

Universidad Politécnica de Chiapas

Ingeniería en Tecnologías de la Información e Innovación Digital

[Programacion Para moviles]

[C2 – A2 - Notificaciones]

[Nomenclatura del nombre de archivo C2 –A2- Notificaciones-223216-Daniel Peregrino Perez.pdf]

[Alumno -Daniel Peregrino Perez] - [223216]

Docente: [José Alonso Macias Montoya]

Fecha de entrega: [15/06/2025]

1. DESCRIPCIÓN DE LA ACTIVIDAD

1.1. Enunciado del problema

Desarrollar una aplicación móvil nativa para Android desde cero utilizando Android Studio y el lenguaje Kotlin. El objetivo principal de la aplicación es ser capaz de recibir y mostrar notificaciones push enviadas desde la plataforma Firebase Cloud Messaging (FCM). La funcionalidad crítica es que las notificaciones deben ser recibidas y mostradas correctamente incluso cuando la aplicación se encuentra en segundo plano o

completamente cerrada, asegurando una comunicación persistente con el usuario.

1.2. Objetivos de aprendizaje

- **Comprender y configurar un proyecto en la Consola de Firebase**, incluyendo la creación del proyecto, el registro de una aplicación Android y la obtención de los archivos de configuración (google-services.json) y claves (SHA-1).
- **Integrar el SDK de Firebase en un proyecto de Android Studio**, configurando correctamente los archivos build.gradle a nivel de proyecto y de módulo para incluir las dependencias necesarias de Firebase Messaging y Analytics.
- **Implementar un FirebaseMessagingService** para gestionar la recepción de mensajes entrantes y la generación de tokens de registro del dispositivo.
- **Dominar el manejo de permisos en Android**, específicamente el permiso POST_NOTIFICATIONS requerido a partir de Android 13 (API 33) para poder mostrar notificaciones.
- **Obtener y gestionar el token de registro de FCM** de un dispositivo, entendiendo su importancia como identificador único para el envío de notificaciones dirigidas.
- **Enviar notificaciones de prueba desde la Consola de Firebase** para validar el funcionamiento de la aplicación tanto en primer plano como en segundo plano y cerrada.
- **Diagnosticar y solucionar errores comunes** de configuración y ejecución, como IllegalStateException por temas incorrectos y problemas de recepción de notificaciones debidos a la optimización de batería del dispositivo.

Definición clara

El problema a resolver es la implementación completa de un sistema de notificaciones push en Android. Esto implica no solo la codificación del cliente (la app), sino también la configuración de la infraestructura del servidor (Firebase) y la correcta gestión del ciclo de vida y los permisos de la app para garantizar una entrega de notificaciones fiable y robusta en cualquier estado de la aplicación.

Objetivos específicos

- Ser capaz de conectar una app Android a un proyecto de Firebase sin errores.
- Ser capaz de escribir el código Kotlin necesario para solicitar el permiso de notificaciones y obtener el token FCM.
- Ser capaz de crear y registrar un servicio en AndroidManifest.xml que escuche eventos de mensajería.
- Ser capaz de enviar una notificación a un dispositivo específico usando su token desde la Consola de Firebase.
- Ser capaz de verificar que la notificación se recibe con la app cerrada.

2. FUNDAMENTOS TEÓRICOS

Firebase Cloud Messaging (FCM) es un servicio multiplataforma de mensajería en la nube proporcionado por Google que permite enviar notificaciones y mensajes de datos de manera fiable y sin costo. Actúa como un intermediario entre un servidor de aplicaciones y los dispositivos cliente (en este caso, nuestra app de Android). El servidor envía un mensaje a los servidores de FCM, y FCM se encarga de entregarlo al dispositivo correcto.

Componentes clave:

- **Token de registro (FCM Token):** Es un identificador único generado por la librería cliente de FCM en la app. Este token representa la "dirección" del dispositivo y la app. Para enviar una notificación a un dispositivo específico, se debe conocer su token.
- **FirebaseMessagingService:** Es una clase de servicio de Android que se extiende para manejar la comunicación con FCM. Sus dos métodos más importantes son `onMessageReceived()`, que se activa cuando llega un mensaje con la app en primer plano, y `onNewToken()`, que se activa cuando se genera o actualiza el token del dispositivo.
- **Notificaciones vs. Mensajes de datos:**
 - **Mensajes de notificación:** Son manejados automáticamente por el SDK de FCM cuando la app está en segundo plano o cerrada. Tienen un título y un cuerpo predefinidos y se muestran en la bandeja del sistema. Son ideales

para mensajes informativos simples.

