

Módulo 3. Reportes y análisis avanzados en apps

☰ [Análisis del comportamiento de usuarios en aplicaciones](#)

☰ [Conversión y retención en aplicaciones](#)

☰ [Referencias](#)

☰ [Descarga en PDF](#)

Análisis del comportamiento de usuarios en aplicaciones

A esta altura del recorrido ya contamos con una comprensión clara sobre cómo implementar y validar la medición tanto en sitios mobile como en aplicaciones. Abordamos el uso de GA4 en entornos mobile web, exploramos métricas relevantes, eventos personalizados, conversiones, así como también aspectos relacionados con la identificación del usuario en múltiples dispositivos. Luego, profundizamos en la analítica de aplicaciones con Firebase, prestando atención a la integración con GA4, la implementación de SDK en Android e iOS, la configuración de eventos *in-app* y la medición de monetización y audiencias. Esta primera etapa nos permitió configurar correctamente la medición y entender cómo se recogen los datos.

En esta unidad, damos un paso más: nos enfocamos en el análisis del comportamiento de los usuarios a partir de los datos recolectados. ¿Qué hacen las personas dentro de una aplicación? ¿Qué pantallas visitan con mayor frecuencia? ¿Cuáles son los

recorridos más comunes? ¿Dónde abandonan? Estas preguntas, entre muchas otras, permiten pasar del registro a la interpretación, es decir, utilizar la información para tomar decisiones que mejoren la experiencia y los resultados.

Trabajaremos dos herramientas para lograr ese análisis. Por un lado, los reportes integrados y personalizados disponibles en las plataformas de analítica, que permiten visualizar de manera clara patrones de comportamiento. Por otro lado, abordaremos el uso de BigQuery como solución avanzada para análisis más complejos y flexibles, especialmente útil cuando se requiere trabajar con grandes volúmenes de datos, segmentaciones específicas o integraciones con otras fuentes. En este sentido, la unidad propone desarrollar habilidades para identificar patrones de uso, analizar interacciones y generar reportes significativos que contribuyan a mejorar el diseño, la navegación y el rendimiento de las aplicaciones móviles.

Reportes y análisis en aplicaciones

Como se trabajó en los módulos anteriores, tanto Firebase como Google Analytics 4 (GA4) permiten registrar información detallada sobre el comportamiento de los usuarios dentro de una aplicación móvil. A través de eventos, parámetros y propiedades, estas herramientas recolectan datos sobre las acciones que las

personas realizan mientras navegan por una página o una *app*: qué pantallas visitan, qué botones tocan, en qué momento completan una acción o cuándo abandonan.

Para organizar y visualizar esta información de forma accesible, ambas plataformas generan distintos tipos de reportes. Algunos de estos reportes son generados automáticamente, sin necesidad de configuración, y ofrecen una primera lectura sobre el uso general de la aplicación. Por ejemplo, el panel de inicio de ***Firestore*** presenta un resumen con los indicadores más consultados sobre actividad reciente. Este panel, visible apenas se ingresa al entorno de Firestore, actúa como un tablero general del estado de la aplicación. Aunque no permite personalización directa, muestra una selección de métricas clave y eventos destacados que permiten tener un panorama rápido del funcionamiento general.

Figura 1. Vista general del panel de inicio en Firestore Analytics



Fuente: Actualizatec, s.f., <https://goo.su/VbUPW4U>

En el gráfico superior izquierdo se visualiza la cantidad de usuarios activos por día. A su lado, la métrica de retención del día 1 muestra qué porcentaje de personas regresaron luego de haber usado la *app* por primera vez. En paralelo, se representa el ingreso estimado generado por eventos relacionados con monetización, como compras o suscripciones.

Más abajo, se resumen datos vinculados con el nivel de interacción: el tiempo promedio que las personas permanecen activas, la cantidad de sesiones con interacción por usuario y la duración media por sesión. Estos indicadores permiten evaluar el nivel de compromiso de quienes usan la aplicación.

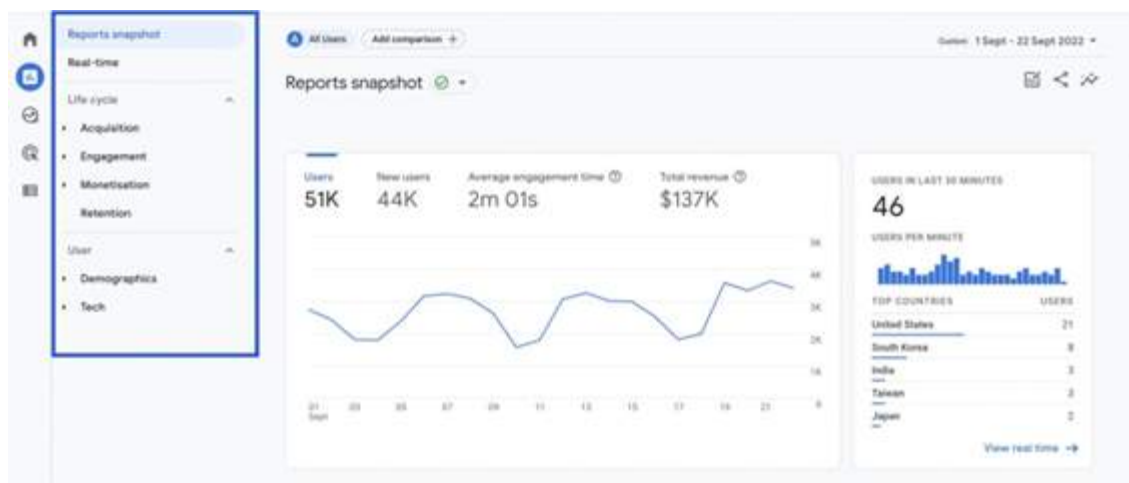
A la derecha del panel, se presentan los eventos registrados con mayor frecuencia (como inicio de sesión o visualización de

pantalla) y aquellos que fueron marcados como conversiones. Estas acciones son especialmente relevantes para los objetivos del negocio y suelen analizarse con mayor profundidad.

Este panel inicial funciona como una puerta de entrada al análisis. Permite identificar si hay cambios relevantes en los principales indicadores y, a partir de allí, decidir qué aspectos conviene examinar en reportes más específicos.

Algo similar ocurre en GA4, donde también se presenta una **vista general de informes** al ingresar a la propiedad.

