**Aplicación en Android Studio y Firebase**

TALLER DE   
APLICACIONES MÓVILES

MÓDULO : PRO401-3704-224081-ONL-TALLER DE APLICACIONES MÓVILES

SEMANA : Semana 6

DOCENTE : Cesar Soto Alarcón

ESTUDIANTE : José L. González Medel, Gino M. Villablanca Arias y Bastián N. Jorquera

Introducción

En el desarrollo de aplicaciones móviles, especialmente aquellas que requieren el manejo de datos en tiempo real y autenticación de usuarios, la elección de una infraestructura eficiente es clave para lograr una experiencia óptima. En este proyecto, hemos integrado Firebase en nuestra aplicación Android con el propósito de gestionar tanto el inicio de sesión de los usuarios como los datos de geolocalización en tiempo real. Esto nos lleva a plantearnos la siguiente pregunta: ¿Cuáles son los beneficios de integrar Firebase en una aplicación Android para gestionar la autenticación y los datos en tiempo real?

Firebase es una plataforma de desarrollo de aplicaciones en la nube proporcionada por Google, que ha ganado popularidad por su capacidad para ofrecer servicios backend sin la necesidad de configurar servidores propios. Entre sus características principales se encuentra Firebase Authentication, una solución que permite implementar sistemas de autenticación con múltiples proveedores (correo electrónico, Google, Facebook, etc.), simplificando la gestión de usuarios en la aplicación. En nuestro caso, utilizamos Firebase Authentication para permitir que los usuarios inicien sesión en la aplicación de forma rápida y segura con solo un correo electrónico y una contraseña. Esto no solo redujo el tiempo de desarrollo, sino que también mejoró la seguridad, ya que Firebase se encarga del almacenamiento y cifrado de credenciales de manera confiable.

Otro componente crucial en nuestro proyecto fue Firebase Realtime Database, que nos permitió almacenar y sincronizar datos de ubicación en tiempo real. Al integrar esta base de datos en la aplicación, los cambios en la ubicación se actualizan automáticamente en todos los dispositivos conectados. Este aspecto es particularmente útil en aplicaciones que requieren datos actualizados constantemente, como aplicaciones de mapas, seguimiento de usuarios o sistemas de gestión de flotas. Además, Firebase ofrece una base de datos no relacional que se ajusta dinámicamente a la escala de la aplicación, permitiendo que crezca sin necesidad de modificaciones complejas en la estructura de almacenamiento.

La combinación de Firebase Authentication y Realtime Database no solo ofrece una solución robusta y escalable para la autenticación y el manejo de datos en tiempo real, sino que también mejora la seguridad y el rendimiento de la aplicación. Firebase maneja automáticamente el manejo de tokens, la persistencia de datos y la autenticación, lo que reduce la carga de trabajo en el desarrollo y evita la necesidad de preocuparse por los problemas de infraestructura subyacente.

Desarrollo

El desarrollo de esta aplicación la podemos dividir en 3 grandes etapas:

1. Configuración del desarrollo del proyecto en Android Studio y Firebase
2. Programación de Módulos e integración
3. Pruebas y ajustes

Detallando cada paso, este sería el flujo de desarrollo:

1. **Crear proyecto y configurar sus detalles en Android Studio**
   1. Indicar el SDK mínimo “Android 8.0”, por requerimiento
2. **Configurar archivos con dependencias (build.gradle.kts) tanto a nivel de proyecto, como de módulo**
   1. Las líneas para añadir las indica la plataforma Firebase (ver siguientes pasos: Agregar SDK Firebase)
3. **Configurar AndroidManifest** 
   1. En AndroidManifest.xml añadir permiso de ubicación (Uso de GPS)
   2. Añadir inmediatamente el activity “MenuActivity”
4. **Configurar proyecto en Firebase**
   1. Crear Proyecto (con cuenta Google)
   2. Agregar App Android
   3. Registrar app con mismo nombre del package.
   4. Descargar “google-service.json” y colocarlo en la carpeta /app