- Mensajes de datos: Llevan una carga útil de pares clave-valor personalizados. Siempre se entregan al método `onMessageReceived()` de la app, dándole al desarrollador control total sobre cómo procesar el mensaje, sin importar el estado de la app.
- Permiso `POST_NOTIFICATIONS`: A partir de Android 13 (API 33), las aplicaciones deben solicitar explícitamente el permiso del usuario en tiempo de ejecución para poder mostrar notificaciones. Sin este permiso, ninguna notificación será visible.

3. DESARROLLO DE LA ACTIVIDAD

3.1. Desarrollo

El proceso se dividió en cuatro fases principales:

1. Configuración de Firebase: Se creó un nuevo proyecto en la Consola de Firebase. Se registró la aplicación Android proporcionando el nombre del paquete (`com.example.miappdenotificaciones`) y el certificado de firma SHA-1, obtenido desde Android Studio con el comando `./gradlew signingReport`. Se descargó el archivo de configuración `google-services.json` y se colocó en la carpeta `app/` del proyecto.
2. Configuración de Android Studio: Se modificaron los archivos `build.gradle` para añadir los plugins y dependencias de Firebase.
 - `build.gradle` (Proyecto): Se añadió el plugin `com.google.gms.google-services`.
 - `build.gradle` (Módulo: `app`): Se aplicó el plugin y se añadieron las dependencias `firebase-bom`, `firebase-messaging-ktx` y `firebase-analytics-ktx`.
3. Implementación del Código Kotlin:
 - Se creó la clase `MyFirebaseMessagingService` que hereda de `FirebaseMessagingService`. Se sobrescribieron los métodos `onMessageReceived()` y `onNewToken()` para registrar en Logcat la llegada de mensajes y la obtención del token.
 - Se registró este servicio en el `AndroidManifest.xml` con el intent-filter apropiado para `com.google.firebase.MESSAGING_EVENT`.
 - En el `AndroidManifest.xml`, se añadió el permiso `<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />`.

- En MainActivity.kt, se implementó la lógica para solicitar el permiso de notificaciones en tiempo de ejecución usando `registerForActivityResult` y `ActivityResultContracts.RequestPermission()`.
- También en MainActivity.kt, se añadió el código para obtener el token de FCM usando `FirebaseMessaging.getInstance().token` y mostrarlo en Logcat para poder usarlo en las pruebas.

4. Pruebas y Depuración:

- Se ejecutó la aplicación en un emulador/dispositivo físico. Se aceptó el diálogo de permiso de notificaciones.
- Se copió el token de FCM impreso en la consola Logcat.
- Se accedió a la sección Cloud Messaging de la Consola de Firebase, se redactó una notificación de prueba y se envió al token copiado.
- Se verificó la recepción de la notificación con la app en primer plano, en segundo plano (minimizada) y cerrada (deslizada para cerrarla desde las apps recientes).

[Ir a la documentación](#)

× Agregar Firebase a tu app para Android

1 Registrar app

Nombre del paquete de Android ?

com.example.miappdenotificaciones

Sobrenombre de la app(opcional) ?

App de Notificaciones

Certificado de firma SHA-1 de depuración (opcional) ?

80:9F:78:0D:A9:0A:A6:CF:5D:45:4F:A8:09:C0:E3:4B:4I

Required for Dynamic Links, and Google Sign-In or phone number support in Auth.
Edit SHA-1s in Settings.



2 Descargar y, luego, agregar el archivo de configuración

3 Agregar el SDK de Firebase

4 Próximos pasos

Android

com.example.miappdenotificaciones (androidTest)

com.example.miappdenotificaciones (test)

res

drawable

layout

activity_main.xml

mipmap

values

xml

Gradle Scripts

build.gradle.kts (Project: miappdenotificaciones)

build.gradle.kts (Module :app)

proguard-rules.pro (ProGuard Rules for ":app")

gradle.properties (Project Properties)

gradle-wrapper.properties (Gradle Version)

libs.versions.toml (Version Catalog)

local.properties (SDK Location)

settings.gradle.kts (Project Settings)

MainActivity.kt

build.gradle.kts (miappdenotificaciones)

build.gradle.kts (:app)

activity_main.xml

And

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work prop... Sync Now Ignore these changes

9

android {

41

}

42

}

43

}

44

dependencies {

45

implementation(platform("com.google.firebase:firebase-bom:33.15.0"))

46

implementation("com.google.firebase:firebase-analytics")