Figura 2. Vista general de informes en GA4



Fuente: Icrono, s.f., <https://goo.su/fMPHl>

Esta pantalla resume indicadores clave como usuarios, usuarios nuevos, tiempo de interacción medio e ingresos. Aunque ofrece menos detalle que otros reportes específicos, cumple una función similar a la de Firebase: brindar una primera lectura rápida sobre el comportamiento general en la app o sitio.

Ahora bien, en ciertos casos necesitamos ir más allá de estos paneles generales para responder preguntas más puntuales sobre el comportamiento de los usuarios. Por ejemplo, ¿qué tan diferente es la navegación entre quienes se registraron y quienes no?, ¿en qué pantalla se interrumpe con más frecuencia un flujo de compra?, ¿cuántas personas usan una funcionalidad específica después de una campaña de instalación?

Frente a este tipo de preguntas, los paneles automáticos ya no alcanzan. Es necesario contar con reportes que puedan construirse a medida, combinando eventos, filtros y comparaciones según los objetivos de análisis. Estos reportes permiten adaptar la lectura de los datos a necesidades concretas del equipo de producto, *marketing* o negocio.

A modo de ejemplo, supongamos que queremos construir un reporte que nos permita analizar las pantallas de destino dentro de una aplicación. En este caso, entendemos como pantalla de destino a aquella en la que comienza la interacción del usuario luego de abrir la app, ya sea por acceso directo, desde una campaña o una notificación. Analizar este dato permite entender qué tan efectivas son esas entradas iniciales y si conducen o no a interacciones posteriores valiosas, como una compra, una suscripción o la navegación hacia otras secciones.

Como ya vimos, Firebase y GA4 pueden integrarse para recolectar y analizar eventos dentro de una app. En este ejemplo, vamos a utilizar la sección de exploraciones de GA4 para crear un informe personalizado que nos muestre cuántos usuarios comenzaron su sesión en determinadas pantallas, cuántas veces se produjeron esas entradas y si esas visitas derivaron en conversiones.

El procedimiento para configurar este informe paso a paso es el siguiente:

1

Ingresar a la sección «Explorar» y seleccionar «Formato libre»

Dentro del menú izquierdo de GA4, al ingresar a la sección «Explorar», se despliega una serie de opciones para crear análisis personalizados.

Figura 1. Selección de «Formato libre» dentro de la sección «Explorar» en GA4



Fuente: Jordán, s.f., <https://goo.su/DwLLnP>

En este caso, seleccionamos el tipo de exploración llamado «Formato libre», ya que permite combinar eventos, métricas y dimensiones en una misma vista de forma ágil.

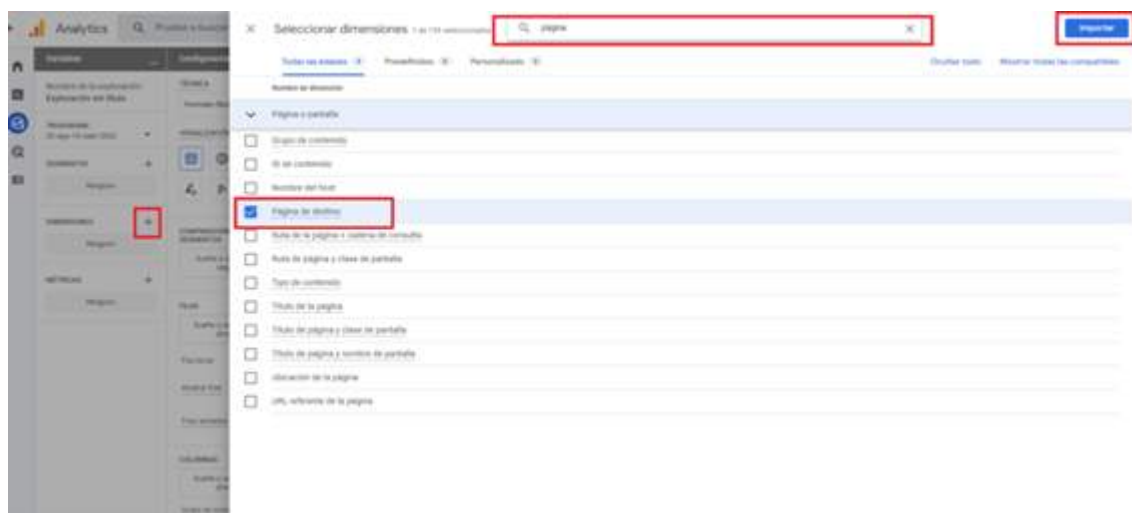
2

Agrega dimensiones

Hacemos clic en el botón «+» dentro del recuadro de «Dimensiones». Usamos el buscador para localizar la dimensión «Pantalla de destino» y la seleccionamos. Esta dimensión

representa el primer punto de contacto del usuario dentro de la app, es decir, la pantalla que ve al iniciar una sesión. Una vez seleccionada la dimensión, hacemos clic en «Importar» para incorporarla al informe.

Figura 2. Agregar la dimensión «Pantalla de destino» en GA4



Fuente: Jordán, s.f., <https://goo.su/DwLLnP>

3

Agregar métricas

Al igual que en el paso anterior, hacemos clic en el botón «+», pero esta vez dentro del recuadro de «Métricas». Desde el buscador, seleccionamos las siguientes métricas relevantes para el análisis:

- «Entradas»
- «Total de usuarios»
- «Sesiones»
- «Compras» (o cualquier otro evento que represente una conversión en la *app*)

Estas métricas nos van a permitir observar no solo cuántas veces los usuarios ingresaron desde cada pantalla de destino, sino también si esa navegación derivó en interacciones valiosas para el negocio. Una vez seleccionadas, hacemos clic en «Importar».

3

Organizar dimensiones y métricas en el informe

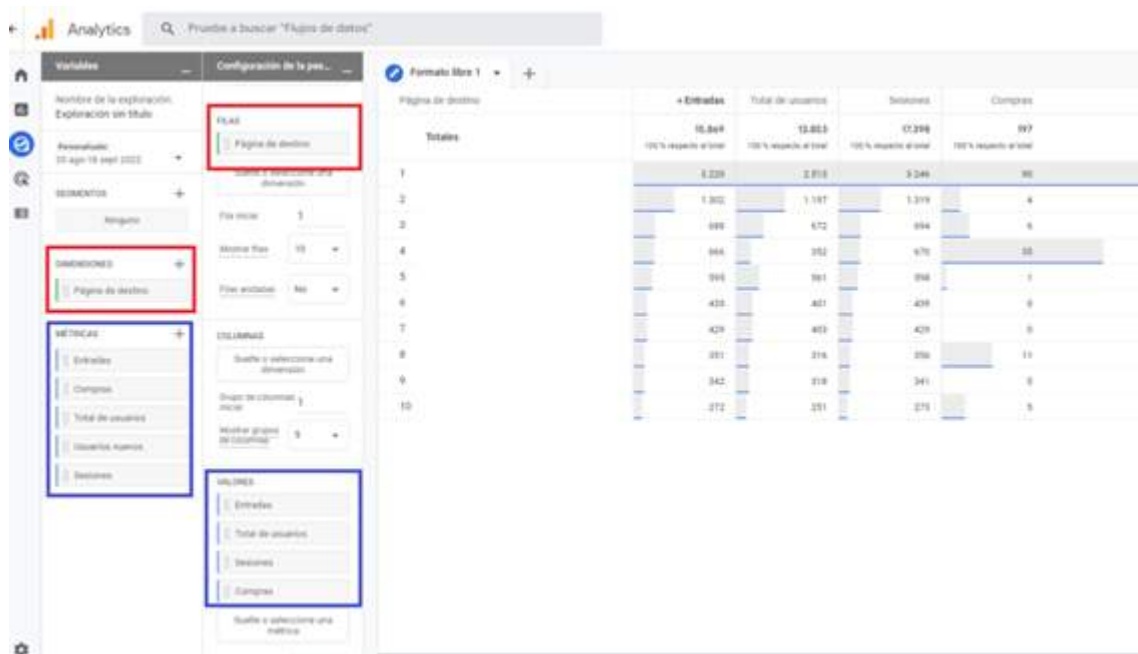
Una vez importadas, las dimensiones y métricas aparecen disponibles en el panel izquierdo. Desde ahí, simplemente arrastramos cada elemento a su lugar correspondiente.

En este ejemplo:

- En «Filas», colocamos la dimensión «Página de destino».

- En «Valores», sumamos las métricas «Entradas», «Total de usuarios», «Sesiones» y «Compras».

Figura 4. Informe personalizado en GA4 con métricas aplicadas por pantalla de destino



Fuente: Jordán, s.f., <https://goo.su/DwLLnP>

De este modo, el informe personalizado permite observar qué pantallas de la app funcionaron como página de destino, cuántas veces fueron visualizadas y qué resultados se generaron a partir de esas visitas. A partir de esta base, es posible incorporar otras métricas relevantes para el análisis, según los objetivos del proyecto. Por ejemplo, si se quisiera conocer cuántas personas

que llegaron a una pantalla específica completaron un evento determinado, puede agregarse la métrica correspondiente y aplicar un filtro para centrarse en ese comportamiento. Esto permite adaptar el reporte a distintas preguntas analíticas, manteniendo una estructura clara y flexible.

Ahora bien, cuando los reportes disponibles en el panel de GA4 o Firebase no alcanzan para analizar relaciones más complejas entre eventos, segmentos o fuentes de datos, es posible dar un paso más en el análisis. En estos casos, BigQuery se presenta como una herramienta que permite trabajar directamente con los datos crudos recolectados por Firebase o GA4, abriendo nuevas posibilidades de exploración y modelado.

BigQuery para análisis de datos de aplicaciones

BigQuery es un servicio de análisis de datos en la nube desarrollado por Google, diseñado para almacenar, manipular y consultar grandes volúmenes de datos de forma eficiente y escalable. Esta herramienta, que forma parte de la plataforma Google Cloud, ofrece un entorno gestionado que permite ejecutar consultas sin necesidad de administrar servidores. Su modelo de pago por uso permite a las organizaciones acceder a

potentes capacidades analíticas, pagando únicamente por los recursos que utilizan.

Pero, ¿por qué sería necesario usar una herramienta como BigQuery si ya contamos con reportes en GA4 o Firebase? ¿Qué permite hacer que otras herramientas no contemplan? ¿Cuándo conviene recurrir a este tipo de análisis? Estas preguntas aparecen cuando los equipos necesitan integrar datos de la *app* con otras fuentes o realizar consultas que van más allá de los paneles estándar.

Por ejemplo, en una aplicación universitaria se podría analizar cuántas personas abrieron la *app*, se registraron y nunca completaron una inscripción en materias. Si además se desea cruzar esa información con un sistema interno de gestión académica (CRM), donde están los registros reales de inscripción, los reportes de GA4 o Firebase —aunque sean personalizados— no alcanzan para responder esa inquietud. En estos casos, BigQuery permite acceder a los registros detallados de eventos recolectados, combinarlos con datos externos y construir análisis a medida que respondan con precisión a las necesidades del equipo.

Para permitir este tipo de análisis complejos, BigQuery se apoya en una arquitectura pensada para el trabajo con grandes volúmenes de datos. Su funcionamiento combina tres elementos

principales. En primer lugar, el **almacenamiento distribuido**, que guarda los datos de manera replicada en distintos servidores, asegurando disponibilidad, tolerancia a fallos y acceso eficiente a la información. En segundo lugar, la **capa de redistribución de datos en memoria**, que actúa como una capa de procesamiento intermedio y se encarga de reorganizar y mover los datos internamente según lo que cada consulta necesita, permitiendo ejecutar operaciones como uniones, agrupamientos y agregaciones de forma paralela. Finalmente, una **capacidad de cómputo altamente escalable**, que se ajusta automáticamente en función de la carga de trabajo y la complejidad del análisis. Todos estos componentes se encuentran conectados por una **red de alta velocidad**, infraestructura interna de Google que permite transferir grandes volúmenes de datos entre almacenamiento y cómputo sin generar cuellos de botella, posibilitando la ejecución de consultas complejas en segundos, incluso cuando requieren unir millones de registros provenientes de múltiples fuentes.

Figura 5. Arquitectura general de BigQuery: almacenamiento distribuido y procesamiento escalable



Fuente: Google Cloud, s.f., <https://goo.su/qYrUC1>

Entonces, si pensamos en el ejemplo de la universidad, los datos sobre el uso de la app (pantallas vistas, registros, sesiones) pueden estar almacenados en Firebase y los datos académicos (como la inscripción a materias o la sede de cursada) en un sistema CRM interno. Cada uno de estos conjuntos se guarda por separado y en estructuras distintas. Gracias al modelo distribuido de BigQuery, esta información puede cargarse en tablas separadas, pero consultarse en conjunto: por ejemplo, es posible unir los eventos de quienes se registraron en la app con los datos de inscripción académica para detectar cuántas personas no

completaron el proceso. Mientras que una tabla almacena los eventos de navegación (como la apertura de pantalla o el clic en un botón), otra contiene los datos académicos, como la carrera elegida o la fecha de inscripción. La arquitectura distribuida permite que estas fuentes se procesen en paralelo, filtrando, agrupando o calculando métricas sin necesidad de mover los datos manualmente. En pocos segundos, BigQuery puede responder preguntas específicas, como cuántas personas de una determinada carrera se registraron en la app, pero no se anotaron en ninguna materia en un período dado.

