Desarrollo de Software en Móvil (DSM)

FORO 2

Docente: Alexander Siguenza

Apellidos	Nombres	Carné
Guevara Rodríguez	Oscar Alexander	GR222756

Repositorio: https://github.com/OscarG673/DSM-foro2

Video: https://youtu.be/Dl2JCdolDyI

Contenido

Autenticación

La mayoría de las aplicaciones móviles requieren identificar a los usuarios cuando se trata de aplicaciones que interactuarán con datos de usuarios en específicos o simplemente es necesario un registro. El conocer a un usuario permite que una aplicación puede guardar sus datos en una base de datos de forma segura y de esta manera la aplicación pueda proporcionar la misma experiencia personalizada en todos los dispositivos que posea el usuario.

Para ello existen diversas herramientas que ayudan a proporcionar servicios de backend, SDK (Software Development Kit) fáciles de usar y bibliotecas de interfaz de usuario ya elaboradas, con el fin de facilitar la autenticación de los usuarios en cualquier aplicación móvil.

La que se tocará en esta investigación es Firebase, la cual es una plataforma en la nube para el desarrollo de aplicaciones web y móvil. En este caso nos centraremos en la autenticación, y para ello, se verá de forma detallada la herramienta de Firebase llamada Firebase Authentication.

Existen muchas formas de autenticación que admite Firebase Authentication, entre las cuales están:

Tipos de autenticación con Firebase

Auntenticación por correo y contraseña

Es una autenticación clásica, la cual permite que los usuarios puedan autenticarse con un correo electrónico y una contraseña. Firebase se encarga de gestionar los usuarios que utilicen su correo y contraseña para acceder, así como también se encarga del proceso de restablecer contraseña, por medio del envío de correos electrónicos de restablecimiento.

Ventajas	Desventajas
No depende de terceros	Requiere de la gestión de contraseñas por parte del usuario, ya sea por olvidos.
Es sencillo de implementar	

Consideraciones:

Se recomienda implementar una autenticación de 2 factores para mayor seguridad (2FA)

Utilización de reglas de seguridad fuertes para contraseñas.

Integración con proveedores de identidad federada

Autentica a los usuarios por medio de la integración con proveedores de identidad federada, es decir, Firebase proporciona métodos que permiten a los usuarios acceder con sus cuentas de Google, Facebook, X y Github.

Ventajas	Desventajas
Experiencia rápida, sin necesidad de almacenar contraseñas y que el usuario deba recordarlas.	Depende del servicio de terceros
	Configuración adicional en la consola de cada proveedor
	Uso gratuito limitado por algunos de los proveedores.

Consideraciones:

Registro de la aplicación en cada plataforma

Manejo de tokens de acceso.

Autenticación con número de teléfono

Envía mensajes de texto (SMS) a los usuarios para que puedan autenticarse, por medio de un código enviado, sin necesidad de utilizar contraseñas.

Ventajas	Desvenntajas
Rápido y fácil para el usuario	El envío de SMS representa costos adicionales.
Ideal para regiones donde el correo electrónico no es utilizado normalmente	Es más vulnerable a ataques como SIM swapping

Consideraciones:

Utilización de recaptcha para prevenir spam.

Límites en pruebas de autenticación.

Integración de sistemas de autenticación personalizados

Permite integrar un propio sistema de autenticación con el SDK de Firebase por medio de tokens personalizados, siendo muy ideal para sistemas empresariales o de backend propio.

Ventajas	Desventajas
Flexibilidad	Complejidad en la implementación
Integración de usuarios existentes de otros sistemas	Necesidad de un servidor seguro para la generación de los tokens

Consideraciones:

Se debe tomar en cuenta la gestión de la seguridad del backend que emite los tokens.

No es una opción que deba considerarse si solo se utiliza Firebase

Autenticación anónima

Crea cuentas anónimas de forma temporal que permiten el uso de funciones que requieran autenticación sin exigir que los usuarios accedan primero. En caso de que posteriormente el usuario decida registrarse, se podrá actualizar la cuenta anónima y convertirla en una normal, de manera que el usuario podrá continuar con su actividad donde la haya interrumpido.

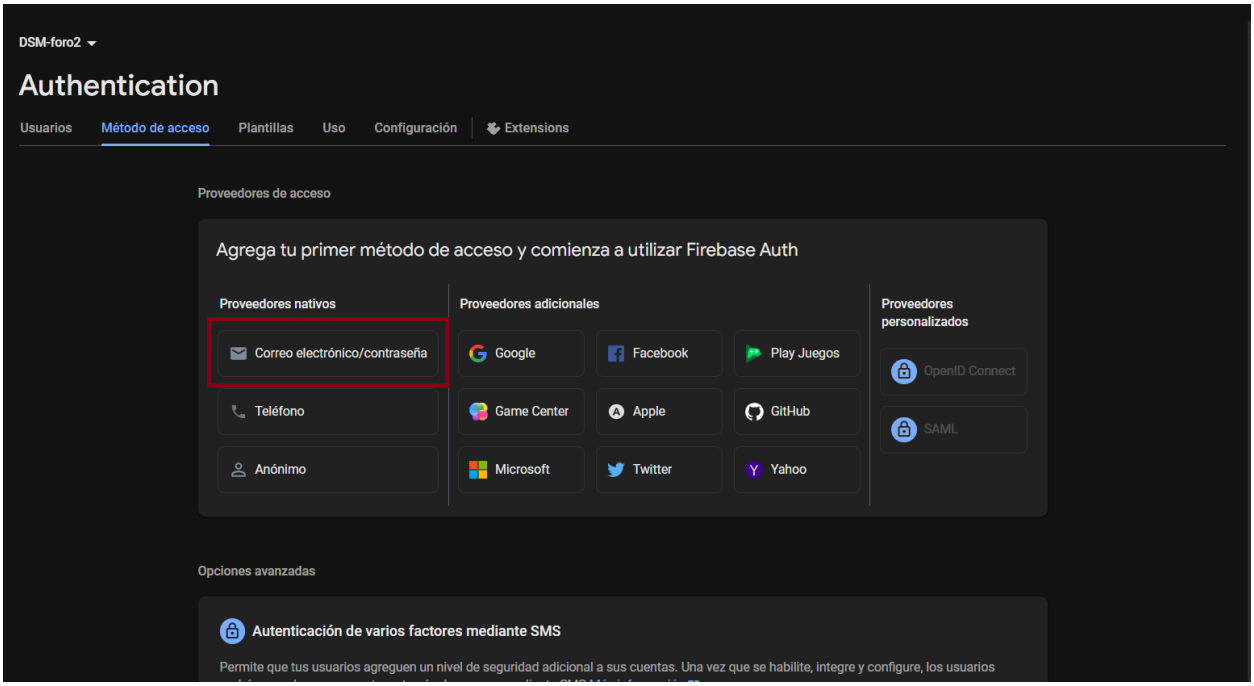
Ventajas	Desventajas
Permite conservar datos cuando el usuario decide registrarse	La sesión se pierde si el usuario densistala la aplicación
	No es para aplicaciones que requieren identificación permanente

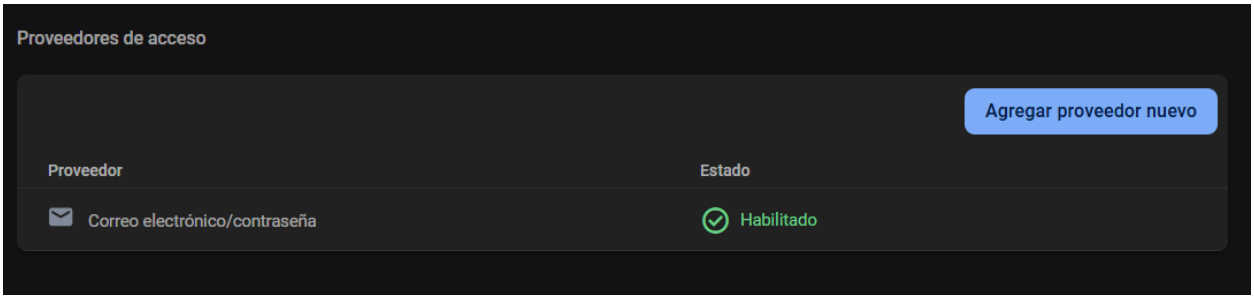
Consideraciones:

Se recomienda vincular la cuenta anónima con un método permanente cuando sea posible,

Integración de login por correo electrónico a la aplicación

1. Se debe crear un nuevo proyecto en firebase y agregar la aplicación una vez hecho esto, se debe habilitar la autenticación de Email/Password





2. Agregar las dependencias necesarias al archivo “build.grandle.kts”

```
implementation(platform("com.google.firebase:firebase-bom:33.12.0"))
implementation("com.google.firebase:firebase-auth")
implementation("androidx.credentials:credentials:1.3.0")
implementation("androidx.credentials:credentials-play-services-auth:1.3.0")
implementation("com.google.android.libraries.identity.googleid:googleid:1.1.1")
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.9.0")
implementation("com.google.android.gms:play-services-auth:20.7.0")
implementation("com.google.firebase:firebase-auth-ktx:22.1.1")
implementation("com.google.android.gms:play-services-auth:20.7.0")
```

3. Inicializar FirebaseAuth en el código de la pantalla de login

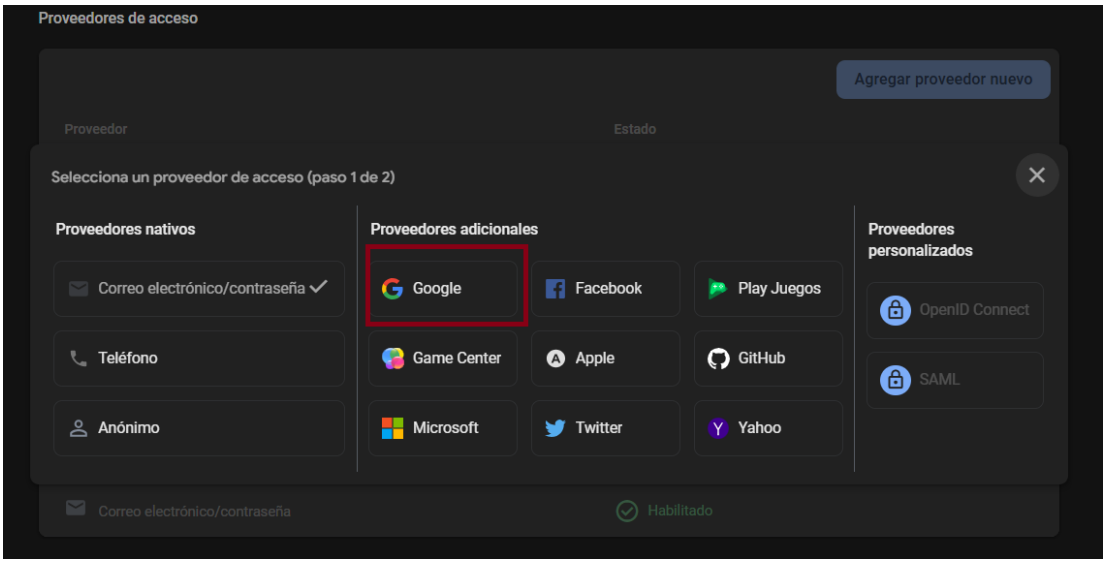
```
private lateinit var auth: FirebaseAuth
private lateinit var googleSignInClient: GoogleSignInClient
private val RC_SIGN_IN = 1
```

