

Universidad Nacional de La Matanza



Google Analytics for Firebase

Intro

Herramienta de medición que proporciona estadísticas sobre el uso de las apps y la participación de los usuarios.

Permite entender cómo se comportan los usuarios y así tomar decisiones sobre cambios o mejoras en nuestra app



Google Analytics
for Firebase

Funcionalidades

Registro de eventos

Definir propiedades y segmentos de usuarios

Seguimiento de pantallas

Generación de informes

Integración con otras herramientas

Crashlytics

Firebase remote config

[otras más, como Google Tag Manager, Big Query; pero a estas no le daremos foco en la cátedra]

Configuración

Habilitar en Firebase

Al crear un proyecto nuevo, tildar la opción de Analytics

Para verificar o activar en un proyecto ya creado

Página de configuración -> Integraciones -> Google Analytics

Configuración

En el archivo build.gradle, a nivel de app

```
dependencies {  
    // Import the BoM for the Firebase platform  
    // BoM permite que la app siempre utilice las versiones compatibles de las bibliotecas  
    implementation platform('com.google.firebase:firebase-bom:27.1.0')  
  
    // Declarar la dependencia de Analytics CON BoM  
    implementation 'com.google.firebase:firebase-analytics-ktx'  
  
    // Declarar la dependencia de Analytics SIN BoM  
    // sin BoM, se debe especificar la versión de la biblioteca importada  
    implementation 'com.google.firebase:firebase-analytics-ktx:18.0.3'  
}
```


Configuración

En la actividad donde aplicaremos Analytics

Agregar el import:

```
com.google.firebase.analytics.FirebaseAnalytics
```

Declarar el objeto:

```
private lateinit var firebaseAnalytics: FirebaseAnalytics
```

Finalmente, en el método onCreate():

```
firebaseAnalytics = Firebase.analytics
```

Registro de eventos

Registro de eventos

Los eventos proporcionan información sobre lo que sucede en nuestra app (acciones del usuario, eventos del sistema, errores)

Algunos eventos automáticos:

- screen_view
- session_start
- first_open
- app_update
- app_exception
- error

Registro de eventos

Método logEvent()

```
firebaseAnalytics.logEvent(FirebaseAnalytics.Event.SELECT_ITEM) {  
    param(FirebaseAnalytics.Param.ITEM_ID, id)  
    param(FirebaseAnalytics.Param.ITEM_NAME, name)  
    param(FirebaseAnalytics.Param.CONTENT_TYPE, "image")  
}
```

Registro de eventos

Eventos personalizados

```
firebaseAnalytics.logEvent("PartidaFinalizada") {  
    param("ganador", "retador u oponente")  
}
```

Registro de eventos

Parámetros predeterminados

```
val parameters = Bundle().apply {  
    this.putString("level_name", "Caverns01")  
    this.putInt("position", 0)  
}  
  
firebaseAnalytics.setDefaultEventParameters(parameters)
```

Propiedades de usuarios

Propiedades de usuario

Atributos que definimos para describir segmentos de usuarios.

- **Edad**
- **Género**
- **Interés**
- País
- Marca del dispositivo
- Modelo del dispositivo
- Tienda de apps
- Versión de la app
- Versión de SO
- Plataforma
- Idioma
- Nuevo o establecido

Propiedades de usuario

Analytics permite crear hasta 25 propiedades personalizadas.

Para crear una propiedad:

Ir a la Consola de Firebase, en la sección **Analytics**, menú **Custom Definitions**

Para configurar una propiedad en el código:

```
// Tipo de membresía: normal, premium  
firebaseAnalytics.setUserProperty("membership", membresia)
```

Seguimiento de pantallas

Seguimiento de pantallas

Automático en cada `Activity`, guardado con el evento `screen_view`

Si hay un `Activity` por pantalla, realiza el seguimiento de transición de las mismas

Seguimiento de pantallas

Seguimiento manual, mediante el evento *logEvent*

```
firebaseAnalytics.logEvent(FirebaseAnalytics.Event.SCREEN_VIEW) {  
    param(FirebaseAnalytics.Param.SCREEN_NAME, screenName)  
    param(FirebaseAnalytics.Param.SCREEN_CLASS, "MainActivity")  
}
```

Otras funciones

Otras funciones - ID de usuario

Opcional, para identificar eventos que pertenezcan al mismo usuario. No es el id interno de nuestra app, debemos generar un identificador para Analytics

Después de configurar el ID de usuario, todos los eventos futuros se etiquetarán automáticamente con ese valor, pero no afectará a los eventos previamente registrados.

```
FirebaseAnalytics.setUserId("123456");
```

Otras funciones - medición de e-commerce

Utilizando los eventos predefinidos para comercio:

- VIEW_ITEM_LIST
- ADD_TO_WISHLIST
- BEGIN_CHECKOUT
- PURCHASE
- SELECT_ITEM
- VIEW_CART
- ADD_PAYMENT_INFO
- SELECT_PROMOTION

```
firebaseAnalytics.logEvent(FirebaseAnalytics.Event.VIEW_ITEM) {  
    param(FirebaseAnalytics.Param.CURRENCY, "USD")  
    param(FirebaseAnalytics.Param.VALUE, 9.99)  
    param(FirebaseAnalytics.Param.ITEMS, arrayOf(items))  
}
```

Bibliografía

- [Documentación oficial de Google Analytics for Firebase](#)
- [Eventos predefinidos](#)
- [Parámetros predefinidos](#)

¿Preguntas?