También se puede utilizar BigQuery para analizar la efectividad de campañas de comunicación. Por ejemplo, si se envía una notificación a estudiantes que no se inscribieron aún, es posible medir cuántos de ellos vuelven a abrir la app, cuántos avanzan hasta el formulario de inscripción y en qué punto abandonan. Este análisis requiere combinar datos de campañas enviadas (por ejemplo, desde un sistema de marketing) con eventos registrados en la app, lo que es posible al centralizar todo en BigQuery.

Sin embargo, el valor de trabajar con BigQuery no está solo en acceder a datos detallados, sino en poder transformarlos en conocimiento útil. Esto permite anticipar comportamientos,

identificar patrones de uso según características del usuario o priorizar mejoras en la app basadas en evidencia. Si, por ejemplo, se detecta que una gran cantidad de usuarios abandona el proceso de inscripción después de visitar cierta pantalla, es posible analizar si ese comportamiento varía según la carrera elegida, el dispositivo utilizado o el horario de uso. Con esa información, el equipo puede decidir si es necesario ajustar la interfaz, enviar recordatorios personalizados o revisar el contenido disponible en esa etapa.

Para finalizar, es importante señalar que BigQuery no está pensado para cualquier tipo de usuario. Generalmente, su uso queda en manos de perfiles técnicos, como analistas, data engineers o científicos de datos, que tienen la capacidad de escribir consultas en SQL (un lenguaje diseñado específicamente para consultar y manipular bases de datos), integrar fuentes externas y generar estructuras de datos reutilizables. Estos perfiles trabajan junto a equipos de producto, marketing o desarrollo, ayudando a traducir los datos en decisiones informadas. De este modo, BigQuery se ubica como una pieza clave dentro de un flujo de trabajo colaborativo, donde el análisis profundo complementa la mirada más operativa de los reportes automatizados.

CONTINUAR

Conversión y retención en aplicaciones

Hasta aquí se analizaron distintas formas de medir el comportamiento de los usuarios en aplicaciones móviles, tanto a través de reportes predeterminados como mediante la creación de informes personalizados que permiten adaptar el análisis a preguntas específicas. En esta unidad, el recorrido se orienta hacia dos aspectos centrales del ciclo de vida del usuario: la conversión y la retención.

Ambos conceptos están directamente relacionados con el rendimiento de una aplicación, ya que permiten entender qué tan efectivo es el recorrido propuesto (desde la descarga hasta la realización de una acción deseada), y qué tan sostenido es el vínculo con quienes ya interactuaron. Para eso, se presentarán herramientas como los *funnels*, las cohortes y los análisis de retención, que ayudan a interpretar patrones de uso, identificar etapas sensibles del uso y tomar decisiones para mejorar la experiencia y los resultados.

Estas herramientas —disponibles tanto en GA4 como en Firebase— ofrecen múltiples opciones para analizar los recorridos de los usuarios y tomar decisiones basadas en evidencia. Sin embargo, también presentan ciertos límites cuando se busca mayor profundidad o flexibilidad en los análisis. En este punto, BigQuery se convierte en una alternativa clave: su capacidad para acceder al detalle de los eventos, cruzarlos con otras fuentes de datos y construir métricas personalizadas permite extender el análisis de *funnels*, cohortes y retención más allá de lo que permiten los paneles estándar. A lo largo de esta unidad, se mostrarán ejemplos concretos de estos enfoques, destacando en qué casos conviene utilizar las herramientas predeterminadas y cuándo es útil recurrir a soluciones más avanzadas.

Funnels, cohortes y retención

Imaginemos el recorrido de una persona que descarga una app, crea una cuenta, explora algunas funcionalidades, abandona sin realizar ninguna acción relevante... Y vuelve unos días después, esta vez para completar una compra o iniciar un curso. ¿Qué factores influyen en ese comportamiento? ¿Qué tan común es ese patrón entre quienes se descargan la aplicación? ¿En qué momento del recorrido suelen detenerse otros usuarios?

Para responder este tipo de preguntas, existen tres enfoques que permiten observar el uso de una aplicación desde distintos ángulos: los funnels, las cohortes y los análisis de retención. Estas herramientas permiten interpretar con mayor detalle cómo interactúan las personas con una app, en qué etapas abandonan, cómo varía su comportamiento según el momento de ingreso, y cuántos vuelven después de una primera visita. A lo largo de esta sección, se analizarán por separado estas formas de lectura, con ejemplos que permitan comprender su utilidad en distintos contextos.

Análisis de funnels: cómo interpretar el paso a paso del usuario

El concepto de *funnel* (o embudo) proviene del *marketing* y se refiere a las distintas etapas que atraviesa una persona desde el primer contacto con un producto hasta la conversión final. En el caso de una app, este embudo tiene algunas particularidades: no se trata solo de atraer a un usuario, sino de lograr que descargue la aplicación, la utilice y se mantenga activo en el tiempo.

