



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Monitorización y verificación de aplicaciones Android

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Javier Fernández Estévez

Tutores: Juan Carlos Ruiz García/David de Andrés Martínez

2015/2016

Resumen

En este trabajo se describen y analizan las posibilidades que ofrece la herramienta de monitorización *Google Analytics* a la hora de poder comprobar cómo está funcionando una aplicación, es decir, cómo los usuarios finales hacen uso de ella. Además, se describe el proceso de implementación y despliegue de las sondas software que deben incluirse en el producto para poder ser, una vez distribuido, monitorizado a través de esta herramienta de monitorización. También se detalla el proceso a seguir para publicar una aplicación móvil en el mercado de apps de Android y, para acabar se muestra qué tipo de resultados proporciona la herramienta *Google Analytics* y cómo pueden interpretarse los mismos con vistas a planificar el esfuerzo de desarrollo y mejorar la funcionalidad de la aplicación móvil en cuestión. Todo este trabajo se ejemplifica a través de una app móvil real, llamada *QuotationShake*, que ha sido instrumentada convenientemente y publicada en el mercado para aplicaciones móviles oficial de Android, el mercado conocido como Google Play.

Palabras clave: Android, google analytics, mercado de aplicaciones móviles, aplicación móvil.

The present work reports on how to use the features provided by the *Google Analytics* tool to monitor and profile the use of mobile applications (apps). In addition, it is also described the underlying implementation process required to deploy the monitoring software probes required by the analytics tool. The publication process of the resulting instrumented mobile app is then detailed and it is shown which type of information can be provided by *Google Analytics* and how to exploit such information in order to influence the development process and improve the app functionality. The feasibility and interest of this work is exemplified through a real mobile application, the *QuotationShake* app, that has been published, and it is currently available, at the official Android app market, the so-called *Google Play*.

Keywords: Android, google analytics, mobile market, app.



Tabla de contenidos

1. Introducción/Motivación.....	6
2. Más allá del desarrollo de apps	8
2.1. Publicación de una aplicación móvil para Android.....	9
2.2. Seguimiento de una aplicación móvil para Android	11
3. Caso de estudio QuotationShake	19
3.1. Seguir QuotationShake con Google Analytics	21
4. Publicar QuotationShake	33
5. Resultados	37
6. Conclusiones	40
Referencias	42

1. Introducción/Motivación

Internet se ha convertido en algo que todos usamos a diario. Cada día millones de personas usan internet y entran en páginas *web* tanto para buscar información como contenido multimedia. El crecimiento del uso de estas páginas, y su uso por millones de personas, ha convertido a la analítica *web* en una disciplina muy importante para todas aquellas empresas que quieren monitorizar estos sitios. La analítica *web* mide detalles concretos tales como, cuántas personas han visitado tu *web*, cómo llegaron esos visitantes, qué palabras clave buscaron para encontrar la página, cuánto tiempo permanecieron en una página determinada o en toda la *web* y en qué enlaces han hecho clic. Con esta información, puedes determinar qué áreas de tu página son más populares y cuáles no consiguen tanto tráfico, para así saber cómo agilizarla creando una mejor experiencia de usuario.

Ahora bien, en los últimos años el uso de *Smartphone* ha crecido de manera increíble en los últimos años, hasta el punto que ahora tener uno se ha convertido en algo casi indispensable.

El auge de los dispositivos móviles ha traído consigo el lanzamiento de un gran número de aplicaciones para móviles, comúnmente denominadas *apps*. Las empresas también han sabido aprovechar esta revolución, ya que se han convertido en una nueva fuente de entrada de clientes, ya sea ofreciendo servicios a través de ellas, creando simples *apps* que refuerzan la marca de una compañía o simplemente, introduciendo publicidad a través de aplicaciones de éxito. Pero no solo las empresas han aprovechado este crecimiento, muchos usuarios han visto en estas *apps* una forma de llevar a cabo sus ideas consiguiendo con ellas un gran éxito que les beneficie de forma económica.

Por esto, es muy importante que las compañías o particulares conozcan cómo los usuarios usan su aplicación. Por ello las herramientas de *Web Analytics* se han visto obligadas a crear nuevas versiones para integrarlas en las aplicaciones.

Para ver cómo funcionan estas herramientas y más en concreto *Google Analytics* está hecho este proyecto. Saber cómo funciona esta herramienta pasa por saber cómo empezar a usarla, que pasos hay que seguir para que tu código de la aplicación la implemente, que puedes medir dentro de tu *app*, como *Google Analytics* muestra toda esta información y por último, como se deben interpretar estos resultados para que resulten útiles. Y no solo esto, si no también conocer los procesos siguientes al desarrollo de una aplicación como es su publicación.

Vamos a dividir este trabajo de la siguiente manera, primero veremos lo que viene después del desarrollo de una aplicación, esto es el proceso de publicación y análisis de cómo está funcionando la *app*. Es importante saber que el ciclo a la hora de llevar a cabo una aplicación no acaba una vez se desarrolla, si no que sigue una vez se ha creado, de eso hablaremos en este primer punto.

Después de esto, nos centraremos en conocer la *app* sobre la que integraremos *Google Analytics*. Veremos cómo funciona, cuál es su estructura, qué aspectos de su uso y funcionamiento nos interesa supervisar y en que partes hemos decidido insertar el código necesario para hacerlo. El paso siguiente será el describir como ha sido el proceso de publicación de dicha aplicación. Veremos todos los pasos que se han de seguir, y que hemos seguido, para que la *app* esté disponible en el *market Google Play*.

Y para acabar, analizaremos los datos que hemos recabado de nuestra aplicación una vez ésta descargada, instalada y en ejecución en distintos dispositivos móviles. Analizaremos qué información recibimos y cómo interpretarla y en qué medida los resultados obtenidos afectan o pueden afectar a la *app*.

2. Más allá del desarrollo de apps

A menudo ocurre que se suelen centrar todos los esfuerzos en el desarrollo, en la creación de la aplicación olvidando un poco el resto. Creamos una *app*, funciona bien, pero, ¿Ahora qué? ¿Cuál es el siguiente paso? ¿Nos quedamos ahí? Evidentemente no, y en este apartado vamos a ver que se debe hacer una vez creada una aplicación.

No se debe caer en el error de centrar todos los esfuerzos en el desarrollo de una aplicación sin tener en cuenta lo que viene después. Pero la pregunta es, ¿Que viene después? Pues bien, lo que viene a continuación se puede ver en la imagen que tenemos justo debajo.