4. Implementar el login con correo por medio de la función “signInWithEmailAndPassword”

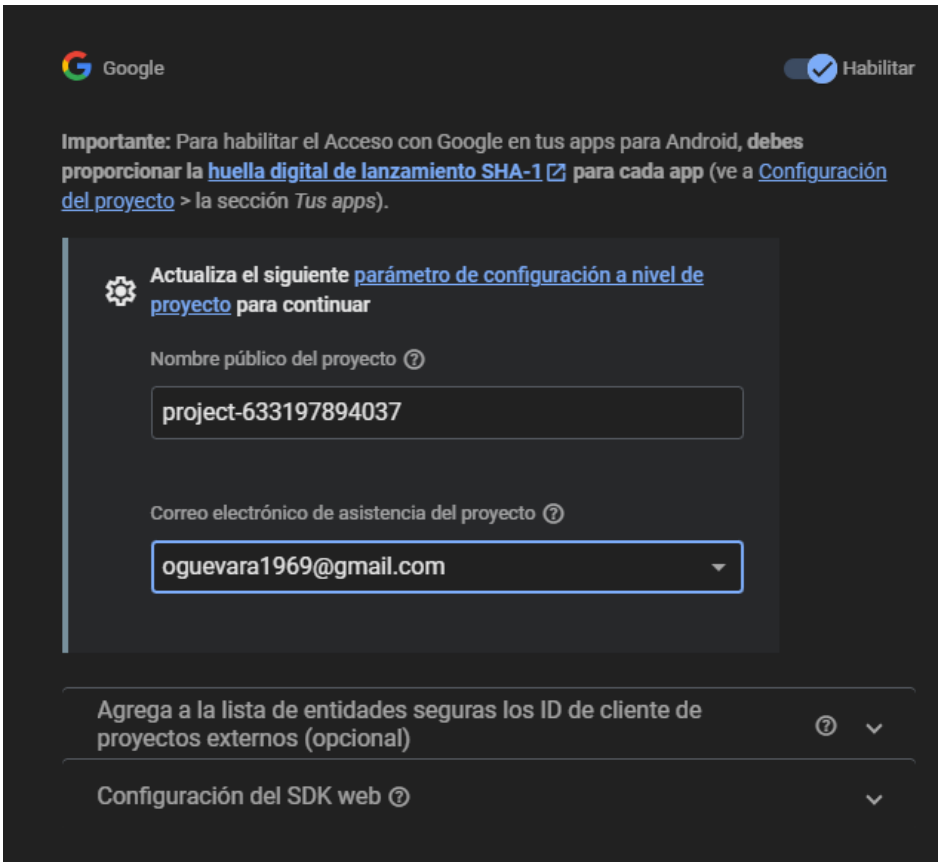
```
private fun signInWithEmail(email: String, password: String) {
    auth.signInWithEmailAndPassword(email, password)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                goToMain()
            } else {
                Toast.makeText(this, "Error: ${task.exception?.message}", Toast.LENGTH_SHORT).show()
            }
        }
}
```

Implementación de autenticación con Google

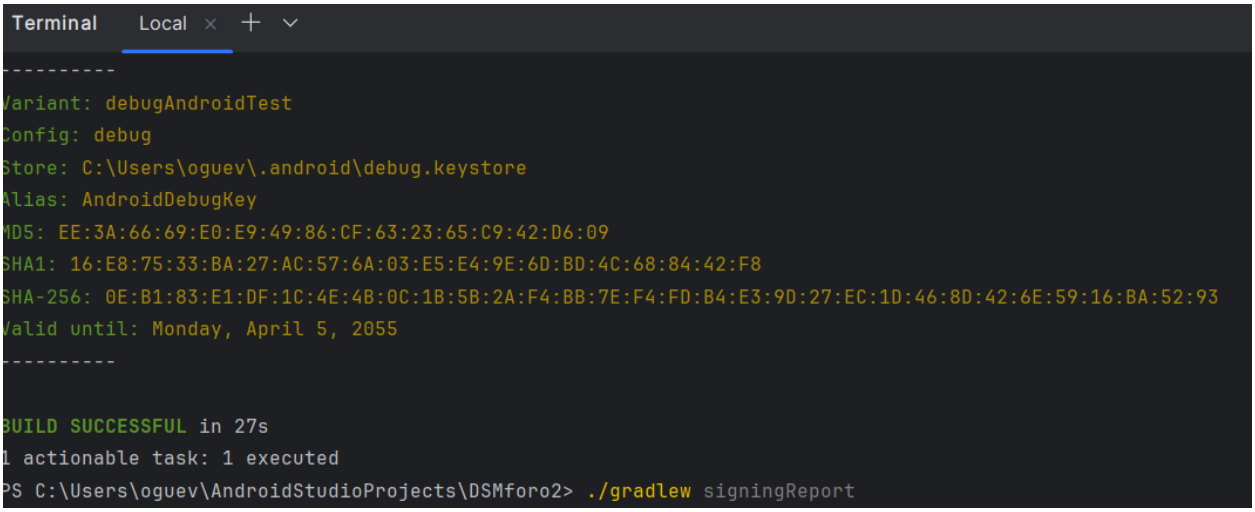
1. Agregar autenticación por medio de Google desde Firebase