Figura 6. *Funnel* de conversión para apps



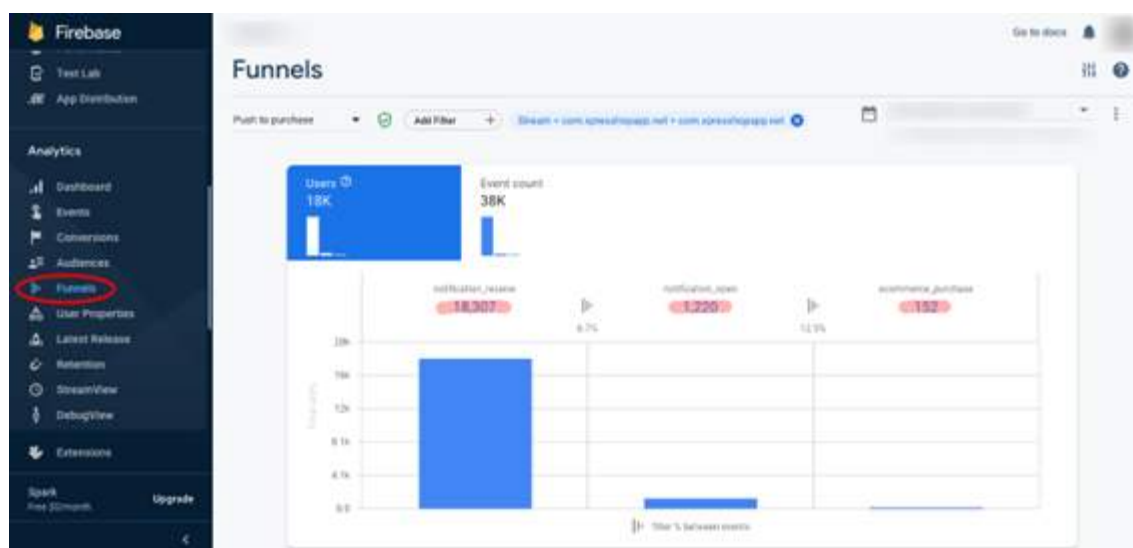
Fuente: EMMA, s.f., <https://goo.su/CwIHj5>

Como se observa en la figura, el recorrido del usuario puede dividirse en distintas etapas, desde la exposición inicial hasta la retención. Cada paso incluye una acción concreta: desde conocer la *app* por una campaña, encontrarla en una tienda, revisar su ficha, instalarla y, finalmente, seguir usándola. Esta representación permite entender cómo se va filtrando el público en cada instancia y qué aspectos influyen en la continuidad o el abandono. Analizar estos pasos ayuda a identificar dónde se pierde más interacción y qué se puede ajustar para mejorar los resultados.

Entendido este recorrido, el siguiente paso es ver cómo puede medirse dentro de las herramientas que ya conocemos. En este caso, trabajaremos con Firebase y GA4, que permiten construir embudos para analizar el paso a paso de los usuarios dentro de una *app*. A partir de eventos recolectados automáticamente o definidos por el equipo, es posible identificar cuántas personas completan cada etapa y dónde se producen los principales abandonos.

El análisis de *funnels* es una de las funcionalidades disponibles dentro del segmento «Analytics» de Firebase. Allí, en la sección «Funnels» del panel lateral (ver figura 7), se puede visualizar el avance de los usuarios entre distintos eventos, como recibir una notificación, abrirla y concretar una compra.

Figura 7. Visualización de *funnels* en Firebase



Si observamos la imagen, podemos notar que se muestra el recorrido de los usuarios a través de tres eventos distintos: la recepción de una notificación (`notification_receive`), su apertura (`notification_open`) y, finalmente, la realización de una compra (`ecommerce_purchase`). Cada barra representa la cantidad de usuarios que realizó ese evento en particular, sin necesidad de haber completado los pasos anteriores.

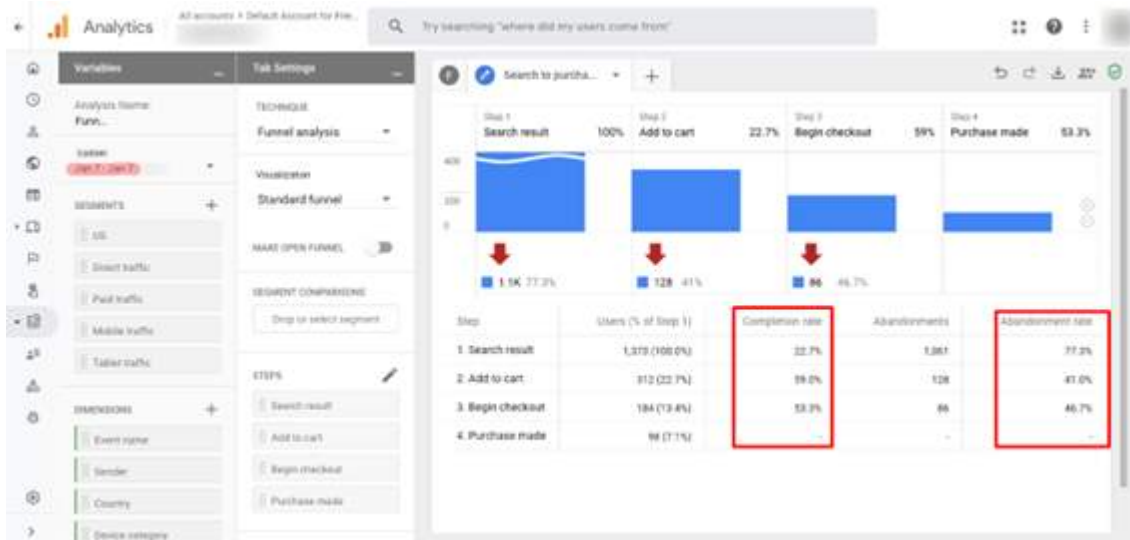
Para crear este embudo, es necesario seleccionar manualmente los eventos que se desean analizar. Por ejemplo, en este caso se eligieron eventos que permiten observar el impacto de una campaña de notificaciones *push* sobre las compras dentro de la *app*. Esta configuración resulta útil para identificar qué porcentaje de usuarios pasa de recibir una notificación a abrirla, y cuántos de ellos efectivamente concretan una acción valiosa, como comprar.

Ahora bien, este tipo de embudo se conoce como **embudo abierto**, ya que los eventos que lo componen no están conectados entre sí: un usuario puede haber realizado solo el tercer evento (por ejemplo, una compra) sin haber pasado necesariamente por los anteriores (como recibir o abrir una notificación).

En caso de que se quiera analizar un recorrido en el que cada paso depende del anterior —por ejemplo, solo considerar usuarios que buscaron un producto, luego lo agregaron al carrito, iniciaron el proceso de compra y finalmente compraron— se necesita construir un **embudo cerrado**. Esta opción no está disponible directamente en Firebase, pero sí puede implementarse en GA4, donde es posible definir pasos interdependientes y ver tasas de conversión más precisas.

En la siguiente figura podemos ver un ejemplo de embudo cerrado construido en GA4. El embudo muestra el recorrido de los usuarios desde que realizan una búsqueda hasta que completan una compra. Cada paso está representado por un evento específico: **Search result** (el usuario realiza una búsqueda dentro de la app y se le muestra un listado de resultados), **Add to cart** (el usuario selecciona un producto y lo agrega al carrito), **Begin checkout** (el usuario inicia el proceso de pago, completando datos como dirección o método de envío) y **Purchase made** (el usuario finaliza la transacción y realiza la compra).