Como vemos, los siguientes pasos son la publicación y el seguimiento de la *app*. Pero, ¿Por qué un ciclo? La respuesta es sencilla. Una vez publicada la aplicación, y podemos ver cómo se comporta una vez se use, debemos ver que partes son las que funcionan mejor o peor, y que partes se deben mejorar, potenciar o incluso eliminar. Esto lleva a volver al desarrollo para tener que modificar lo que veamos oportuno para hacer nuestra *app* mucho mejor, lo que hace necesario volver a publicarla o actualizarla para ver de nuevo como se comporta volviendo a repetir el ciclo. Este ciclo nunca acaba, siempre se pueden hacer mejoras e incorporar nuevas ideas para que tu *app* crezca.

Por esto aquí vamos a hablar del proceso de publicación, veremos las alternativas que tenemos y cuál es la que hemos elegido nosotros y porque. Haremos un repaso de los pasos a seguir para que esté lista para los usuarios finales. Además de esto, veremos que herramientas tenemos a la hora de seguir una aplicación, veremos algunas de ellas y diremos el porqué de haber elegido *Google Analytics*.

2.1. Publicación de una aplicación móvil para Android

Una vez tenemos desarrollada nuestra *app*, llega el momento de plantearnos donde publicarla. Existen diferentes alternativas tanto para publicar en Android como en iOS. En estas plataformas existen dos principales *markets* donde publicar tu aplicación, *Google Play Store* para Android, y *AppStore* para iOS. Estas dos plataformas son las líderes en sus respectivos sistemas ya que existen desde el principio de los Smartphone, pero con el paso del tiempo han surgido alternativas tanto para Android como para iOS y aquí vamos a ver alguna de esas alternativas, además claro de analizar tanto la *AppStore* como la *Google Play Store*. Nos centraremos quizás más en las alternativas de Android dado que donde que nuestra *app* va dirigida a este sistema.

Para empezar tenemos que saber que es un *market*, pues bien, se trata de una plataforma destinada a distribuir aplicaciones para móviles compatibles con un sistemas operativos concreto, como puede ser Android o iOS [6]. Debido al uso de estos sistemas operativos los principales *markets* y los más reconocidos están destinados a ellos. Ahora vamos a hacer un repaso de los *markets* más importantes para estos sistemas operativos y diremos en cuál de ellos subiremos nuestra *app* de prueba. Los *markets* que vamos a repasar aquí son la App Store para iOS, y Google Play Store y Amazon *AppStore* para Android.

AppStore

Es una plataforma fundada por Apple, uno de los principales fabricantes de tecnología móvil, y al igual que la *Play Store* ocupa un papel crucial dentro de los principales *markets* de *apps*, junto a Google, reinan este mercado.

La *AppStore* se caracterizan por su rigidez y por ser muy meticulosos con las aplicaciones que admiten en su *market*, ya que además de tener políticas de contenidos, también tienen muy en cuenta el diseño de la aplicación, el cual, debe ser lo más parecido posible al estilo seguido por esta compañía. Una de las grandes ventajas de publicar aplicaciones en Apple, es la de la fidelidad del usuario, ya que los usuarios confían en Apple y descargan las aplicaciones de pago que recomienda, por lo tanto, es rentable monetizar las aplicaciones. Un punto negativo de esta *AppStore* es el coste de la creación de una cuenta como desarrollador, la cual, tiene un coste de 90\$/anuales aproximadamente, por lo que hay que tener claro que se va a rentabilizar ya que puede ser un gasto importante [6].

Google Play Store

Es una plataforma fundada por Google y podemos decir que ya desde casi el principio ocupa un papel muy relevante dentro del mercado de los principales *markets* de *apps* móviles. Una de las características principales de la *Google Play Store*, es el relativamente poco control a la hora de publicar aplicaciones como desarrollador, a diferencia de la *AppStore*, donde se revisan las aplicaciones antes de publicarlas. Esto no quiere decir que se pueda publicar cualquier tipo de aplicación, ya que si se infringe la política de contenido o cualquier otra política del sistema y



eres denunciado por algún usuario de tu aplicación, o eres revisado aleatoriamente por uno de los trabajadores de Google, pueden advertirte e incluso suspenderte una *app* o una cuenta, por lo que siempre es recomendable hacer aplicaciones que aporten un valor al usuario y cumplan con las políticas de la tienda. Otra de las grandes ventajas de la Google Play, es el coste de creación de cuenta como *developer*, ya que por 25\$ tienes una cuenta vitalicia donde publicar aplicaciones durante el resto de tu vida [6].

Amazon AppStore

Es una plataforma fundada por Amazon. Han aprovechado el potencial de su mercado ya establecido y lo han ampliado con un *market* propio de aplicaciones. Una de las principales virtudes de la *Amazon AppStore*, es la posibilidad de publicar aplicaciones de Android, gracias a ello, han atraído a muchos desarrolladores de la *Play Store*, los cuales, pueden desarrollar aplicaciones que más adelante pueden publicar en ambos mercados gracias a la posibilidad de desarrollar con un formato único. Esto ha hecho que por ello, se considere entre los principales *markets* de *apps*. Se trata de un *market* no tan poblado, y donde no hay tanta competencia [6].

Después de ver los diferentes mercados, vemos que los que contienen más *apps* son *Google Play Store* y *AppStore* donde cada una tiene alrededor de millón y medio de aplicaciones disponibles, mientras que *Amazon AppStore* no llega al medio millón siendo este el tercero que más *apps* tiene disponibles [7]. Decidimos subir nuestra *app* a la *Google Play Store* debido a que es sin duda el principal *market* que existe en este momento y en el que puedes llegar a más personas si subes ahí tu aplicación. Además, puedes subir tu aplicación de manera sencilla y más o menos rápida. A todo esto cabe añadir que la principal razón de usar la *Google Play Store* es el hecho de que desde *Google Analytics* puedes enlazar fácilmente tu cuenta de *Google Play* y recibir información adicional de tu aplicación que puede resultar muy útil.

2.2. Seguimiento de una aplicación móvil para Android

En este apartado vamos a realizar una pequeña descripción de lo que es *Google Analytics* y además vamos a ver algunas alternativas a esta herramienta. Pero antes de esto, primero vamos a saber que es una herramienta *mobile analytics* para *apps* además de repasar las razones por la que se usan y son útiles.