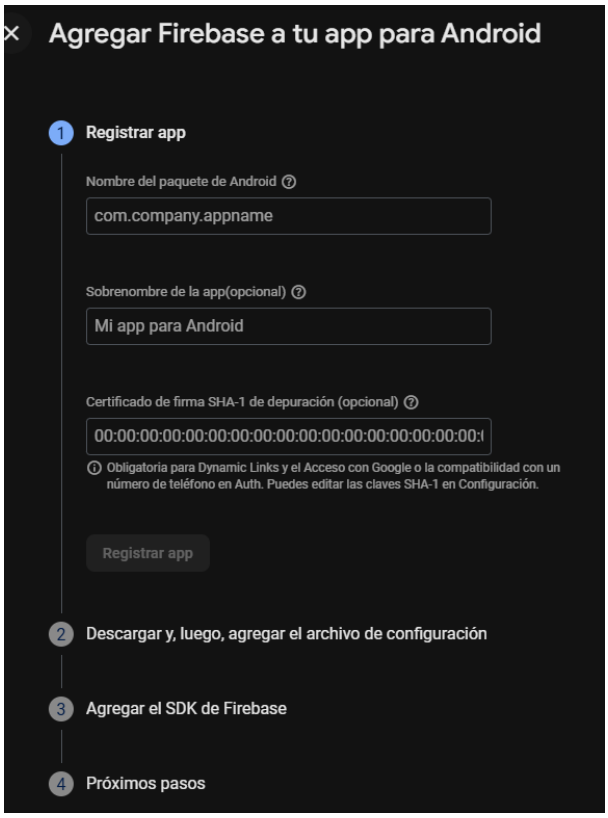
2. Agregar correo electrónico de asistencia de proyecto (se empezará a solicitar la huella digital SHA-1)