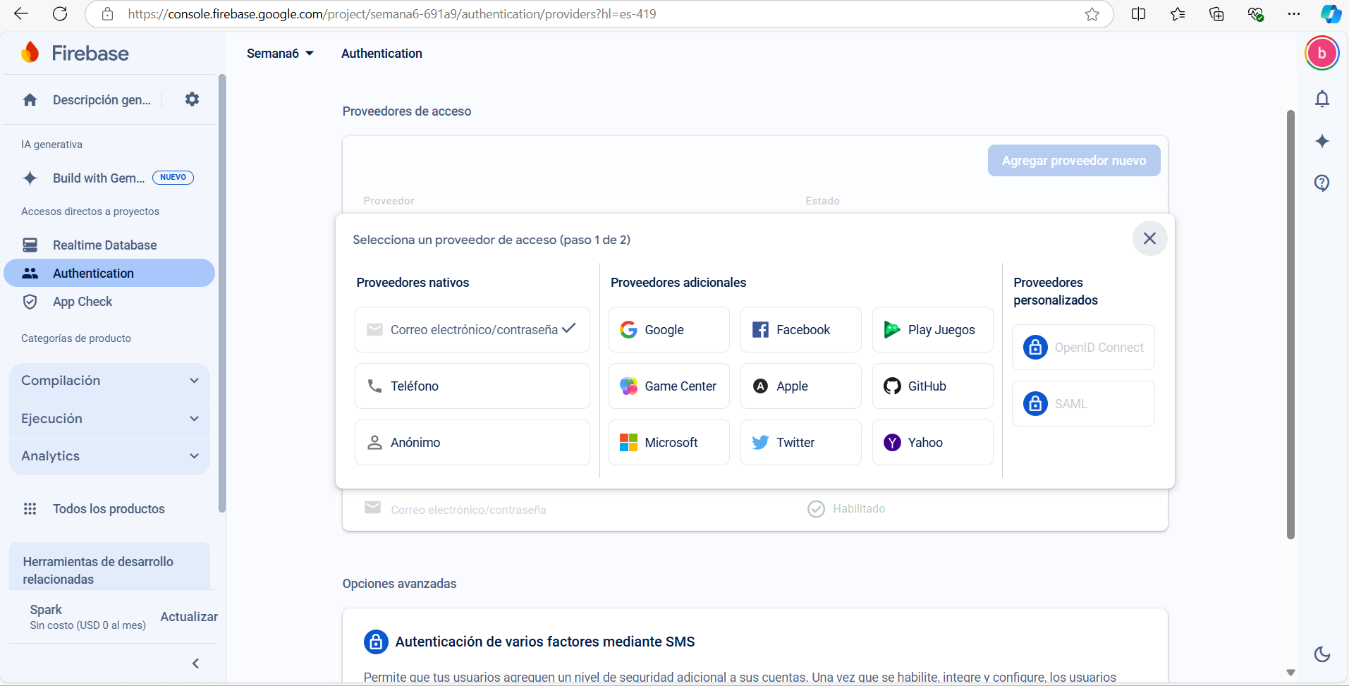
Agregar los SDK de Firebase

**Screens screenshot of a computer screen

Description automatically generated**

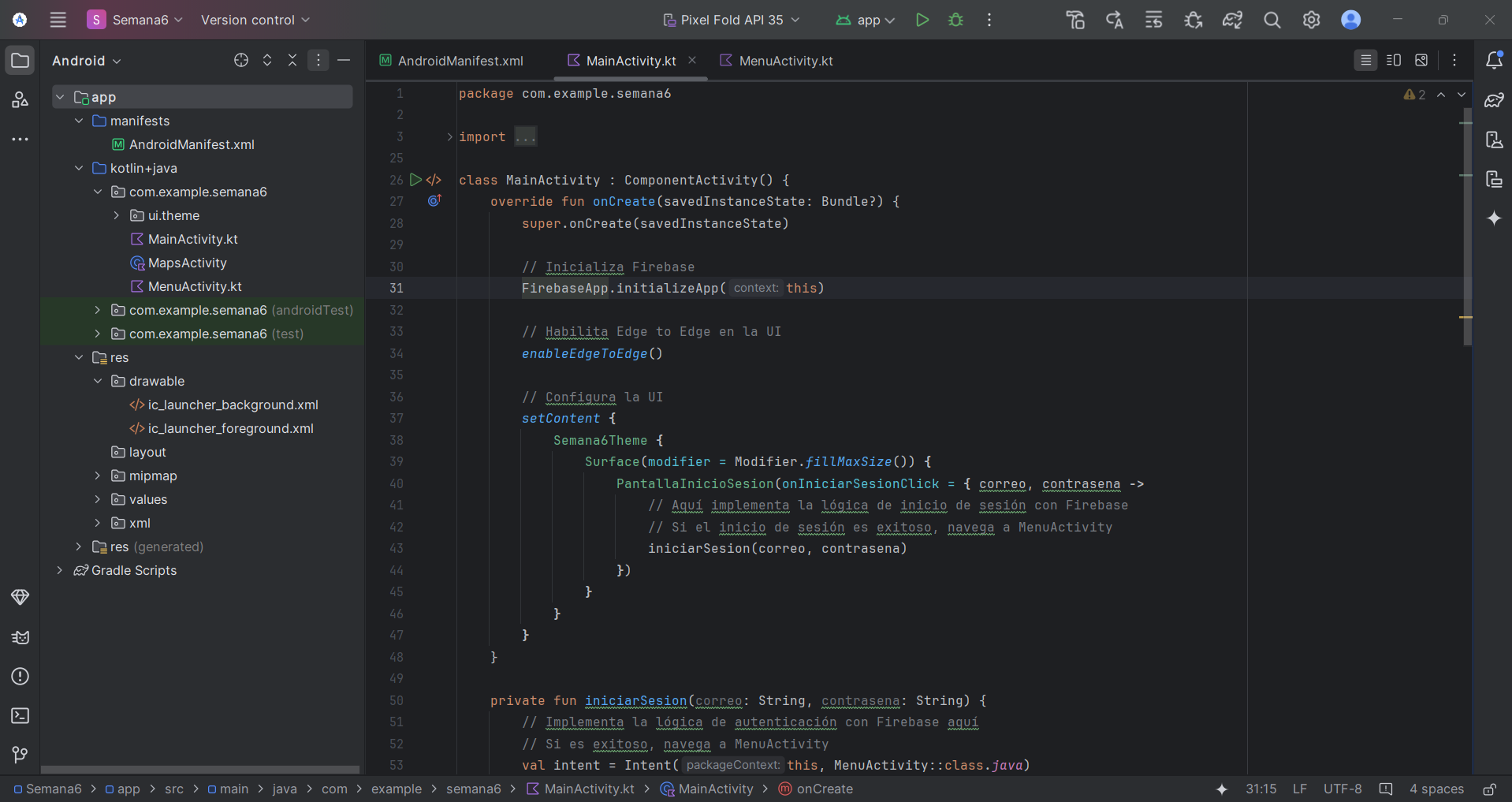
**5.- Configurar Firebase Auth**

* 1. Elegir la opción de correo y contraseña
  2. Interfaz de usuario gráfica, Texto, Aplicación, Word

     Descripción generada automáticamenteEn la pestaña de usuarios, podemos crear nuestros registros correo/contraseña que luego servirán para validar los login.

**6.- Código para MainActivity.kt**

* 1. Instanciar FirebaseAuth[[1]](#footnote-1)
  2. Autenticar usuario (correo y password) con Firebase[[2]](#footnote-2)
  3. Como se utiliza Jetpack Compose, se crea un ‘Composable’ de la vista de esta pestaña de login.
  4. Verificar que cada componente haya sido importado.



**7.- Código para MenuActivity.kt**

* 1. Crear un nuevo archivo kotlin en el mismo package.
  2. Agregar métodos necesarios y ‘Composables’ con los componentes necesarios.
     1. Botón para obtener coordenadas (Pedirá permisos de uso de GPS la primera vez)
     2. Tabla de coordenadas, donde se cargarán los datos de la DB

Captura de pantalla de computadora

Descripción generada automáticamente

**8.- Configurar Firebase RealTime DataBase**

* 1. Desde consola de Firebase, crear Realtime Database
  2. Elegir ‘Comenzar en modo prueba’
  3. Instanciar RealTime Database[[3]](#footnote-3)

**9.- Build Project!**

* 1. Crear una build limpia
  2. Identificar errores y alertas en archivos.
  3. Revisar posibles errores con Logcat

### **Listado de requerimientos**

## Funcionales:

* Autenticar login con registros en Firebase Authentication.
* Identificar coordenadas del móvil que ejecuta la aplicación,
* Almacenar en Firebase Realtime Database, el registro de coordenadas.