Estas herramientas de *analytics* permiten conocer el comportamiento de los usuarios que usan una aplicación en sus dispositivos móviles. El objetivo es averiguar qué aspectos de una aplicación funcionan mejor o peor para trabajar más esos aspectos que no funcionan del todo bien e intentar mejorarlos o intentar potenciar esos otros aspectos que si están funcionando bien. En base a esta información, podremos mejorar nuestra *app* e intentar llegar así a más usuarios.

No es algo nuevo que el uso de dispositivos móviles sigue creciendo. Por tanto, empresas, gente particular o empresas más especializadas en el mundo de la tecnología aprovechan este crecimiento lanzando sus propias aplicaciones para móviles. Todos ellos tendrán objetivos distintos a la hora de lanzar su *app*, a las empresas, por ejemplo les puede interesar porque lo ven como una nueva fuente de entrada de clientes, ya sea ofreciendo servicios a través de ellas para que refuercen la marca de su compañía o simplemente, introduciendo publicidad a través de aplicaciones de éxito. Mientras que para gente particular es una forma de desarrollar una buena idea y ganar así dinero gracias a ella. Si sabemos cómo se utiliza una *app*, podremos saber cómo mejorarla. Podremos conocer cuánto utilizan la *app*, cuándo y de qué manera. Con todo este conocimiento podemos mejorar la experiencia de los usuarios. Otra información importante que proporcionan estas herramientas es saber sobre que dispositivo se usa nuestra aplicación para enfocar nuestras campañas de marketing o para priorizar el desarrollo de nuevas versiones.

Cuanta más información consigamos recoger sobre los usuarios, mejor será la experiencia de ellos. Si mejoramos la experiencia, incrementaremos su fidelidad y volverán a usar la *app*. Optimizando esta experiencia, obtendremos mejores resultados y conseguiremos mejorar nuestra monetización.

En resumen, sea cual sea el fin de la aplicación, es importante conocer como está funcionando y si está cumpliendo los objetivos que se han marcado para ella. Por ello es importante el uso de estas herramientas, porque te ayudan a comprender la aplicación ofreciendo información sobre la misma.

Una vez ya sabemos lo que son y que utilidad tienen las herramientas de *analytics*, vamos a pasar a enumerar y describir algunas de estas herramientas. Una vez hecho esto, las compararemos en base a unas características que definiremos, y por último haremos una comparativa más general de todas ellas y diremos por qué hemos elegido *Google Analytics*.



Google Analytics

Empezamos con *Google Analytics*. Gracias a *Google Analytics* se pueden analizar métricas tanto en Android como en iOS en todas las fases del proceso: adquisición, interacción y resultados. Esta plataforma ofrece funciones avanzadas como analítica en tiempo real, segmentación avanzada, además de la posibilidad de efectuar un seguimiento de sus propios parámetros y métricas personalizados.

Puedes saber cómo los usuarios obtienen la aplicación. Esto ayuda a conseguir una visión completa del proceso de obtención de usuarios, incluidos el coste y los próximos pasos a seguir. Por ejemplo, los informes de usuarios nuevos y activos muestran el número de usuarios que ejecutan su aplicación cada día, las fuentes de tráfico de *Google Play* permiten conocer qué fuente de tráfico aporta más usuarios nuevos, y la visión general de los dispositivos muestra los principales dispositivos móviles y versiones de sistema operativo en los que se ejecuta su aplicación.

Google Analytics ofrece informes con datos clave sobre el poder de atracción de las aplicaciones y sobre el comportamiento de los usuarios. Gracias a este sistema de seguimiento de eventos especializado, se puede supervisar cualquier evento de interés. El flujo de interacción es una representación visual de los pasos que realizan los usuarios en la aplicación. Muestra las pantallas concretas que visualizan, así como las acciones que llevan a cabo. Los informes del comportamiento de usuario evalúan la fidelización, la frecuencia y el nivel de interacción. Los bloqueos de la aplicación permiten resolver las incidencias que se producen en los dispositivos y los sistemas operativos, al detectar las tendencias de los bloqueos y las excepciones. Conocer el comportamiento en la aplicación resulta fundamental para desarrollar y mejorar la experiencia que ofrece.

Una última función es el poder saber si se los usuarios están cumpliendo los objetivos marcados. Las aplicaciones se rigen por objetivos. Las conversiones por objetivos le permiten establecer eventos de conversión en la aplicación (por ejemplo, el tiempo que el usuario ha pasado en la aplicación o una compra en la aplicación, un clic en un anuncio, etc.) para evaluar el éxito obtenido y, en última instancia, conocer el valor del cliente. El sistema de comercio electrónico de esta herramienta permite realizar un seguimiento de las compras integradas en la aplicación y proporciona datos clave para vender bienes virtuales o tangibles adquiridos en la aplicación, ya que muestra el número de compras y los ingresos generados.



Figura 1.1. Ejemplo Google Analytics

Countly

Countly es una herramienta de analítica móvil multiplataforma, de código libre, que entre sus características incluye el análisis en tiempo real. Su precio es de 125\$ si no se superan una serie de sesiones o eventos al mes, 250\$ para ampliar el límite, o precio negociado por encima de este nuevo límite. Además, tiene una versión de prueba gratis limitada.

Dispone de una *Community Edition* gratuita para uso personal o interno de la empresa de desarrollo, con la limitación de que solo dispone de soporte de la comunidad, debe alojarse en un servidor propio y no posee funciones como los embudos o la segmentación, lo que la limita bastante para hacer un análisis en profundidad.

Los datos ofrecidos por *Countly* aparecen en su panel de control donde se da un resumen general de los ingresos, ingresos promedio por usuario, el número de usuarios de pago y más. Con la función de analíticas de usuario de *Countly*, se puede obtener información de todos y cada uno de los usuarios, y tomar decisiones en base a esa información. Puedes realizar un seguimiento de lo que necesitas saber acerca de los clientes y cómo utilizan las aplicaciones, saber quién ha iniciado sesión y de que ha hecho durante ella y crear filtros y seguir a los usuarios a través de él.

Countly puede analizar *apps* de Android, iOS, Windows Phone y Blackberry y también soporta las aplicaciones multiplataforma hechas con Unity3D. Incluso existen SDKs desarrollados por la comunidad para poder usarlos [3].