47

}

48

implementation(libs.androidx.core.ktx)

49

implementation(libs.androidx.lifecycle.runtime.ktx)

50

implementation(libs.androidx.activity.compose)

51

implementation(platform(libs.androidx.compose.bom))

52

implementation(libs.androidx.ui)

53

implementation(libs.androidx.ui.graphics)

54

implementation(libs.androidx.ui.tooling.preview)

55

implementation(libs.androidx.material3)

56

testImplementation(libs.junit)

57

androidTestImplementation(libs.androidx.junit)

58

androidTestImplementation(libs.androidx.espresso.core)

59

androidTestImplementation(platform(libs.androidx.compose.bom))

Terminal

Local

MD5: 5C:A9:02:F4:1F:0C:56:B9:80:BE:00:24:AB:0D:50:0C

SHA1: 80:9F:78:0D:A9:0A:A6:CF:5D:45:4F:A8:09:C0:E3:4B:48:B7:66:B6

SHA-256: 4B:57:A0:EC:80:CE:82:A5:81:07:90:3B:0A:D4:3A:18:77:0A:0F:DD:35:02:46:E9:80:22:11:42:F8:54:0B:A4

Valid until: miércoles, 5 de mayo de 2055

BUILD SUCCESSFUL in 11s

1 actionable task: 1 executed

PS C:\Users\Marty\AndroidStudioProjects\miappdenotificaciones>

miappdenotificaciones > app > build.gradle.kts > dependencies

46:61 LF UTF-8 4 spaces

Guía para Crear App And...

MiAppDeNotificaciones

Tareas pendientes

The Complete Guide To T...

universidad politécnica de...

Inicio

Netflix

PayPal: Resumen

Cursos en línea: A...

SUBES - 2024

Monster Hunter W...

animeclips.online

(1) HACER VIDEOS...

Top topics - Outlie...

Outlier Playground

Render Dashboard

Firestore Database

Machine Learning

Messaging

AI Logic

Compilación

Ejecución

Analytics

IA

Todos los productos

Herramientas de desarrollo relacionadas

Firebase Studio

Spark

Actualizar

MiAppDeNotificaciones

Messaging

¿Necesitas ayuda para configurar Messaging? Preguntarle a Gemini

Campañas

Informes

Gemini puede ayudarte a analizar el rendimiento de tu campaña y recomendar los próximos pasos.

Es posible que Gemini muestre información imprecisa, por lo que debes verificarla. Obtén más información sobre cómo Gemini en Firebase usa tus datos.

Generar estadísticas de IA

Filtrar campañas

Buscar campañas por nombre, descripc...

Experimento nuevo

Campaña nueva

Campaña	Inicio	Fin	Estado	Segmentación	Última actualización	Envíos o impresiones	Clics o aperturas
Notificacion1.1 Hola mundo	16 jun 2025 5:17:26 p.m.	-	Activa		16 jun 2025	<1,000	0%

0 de 10 notificaciones recurrentes

4. RESULTADOS

4.1. Resultados obtenidos

- Se ha construido exitosamente una aplicación Android funcional que se integra con Firebase Cloud Messaging.
- La aplicación es capaz de recibir y mostrar notificaciones push enviadas desde la consola de Firebase.
- Se ha verificado que la funcionalidad principal opera correctamente en los tres estados de la aplicación: primer plano, segundo plano y cerrada.
- Se ha obtenido el token de registro del dispositivo, lo que permite el envío de notificaciones dirigidas.
- Se ha implementado correctamente el flujo de solicitud de permisos para Android 13+, asegurando la compatibilidad con las versiones más recientes del sistema operativo.
- Se solucionó un error inicial de `IllegalStateException` ajustando el tema de la aplicación a uno derivado de `Theme.AppCompat` (específicamente `Theme.Material3.DayNight.NoActionBar`) en el archivo `themes.xml`.

📶 Internet >

🔌 Bluetooth

🔇 No interrumpir

🔄 Rotación autom..

🤖 Sistema Android ^

Depuración USB conectada

Toca para desactivar la depuración USB

🔊 Notificacion1.1 · ahora 🔔

Hola mundo v

Silenciadas ✕

🔊 Está cifrando el video de respald.. · 10 min

1 restante v

🔊 Subir Escanear Benefic... mensaje v

🤖 Cargando dispositivo mediante USB v

Administrar

Borrar todo

⋮

5. CONCLUSIONES

El desarrollo de esta aplicación ha permitido consolidar conocimientos fundamentales sobre la integración de servicios de terceros como Firebase en Android. Se ha comprendido que la implementación de notificaciones push es un proceso que involucra tanto la configuración del backend (Firebase) como la codificación cuidadosa en el cliente (la app), prestando especial atención a los permisos y al ciclo de vida.