Figura 8. Embudo cerrado de conversión en GA4: desde la búsqueda hasta la compra



Fuente: Radhakrishnan, s.f., <https://goo.su/VsfCzA>

A la derecha, se incluyen métricas clave como la tasa de finalización (Completion rate) y la tasa de abandono (Abandonment rate), que permiten evaluar en qué etapa se pierden más usuarios. Por ejemplo, se observa que el 77,3% abandona después de buscar, y solo el 7,1 % completa la compra.

Para construir este embudo, hay que ingresar a la sección de exploraciones en GA4, elegir la técnica «Funnel analysis», definir los eventos que representan cada paso del proceso y seleccionar la opción de embudo cerrado (desactivando la opción «Make open funnel»). Esto permite analizar la conversión de manera secuencial, ya que los usuarios deben completar cada paso para avanzar al siguiente.

Ahora bien, aunque GA4 permite medir con bastante precisión el comportamiento dentro de la app —incluyendo tasas de conversión, tiempos entre pasos y características de los usuarios — hay situaciones en las que ese análisis no alcanza. Por ejemplo, si se quiere saber cuántas personas abandonaron el proceso de compra en la app *después* de haberse contactado previamente por WhatsApp, o si quienes no compraron habían recibido antes un mail con un cupón, esa información no está dentro de GA4.

En estos casos, BigQuery permite unir los datos recolectados por la app con otras fuentes externas (como un CRM, una plataforma de mensajería o una herramienta de email marketing) y construir reportes integrados. De este modo, es posible responder preguntas más complejas que no se limitan al uso de la app en sí, sino que contemplan el recorrido completo del usuario a través de distintos canales y que, en definitiva, afectan directamente el funnel de conversión dentro de la app. Esto permite tomar decisiones basadas en una visión más completa del comportamiento, no solo en lo que ocurre dentro de la aplicación.

Análisis de cohortes: cómo detectar patrones de comportamiento a lo largo del tiempo

El análisis de cohortes es una técnica que permite observar cómo se comportan distintos grupos de personas a lo largo del tiempo. Cada cohorte se define a partir de una característica común — por ejemplo, el día en que comenzaron a usar un servicio, realizaron una compra o se registraron en una plataforma— y luego se compara su comportamiento en distintos momentos. Esta forma de análisis ayuda a identificar patrones, evaluar la evolución de ciertos grupos y detectar diferencias significativas entre ellos.

Llevado al ámbito de una aplicación, el análisis de cohortes permite observar cómo evoluciona el comportamiento de grupos de usuarios que comparten un punto de partida común dentro de la *app*. Por ejemplo, se puede analizar si las personas que instalaron la *app* durante una campaña específica vuelven a usarla después de una semana, o si quienes se registraron en un mismo día mantienen un nivel similar de actividad. Este enfoque ayuda a entender la retención —concepto que desarrollaremos más adelante—, comparar resultados entre cohortes y tomar decisiones basadas en la evolución real del uso.

Para analizar cohortes dentro de una *app*, Firebase ofrece un informe específico, pero está orientado principalmente a medir cuántos ingresos generan los usuarios con el tiempo. Para poder

usar este informe, es necesario vincular la app con otras plataformas, como **AdMob** (que gestiona la publicidad dentro de las apps) o **Google Play** (que registra las compras y suscripciones). Esto se debe a que el informe se basa en datos de monetización, es decir, cuánto dinero deja cada grupo de usuarios según la fecha en que instalaron la *app*.

Sin embargo, si el objetivo es analizar el comportamiento en general —como la frecuencia de uso, el cumplimiento de ciertos eventos o la tasa de retención a lo largo del tiempo—, se puede recurrir a Google Analytics 4, que también permite trabajar con cohortes basadas en eventos y condiciones personalizadas. Y para preguntas aún más específicas o que involucren múltiples fuentes de datos, BigQuery resulta especialmente útil, ya que permite crear cohortes a medida según cualquier criterio relevante para el análisis.

Dentro de Google Analytics 4, esta exploración se configura eligiendo dos elementos: el **criterio de inclusión**, que determina quiénes integran cada cohorte (por ejemplo, todos los usuarios que instalaron la *app* o realizaron una primera compra en un período determinado); y el **criterio de retorno**, que indica qué comportamiento se desea medir (como volver a abrir la *app*, realizar otra compra o completar una conversión específica). GA4 permite aplicar distintos niveles de granularidad —diaria, semanal o mensual— para observar cómo se comportan los

usuarios en intervalos de tiempo definidos desde el momento en que ingresaron a la cohorte.

Una exploración típica de cohortes en una *app* podría centrarse, por ejemplo, en los usuarios que descargaron la aplicación durante una determinada semana y que luego realizaron compras en las semanas siguientes. En la imagen podemos ver este tipo de análisis: cada fila representa una cohorte distinta, agrupada según la semana de adquisición de los usuarios (columna izquierda), mientras que las columnas muestran las semanas posteriores (de la 0 a la 9).

Figura 9. Ejemplo de análisis de cohortes en GA4: transacciones semanales según fecha de adquisición

Cada celda es la suma de la métrica "Transacciones" de los usuarios que cumplieron el criterio de devoción "purchase" ese período (semana), después del criterio de inclusión "Primer contacto (fecha de adquisición)".

	SEMANA 0	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9
Todos los usuarios Transacciones	274	40	13	2	5	8	4	0	0	1
1 ago-5 ago 2023 2,671 usuarios	19	6	0	0	2	1	2	0	0	1
6 ago-12 ago 2023 4,567 usuarios	26	3	3	1	1	2	0	0	0	0
13 ago-19 ago 2023 3,915 usuarios	21	4	4	0	2	2	2	0	0	0
20 ago-26 ago 2023 2,383 usuarios	15	3	1	1	0	2	0	0	0	0
27 ago-2 sept 2023 3,683 usuarios	46	5	0	0	2	1	0	0	0	0
3 sept-9 sept 2023 0,744 usuarios	30	4	1	0	0	0	0	0	0	0
10 sept-16 sept 2023 2,202 usuarios	30	6	4	0	0	0	0	0	0	0
17 sept-23 sept 2023 4,159 usuarios	37	0	0	0	0	0	0	0	0	0
24 sept-30 sept 2023 6,179 usuarios	42	1	0	0	0	0	0	0	0	0
1 oct-2 oct 2023 1,365 usuarios	9	0	0	0	0	0	0	0	0	0

Fuente: Why Ads, s.f., <https://goo.su/zr1NE>

Cada celda indica la cantidad de usuarios de esa cohorte que realizó una transacción (evento purchase) en la semana correspondiente. Por ejemplo, la cohorte de usuarios que instaló la *app* entre el 27 de agosto y el 2 de septiembre generó 46 compras en la misma semana en que se unieron (semana 0), 5 compras en la semana siguiente (semana 1), y 2 compras en la semana 4.