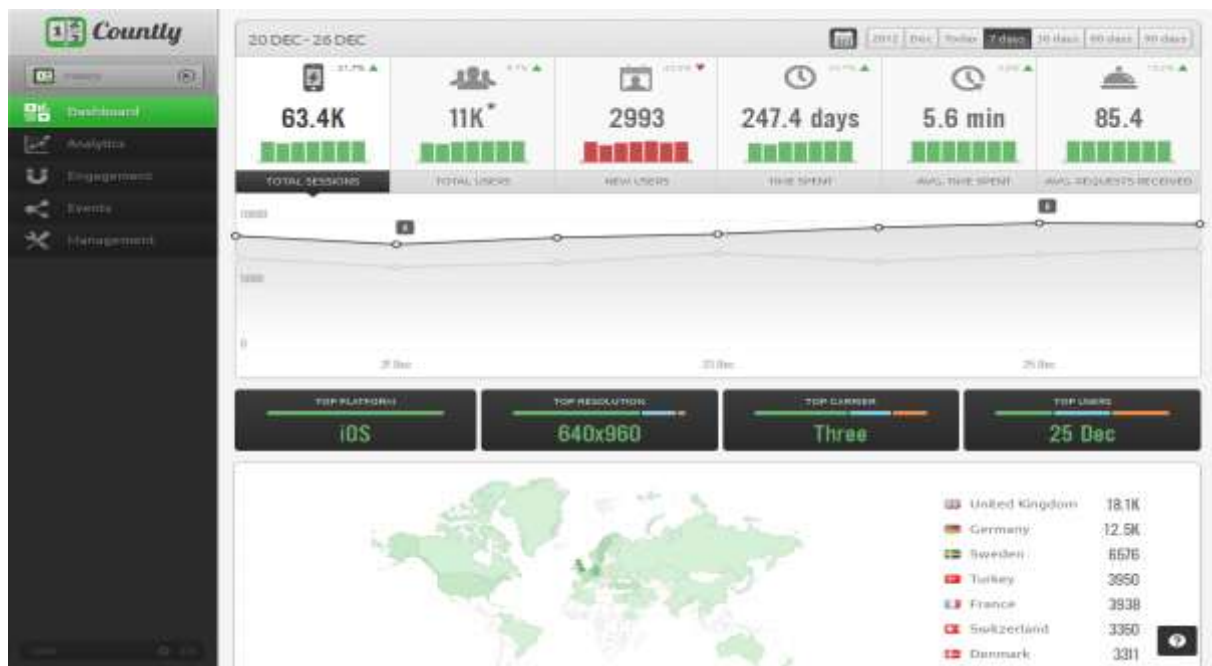


Figura 1.2. Ejemplo Countly

Localytics

Localytics divide su solución en dos modalidades, según se desee sólo analítica o analítica y marketing. *Localytics* permite la segmentación personalizada, o el seguimiento de campañas. Pero añadiendo test A/B para saber que contenido funciona mejor dentro de la aplicación. Permite el envío de mensajes In-App o mensajes Push basadas en segmentos para enviar campañas personalizadas a los usuarios [3].

En la parte de analítica dispone de opciones de segmentación, embudos, análisis del valor del ciclo de vida del usuario y el compromiso en tiempo real.

Para usar *Localytics* hay que pagar una cuota mensual que depende de lo numerosa que sea la audiencia de la aplicación. Si la aplicación tiene menos de 10.000 usuarios activos al mes la herramienta es gratuita. Pero a partir de esa cifra los precios oscilan desde los \$200 a los \$1500 al mes, siempre en función de los usuarios activos. Existe una versión de prueba de 30 días, útil para comprobar si es el tipo de herramienta que se ajusta a las necesidades de la compañía [4].

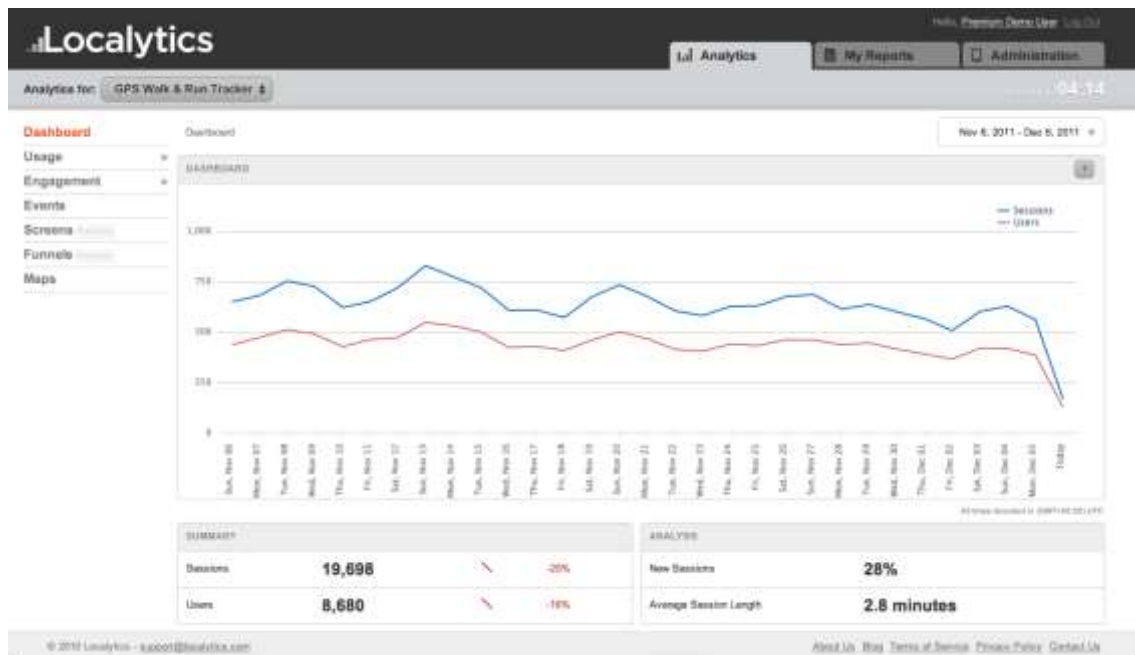


Figura 1.3. Ejemplo Localytics

Flurry

Flurry aparece en 2008 como una de las primeras herramientas de *Mobile Analytics*. Presenta dos ventajas principales. La primera es que está disponible para las principales tecnologías de desarrollo de *apps*: iOS, Android, Windows Phone y más. La segunda es que se trata de un producto totalmente gratuito.

