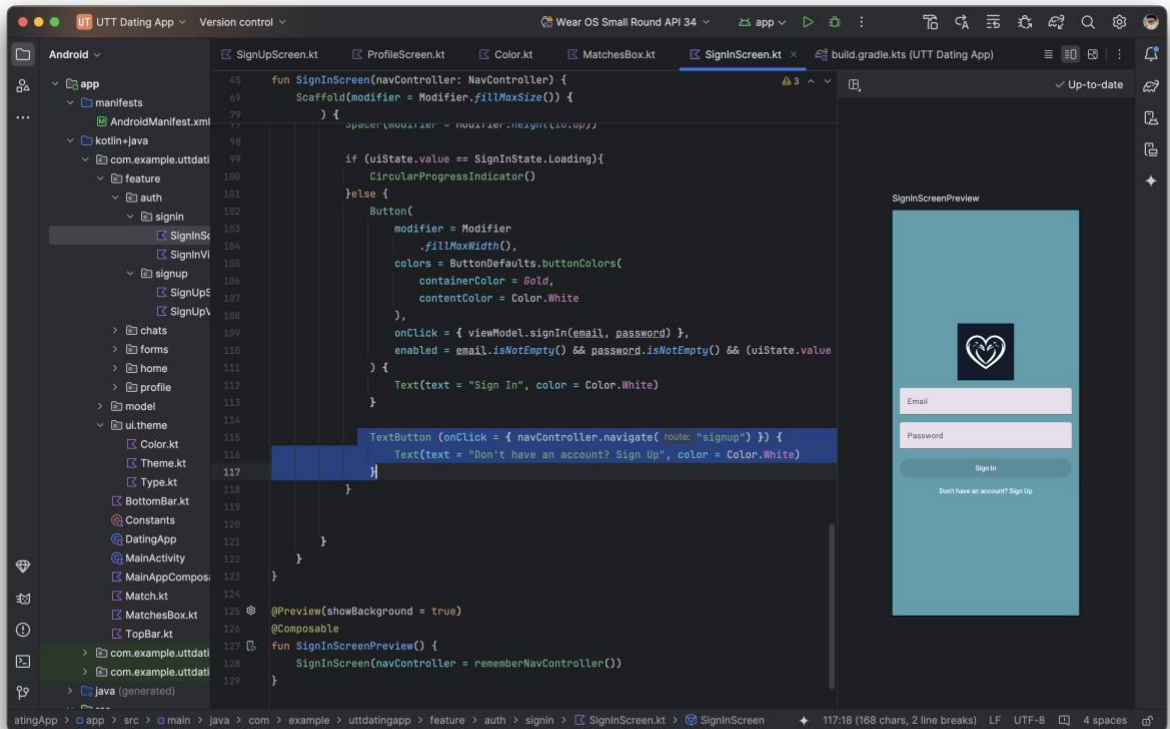
- **Agregar dependencias a Gradle:** En tu archivo **build.gradle** (módulo de la app), agrega las siguientes dependencias:



```
implementation(libs.dagger.hilt.android)
implementation(libs.firebase.crashlytics.buildtools)
kapt(libs.dagger.hilt.compiler)
implementation(libs.dagger.hilt.compose)
implementation(libs.coil)
implementation(platform("com.google.firebase:firebase-bom:32.8.0"))
implementation("com.google.firebase:firebase-analytics")
implementation("com.google.firebase:firebase-auth")
implementation("com.google.firebase:firebase-database")
implementation("com.google.firebase:firebase-storage")
```

En el caso de este proyecto se utilizan otras librerías extras ya que se utilizan otras funcionalidades de Firebase, no solo la autenticación.

- **Realizar el formulario de login y register:** Se necesita tener un formulario de login para poder mandar los datos a Firebase para poder autenticar a los usuarios:



- **Crear los ViewModel:** Cada screen de login y register, necesita de un viewmodel, este es necesario para comunicarse con firebase y mandar los datos para autenticar o registrar a nuestros usuarios:

Para ello se creo la función de signIn:

```
@HiltViewModel
class SignInViewModel @Inject constructor() : ViewModel() {
    private val _state = MutableStateFlow<SignInState>(SignInState.Nothing)
    val state = _state.asStateFlow()

    fun signIn(email: String, password: String) {
        _state.value = SignInState.Loading
        //Firebase Auth
        FirebaseAuth.getInstance().signInWithEmailAndPassword(email, password)
            .addOnCompleteListener { task ->
                if (task.isSuccessful) {
                    task.result?.user?.let {
                        _state.value = SignInState.Success
                        return@addOnCompleteListener
                    }
                    _state.value = SignInState.Error
                } else {
                    _state.value = SignInState.Error
                }
            }
    }
}

sealed class SignInState {
    object Nothing : SignInState()
    object Loading : SignInState()
    object Error: SignInState()
    object Success : SignInState()
}
```

Se añadio una clase en donde se encuentran objetos del estado de las peticiones, esto para que la screen pueda saber que mostrar dependiendo al estado que devuelva la función.

Para el register, utilizamos Real Time Database de Firebase para guardar los datos de nuestros usuarios dentro de la base de datos y no solo en el apartado de autenticación, los guardamos con el mismo ID que crea firebase por defecto al ser autenticados:

```
fun signUp(name: String, email: String, password: String) {
    _state.value = SignUpState.Loading

    val auth = FirebaseAuth.getInstance()
    val database = Firebase.database.reference

    // Register the user
    auth.createUserWithEmailAndPassword(email, password)
        .addOnCompleteListener { task ->
            if (task.isSuccessful) {
                val user = task.result.user

                user?.let {
                    // Update user's display name in Firebase Authentication
                    it.updateProfile(
                        com.google.firebase.auth.UserProfileChangeRequest.Builder()
                            .setDisplayName(name)
                            .build()
                    ).addOnCompleteListener { profileUpdateTask ->
                        if (profileUpdateTask.isSuccessful) {
                            // Create an entry for the user in Realtime Database
                            val userData = mapOf(
                                "profile" to mapOf(
                                    "user_id" to it.uid,
                                    "name" to name
                                ),
                                "chat" to mapOf<String, Any>() // Empty chat structure
                            )
                        }
                    }
                }
            }
        }
```

- Para utilizar la autenticación de firebase, es necesario tener activada dicha funcionalidad en nuestra consola:

