Módulo 2. Analítica de apps con Firebase

Fundamentos de analítica de apps con Firebase

En el módulo anterior mencionamos que **más del 59% del tráfico de internet a nivel global proviene de dispositivos móviles**, lo que evidencia la dimensión que ha tomado el acceso desde estos dispositivos en el ecosistema digital actual (StatCounter, 2025). Sin embargo, este número solo cuenta parte de la historia: **la gran mayoría del tiempo que las personas pasan en sus teléfonos inteligentes no están en el navegador, sino dentro de las aplicaciones móviles**. Estudios recientes señalan que aproximadamente **el 88% del tiempo de uso de dispositivos móviles se dedica a aplicaciones nativas**, evidenciando que las experiencias *in-app* dominan el comportamiento digital de los usuarios frente al tiempo dedicado a la navegación web en el teléfono (We Are Social, 2025).

Este nivel de compromiso con las aplicaciones tiene implicancias profundas tanto para las organizaciones como para los profesionales que trabajan con datos. Las *apps* móviles no son solo un canal de acceso: son un espacio donde se generan interacciones continuas, personales y contextuales. Los usuarios

interactúan con múltiples aplicaciones cada día, ya sea para comunicarse, consumir contenido, realizar compras o gestionar aspectos de su vida cotidiana. Entender qué hacen, cómo lo hacen y **por qué ciertos patrones de uso se presentan** requiere instrumentos de medición especializados y adaptados a la complejidad de estos entornos.

Es en este contexto que **Firebase Analytics** se presenta como una herramienta diseñada para capturar con precisión las acciones que ocurren dentro de las aplicaciones móviles. Más allá de las métricas básicas de tráfico web, Firebase provee un enfoque centrado en eventos y comportamientos *in-app*, alineado con las necesidades analíticas propias de las aplicaciones nativas. Esta unidad se propone construir las bases conceptuales y metodológicas para comprender qué es Firebase Analytics, qué datos puede aportar y por qué su integración con Google Analytics 4 (GA4) es estratégica para obtener una visión unificada de la presencia digital de una marca o producto.

A continuación, abordaremos primero una introducción a los conceptos y funcionalidades de Firebase Analytics, poniendo énfasis en cómo se estructura la medición dentro de este ecosistema. Luego veremos con detalle **cómo se integra Firebase con GA4**, lo que permite unificar la analítica de aplicaciones con otras plataformas digitales. Esta comprensión es preparatoria para las unidades siguientes, donde se

desarrollarán aspectos técnicos de implementación y análisis avanzado.

Introducción a Firebase Analytics

Firebase es una plataforma desarrollada por Google que permite a los desarrolladores crear aplicaciones móviles y web de alta calidad, rendimiento y seguridad. Su principal objetivo es ofrecer un entorno integral donde se puedan medir y optimizar interacciones de los usuarios, gestionar datos en tiempo real y mantener la aplicación estable y confiable en distintos dispositivos. De este modo, Firebase se convierte en una solución que conecta la teoría del análisis de datos con la práctica diaria del desarrollo de apps.

Para que Firebase funcione dentro de una aplicación, se utiliza un **SDK (Software Development Kit)**, que es un conjunto de herramientas y librerías que se integran directamente en el código de la *app*. En términos prácticos, esto significa que cuando se añade el SDK de Firebase en una app Android o iOS, automáticamente se habilitan funciones como la recolección de eventos, medición de pantallas, autenticación de usuarios y sincronización de datos en tiempo real. Por ejemplo, si un usuario abre la app, el SDK registra esa acción como un evento y

lo envía a Firebase Analytics, permitiendo al equipo de desarrollo analizar cómo se comportan los usuarios dentro de la aplicación.

Firebase organiza sus servicios en cuatro segmentos principales que permiten cubrir todas las necesidades de una *app*: desde el desarrollo, la medición de usuarios, la interacción con ellos, hasta la monetización. Para comprender mejor estos segmentos y sus funciones, se puede representar de forma clara en la siguiente tabla:

Tabla 1. Segmentos de servicio de Firebase

Segmento	Función	Ejemplo
Analytics	Medir y analizar el comportamiento de los usuarios	Registrar eventos como aperturas de pantalla, clics o conversiones dentro de la app
Develop	Proveer herramientas para crear y optimizar la aplicación	Cloud Messaging, Realtime Database, Crash Reporting, Authentication

Grow	Incrementar la base de usuarios y la interacción	Enviar notificaciones personalizadas, mejorar visibilidad de la app en búsquedas, gestionar referencias
Earn	Monetizar la aplicación mediante publicidad	Mostrar anuncios segmentados con AdMob sin afectar la experiencia del usuario

Fuente: elaboración propia

Gracias al SDK, todos estos servicios se integran de manera automática en la aplicación, permitiendo que los datos se sincronicen en tiempo real con la infraestructura de Google. Esto asegura que cualquier cambio o actualización se refleje inmediatamente en los dispositivos de los usuarios y proporciona información precisa para la toma de decisiones sobre diseño de interfaz y experiencia de usuario, rendimiento y estrategias de monetización.

De esta forma, Firebase Analytics se convierte en el núcleo de la plataforma, permitiendo comprender en detalle cómo los usuarios interactúan con la *app*, qué funcionalidades son más utilizadas y dónde existen oportunidades de mejora. Su integración práctica mediante el SDK facilita que los desarrolladores puedan implementar estas mediciones sin interrumpir la experiencia de los usuarios, conectando directamente la teoría con la práctica en el desarrollo de aplicaciones móviles.

Ahora bien, en los cuatro segmentos de Firebase —**Analytics, Develop, Grow y Earn**— se agrupan múltiples funciones que permiten cubrir todas las necesidades de una aplicación, desde su desarrollo y optimización, hasta la medición de usuarios, la interacción con ellos y la monetización. Cada segmento contiene herramientas específicas que facilitan tareas concretas y mejoran la experiencia del usuario dentro de la *app*.

Entre todas las funcionalidades disponibles en Firebase, en este módulo nos centraremos específicamente en las funcionalidades del segmento «Analytics», dado que son las que permiten medir el uso de la aplicación, registrar eventos, analizar interacciones in-app y comprender cómo los usuarios se comportan dentro de una *app* móvil.

Realtime

La funcionalidad «Realtime» dentro del segmento «Analytics» permite observar lo que ocurre en la app en tiempo real, es decir, mientras los usuarios están activos. Esta sección del panel muestra información instantánea sobre el comportamiento de quienes usan la aplicación en ese momento, sin necesidad de esperar los reportes procesados por Firebase.

En la figura 1 se muestra la vista completa de esta sección. A la izquierda, en el menú de navegación, aparece el acceso directo a «Realtime», junto con las demás funciones de medición. A la derecha, se observa el tablero con distintos datos relevantes:

Figura 1. Vista de la sección «Realtime» en Firebase Analytics

Fuente: Actualizatec, s.f., https://goo.su/q3T3PT

El gráfico de barras en la parte superior del panel indica **el número de eventos por minuto** registrados en los últimos 30 minutos. Esto permite ver de forma clara si hay actividad sostenida, picos de uso o momentos sin interacción. Justo debajo, se presenta un gráfico circular que resume la **categoría de dispositivo** desde el cual los usuarios acceden a la *app* en ese momento, diferenciando, por ejemplo, entre teléfonos móviles y tabletas. Además, el mapa de calor ubicado a la derecha muestra en qué partes del mundo se encuentran los usuarios activos, representando su ubicación con puntos azules. Esto es especialmente útil para saber desde qué regiones se está utilizando la aplicación, lo que puede tener implicancias en decisiones sobre idioma, horario de publicación de contenido o incluso soporte técnico.

Como se observa, «Realtime» permite una primera validación de que los eventos están llegando correctamente, pero también brinda una visión inmediata sobre la actividad, los dispositivos y la distribución geográfica de los usuarios activos en la *app*.

Events

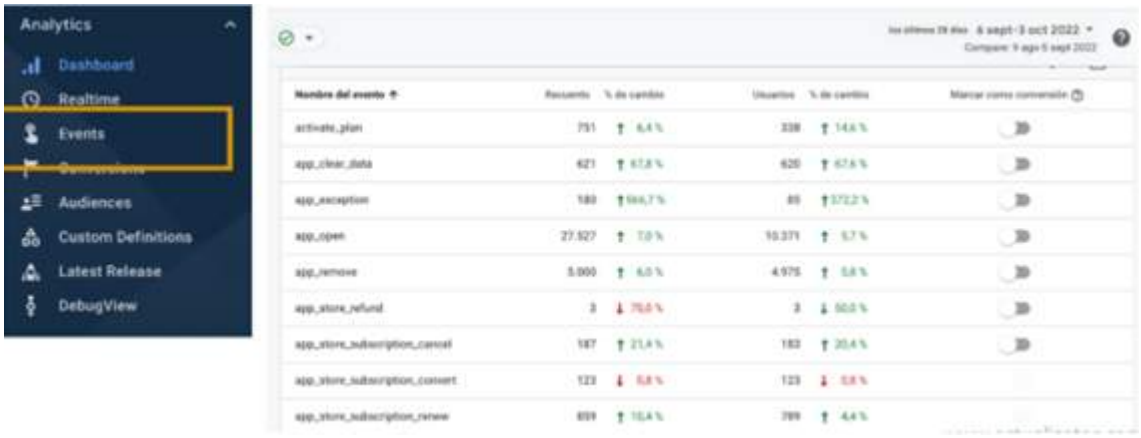
Como se verá en la unidad 2, **los eventos se envían desde el SDK de Firebase**, ya sea de manera automática o definidos manualmente por el equipo de desarrollo. Sin embargo, **para**

que los datos se visualicen correctamente en esta sección del panel, es necesario configurar las dimensiones y métricas asociadas a esos eventos. Si no se realiza esa configuración en el entorno de Firebase, los datos pueden estar siendo recolectados, pero no aparecerán en los reportes. Por eso, la configuración dentro del panel es una etapa clave en el proceso de medición. Por el momento, **nos centraremos únicamente en comprender la funcionalidad «Events» dentro de Firebase Analytics**, sin entrar todavía en su implementación técnica.

La funcionalidad **«Events»** permite acceder al listado completo de eventos que se están registrando dentro de la *app*. Recordemos que los eventos son las acciones que realizan los usuarios mientras navegan; en este caso, mientras usan la aplicación: abrir la *app*, suscribirse, borrar datos, hacer una compra, cancelar una suscripción, entre otros. Esta sección del panel permite visualizar qué eventos se están activando, con qué frecuencia y cuántos usuarios los ejecutaron en un período determinado.

La figura 2 muestra cómo se presenta esta sección en el panel de Firebase. A la izquierda, en el menú de navegación, se encuentra el acceso directo a «Events». A la derecha, aparece la tabla que resume los eventos registrados en la *app*, ordenados por nombre.

Figura 2. Vista de la sección «Events» en Firebase Analytics



Fuente: Actualizatec, s.f., https://goo.su/q3T3PT

Cada fila corresponde a un evento distinto, como app_open, app_remove, app_exception, entre otros. Las columnas muestran la siguiente información:

- **Recuento**. El número total de veces que se ejecutó ese evento en el período analizado.
- **Porcentaje de cambio**: la variación respecto del período anterior, lo que permite analizar tendencias.
- **Usuarios**: cantidad de usuarios únicos que ejecutaron ese evento.
- **Marcar como conversión**: un interruptor que permite indicar si ese evento debe considerarse una conversión, es

decir, una acción relevante para los objetivos del negocio (por ejemplo, una compra o un registro).

Este panel permite conocer en detalle cómo interactúan los usuarios con la *app*, qué funcionalidades se utilizan más y cómo cambian esos comportamientos en el tiempo. Además, permite detectar comportamientos inesperados, como errores o caídas en acciones importantes.

Audiences

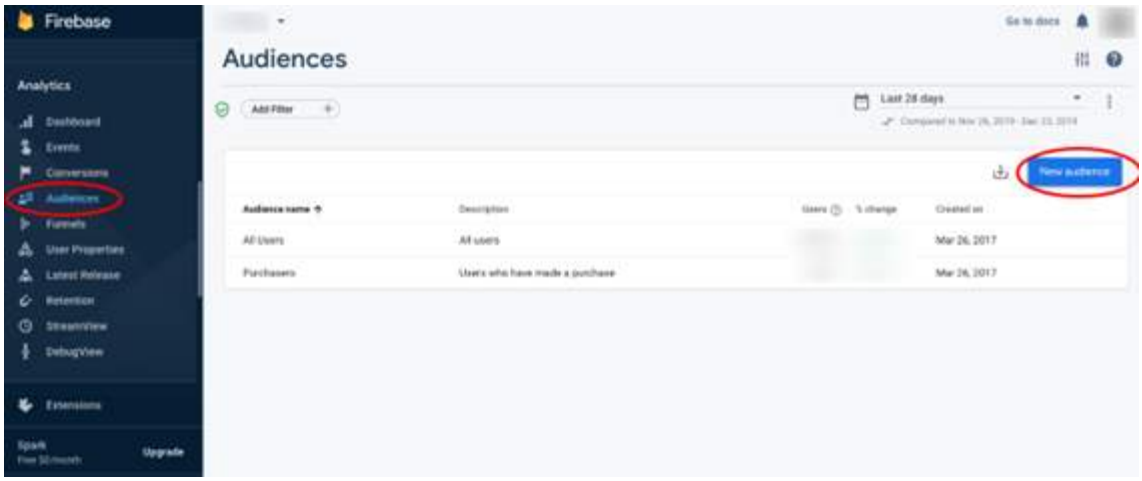
La funcionalidad **«Audiences»** permite segmentar a los usuarios de la *app* en distintos grupos, según sus características o comportamientos. Esta segmentación es clave para interpretar mejor a los usuarios, personalizar la comunicación, diseñar campañas relevantes y tomar decisiones basadas en datos reales. A diferencia de los eventos, que capturan acciones, las audiencias agrupan usuarios según criterios definidos por quienes gestionan el proyecto.

Una **audiencia** puede definirse, por ejemplo, a partir de un evento como `ecommerce_purchase` (usuarios que realizaron una compra), `first_open` (usuarios que abrieron la aplicación por primera vez), o `add_to_cart` (usuarios que añadieron un producto al carrito). También puede crearse a partir de **propiedades del usuario**, como edad, idioma, ubicación o género. Además, es

posible **combinar múltiples eventos y propiedades** para definir grupos más precisos.

La figura 3 muestra la vista principal de esta funcionalidad en el panel de Firebase. A la izquierda se encuentra el acceso a «Audiences», mientras que en el área principal se visualiza el listado de audiencias creadas. En el ejemplo se presentan dos: *All Users* y *Purchasers*. Esta última agrupa a quienes han realizado al menos una compra. Desde el botón «New audience» es posible crear nuevas audiencias según los criterios que se consideren relevantes para el análisis o para campañas de *remarketing*.

Figura 3. Vista de la sección «Audiences» en Firebase Analytics



Fuente: Radhakrishnan, s.f., https://goo.su/yClTNc

Crear audiencias permite dirigir estrategias más efectivas hacia grupos con intereses específicos. Por ejemplo, si una audiencia está compuesta mayoritariamente por mujeres de entre 20 y 40 años interesadas en libros, se pueden ofrecer promociones y contenido alineado con ese perfil. También es posible experimentar con distintos mensajes o funcionalidades según la audiencia, y medir cómo responde cada grupo.

En la siguiente unidad nos detendremos en **cómo configurar audiencias desde el panel de Firebase**, combinando eventos y propiedades de usuario para lograr segmentaciones útiles y accionables.

Conversions

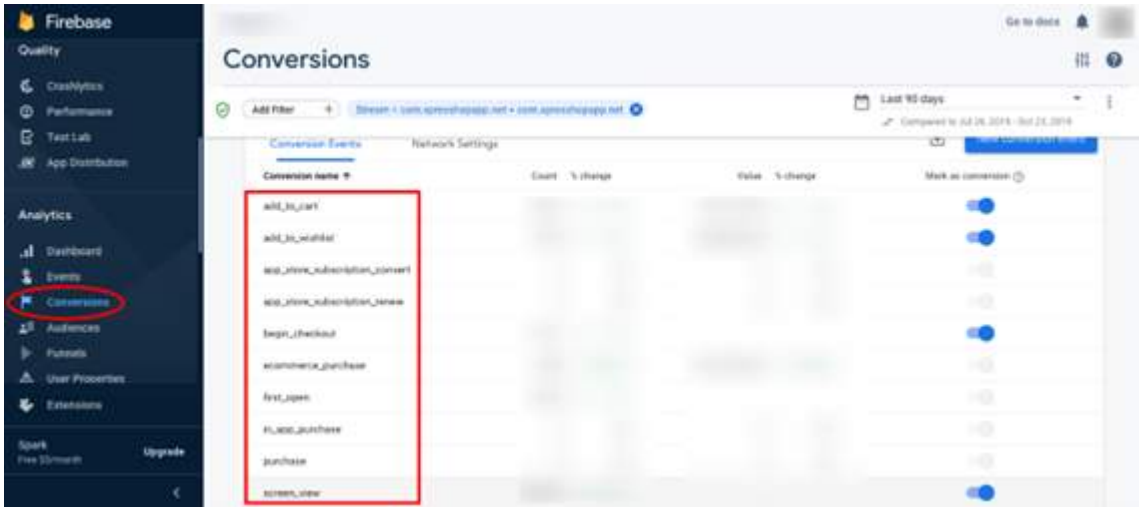
Las conversiones en Firebase permiten identificar qué acciones dentro de una *app* tienen un valor estratégico para el negocio, como una descarga, una compra o la suscripción a un servicio. A través de la sección **«Conversions»**, es posible monitorear estos eventos relevantes y analizar cuántos usuarios los ejecutaron,

cómo evolucionan en el tiempo y cuál es su impacto en el crecimiento del proyecto.

A diferencia de otros eventos, los eventos marcados como conversiones son seguidos de forma prioritaria, ya que representan objetivos clave. Firebase no define qué debe considerarse una conversión; es el equipo de desarrollo quien elige qué eventos activar como tales. Esto permite adaptar la medición a las metas específicas de cada aplicación.

La figura 4 muestra cómo se visualiza esta funcionalidad dentro del panel de Firebase. En la columna «Conversion name» se listan todos los eventos disponibles que podrían marcarse como conversión, como `add_to_cart`, `begin_checkout`, `purchase` o `first_open`. En la parte derecha de la tabla, se puede activar o desactivar la opción «Mark as conversion» (marcar como conversión) para cada evento. A partir de esa selección, el evento comienza a ser considerado como conversión y se incorpora a los reportes y análisis comparativos.

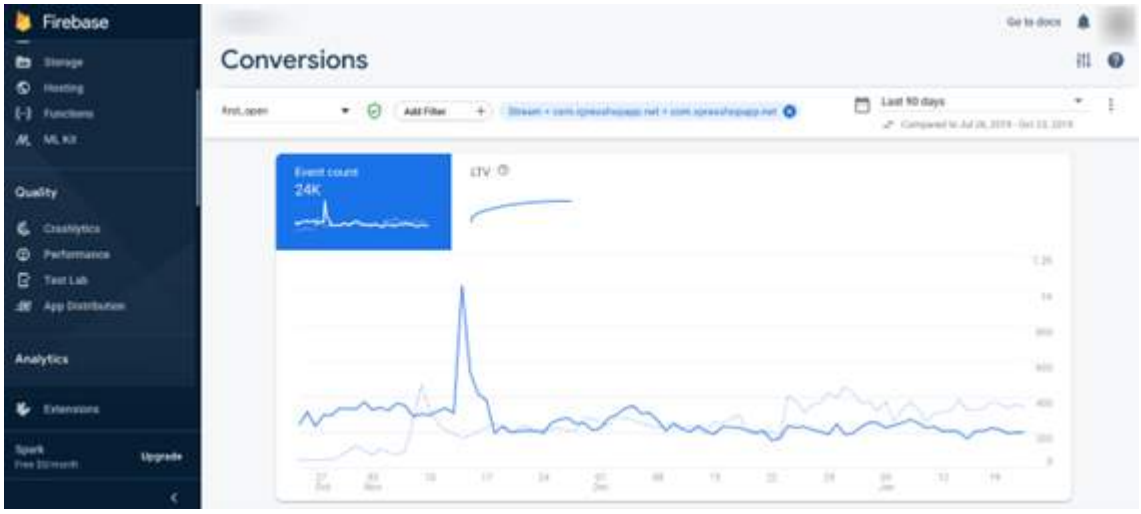
Figura 4. Sección «Conversions» y selección de eventos marcados como conversión



Fuente: Radhakrishnan, s.f.a., https://goo.su/CMBCF

Una vez que un evento se ha definido como conversión, Firebase permite analizar su evolución en profundidad. Por ejemplo, si se selecciona el evento first_open —que mide cuántos usuarios descargaron la *app* y la abrieron por primera vez— es posible ver un desglose detallado del comportamiento de ese evento a lo largo del tiempo. La figura 5 muestra el gráfico correspondiente: se observa que en los últimos 90 días se registraron aproximadamente 24000 first_open. El pico en la curva podría deberse a una acción promocional, un cambio en la estrategia de adquisición o el lanzamiento de una nueva funcionalidad.

Figura 5. Análisis del evento first_open como conversión



Fuente: Radhakrishnan, s.f.a., https://goo.su/CMBCF

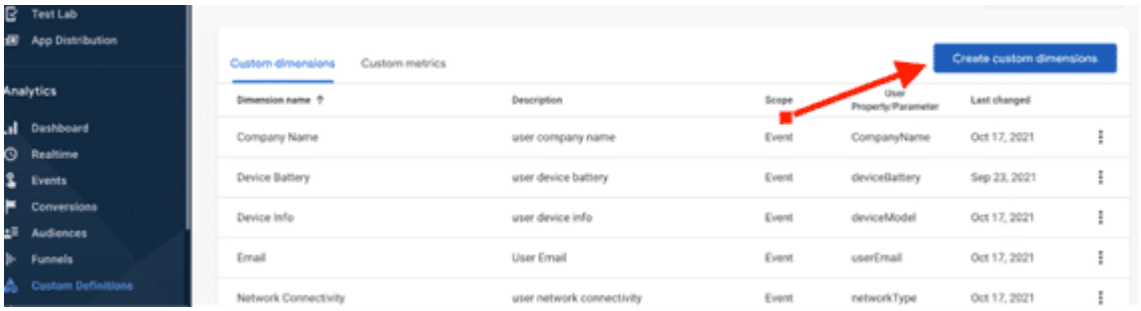
Si se hace clic sobre cualquiera de los eventos marcados como conversión, el panel redirige automáticamente a una vista detallada (como la que se muestra en la figura 2), donde se puede observar el recuento de eventos, la variación respecto de períodos anteriores, el valor económico asociado —si aplica— y otros filtros que permiten refinar el análisis. De este modo, «Conversions» se convierte en una herramienta clave para comprender qué acciones tienen mayor impacto en los objetivos de la aplicación y ajustar las estrategias en función de los resultados obtenidos.

Custom Definitions

Firebase permite recolectar una gran cantidad de datos automáticamente, pero en muchos casos los equipos de desarrollo necesitan registrar información específica que no está incluida por defecto. La funcionalidad **«Custom Definitions»** permite crear esas métricas y dimensiones personalizadas para que puedan aparecer en los reportes y ser analizadas. Por ejemplo, si una *app* envía el nivel de batería del dispositivo como parte de un evento, este dato no se verá reflejado en Firebase hasta que se configure como dimensión personalizada.

La figura 6 muestra el panel donde se crean y gestionan estas definiciones.

Figura 6. Vista principal de la funcionalidad «Custom Definitions»



Fuente: Stack Overflow, s.f., https://goo.su/KoWFB

En el listado se ven ejemplos de dimensiones ya creadas, como Device Battery o User Email. Para agregar una nueva, hay que hacer clic en el botón **«Create custom dimensions»,** buscar el nombre del parámetro que se está recolectando y asignarle una etiqueta clara para su visualización —recordemos que el parámetro es un dato específico que se recolecta en relación con una acción o con el perfil del usuario, como por ejemplo la ciudad, el tipo de red o el modelo del dispositivo—. Este paso es indispensable: si no se realiza esta configuración, los datos no aparecerán en los informes aunque estén siendo recolectados correctamente.

User Properties

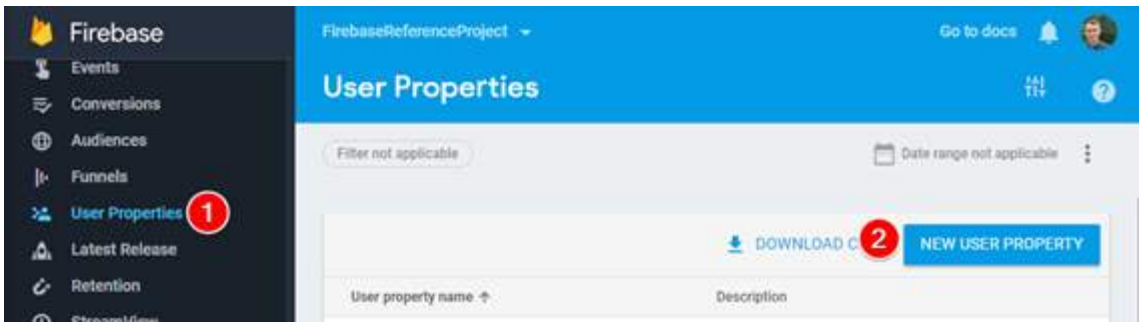
La funcionalidad **«User Properties»** permite asociar atributos específicos a los usuarios de una *app*, con el fin de segmentar audiencias y comprender mejor el comportamiento dentro de Firebase Analytics. A diferencia de los eventos, que registran acciones puntuales, las propiedades del usuario describen **características generales y persistentes**, como idioma preferido, tipo de plan, ciudad de conexión o dispositivo utilizado.

Por ejemplo, se podría definir una propiedad «idioma» para identificar qué usuarios usan la *app* en idioma «español» y luego analizar su comportamiento frente a quienes utilizan otro idioma. También es posible establecer propiedades como «tipo de

usuario» para diferenciar entre usuarios gratuitos y premium, y así observar diferencias en tasas de conversión, retención u otras métricas clave.

En la figura 7, se muestra el panel de Firebase donde se gestionan las propiedades del usuario.

Figura 7. Panel de gestión de propiedades de usuario en Firebase Analytics



Fuente: Lee, s.f., https://goo.su/voeiUjn

Desde el botón «New User Property» es posible crear una nueva propiedad personalizada, asignándole un nombre y una descripción. Al hacer clic, se abre un formulario simple donde se debe definir el nombre de la propiedad (por ejemplo, tipo_usuario, idioma, plan_actual) y, de manera opcional, una breve descripción que indique su propósito.

Estas propiedades quedan registradas en el sistema y luego se pueden utilizar como filtros en los reportes o para construir audiencias específicas. Esto permite comparar comportamientos entre distintos grupos de usuarios dentro de la app, como por ejemplo quienes usan un idioma determinado, o quienes acceden con una versión gratuita versus una versión paga.

DebugView

La funcionalidad «DebugView» permite visualizar en tiempo real los eventos que se están generando desde una instancia específica de la *app* durante una sesión de prueba. Esto resulta especialmente útil durante el desarrollo o ajuste de la medición, ya que permite verificar si los eventos se están registrando correctamente, en qué orden se activan y con qué parámetros llegan.

A diferencia de los reportes generales, donde la información puede tardar en procesarse, en «DebugView» los eventos se muestran de forma inmediata, permitiendo validar el funcionamiento del sistema de medición sin esperar a que los datos se reflejen en los paneles agregados. En la siguiente unidad exploraremos en detalle cómo activar esta función y cómo

interpretar la información que ofrece para el análisis de eventos y parámetros.

Como vimos, Firebase Analytics ofrece múltiples funcionalidades clave que permiten medir en profundidad el comportamiento de los usuarios dentro de una *app*. Herramientas como «Realtime», «Events», «Conversions», «Audiences», «Custom Definitions», «User Properties» y «DebugView» conforman un ecosistema completo para capturar datos, segmentar audiencias y tomar decisiones basadas en evidencia.

Ahora bien, para que toda esta información cobre verdadero valor analítico, es fundamental integrarla con una plataforma capaz de visualizar, interpretar y combinar los datos con otras fuentes. En este sentido, la conexión entre Firebase y Google Analytics 4 (GA4) permite ampliar las capacidades de análisis, ya que los datos recolectados en la *app* se sincronizan automáticamente con GA4 y pueden combinarse con datos de sitios web u otras plataformas. En el próximo apartado veremos cómo se realiza esta integración y qué ventajas ofrece para una visión unificada del comportamiento del usuario.

Integración GA4 + Firebase

Cuando se trabaja con Firebase, los datos recolectados sobre el uso de una *app* (como aperturas, compras, registros o cancelaciones) quedan almacenados en su propio entorno. Sin embargo, Firebase no permite crear reportes comparativos complejos, ni analizar embudos o cruzar información con campañas publicitarias o con el uso de un sitio web. Por eso, es recomendable vincular el proyecto de Firebase con Google Analytics 4 (GA4), que es la plataforma de análisis más completa de Google.

Con esta integración, todo lo que se mide en la *app* —como el evento `first_open` que registra cuándo una persona abre la aplicación por primera vez, o `purchase`, que detecta cuándo alguien compra algo— se envía también a GA4. Desde allí, se pueden crear reportes personalizados, comparar cuántas personas compran según el canal de adquisición, o ver si los usuarios que usan iOS se comportan distinto a los de Android. Por ejemplo, una tienda digital podría ver en GA4 que los usuarios que llegan desde una campaña en redes sociales tardan más en concretar una compra que los que llegan desde una búsqueda en Google.

Veamos, a continuación, cómo realizar esta integración paso a paso.

Crear una propiedad en Google Analytics 4

Antes de vincular Firebase con GA4, es necesario tener creada una propiedad en Google Analytics 4. Este paso se realiza desde la cuenta de Google Analytics, seleccionando la opción «Crear propiedad» en el panel de administración. Por ejemplo, supongamos que estamos trabajando con una aplicación de entrenamiento físico llamada MiEntrenoApp, que permite a los usuarios seguir rutinas personalizadas y ofrece una suscripción paga para desbloquear contenido exclusivo. Al crear la propiedad en GA4, la nombramos «MiEntrenoApp Analytics», elegimos la zona horaria local y configuramos la moneda en pesos argentinos (ARS), ya que la app ofrece planes pagos. Esta propiedad quedará lista para ser vinculada con Firebase en los pasos siguientes.

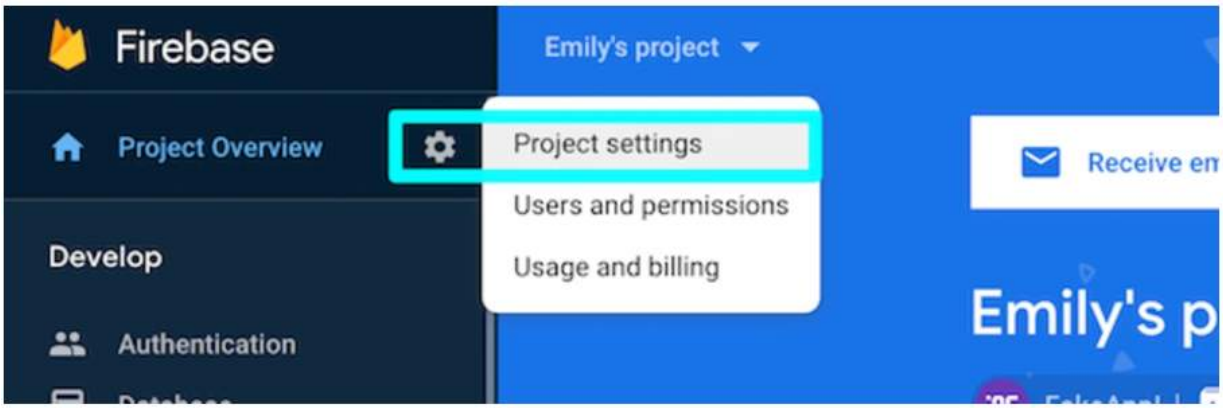
Vincular Firebase con GA4

Una vez que ya tenemos creada la propiedad en Google Analytics 4, el siguiente paso es vincularla desde la consola de Firebase. Esta vinculación permite que los datos recolectados por Firebase (eventos, propiedades del usuario, audiencias, etc.) se envíen automáticamente a GA4 para un análisis más avanzado y una visualización integrada con otras fuentes de datos.

Para realizar esta integración, se debe ingresar a la consola de Firebase, hacer clic en el ícono de configuración que aparece

junto al nombre del proyecto y seleccionar la opción «Project settings», como se muestra en la figura 8.

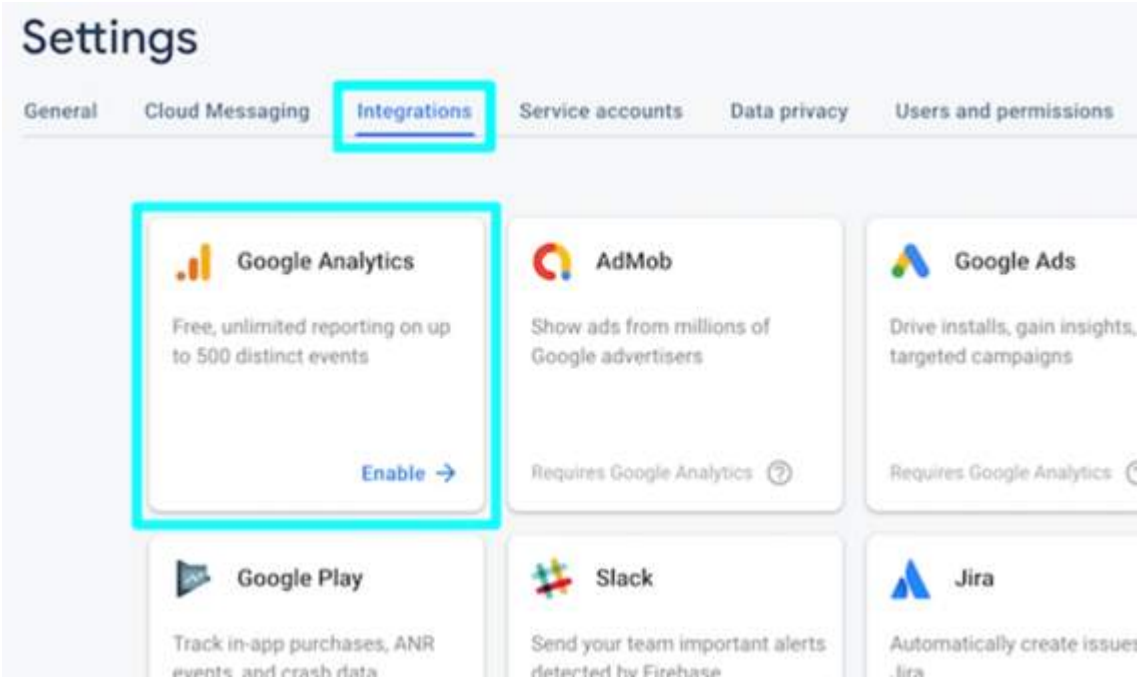
Figura 8. Acceso a la configuración del proyecto en Firebase



Fuente: Chola, 2024, https://goo.su/cegUWu

Dentro de los ajustes del proyecto, se debe seleccionar la pestaña «Integrations». Allí aparecerá una sección específica para Google Analytics, desde donde se puede iniciar el proceso de conexión haciendo clic en «Enable», como se observa en la figura 9.

Figura 9. Integración de Google Analytics en los ajustes del proyecto Firebase



Fuente: Chola, 2024, https://goo.su/cegUWu

En ese momento, Firebase solicitará seleccionar una propiedad de Google Analytics existente para completar la vinculación. Retomando el ejemplo anterior, en este caso deberíamos seleccionar la propiedad «MiEntrenoApp Analytics», que fue creada previamente desde GA4. Una vez completado este paso, los datos comenzarán a fluir entre ambas plataformas de forma automática.

Implementar el SDK de Firebase

Una vez vinculada la propiedad de GA4 con Firebase, es necesario integrar el SDK de Firebase en la aplicación móvil. Recordemos que un SDK (Software Development Kit) es un conjunto de herramientas y librerías que permiten agregar funcionalidades específicas a una aplicación. En este caso, el SDK de Firebase habilita la recolección automática de datos dentro de la *app*.

Esta integración permite que se activen funciones como el registro de eventos, el seguimiento de pantallas, la recopilación de propiedades del usuario, entre otras. El SDK actúa como el canal técnico que envía los datos desde la *app* hacia Firebase y, por extensión, hacia GA4. Si la *app* fue desarrollada para Android, se debe agregar el SDK correspondiente en Android Studio; si es para iOS, se hace desde Xcode.

Ahora bien, además de los eventos automáticos, el equipo de desarrollo también puede **definir eventos personalizados** para medir acciones según los objetivos del negocio. Por ejemplo, en *MiEntrenoApp* podríamos querer registrar cuándo un usuario completa una rutina (`completar_rutina`), accede a un contenido exclusivo (`ver_contenido_premium`) o comparte su progreso (`compartir_logro`). Estos eventos se escriben directamente en el código usando funciones del SDK, y luego aparecen en el panel de Firebase para ser analizados, segmentados o marcados como conversiones. En la unidad 2 nos detendremos especialmente en

este punto para explicar la diferencia entre recolectar datos desde dispositivos Android y desde dispositivos iOS.

Marcar eventos como conversiones

Como dijimos anteriormente, los eventos son acciones que realizan los usuarios dentro de la *app* y se registran en Firebase automáticamente o a través del SDK —tema que veremos en el módulo 2—. Ahora bien, desde el panel de Firebase es posible definir **cuáles de esos eventos se consideran conversiones**, es decir, acciones relevantes para los objetivos del negocio, como una compra, una suscripción o un registro.

Este paso, como mencionamos, se realiza desde la sección «Conversions» dentro del segmento «Analytics» de Firebase. Allí se muestra un listado con todos los eventos registrados en el sistema, y se puede activar la opción «Mark as conversion» para aquellos que se quieran destacar como tales. Por ejemplo, en el caso de *MiEntrenoApp*, si el evento `compra_plan` ya está llegando correctamente al sistema, simplemente debemos marcarlo como conversión desde el panel. A partir de ese momento, Firebase comenzará a mostrar reportes específicos sobre cuántas veces ocurrió, qué usuarios lo realizaron y en qué períodos.

Verificar la integración y monitorear los datos

Una vez que Firebase está vinculado correctamente con la propiedad de GA4, es fundamental asegurarse de que los datos se estén enviando y procesando de manera adecuada. Esta verificación se realiza dentro del panel de Google Analytics 4, en la propiedad vinculada.

Continuando con el ejemplo, ingresamos a **GA4** y accedemos a la propiedad «MiEntrenoApp Analytics». Desde allí, hay dos formas principales de revisar la llegada de datos:

- **Secciones de análisis.** En el menú lateral izquierdo, se puede ingresar a los apartados de «Eventos», «Conversiones» y «Audiencias», donde se visualizarán todos los datos enviados desde Firebase. Por ejemplo, si se configuró el evento `purchase` como conversión, debe aparecer listado allí, junto con la cantidad de veces que se activó y el número de usuarios que lo ejecutaron.
- **Panel de flujo de datos.** Desde el panel de administración (ícono de engranaje), se debe hacer clic en «Flujo de datos», seleccionar el flujo correspondiente a la *app* (por ejemplo, Android o iOS), y luego hacer clic en «Ver detalles del flujo». En esta sección se puede observar e**l** estado de la conexión con Firebase, los eventos que ya están siendo recibidos y los parámetros y propiedades del usuario que llegan asociados a esos eventos.

Este paso permite validar que la integración se haya realizado correctamente y que GA4 esté recibiendo información útil para el análisis del comportamiento de los usuarios en la aplicación.

Con esto finalizamos el recorrido por las principales funcionalidades del segmento «Analytics» de Firebase. En la próxima unidad nos centraremos en cómo implementar técnicamente estas mediciones dentro de una *app*, abordando la integración del SDK, la configuración de eventos y la validación de datos en entornos de desarrollo.

CONTINUAR

Implementación técnica y análisis avanzado en apps

En la unidad anterior exploramos las funcionalidades clave del segmento «Analytics» de Firebase y su integración con Google Analytics 4 (GA4). Vimos cómo se registran y visualizan eventos, conversiones, audiencias, propiedades de usuario y dimensiones personalizadas desde el panel de Firebase, y cómo estos datos permiten comprender el comportamiento de los usuarios dentro de una app. También mencionamos que, para que estos datos se reflejen correctamente en los reportes, es necesario implementar ciertas configuraciones técnicas desde el entorno de desarrollo.

En esta unidad nos enfocaremos en esa implementación técnica. Abordaremos cómo integrar el SDK de Firebase en una aplicación móvil, cómo definir eventos personalizados desde el código, y qué diferencias existen entre la medición en Android y en iOS. Además, veremos cómo validar en tiempo real la recolección de datos usando la funcionalidad «DebugView», lo que permitirá

asegurar que las configuraciones realizadas funcionan correctamente antes del lanzamiento o actualización de una *app*.

SDK de Android e iOS

El SDK (Software Development Kit) de Firebase es el conjunto de herramientas que permite conectar una app con los servicios de Firebase. Como vimos en la unidad anterior, esta integración habilita la recolección automática de eventos, el registro de propiedades del usuario y el envío de datos para su análisis. El SDK actúa como un puente entre la aplicación y la plataforma de Firebase, permitiendo que todo el sistema de medición funcione correctamente sin necesidad de desarrollos complejos.

Sin embargo, esta implementación no es idéntica en todos los casos. Las aplicaciones móviles pueden desarrollarse para distintos sistemas operativos, principalmente Android e iOS, y cada uno tiene sus propios entornos, estructuras y requerimientos. Por ese motivo, en esta unidad abordaremos por separado cómo se integra el SDK en Android y cómo se hace en iOS, destacando las diferencias clave que deben tenerse en cuenta en cada caso.

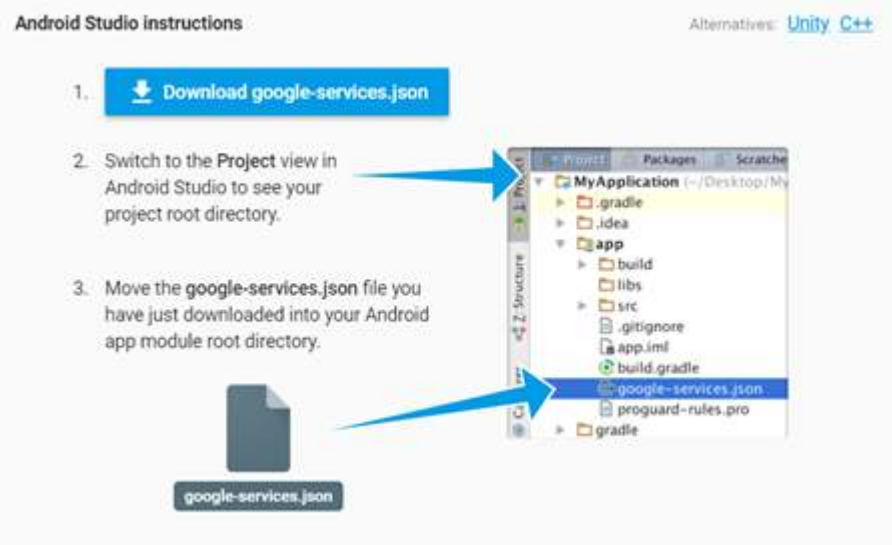
Integración del SDK y configuración de eventos en Android

Para que Firebase pueda comenzar a registrar lo que sucede dentro de una aplicación Android, primero es necesario vincular esa aplicación con la plataforma de Firebase. Este proceso se realiza desde la consola de Firebase, donde se crea un nuevo proyecto y se registra la aplicación que se desea analizar. Durante ese registro, se solicita un dato llamado «nombre del paquete», que es el identificador único de la aplicación dentro del sistema Android. Por ejemplo, si retomamos el caso de *MiEntrenoApp* —la aplicación que presentamos en la unidad anterior—, su nombre de paquete podría ser «com.miempresa.mientreno».

Este nombre se define en el entorno de desarrollo donde se programa la *app*. En el caso de Android, ese entorno suele ser Android Studio, una herramienta gratuita que permite escribir, editar y probar aplicaciones móviles. Aunque no profundizaremos en su uso, es importante saber que el nombre del paquete debe coincidir exactamente con el que fue definido allí, para que la conexión con Firebase funcione correctamente.

Una vez completado ese paso, Firebase mostrará una pantalla con instrucciones para integrar técnicamente la *app*. Como se ve en la figura 10, el sistema guía al usuario para descargar un archivo llamado google-services.json, que contiene los datos de configuración necesarios para vincular Firebase con la app. Este archivo debe colocarse luego en una ubicación específica dentro del entorno de desarrollo Android Studio.

Figura 10. Instrucciones para integrar Firebase en una *app* Android desde la consola



Fuente: Stack Overflow, s.f.a., https://goo.su/Ud98AT

Una vez integrado el SDK, ya es posible comenzar a registrar eventos dentro de la app. Firebase, por defecto, recolecta una serie de eventos de manera automática, como `first_open` (cuando un usuario abre la *app* por primera vez) o `screen_view` (cuando se muestra una pantalla). Sin embargo, muchas veces se necesita registrar **acciones más específicas** que no están contempladas por defecto, como «comenzar una rutina», «agregar un producto al carrito» o «ver un video». A estos se los conoce como **eventos personalizados**.

En Android, esta configuración se realiza desde el entorno de desarrollo Android Studio, utilizando lenguajes como Java o Kotlin. Allí, el equipo de desarrollo puede escribir el código necesario para definir esos eventos personalizados, especificando qué nombre tendrá el evento y qué parámetros adicionales se incluirán (por ejemplo, el tipo de rutina, el nombre del producto o la duración del video). De esta forma, estos eventos se enviarán a Firebase y quedarán disponibles en el panel de informes para su análisis.

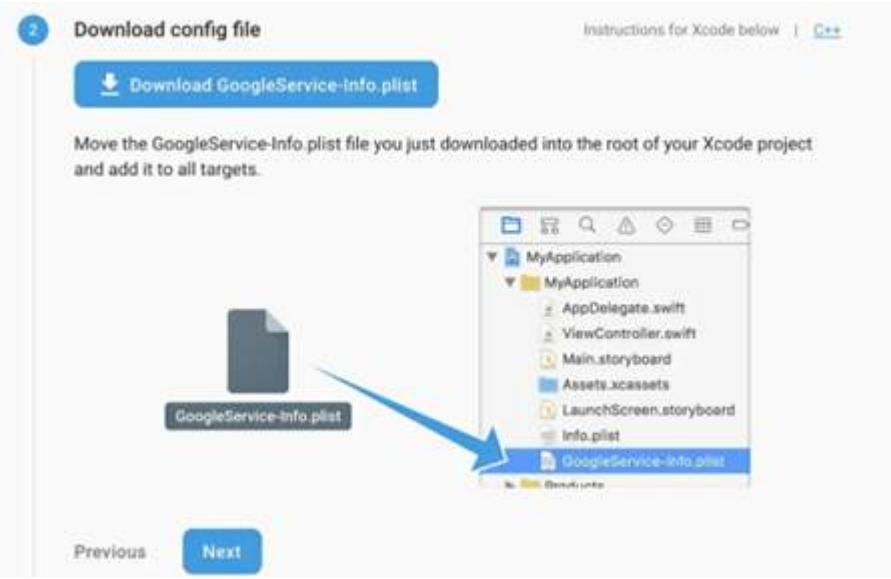
No es necesario saber programar para comprender el proceso: lo importante es entender que cada acción que se quiera medir debe definirse previamente en el código de la aplicación. De esa forma, esos eventos se enviarán a Firebase y quedarán disponibles en el panel de informes para su análisis. Esto permite contar con información clave para tomar decisiones basadas en el comportamiento real de los usuarios.

Integración del SDK y configuración de eventos en iOS

Al igual que en Android, para que Firebase comience a recolectar datos desde una *app* iOS es necesario vincularla primero con Firebase y luego integrar el SDK en el proyecto. El proceso inicial es muy similar al de Android: desde la consola de Firebase se registra la aplicación, se descarga un archivo de configuración y se coloca dentro del proyecto de la *app*.

Cuando se registra una *app* iOS en Firebase, la consola te ofrece un archivo de configuración llamado GoogleService-Info.plist. Este archivo contiene todos los datos necesarios para conectar la *app* con el proyecto de Firebase. En la figura 11 se puede ver este paso: se te muestra el botón para descargar ese archivo y una vista de cómo debería ubicarse dentro de la estructura de carpetas de tu proyecto.

Figura 11. Instrucciones para integrar Firebase en una *app* iOs desde la consola



Fuente: Soluciones Techvoot, 2024, https://goo.su/KZTgUd

Este procedimiento se realiza dentro de Xcode, que es el entorno de desarrollo oficial para crear aplicaciones en dispositivos Apple (iPhone, iPad, etc.). Así como Android Studio se utiliza para apps Android, Xcode permite programar, probar e implementar apps en iOS. Una vez que el archivo de configuración se incorpora al proyecto en Xcode, la app queda conectada con Firebase.

A partir de este punto, también es posible comenzar a registrar eventos dentro de la app. Al igual que en Android, Firebase recolecta algunos eventos de forma automática, pero si se desean medir acciones personalizadas, deben definirse manualmente en el código. La principal diferencia es que, en iOS, esto se hace utilizando el lenguaje Swift o, en algunos casos, Objective-C, dentro del entorno Xcode. Aunque la lógica es la

misma —indicar qué acciones deben medirse y qué información se debe enviar—, la sintaxis y herramientas varían según el sistema operativo.

Medición de pantallas, acciones in-app, monetización y audiencias

El análisis del comportamiento del usuario dentro de una app es fundamental para comprender cómo interactúan con los distintos elementos, qué funcionalidades utilizan con mayor frecuencia y en qué momentos generan ingresos o abandonan la aplicación. A través de Firebase Analytics es posible acceder a múltiples herramientas que permiten observar estas dinámicas en profundidad. En esta sección, analizaremos brevemente cuatro aspectos: la medición de pantallas, el registro de acciones *in-app*, los datos de monetización y la creación de audiencias personalizadas. Cada uno aporta información valiosa para optimizar la experiencia del usuario y tomar decisiones basadas en evidencia.

Medición de pantallas y navegación en la app

La medición de pantallas es una de las funcionalidades más importantes de Firebase Analytics para entender cómo navegan

los usuarios dentro de una *app*. Permite registrar qué secciones se visitan, cuánto tiempo se permanece en cada una y en qué orden se recorren. Esta información es clave para mejorar la experiencia del usuario, detectar puntos de abandono y optimizar el diseño de la interfaz.

Tanto en Android como en iOS, el SDK de Firebase puede registrar automáticamente las visualizaciones de pantalla. Para que esto funcione, las pantallas deben estar correctamente definidas dentro del entorno de desarrollo. En Android, esto ocurre cuando las secciones de la *app* están organizadas como «actividades»; en iOS, el equivalente son los «controladores de vista». Estos son componentes básicos que permiten construir y mostrar las distintas pantallas que ve el usuario. Si la estructura de navegación es más compleja o personalizada, puede ser necesario registrar manualmente los eventos de pantalla (como explicamos anteriormente).

Cada vez que se visualiza una pantalla, se genera un evento llamado `screen_view`. Este evento se acompaña de parámetros como el nombre de la pantalla o el tipo de componente que se mostró. En el panel de Firebase o en GA4, estos eventos permiten analizar qué pantallas se visitan con más frecuencia, cuántos usuarios únicos acceden a cada una y cuánto tiempo permanecen allí. Esta información resulta fundamental para detectar pantallas con baja retención o recorridos poco efectivos.

Volviendo al ejemplo de MiEntrenoApp, imaginemos que un usuario abre la aplicación y recorre tres pantallas: la de bienvenida, la de selección de rutina y la de pago. Firebase registrará automáticamente tres eventos `screen_view` con los nombres «Inicio», «Rutinas» y «Suscripción». Si al analizar los reportes notamos que muchos usuarios llegan a «Rutinas» pero pocos acceden a «Suscripción», podríamos considerar simplificar el proceso de compra o hacer más visible la opción desde pantallas anteriores.

Registro de acciones *in-app* mediante eventos personalizados

Como ya mencionamos, las acciones que los usuarios realizan dentro de una *app*, como hacer clic en un botón, completar una rutina o compartir contenido, no siempre se registran automáticamente. Para poder medir este tipo de interacciones —llamadas acciones *in-app*— es necesario configurar **eventos personalizados**. Esto significa que alguien del equipo de desarrollo debe indicarle explícitamente a Firebase qué acciones queremos medir, para que cada vez que ocurran, se registren como datos en el sistema.

Como vimos, existen eventos automáticos que Firebase registra sin necesidad de configuración, como `first_open` o `app_update`.

Sin embargo, esos eventos no alcanzan para entender todo lo que ocurre dentro de una aplicación concreta. Por eso, Firebase permite que el equipo técnico defina eventos personalizados directamente en el código de la *app*, utilizando una instrucción específica que se incluye en el lugar exacto donde ocurre la acción que queremos medir.

En Android, esta instrucción se escribe en Android Studio usando Java o Kotlin. En iOS, se utiliza Xcode con Swift u Objective-C. En ambos casos, el desarrollador debe usar un comando que se llama `logEvent`, seguido del nombre del evento (por ejemplo, `rutina_completada`) y, si se desea, una serie de parámetros adicionales que permitan describir mejor la acción (por ejemplo, cuánto duró la rutina, en qué nivel estaba el usuario, etc.).

A modo de ejemplo, si se desea medir cuántos usuarios completan una rutina de entrenamiento, el desarrollador debe colocar un `logEvent("rutina_completada")` al final de esa rutina en el código. Eso hará que cada vez que alguien termine una rutina, se envíe esa información a Firebase. Si además queremos saber si fue una rutina básica o avanzada, se puede añadir un parámetro como `nivel: "avanzado"` dentro de ese mismo evento.

Este tipo de configuración convierte a los eventos personalizados en una herramienta muy poderosa. No solo permiten saber qué hace la gente dentro de la app, sino también con qué características, bajo qué condiciones y en qué momento. Todo eso puede luego visualizarse en el panel de Firebase o en GA4, y sirve como base para optimizar el diseño, mejorar la experiencia del usuario o tomar decisiones comerciales basadas en el comportamiento real de las personas.

Monetización y eventos de conversión

Monetizar una aplicación significa implementar estrategias que permitan obtener ingresos a partir del uso que hacen los usuarios. Existen diferentes formas de monetización: se puede ofrecer contenido exclusivo mediante una suscripción, habilitar compras dentro de la app (como productos, funcionalidades adicionales o recursos virtuales) o mostrar anuncios personalizados a los usuarios. La elección del modelo depende del tipo de aplicación, su audiencia y los objetivos del negocio.

Para medir correctamente la monetización dentro de Firebase, es fundamental utilizar **eventos de conversión**. Estos eventos permiten registrar cuándo se concreta una acción que tiene un valor económico para el proyecto, como una compra, el inicio de

una suscripción o la visualización de un anuncio. Al marcar un evento como conversión desde el panel de Firebase —ver figura 4—, se prioriza su seguimiento en los informes y se habilitan métricas, como la tasa de conversión, el valor generado y el recorrido del usuario hasta concretar la acción.

Esta información posibilita tomar decisiones basadas en datos: permite entender qué canales de adquisición son más efectivos, qué tipo de usuarios generan más ingresos o en qué puntos del recorrido se pierde una oportunidad de conversión. Por eso, una estrategia de medición sólida siempre incluye la configuración adecuada de eventos vinculados a la monetización.

Audiencias: segmentación para comprender comportamientos

Tal como explicamos en la unidad 1, la funcionalidad de «Audiencias» en Firebase Analytics permite agrupar usuarios según patrones de comportamiento, propiedades o eventos que hayan ejecutado. Es decir, se establece un conjunto de reglas que definan a los usuarios que queremos observar de forma separada. Por ejemplo, podríamos querer analizar únicamente a quienes usaron la *app* varias veces, pero no realizaron una compra, o a quienes accedieron desde un idioma determinado. A diferencia de los reportes generales, las audiencias permiten **comparar comportamientos entre segmentos** que comparten

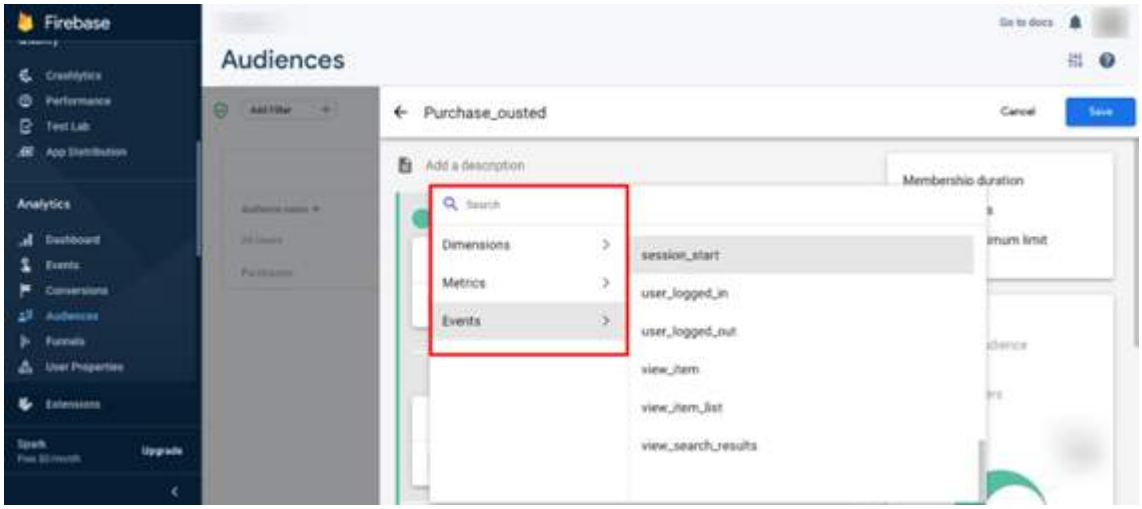
ciertas características o que atravesaron experiencias similares dentro de la *app*.

En la práctica, este proceso se realiza desde el panel de Firebase. Allí se accede a la pestaña de «Audiencias», donde es posible crear nuevas combinaciones de eventos y propiedades (ver figura 3). Firebase ofrece dos caminos: empezar desde cero y definir manualmente cada condición, o utilizar alguna de las audiencias sugeridas, que funcionan como plantillas comunes para proyectos móviles.

Por ejemplo, si quisiéramos entender qué lleva a algunos usuarios a abandonar el proceso de compra, podríamos crear una audiencia llamada `Purchase_outted`. Para construirla, establecemos tres condiciones: que el usuario haya iniciado una sesión (evento `session_start`), que haya añadido al menos un producto al carrito (evento `add_to_cart`), y que no haya ejecutado el evento `ecommerce_purchase`. De esta manera, reunimos en una misma audiencia a las personas que mostraron interés en un producto, pero no completaron la compra.

En la figura 12 se muestra cómo se ve el panel de creación de audiencias en Firebase. Allí se seleccionan las condiciones necesarias para definir cada grupo de usuarios, incluyendo eventos, propiedades del usuario o métricas del sistema.

Figura 12. Creación de una audiencia personalizada en Firebase a partir de eventos



Fuente: Radhakrishnan, s.f., https://goo.su/yClTNc

Una vez creada, esta audiencia puede usarse para visualizar comportamientos en los reportes, comparar con otros segmentos o incluso planificar acciones específicas como notificaciones personalizadas o campañas de remarketing. Esta capacidad de segmentar y analizar de forma diferencial es clave para mejorar la experiencia del usuario, identificar puntos de fricción y optimizar los resultados del negocio.

En última instancia, estas herramientas de medición permiten entender en profundidad cómo las personas interactúan con una app: qué pantallas recorren, qué acciones realizan, cuánto valor generan y cómo se agrupan según sus comportamientos. Ahora

bien, para asegurar que todo se esté registrando correctamente, Firebase cuenta con una funcionalidad llamada «DebugView» que permite validar si se están registrando los eventos en tiempo real. A continuación, nos centraremos en esta herramienta.

DebugView y diagnósticos

Como mencionamos en la unidad 1, una de las funcionalidades del segmento «Analytics» es DebugView. Esta herramienta permite verificar en tiempo real si los eventos que deberían registrarse desde la *app* están llegando correctamente a Firebase. Se utiliza durante las etapas de desarrollo y prueba, ya que muestra cada evento en el momento en que ocurre, con su orden de aparición y los parámetros asociados. Esto permite detectar errores o confirmar que la configuración esté funcionando como se espera.

Para usar DebugView, primero hay que indicarle a Firebase que queremos ver lo que sucede en una app que estamos probando. Para eso, necesitamos activar el **modo de depuración** en el dispositivo o simulador donde estamos ejecutando la *app*.

En el caso de **A**ndroid, se accede a Android Studio, que es el entorno donde se desarrolla y prueba la aplicación. Desde allí, se puede ejecutar la app en un dispositivo físico o en un emulador.

Para que Firebase sepa que estamos haciendo pruebas, es necesario activar el modo de depuración. Esto se hace abriendo una **terminal** dentro de Android Studio y escribiendo el siguiente comando:

```
adb shell setprop debug.firebase.analytics.app <nombre_del_paquete>
```

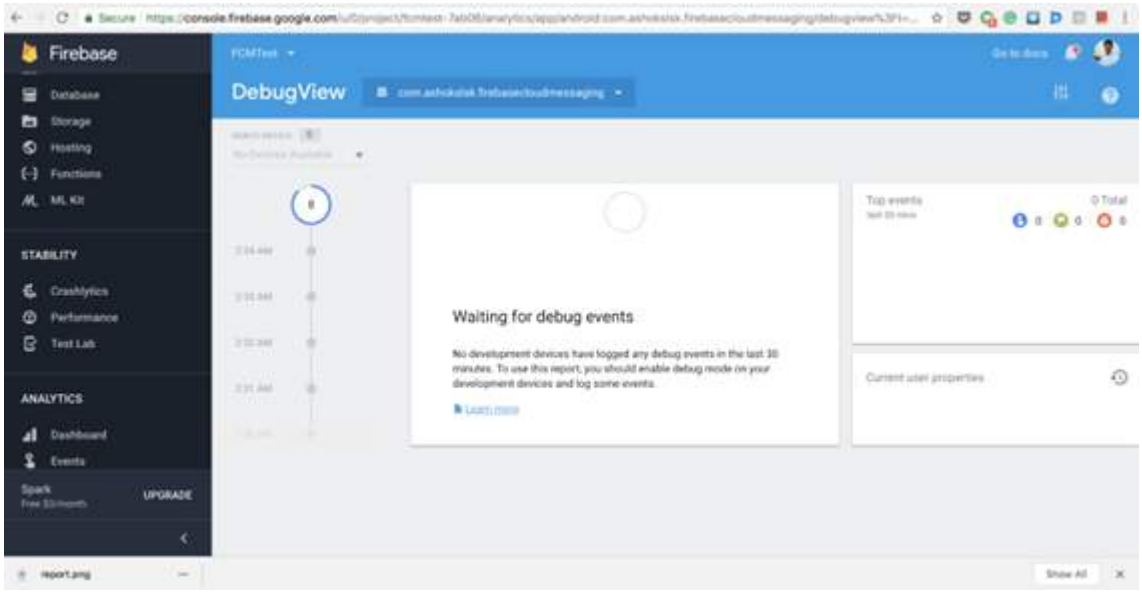
Ese nombre del paquete es el identificador único de la *app*, como vimos antes (por ejemplo, `com.miempresa.mientreno`). Una vez ejecutado el comando, Firebase comienza a registrar los eventos generados en esa sesión como parte de una prueba y los muestra en **DebugView**.

En el caso de iOS, el proceso también se realiza desde el entorno de desarrollo, que en este caso es Xcode. Cuando la *app* se ejecuta desde Xcode en modo de prueba (*modo debug*), Firebase la reconoce automáticamente como una sesión de prueba, sin necesidad de comandos adicionales. A partir de ahí, todos los eventos generados desde esa ejecución se visualizarán en el panel de DebugView.

En la figura 13 se puede ver el aspecto del panel DebugView cuando aún no hay ningún dispositivo conectado ni eventos en curso. Esta vista puede aparecer, por ejemplo, cuando se olvidó

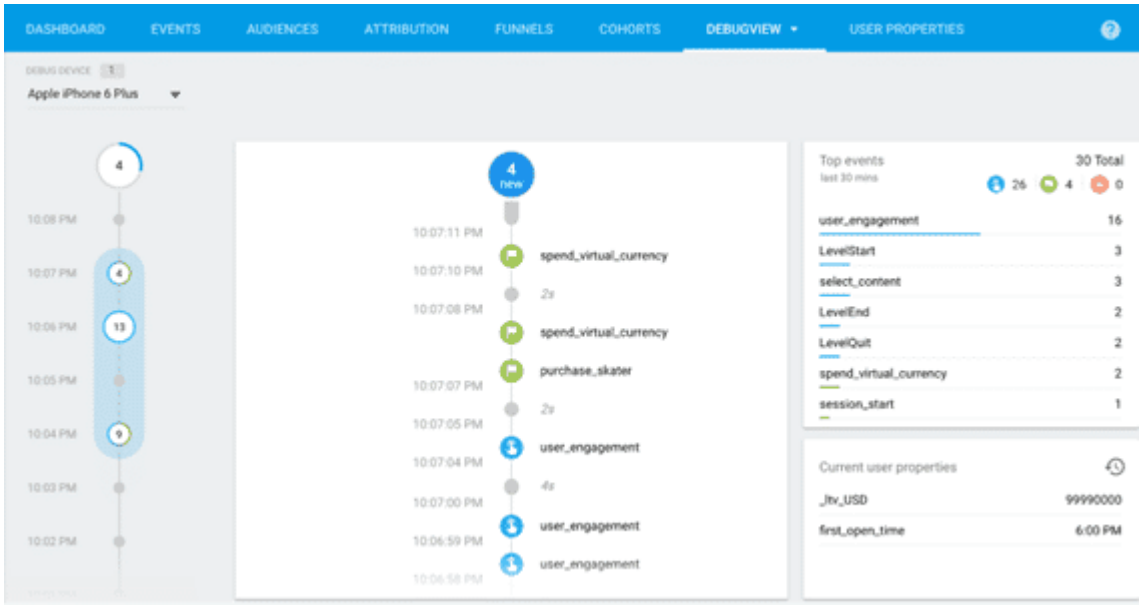
activar el modo de depuración o cuando la app no está generando eventos en ese momento.

Figura 13. Panel DebugView sin eventos registrados



Fuente: Kumar, s.f., https://goo.su/cTcrV7

Una vez que se activa correctamente el modo de depuración y se comienza a usar la app desde el entorno de desarrollo, Firebase empieza a registrar los eventos en tiempo real. En la figura 14 se observa cómo se visualizan los eventos: se muestra una línea de tiempo con cada acción realizada, su orden cronológico y los parámetros asociados. Además, se pueden ver los eventos principales registrados en los últimos 30 minutos y las propiedades del usuario activo.



Fuente: Firebase, s.f., https://goo.su/xKt2s5

Esta información es especialmente valiosa para validar que los eventos personalizados fueron correctamente configurados y están siendo recolectados como se esperaba. Así, antes de lanzar la *app* o una nueva funcionalidad, se puede revisar su correcto

funcionamiento sin necesidad de esperar a que los datos aparezcan en los reportes generales de Firebase.

Con esto finaliza el recorrido técnico sobre la implementación de Firebase y su integración con Google Analytics 4. En el siguiente módulo, nos centraremos en cómo aprovechar los datos recolectados para generar **reportes y análisis avanzados**. Veremos cómo explorar los datos con herramientas como **Exploraciones**, cómo utilizar **BigQuery** para hacer análisis personalizados, y cómo interpretar visualizaciones clave como ***funnels***, **cohortes** o **tasa de retención**. También aprenderemos a identificar posibles **cuellos de conversión** dentro de la *app* y a usar esa información para tomar decisiones basadas en datos reales.

CONTINUAR

Referencias

Actualizatec, (s.f.). *Qué es y cómo utilizar Firebase Analytics para Apps.* https://actualizatec.com/blog/tutorial-firebase-analytics/

Firebase, (s.f.). *Depurar eventos.* https://firebase.google.com/docs/analytics/debugview?hl=es-419

Kumar, A. (s.f.). *Dominando Firebase para el desarrollo de Android*. https://www.oreilly.com/library/view/mastering-firebase-for/9781788624718/d81c1b1d-9fbf-4d12-b8b7-0f419d1e74cc.xhtml

Lee, J. (s.f.). *Filtrado del tráfico de Google/Firebase Analytics por tipo de compilación/entorno en Android.* https://blog.jakelee.co.uk/filtering-google-firebase-analytics-traffic-by-buildtype-environment-on-android/

Radhakrishnan, K. (s.f.). *Crear y segmentar audiencias en Google Firebase.* https://blog.appmaker.xyz/create-and-segment-

audiences-on-google-firebase/

Radhakrishnan, K. (s.f.a.). *Seguimiento de conversiones con Firebase.* https://blog.appmaker.xyz/conversion-tracking-with-firebase/

StatCounter. (2025). *Mobile vs. desktop vs. tablet market share worldwide*. https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide

Stack Overflow, (s.f.). *¿Cómo ver el análisis de eventos de Firebase con parámetros personalizados en un informe CSV?* https://stackoverflow.com/questions/61656770/how-to-see-the-firebase-event-analytics-with-custom-parameters-in-csv-report

Stack Overflow, (s.f.a). *¿Cómo agregar google-services.json en Android?* https://stackoverflow.com/questions/32072568/how-to-add-google-services-json-in-android

We Are Social. *(2025). Digital 2025 Global Overview Report.* https://datareportal.com/reports/digital-2025-global-overview-report

CONTINUAR

Descarga en PDF

PDF

Módulo 2. Analítica de apps con Firebase.pdf
972.5 KB