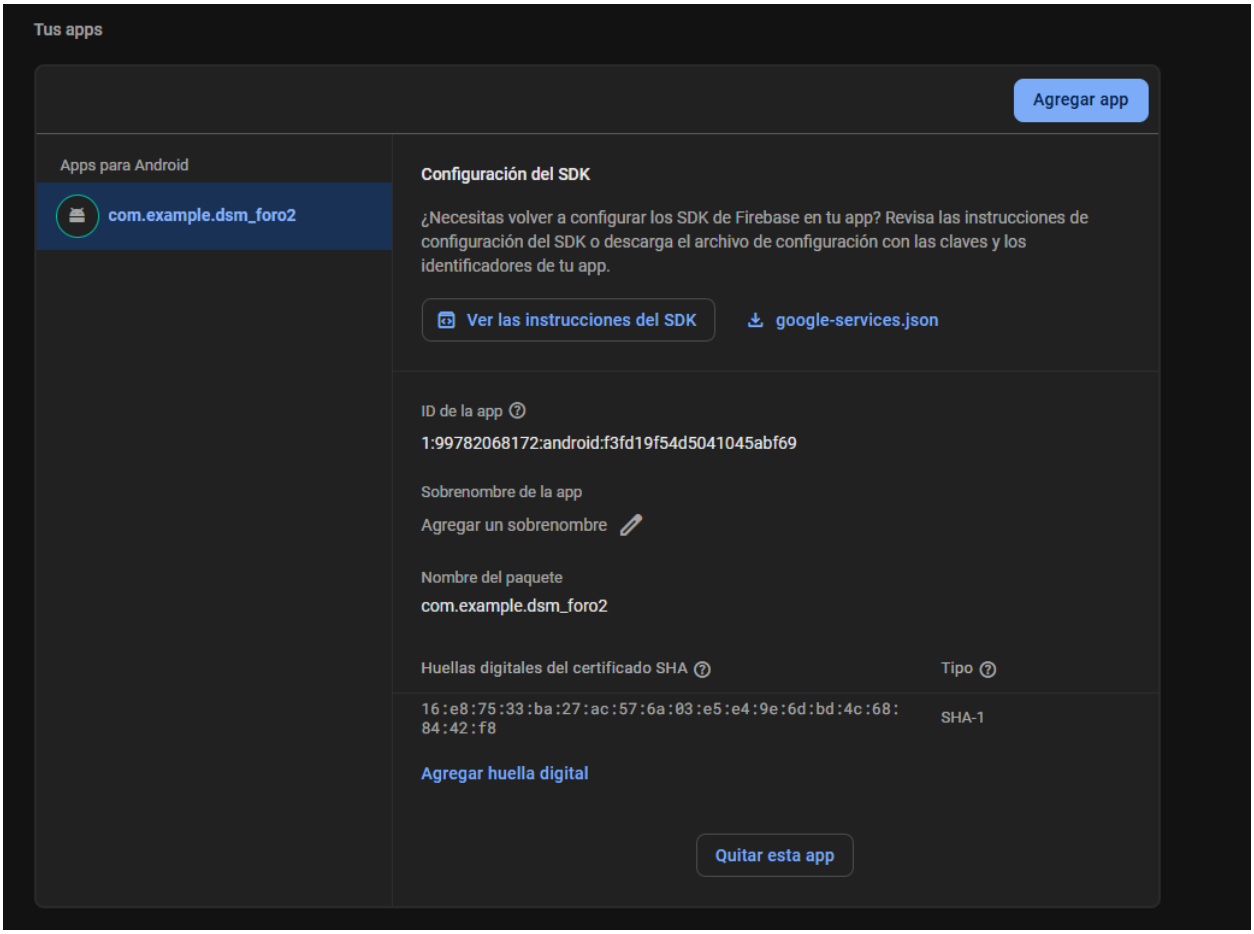
3. Obtener la huella digital por medio del comando ./gradlew signingReport



4. Al momento de enlazar la aplicación con Firebase colocar la huella SHA-1 obtenida



5. Una vez ingresada, descargar el archivo Google-services.json y agregarlo a la carpeta “app” del proyecto



6. Agregar las dependencias necesarias para Firebase Auth

```
implementation(platform("com.google.firebase:firebase-bom:33.12.0"))
implementation("com.google.firebase:firebase-auth")
implementation("androidx.credentials:credentials:1.3.0")
implementation("androidx.credentials:credentials-play-services-auth:1.3.0")
implementation("com.google.android.libraries.identity.googleid:googleid:1.1.1")
implementation("androidx.appcompat:appcompat:1.6.1")
implementation("com.google.android.material:material:1.9.0")
implementation("com.google.android.gms:play-services-auth:20.7.0")
implementation("com.google.firebase:firebase-auth-ktx:22.1.1")
implementation("com.google.android.gms:play-services-auth:20.7.0")
```

7. Agregar la función signInWithCredential la cual convierte el idToken de Google en una credencial válida para Firebase y realiza el login. Si el usuario no existe, Firebase lo creará automáticamente.

```
private fun firebaseAuthWithGoogle(idToken: String) {
    val credential = GoogleAuthProvider.getCredential(idToken, null)
    auth.signInWithCredential(credential)
        .addOnCompleteListener(this) { task ->
            if (task.isSuccessful) {
                goToMain()
            } else {
                Toast.makeText(this, "Error de autenticación con google", Toast.LENGTH_SHORT).show()
            }
        }
}
```

Referencias

Firebase Authentication. (s. f.). Firebase. https://firebase.google.com/docs/auth?hl=es-419

Autenticar con Firebase mediante el uso de cuentas basadas en contraseñas en Android. (s. f.). Firebase. https://firebase.google.com/docs/auth/android/password-auth?hl=es-419

Autentica con Google en Android. (s. f.). Firebase. https://firebase.google.com/docs/auth/android/google-signin?hl=es-419