Aprendizaje Adquirido: Se ha comprendido la arquitectura cliente-servidor de FCM, el rol crucial del token de registro y la importancia del `FirebaseMessagingService` como punto de entrada para los mensajes. La necesidad de manejar permisos explícitamente y de considerar las optimizaciones de batería del sistema operativo son lecciones prácticas clave para el desarrollo de aplicaciones robustas en el mundo real.

Relación con los Objetivos: Todos los objetivos de aprendizaje planteados fueron cumplidos. Se logró configurar, implementar y probar exitosamente un sistema de notificaciones de principio a fin. La experiencia práctica de encontrar y solucionar errores reales (como el del tema y los posibles problemas de recepción) fue invaluable y cumplió con el objetivo de diagnóstico y solución de problemas.

Las notificaciones push son una herramienta esencial para la interacción y retención de usuarios en aplicaciones móviles. Dominar su implementación con FCM es una habilidad indispensable para cualquier desarrollador Android moderno.

6. DIFICULTADES Y SOLUCIONES

- **Identificación del problema (Dificultad encontrada):**

La aplicación fallaba al iniciarse con un error `java.lang.RuntimeException: Unable to start activity...` causado por una `java.lang.IllegalStateException: You need to use a Theme.AppCompat theme (or descendant) with this activity.`

- **Análisis de causas (Investigación de los factores que originaron el problema):**

El problema se originó porque la `MainActivity` heredaba de `AppCompatActivity`, la cual requiere que el tema de la aplicación sea un descendiente de `Theme.AppCompat`. El tema por defecto, o uno mal configurado, no cumplía este requisito, causando el crash al intentar renderizar la UI.

- **Implementación de solución (Aplicación de estrategias para resolver la dificultad):**

Se modificó el archivo `res/values/themes.xml`. El atributo `parent` del estilo principal de la aplicación se cambió de un tema base de Android a `Theme.Material3.DayNight.NoActionBar`, que es un tema moderno y compatible.

- **Verificación (Comprobación de la efectividad de la solución aplicada):**

Tras aplicar el cambio y volver a ejecutar la aplicación, el error desapareció y la `MainActivity` se inició correctamente, permitiendo continuar con el desarrollo.

- **Identificación del problema (Dificultad encontrada):**

Tras implementar el código, las notificaciones no llegaban al dispositivo, a pesar de que la aplicación se ejecutaba sin errores y se obtenía un token de FCM.

- **Análisis de causas (Investigación de los factores que originaron el problema):**

Se identificaron varias causas potenciales:

1. No se había concedido el permiso de notificaciones en el dispositivo.
2. La optimización de batería del sistema operativo estaba impidiendo que el servicio se ejecutara en segundo plano.
3. Se estaba utilizando un token de FCM antiguo o incorrecto en la consola de Firebase.

- **Implementación de solución (Aplicación de estrategias para resolver la dificultad):**

Se siguió un checklist de depuración:

1. Se verificó en **Ajustes > Apps > MiAppDeNotificaciones > Notificaciones** que el permiso estuviera activado.
2. Se fue a **Ajustes > Apps > MiAppDeNotificaciones > Batería** y se cambió el modo a **"Sin restricciones"**.
3. Se copió el token más reciente generado en Logcat y se utilizó para enviar una nueva notificación de prueba.

- **Verificación (Comprobación de la efectividad de la solución aplicada):**

Después de asegurar los permisos y la configuración de la batería, y utilizando el token correcto, las

notificaciones comenzaron a llegar de manera fiable al dispositivo, tanto con la app abierta como cerrada.

7. REFERENCIAS

- Google Developers. (s.f.). *Configura un cliente de Firebase Cloud Messaging en Android*. Obtenido de <https://firebase.google.com/docs/cloud-messaging/android/client>
 - Google Developers. (s.f.). *Recibe mensajes en una app para Android*. Obtenido de <https://firebase.google.com/docs/cloud-messaging/android/receive>
 - Google Developers. (s.f.). *Permisos de notificación en tiempo de ejecución*. Android Developers. Obtenido de <https://developer.android.com/develop/ui/views/notifications/notification-permission>
 - Google Developers. (s.f.). *Uso de temas de Material Design*. Android Developers. Obtenido de <https://developer.android.com/develop/ui/views/theming/material-theme>
-