Con *Flurry* se pueden seguir eventos de todo tipo dentro de la aplicación. Permite la segmentación de usuarios por uso, retención, geografía, edad, sexo e intereses, pero también puedes crear segmentos personalizados. Permite medir y mejorar la conversión de usuarios a través de la definición de *funnels*. También permite la personalización de informes y la definición de alarmas [4].

Por último decir que *Flurry* proporciona análisis de errores de las aplicaciones. Es muy importante detectar y atajar este tipo de problemas para que el usuario esté satisfecho con la *app*. Cuando *Flurry* detecta un fallo en la ejecución de la *app*, envía inmediatamente un email al responsable de la aplicación para que pueda solventar el problema.

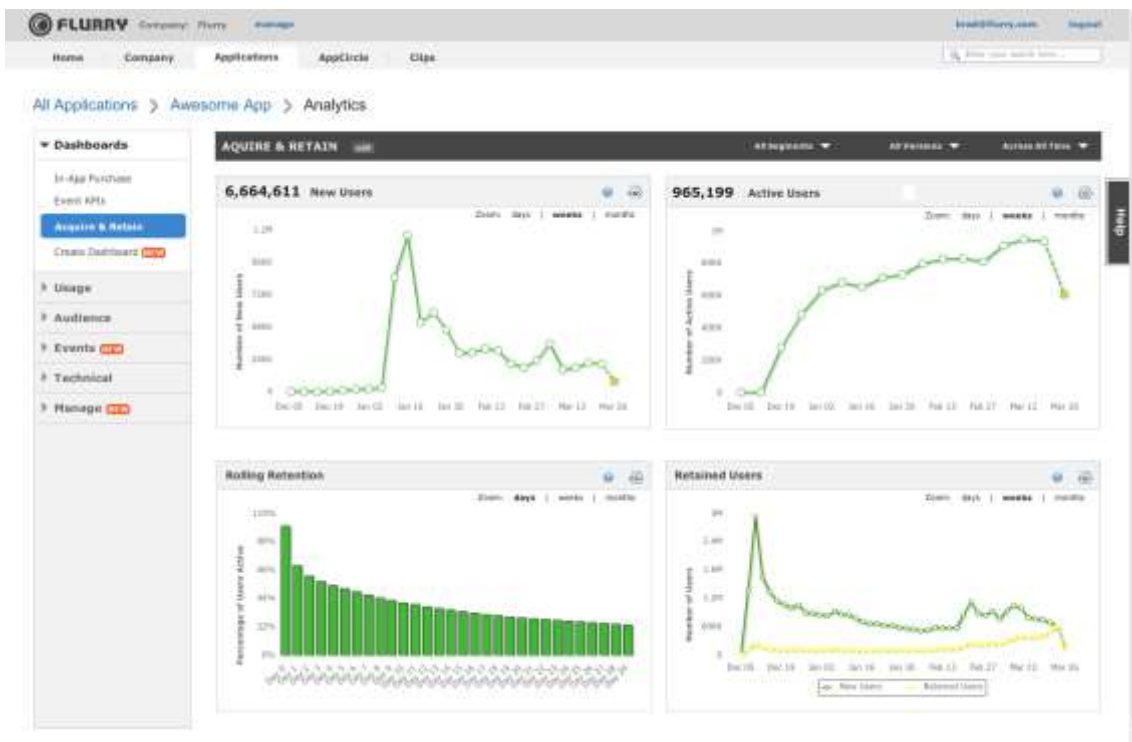


Figura 1.4. Ejemplo Flurry

Honey Tracks Game Analytics

He añadido esta herramienta porque puede resultar interesante mencionar una herramienta centrada en el análisis de juegos.

Honeytracks es una solución de analítica orientada específicamente a juegos que ha sido adquirida recientemente por iQU para integrarlo en una solución más completa, por lo que es previsible que haya cambios importantes en poco tiempo.

iQU estaba más orientada a la parte de marketing, ofreciendo soluciones de publicidad en juegos que ahora dispondrán del complemento de la analítica de *Honeytracks* que ofrece una versión gratuita con seis dashboards pre configurados con una selección de métricas para representar los datos principales de activaciones, retención de usuarios, beneficios, etc. y almacenamiento de los datos de solo un mes. En las versiones de pago (hasta 99€ o personalizado) se incluye la posibilidad de hacer segmentos para analizar comportamientos, generar dashboards propios, hacer test A/B, analizar el proceso de adquisición y llegar a afinar hasta usuarios individuales. Además guarda los datos de 1, 3 o 12 meses, lo que puede resultar poco si queremos analizar tendencias a largo plazo [3].

Es una solución multiplataforma con SDKs para Android, iOS, Flash, C#, PHP, Ruby, Java y Javascript.



Figura 1.5. Ejemplo Honey Track Game Analytics

Una vez ya tenemos definidas las herramientas, vamos a compararlas. Para ello vamos a hacer uno de una tabla y a compararlas teniendo en cuenta varias características vamos a definir ahora.

Primero vamos a ver las diferentes plataformas en las que estas herramientas pueden trabajar. Veremos sobre que sistemas pueden realizar mediciones, iOS, Android, Windows phone, etc. y comprobaremos si el software de las herramientas es libre, o si por el contrario es propietario. También veremos el precio de cada una de estas herramientas, si son completamente gratis, o si son gratis con algunas funciones de pago o si directamente solo ofrecen versiones de pago. Veremos que herramientas ofrecen la posibilidad de notificar a los desarrolladores si ha habido algún tipo de error en la aplicación, entendiendo estos errores como en ejecución o posibles *bugs*. Mostraremos también si estas herramientas ofrecen un seguimiento individual de usuarios, esto es ver si la herramienta es capaz de poder analizar a un usuario sin meterlo dentro de un conjunto de usuarios, si no que ofrece la posibilidad de ofrecer información individualizada de usuarios. Más aspectos importantes a tener en cuenta son si se pueden ver datos en tiempo real, es decir, saber al momento toda la información que necesitamos saber de nuestra aplicación y si es posible medir eventos. Entendemos eventos como todas aquellas acciones que hacen los usuarios. Si sabemos el objetivo de la *app*, los eventos serán muy fáciles de definir. Y estos eventos nos ayudarán a saber por dónde han pasado el usuario. Por último veremos que alternativa o alternativas es capaz de medir los ingresos que está generando nuestra aplicación.

Ahora que tenemos definidos los parámetros por los que compararemos las herramientas, pasamos a ver la tabla comparativa con las distintas herramientas.

Comparativa entre herramientas

	Google Analytics	Countly	Localytics	Flurry
Plataformas	iOS Android	iOS, Android, Windows phone, Blackberry, HTML 5	iOS, Android, Windows phone, Blackberry, HTML 5	iOS, Android, Windows phone, Blackberry, Java ME
Precio	Gratis	Gratis con versión de pago	Gratis con versión de pago	Gratis
Software	Propietario	Libre	Propietario	Propietario
Detección de errores	Si	Si	Si	Si
Seguimiento individual de usuarios	No	No	No	No
Datos en tiempo real	Si	Si	Si	Si
Medición de eventos	Si	Si	Si	Si
Medición ingresos	Si	No	No	No

[5]



En conclusión podemos decir que se trata de aplicaciones muy completas y que con el tiempo irán mejorando. Todas ofrecen más o menos las mismas funcionalidades pero *Google Analytics* parece la opción más completa de todas. No solo ofrece las mismas posibilidades que las demás, si no que ofrece la interfaz más amigable y ofrece datos de una manera clara. La mejor alternativa a *Google Analytics* puede ser *Flurry*, pero resulta difícil de manejar y la información no aparece tan clara. Por lo que *Google Analytics* es la elección en este trabajo para poder analizarla y comprender más su funcionamiento y todas las posibilidades que ofrece.

3. Caso de estudio QuotationShake

Una vez sabemos lo que es un *market* y una herramienta de *analytics* y tenemos claro dónde vamos a publicar nuestra app y cómo vamos a monitorizarla, tenemos que saber con qué aplicación vamos a trabajar. Pues bien, aquí vamos a presentar la aplicación con la que hemos trabajado, veremos su estructura, como funciona y lo que hemos añadido para que sea posible saber cómo los usuarios utilizan esta app.

La aplicación que hemos elegido es *QuotatioShake*. Se trata de una app donde puedes ver citas y frases célebres de diferentes personajes famosos. Su uso es bastante sencillo y tiene una característica que la hace diferente, agitando el dispositivo, se podrá cambiar de cita y que aparezca una nueva. Este es su principal reclamo y la que lo hace destacar entre otras *app* similares. Además de poder ver frases, podrás guardarlas en favoritos, y una vez allí, pulsando en el nombre del autor se muestra una *web* con información de éste. Como vemos es una *app* simple y de fácil uso que nos viene muy bien para este trabajo.

En cuanto a su estructura, tenemos cuatro paquetes diferentes donde se reparten las diferentes clases: *activities*, *database*, *pojo*, *shakingmanagement*, *tasks* y *trackers*. Vamos a ver que contienen cada una.

Dentro de *activities* tenemos las diferentes actividades de la que consta nuestra aplicación. Estás son *DashboardActivity*, *FavouriteActivity*, *AboutActivity*, *QuotationActivity*, *SettingsActivity* y *AppCompatPreferenceActivity*. La actividad principal es *DashboardActivity*, desde aquí simplemente accedemos al resto de actividades de nuestra *app*. Es lo primero que se muestra al abrir la aplicación y desde esta vista accedemos a todas las funciones que podemos realizar. Estás funciones son algunas muy sencillas, como la que realiza *AboutActivity*, que simplemente muestra información de los autores de esta *app*. También tenemos otra típica actividad, *SettingsActivity*, donde podemos configurar diferentes aspectos para que esté todo a nuestro gusto. En *FavouriteActivity* es donde se muestran las diferentes citas que hemos marcado como favorito. Además podremos tanto borrar una cita concreta, como todas las que tenemos. Y por último tenemos *QuotationActivity*, aquí es donde se nos muestra todas las citas, donde podremos verlas, marcarlas como favorito, y donde se hace uso de la peculiaridad de esta *app*, poder agitar el teléfono para mostrar una nueva cita. Solo añadir que desde la actividad de favoritos se puede mantener pulsado en la cita y se muestra una *web* con información de su autor.

Luego tenemos los paquetes *pojo* y *task*, ambos contienen solo una clase. En este primero simplemente tenemos la clase *Quotation*, donde se implementan los *gets* y *sets* necesarios para crear una nueva cita. Y en *task* es donde se implementa la obtención de una cita, pero se hace de una forma asíncrona para no perder funcionalidad dentro de la aplicación.

Dentro del paquete *database* es donde tenemos las clases necesarias para realizar los accesos a una base de datos que necesitamos. En este paquete tenemos una primera clase donde creamos la tabla donde iremos añadiendo las citas que se marquen como favoritas. En la otra clase hacemos uso de esta tabla añadiendo consultando o eliminando citas.



Nos quedan dos paquetes más, *shakingmanagement* y *trackers*. Dentro de *shakingmanagement* tenemos la clase del *listener* que implementa la función que nos permite cambiar una cita agitando el móvil. Y por último, dentro de *trackers* tenemos la clase que configura el o los *trackers* que enviarán a *Google Analytics* toda la información sobre cómo se está utilizando nuestra *app*.

Por último, para comprender mejor como funciona esta *app*, vamos a mostrar un flujo de navegación que hay entre las diferentes pantallas para saber de una manera más gráfica y como se navega por la aplicación.

