

Universidad Nacional de La Matanza



Firebase Cloud Messaging (FCM)

Intro

Las notificaciones surgen porque necesitamos enviar mensajes a los usuarios

Se las conoce como *push notifications* porque se originan desde el servidor hacia la aplicación cliente (push)

Casos de Uso

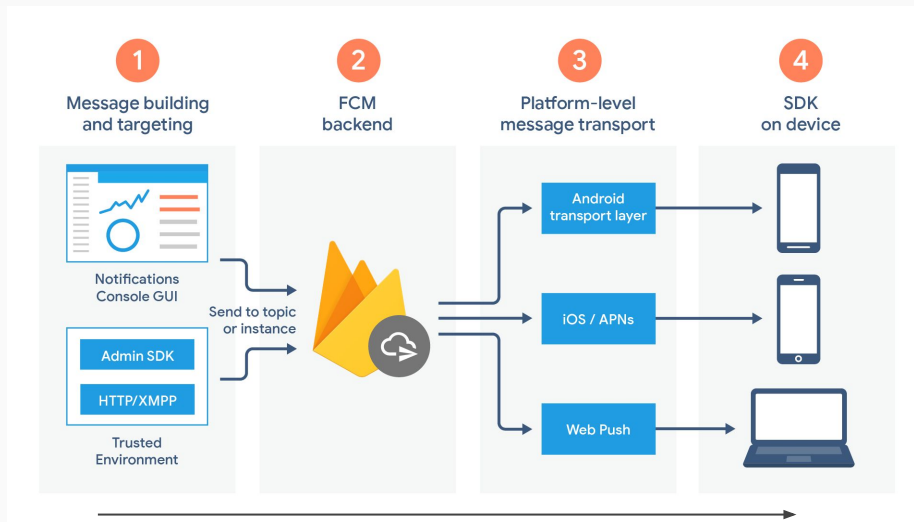
- Como funcionalidad "core" de la app (ej: apps de mensajería)
- Se suelen utilizar como herramienta de marketing: campañas, promociones, eventos)
- Toda información relevante para el usuario: noticias, transacciones bancarias, cambio de turno en juegos, nuevo e-mail, estado de pedidos, etc.

¿Qué vamos a utilizar?



Firebase
Cloud Messaging

¿Cómo funciona?



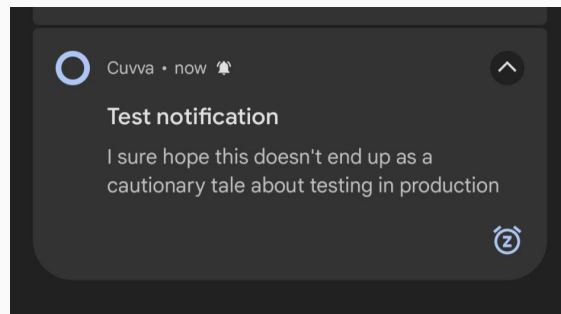
- 1 - Creamos una solicitud de mensaje en la consola de Firebase
- 2 - El servidor de FCM recibe la solicitud y genera el mensaje
- 3 - El mensaje se configura y distribuye de acuerdo a cada plataforma
- 4 - La app mobile/web recibe el mensaje y muestra la notificación en el dispositivo

Ventajas de usar FCM

- Gratis: Es completamente gratuito
- Android, iOS, Web: Permite enviar notificaciones a estas 3 plataformas
- Segmentacion: Podemos elegir a qué grupos de usuarios enviar mensajes
- Facilidad: Nos abstrae de la complejidad de infraestructura
- Bajo consumo de recursos: Sobre todo de batería

Consideraciones

- Los usuarios pueden desactivar las notificaciones
- La entrega de las push notifications no está garantizada (por restricciones de Google/Apple o el mismo SO)
- ¡Nunca hacer pruebas en **producción**!



Integración

- Integrar Firebase modificando los archivos build.gradle y agregando el google-services.json, de acuerdo a lo visto en clases anteriores

En el archivo build.gradle a **nivel app**

```
dependencies {  
  
    // Dependencia de Firebase  
    implementation platform('com.google.firebase:firebase-bom:27.1.0')  
  
    // Dependencia de Firebase Cloud Messaging  
    implementation 'com.google.firebase:firebase-messaging-ktx'  
    ...  
  
    // Otras dependencias  
    ...  
}
```

Para conocer el token, solo basta con llamar al siguiente código:

```
Firebase.messaging.token.addOnCompleteListener {  
    if (it.isSuccessful) {  
        // En este momento conocemos el valor del token  
        Log.d("Notificaciones", it.result!!)  
    }  
}
```

¡Listo!

Nuestra aplicación ya está en condiciones de recibir mensajes de Firebase y mostrar notificaciones cuando los reciba.

Sólo resta ir a la [consola de Firebase](#) y enviar un mensaje de prueba

(Opcional) Integración

Si necesitamos hacer un manejo más personalizado de las notificaciones, lo que tenemos que hacer es crear un servicio especial para esto.

El componente principal para el manejo de los mensajes de FCM es un **servicio** que debe extender de la clase *FirebaseMessagingService*.

Para ello creamos una clase de la siguiente manera:

```
class MiServicioDeNotificaciones : FirebaseMessagingService() {  
  
}
```

Y luego la registramos en el archivo *AndroidManifest* de la siguiente manera:

(Opcional) Integración

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="...">

    <application
        ...>

        <...>

        <service
            android:name=".MiServicioDeNotificaciones"
            android:exported="false">
            <intent-filter>
                <action android:name="com.google.firebase.MESSAGING_EVENT" />
            </intent-filter>
        </service>

    </application>

</manifest>
```

(Opcional) Integración

Una vez que tenemos el servicio creado y definido, necesitamos hacer 2 cosas: escuchar cuando el token se actualiza y recibir los mensajes.

Para ello, sobrescribimos 2 métodos en nuestro servicio de notificaciones:

```
class MiServicioDeNotificaciones : FirebaseMessagingService() {  
  
    // Esta función se llama cada vez que Firebase cambia el token  
    override fun onNewToken(token: String) {  
        super.onNewToken(token)  
        // Guardar el token, enviarlo a nuestro servidor, etc  
    }  
  
    // Esta función se llama cada vez que recibimos un mensaje con datos personalizados  
    // o bien nuestra app esta en foreground (visible)  
    override fun onMessageReceived(message: RemoteMessage) {  
        super.onMessageReceived(message)  
        // Manejar el mensaje recibido  
    }  
  
}
```


Bibliografía

- [Building Mobile At Scale: 5. Push and Background Notifications](#)
- [Documentación Firebase Cloud Messaging](#)
- [Apple Notifications](#)
- [Android Notifications](#)
- [Caso de notificaciones en producción](#)

¿Preguntas?