## No Funcionales:

* Debe ejecutarse desde la versión “Android 8.0 Oreo”
* Debe ser fácil de entender y usar por los usuarios.

# **Historias de Usuarios**

## Historia de Usuario 1: Autenticación

**Como usuario**, necesito poder ingresar a la aplicación usando mi correo y contraseña. Estos estarán registrados previamente en la base de datos por un administrador.

Criterios de Aceptación:

* El usuario debe poder hacer login.
* El sistema debe validar el correo y contraseña.

## **Historia de Usuario 2: Registro de Coordenadas**

Como usuario, debo poder guardar y ver las coordenadas actuales de mi dispositivo.

Criterios de Aceptación:

* El usuario puede visualizar las coordenadas de latitud y longitud actuales.
* El sistema registra y almacena esa información, junto a un timestamp.
* El sistema permite visualizar los registros almacenados con anterioridad.

# **Cronograma**

Día 1: Configuración del entorno.

Día 2: Creación de proyecto en Firebase.

Día 3: Desarrollo de módulo de login, integrado con Firebase.

Día 4: Pruebas unitaria del login. Prueba de integración con Firebase.

Día 5: Desarrollo de módulo de captura de coordenadas.

Día 6: Integración con Firebase. Almacenar registros en DB.

Día 7: Pruebas unitarias captura de coordenadas. Prueba de integración con Firebase.

Día 8-9: Mejoras de la interfaz. Optimización de código. Publicación en Github.

Día 10: Desarrollo de Documentación para este MVP

Conclusión

El uso de Firebase en el desarrollo de aplicaciones Android ha demostrado ser una solución eficiente y poderosa, especialmente cuando se trata de gestionar la autenticación y los datos en tiempo real. A lo largo del desarrollo de esta aplicación, Firebase no solo ha simplificado procesos críticos como el inicio de sesión y la gestión de usuarios, sino que también ha proporcionado una integración fluida para el manejo de datos de ubicación, facilitando la implementación de funcionalidades avanzadas como la obtención de coordenadas GPS.

En comparación con otras soluciones, Firebase destaca por su simplicidad y facilidad de uso, lo que acelera notablemente el ciclo de desarrollo. Herramientas tradicionales como bases de datos SQL o la implementación manual de servidores de autenticación suelen requerir configuraciones complejas y un mantenimiento constante. Con Firebase, estas tareas se manejan en la nube, lo que libera al desarrollador de la necesidad de gestionar la infraestructura. Esto no solo reduce la carga de trabajo, sino que también minimiza los posibles errores humanos y simplifica la depuración.

Además, el hecho de que Firebase sea una plataforma en tiempo real añade otro nivel de funcionalidad a la aplicación. En nuestro caso, esto se refleja en la posibilidad de actualizar datos de ubicación en el backend al instante, permitiendo que los cambios se reflejen inmediatamente en los dispositivos de los usuarios. Esta capacidad es difícil de implementar de manera eficiente con tecnologías tradicionales, donde las conexiones persistentes o las actualizaciones manuales de la base de datos suelen ser más complicadas de mantener

Otro aspecto clave es la seguridad. Firebase cuenta con protocolos robustos de autenticación, como el inicio de sesión con correo electrónico y contraseña, lo que permite gestionar la identidad de los usuarios de manera segura y sin necesidad de implementar sistemas de autenticación desde cero. Las reglas de seguridad de Firebase permiten al desarrollador controlar de manera detallada quién tiene acceso a los datos, lo que añade una capa adicional de protección.

Por último, Firebase también se integra perfectamente con el ecosistema de Android Studio, ofreciendo SDKs que facilitan la conexión y el uso de sus servicios sin la necesidad de configurar múltiples dependencias externas. Esto no solo reduce el tiempo de desarrollo, sino que también asegura una experiencia más coherente y libre de complicaciones.

Bibliografía

Miles Javier, Molina Felipe (S/F) Taller de aplicaciones móviles (no se especifica ciudad ni país de publicación) Material en general correspondientes de la semana uno a la seis

1. https://firebase.google.com/docs/auth/android/start?hl=es-419 [↑](#footnote-ref-1)
2. https://firebase.google.com/docs/auth/android/password-auth?hl=es-419 [↑](#footnote-ref-2)
3. https://firebase.google.com/docs/database/android/start?hl=es-419 [↑](#footnote-ref-3)