Este tipo de visualización permite identificar patrones valiosos: por ejemplo, se puede detectar cuál es el momento más frecuente para concretar una primera compra, si los usuarios vuelven a comprar después de cierto tiempo o si hay cohortes con mejor rendimiento que otras. Además, permite analizar la retención desde una perspectiva de impacto, observando no solo si los usuarios regresan, sino también si continúan generando valor con el tiempo.

Ahora bien, si el análisis busca mayor profundidad —por ejemplo, cruzar la información de cohortes con datos externos como la atención por WhatsApp, campañas de *email* o historial de compras—, BigQuery se convierte en una herramienta clave. A través de consultas personalizadas, es posible construir cohortes basadas en criterios más flexibles que los disponibles en GA4 y combinar múltiples fuentes de datos. Esto permite responder preguntas como las siguientes: ¿los usuarios que interactúan por

canales de soporte tienen una mayor tasa de retención?, ¿existe alguna diferencia en el comportamiento según el país o el modelo de dispositivo?, ¿qué cohortes generan más ingresos totales considerando toda su actividad, dentro y fuera de la *app*?

Con este enfoque, el análisis de cohortes deja de ser solo un seguimiento de eventos en la *app*, y se transforma en una herramienta estratégica para entender el recorrido completo del usuario.

Análisis de retención: cómo medir la permanencia de los usuarios en el tiempo

Uno de los desafíos más importantes para cualquier aplicación no es solo conseguir descargas, sino lograr que las personas vuelvan a usarla después del primer contacto. Ahí es donde entra en juego el análisis de retención: una métrica clave para entender cuántos usuarios regresan a la *app* luego de haberla instalado.

Firestore ofrece una función específica para esto dentro de su panel de Analytics, llamada «Retention», como se ve en la imagen. Esta herramienta permite visualizar de manera clara qué porcentaje de usuarios vuelve a interactuar con la *app* después de la primera semana, la segunda, y así sucesivamente. Cada fila representa una cohorte (grupo de usuarios) que comenzó a usar la *app* en una semana determinada.

Figura 10. Análisis de retención en Firebase



Fuente: Radhakrishnan, s.f.a., <https://goo.su/3GqnxxU>

Por ejemplo, si observamos la cohorte del 8 al 14 de julio, todos esos usuarios comienzan con un 100% en la semana 0, ya que esa fue su semana de incorporación. Luego, Firebase calcula qué porcentaje de ese grupo volvió a abrir la *app* en las semanas posteriores. En este caso, un 16,4% lo hizo en la semana 1 y un 9,8% en la semana 2.

Este tipo de visualización permite detectar tendencias: si muchas cohortes pierden usuarios rápidamente, quizás sea necesario mejorar la experiencia inicial o agregar recordatorios para fomentar el regreso. En cambio, si algunas cohortes muestran

buena retención, puede ser útil analizar qué campañas o funcionalidades estuvieron activas en ese período.

El informe de retención se basa principalmente en el evento automático `user_engagement`, que se activa cada vez que un usuario interactúa activamente con la app durante un período mínimo de tiempo. Este evento es parte del conjunto básico que Firebase registra de forma automática, por lo que los datos de retención estarán disponibles sin necesidad de configuración adicional, siempre y cuando la app tenga correctamente integrada la biblioteca de Firebase Analytics (conocida como SDK). Esto significa que basta con que el SDK esté instalado para empezar a ver los porcentajes de retención de cada cohorte.

Identificación de cuellos de conversión

Hasta aquí se analizó cómo medir el funnel de conversión dentro de una aplicación, cómo interpretar el comportamiento de los usuarios a partir del análisis de cohortes y cómo evaluar la retención a lo largo del tiempo. Estos enfoques permitieron comprender de qué manera las personas avanzan por los

distintos recorridos propuestos, cómo evoluciona su uso después del primer contacto y qué tan sostenido resulta el vínculo con la aplicación.

Ahora bien, disponer de estas mediciones invita a profundizar el análisis del recorrido de los usuarios dentro de la aplicación. ¿En qué puntos se concentra la mayor pérdida de personas? ¿Qué etapas del proceso presentan mayores dificultades? ¿En qué momentos se interrumpe con más frecuencia la progresión hacia una acción de valor?

Para abordar estas preguntas es necesario dar un paso adicional y observar los recorridos de forma integrada, analizando con mayor detalle las transiciones entre eventos, pantallas y etapas del funnel. Para ello, se trabaja con la identificación de los cuellos de conversión, es decir, aquellos puntos del recorrido en los que se concentra la fricción o el abandono y que impactan directamente en el desempeño de la aplicación.

Tanto Firebase como Google Analytics 4 permiten identificar los cuellos de conversión dentro de una aplicación a partir del análisis de eventos, recorridos y tasas de abandono en los distintos flujos de uso. Estas herramientas ofrecen información suficiente para detectar en qué etapas del funnel se concentra la pérdida de usuarios y cómo se comportan las personas frente a acciones relevantes para la aplicación.

A continuación, y asumiendo que la integración entre Google Analytics 4 y Firebase ya se encuentra configurada —tal como se desarrolló en el módulo 2—, se abordará cómo realizar este análisis directamente desde Firebase. En particular, se trabajará con los informes disponibles en la plataforma para analizar recorridos de usuarios y visualizar los puntos del flujo en los que se produce una mayor fricción o interrupción del proceso.

Para comprender cómo se identifican los cuellos de conversión desde **Firestore**, resulta útil partir de un caso concreto y luego trasladar esa lógica a otros recorridos posibles. Pensemos, por ejemplo, en una aplicación educativa cuyo objetivo es que las personas, luego de instalar la app, completen el proceso de registro y confirmen la inscripción a una materia. El recorrido esperado puede incluir acciones como abrir la aplicación, iniciar el registro, completar un formulario con datos personales, seleccionar una materia y confirmar la inscripción.