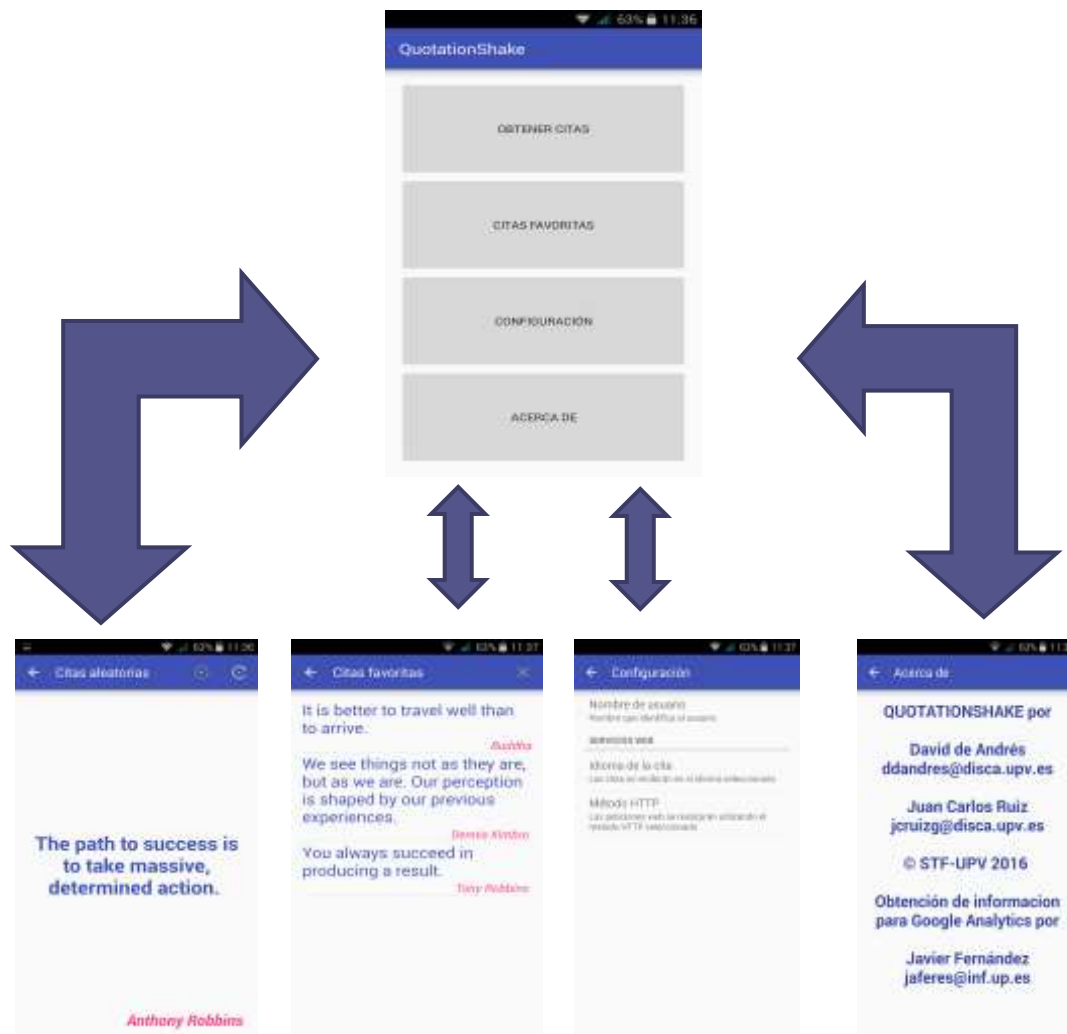


Figura 1.6 Flujo navegación de QuotatiomShake

Como vemos, desde la pantalla principal podemos dirigirnos a cualquier otra pantalla de la *app*. Y pulsando a la flecha de arriba a la derecha podemos volver de nuevo a la pantalla principal.

3.1. Seguir QuotationShake con Google Analytics

Una vez sabemos cómo funciona nuestra aplicación tenemos que preguntar ¿Qué vamos a medir? ¿Qué información queremos captar sobre como el usuario usa esta app? Pues bien, aquí vamos a contestar a estas preguntas y veremos en qué puntos hemos decidido introducir código que nos aporta información útil.

Antes de nada, debemos configurar correctamente *Google Analytics* para poder utilizarlo. Una de las mayores ventajas que tiene esta herramienta es su facilidad de uso e implementación.

Lo primero es hacer una cuenta en la página oficial de *Google Analytics*, es algo sencillo, y si tienes cuenta en Google se vincula automáticamente y no hace falta un registro. Una vez registrado es necesario crear una propiedad. Una propiedad es para *Google Analytics* un recurso asociado a un código de seguimiento. Cuando se realiza un seguimiento, se incluye un ID de propiedad, que se proporciona automáticamente, en el código de seguimiento. Los datos se recopilan en la propiedad correspondiente. Hay que tener en cuenta que si se asocia un ID de propiedad a dos aplicaciones, los datos de ambas aparecen en la misma propiedad en la cuenta de *Google Analytics*. Pero es posible aplicar vistas y filtros para organizar los datos de ambas aplicaciones

Una vez registrado y configurada la propiedad, es necesario configurar nuestro código de la aplicación para que sea capaz de usar *Google Analytics*. Esto se hace descargando el SDK de *Google Analytics* que se encuentra en la página oficial. Una vez descargado se añade a nuestro proyecto de la aplicación. Ahora debemos configurar algunos archivos en nuestro proyecto. Primero modificaremos nuestro *AndroidManifest.xml* y añadiremos un par de permisos.

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

Código 1.1 Permisos necesario para que funcione Google Analytics

Estos permisos son necesarios para que la aplicación tenga acceso a internet, y pueda enviar datos a *Google Analytics*. Una vez hecho esto, es necesario añadir una dependencia para poder usar los servicios de *Google Play*, así que en el archivo *build.gradle* añadiremos esto.

```
compile 'com.google.android.gms:play-services-analytics:8.1.0'
```

Código 1.2 Añade dependencias necesaria a nuestro proyecto



Ahora tenemos que configurar dos archivos XML que sirven para configurar los diferentes parámetros de los *trackers* que añadas en la aplicación. Un archivo servirá como configuración general para todos los *tracker* que uses en la aplicación, y otro más particular y específico por si quieres cambiar en valor de un parámetro en diferentes *trackers*.

Estos parámetros pueden tener diferentes valores en función de lo que necesites. Vamos a repasar estos parámetros y sus funciones.

Empezamos por los que suelen ser los parámetros globales y que van a tener todos los *trackers* de tu aplicación.

- `ga_LogLevel`: Indica el nivel de detalle del objeto de seguimiento. Estos detalles son información de errores o datos, como el tipo del objeto seguimiento. Los valores válidos, de menos a más detalle son: *error*, *warning*, *info* y *verbose*. De forma predeterminada, se configura *warning*.
- `ga_dispatchPeriod`: Frecuencia de envío de datos de manera automática en segundos. Por defecto es 30 minutos.
- `ga_dryRun`: Si se configura como "true", se habilita el modo de ejecución segura. El valor predeterminado es "false". En el modo de ejecución segura, no se envían los hits a los servidores de *Google Analytics*. Esto se utiliza para depurar llamadas al SDK de *Google Analytics* sin contaminar los datos registrados.

Vamos ahora con parámetros más concretos que pueden diferenciar el uso de un *tracker* de otro.