En este escenario, el análisis del cuello de conversión requiere definir qué eventos representan cada una de estas etapas del recorrido. Algunos de estos eventos se registran de forma automática, mientras que otros deben configurarse específicamente durante el desarrollo de la app, tal como se explicó en el módulo anterior. Para este flujo, podrían considerarse los siguientes eventos:

Tabla 1. Eventos que componen el recorrido de conversión analizado en Firebase

Evento	Descripción
<code>first_open</code>	Se registra la primera vez que una persona abre la aplicación luego de instalarla. Permite identificar el punto inicial del contacto con la app.
<code>screen_view</code>	Se activa cada vez que el usuario visualiza una pantalla. Resulta útil para reconstruir la navegación general y asociar los eventos del flujo a pantallas concretas.
<code>sign_up_start</code> (o evento equivalente definido por el equipo)	Representa el inicio del proceso de registro dentro de la aplicación.
<code>form_submit</code> (o evento personalizado similar)	Indica el envío del formulario con los datos personales requeridos para el registro.

<code>select_content</code> (o evento específico de selección)	Se utiliza para registrar la elección de una materia dentro de la app.
<code>sign_up</code> (o evento de confirmación)	Marca la finalización del proceso de inscripción y representa la acción esperada al final del recorrido.

Fuente: elaboración propia

Una vez definidos los eventos que componen el recorrido, el análisis de cuellos de conversión se centra en observar cómo se distribuye el avance de los usuarios entre cada una de estas etapas. A partir de los informes de *funnels* disponibles en Firebase —y analizamos previamente—, es posible visualizar cuántas personas ejecutan cada evento y cómo varía ese número a medida que el flujo progresa.

Siguiendo con el ejemplo de la aplicación educativa, supongamos que el análisis muestra un alto volumen de usuarios registrados en el evento `first_open`, lo que indica que la app logra ser abierta luego de la instalación. A continuación, una proporción significativa de estas personas visualiza distintas pantallas, reflejado en el evento `screen_view`, lo que sugiere una

exploración inicial del entorno. Sin embargo, al observar el evento `sign_up_start`, se detecta una caída marcada en la cantidad de usuarios que efectivamente inician el proceso de registro. Este primer descenso permite identificar un posible cuello de conversión asociado al momento en que se propone al usuario comenzar el registro.

En este caso, el cuello no se encuentra en la instalación ni en la navegación inicial, sino en la transición hacia una acción más comprometida. El análisis permite inferir que, aunque las personas ingresan y recorren la *app*, una parte relevante no avanza hacia el registro, lo que puede estar vinculado a la forma en que se presenta esa opción, al momento en que aparece o a las expectativas generadas en las pantallas previas.

Supongamos ahora que el evento `sign_up_start` presenta un volumen razonable, pero que el evento `form_submit` muestra una nueva caída pronunciada. Aquí, el cuello de conversión se desplaza hacia una etapa posterior del recorrido. El análisis indica que los usuarios comienzan el registro, pero no completan el envío del formulario. Este comportamiento suele estar asociado a fricciones propias de esa instancia, como la cantidad de campos solicitados, la complejidad de los datos requeridos o la experiencia de uso en esa pantalla específica.

En un tercer escenario, puede observarse que tanto el inicio del registro como el envío del formulario mantienen una continuidad aceptable, pero que el evento `select_content` presenta una disminución significativa. En este caso, el cuello de conversión se ubica en la etapa de selección de la materia, lo que permite focalizar el análisis en cómo se presenta la oferta académica, si las opciones resultan claras o si el usuario encuentra dificultades para avanzar en esa decisión.

Finalmente, si el mayor descenso se produce entre `select_content` y `sign_up`, el cuello de conversión se concentra en la confirmación final de la inscripción. Este tipo de caída suele indicar fricción en el último paso del proceso, donde los usuarios ya invirtieron tiempo en completar el recorrido, pero no terminan de concretar la acción esperada.

En todos estos casos, el valor del análisis reside en poder ubicar con precisión el punto del recorrido donde se interrumpe el avance. Firebase permite observar estas diferencias entre eventos de manera clara y compararlas dentro de un mismo flujo, facilitando la identificación de los cuellos de conversión. A partir de esta información, el análisis puede profundizarse incorporando otras variables, como el dispositivo utilizado, el momento del día o el origen del usuario, para comprender mejor el contexto en el que se produce la interrupción. De este modo, los eventos definidos previamente se transforman en una

herramienta para interpretar el comportamiento real de los usuarios y detectar, con base en datos, qué etapas del recorrido requieren mayor atención dentro del diseño y la lógica de la aplicación.

Para finalizar esta unidad, cabe decir que el trabajo con eventos, *funnels*, cohortes, retención y análisis en profundidad permite comprender cómo interactúan las personas con una *app*, dónde se producen interrupciones y cómo utilizar esa información para mejorar la experiencia y el rendimiento. Este enfoque analítico sienta las bases para avanzar, en el próximo módulo, hacia la evaluación del impacto de estas interacciones desde una perspectiva de negocio, incorporando el análisis del ROI, los modelos de atribución y los desafíos de la medición en entornos mobile atravesados por restricciones de privacidad.

[CONTINUAR](#)

Referencias

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

EMMA, (s.f.). *Funnel para apps. Descubre e interpreta la caída de tus usuarios.* <https://emma.io/blog/funnel-para-apps-descubre-e-interpreta-la-caida-de-tus-usuarios/>

Google Cloud, (s.f.). *Descripción general de BigQuery.* <https://docs.cloud.google.com/bigquery/docs/introduction?hl=es-419>

Icrono, (2024). *Cómo funcionan los informes en GA4.* <https://icrono.com/tendenciasmarketing/como-funcionan-los-informes-en-ga4/>

Jordan, S. (s.f.). *Cómo crear informes personalizados de Google Analytics 4 (paso a paso).*

<https://vivaconversion.com/blog/analitica-web/como-crear-informes-personalizados-de-google-analytics-4-paso-a-paso/>

Radhakrishnan, K. (s.f.). *Embudos y análisis de Firebase*. <https://blog.appmaker.xyz/firebase-funnels-and-analysis/>

Radhakrishnan, K. (s.f.a.). *Aumentar la retención de usuarios de Firebase*. <https://blog.appmaker.xyz/increasing-firebase-retention-of-users/>

Why Ads, (s.f.). *Qué es el modelo de cohortes en GA4*. <https://whyadsmedia.com/diccionario/modelo-de-cohortes/>

CONTINUAR

Lección 4 de 4

Descarga en PDF