- `ga_trackingId`: El ID de seguimiento de *Google Analytics* al que se enviarán los datos. Si quieres inhabilitar el seguimiento, no se proporciona este valor.
- `ga_appName`: El nombre de la aplicación que se usará en los informes.
- `ga_appVersion`: Versión de la aplicación que se usará en los informes.
- `ga_autoActivityTracking`: Si se configura como true, se realizará el seguimiento automático de las vistas (actividades). El valor predeterminado es false.
- `ga_reportUncaughtExceptions`: Realiza un seguimiento automático de las excepciones cada vez que se produce una no detectada en la aplicación. El valor predeterminado es false.
- `ga_sessionTimeout`: El tiempo (en segundos) que la aplicación puede estar en segundo plano antes de que finalice la sesión. El valor predeterminado es de 30 segundos.

Por último tenemos que crear la clase a la que llamaremos cuando queremos iniciar un nuevo *tracker* para enviar información a *Google Analytics*. Esta es la clase implementada en el paquete *trackers* que hemos visto antes.

```
private AnalyticsTrackers(Context context) { mContext = context.getApplicationContext(); }

public synchronized Tracker get(Target target) {
    if (!mTrackers.containsKey(target)) {
        Tracker tracker;
        switch (target) {
            case APP:
                GoogleAnalytics.getInstance(mContext).dispatchLocalHits();
                tracker = GoogleAnalytics.getInstance(mContext).newTracker(R.xml.app_tracker);
                tracker.enableAdvertisingIdCollection(true);
                tracker.enableAutoActivityTracking(true);
                tracker.enableExceptionReporting(true);
                break;
            default:
                throw new IllegalArgumentException("Unhandled analytics target " + target);
        }
        mTrackers.put(target, tracker);
    }

    return mTrackers.get(target);
}
```

Código 1.3 Código que crea el tracker que envía información a Google Analytics

Esta es la parte principal de esta clase. Aquí vemos el método *get* que devuelve el *tracker* que usaremos para enviar nuestros datos. También desde aquí, desde el código, se pueden asignar valores a parámetros. Como vemos, primero llamas a una instancia de *Google Analytics* y luego inicializamos el *tracker* con el archivo XML donde tengamos su configuración.

Una vez hecho todo esto, ya estamos preparados para enviar datos a *Google Analytics* para después comprobar el funcionamiento de nuestra *app*.

Ya estamos preparados para recoger información, pero ¿Qué información recogemos? Pues bien, ahora vamos a mostrar todas las partes en las que hemos decidido implementar un *tracker* que nos enviará información útil de cómo funciona la *app*.

Usuarios

Es importante conocer cuántos usuarios están utilizando tu *app*, saber a qué número de personas está llegando. Para saber esto, por defecto, *Google Analytics* asigna un id por cada nuevo dispositivo que hace uso de la aplicación. Al registrar un nuevo dispositivo, se le asigna un número que pasa a ser su identificador. El problema está en que si un mismo usuario accede desde otro dispositivo diferente, a pesar de ser el mismo usuario, se cuenta como uno diferente. Esto se puede solucionar usando el User-ID. Esta función te permitirá identificar a un mismo usuario sin importar los dispositivos que use. Pero, ¿cómo usar User-ID? Pues bien, primero tenemos que activar esta opción desde *Google Analytics*, tenemos que configurar nuestra herramienta para que reconozca que vamos a hacer uso de esta función. Una vez hecho, nosotros somos los encargados de asignar un ID, esto es muy importante, *Google Analytics* no te proporciona este ID, te proporciona la opción de mandarlo por medio de un *tracker*, pero debe de ser el desarrollador el encargado de asignar este identificador.



Este identificador puede ser cualquier cosa, en nuestro caso, la hora que tiene el sistema en segundos en el momento en que se asigna, seguido de un número aleatorio, así conseguimos que cada id sea único. Una vez ya lo tenemos, lo mandamos a *Google Analytics* para que lo reconozca como usuario nuevo. Pero tenemos que tener en cuenta que debemos guardarnos de alguna manera este identificador para que cuando se vuelva a conectar reconozca al usuario y no crea que es uno nuevo. La forma que tenemos de guardarnos este id es hacerlo en las preferencias, creamos un nuevo campo con el valor del identificador, y cada vez que entramos a la aplicación, vemos si este campo existe, si es así, mandamos de nuevo el id para que se registre que el usuario ha entrado de nuevo y si no existiera, simplemente lo creamos.

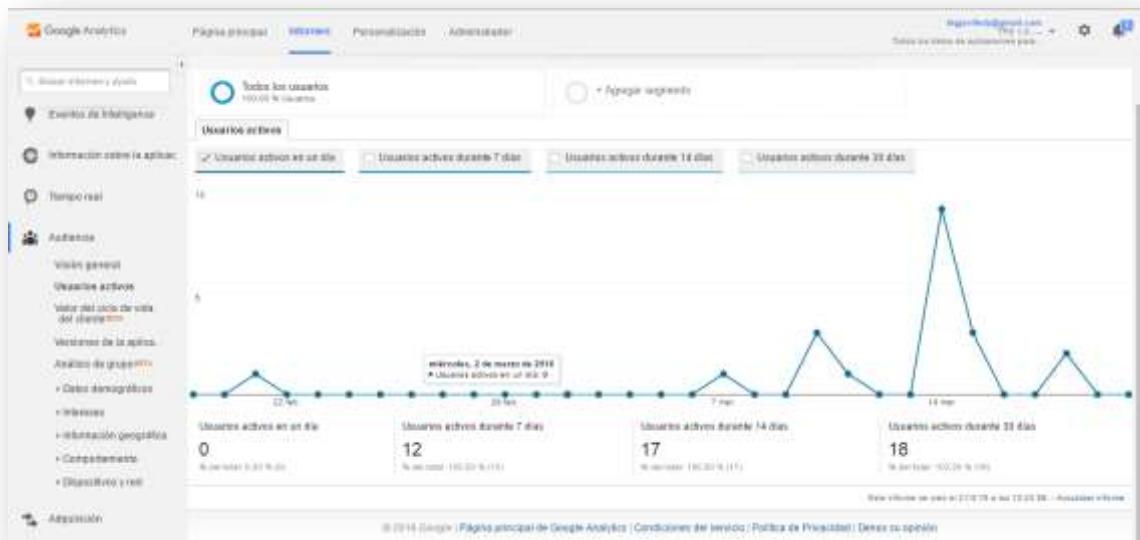


Figura 1.7 Ejemplo muestra de estadísticas de usuario

En cuanto a la información que *Google Analytics* nos muestra los usuarios simplemente nos da cifras de cuantos usuarios se han registrado y de cuantos han entrado en la aplicación en el último día, en las últimas semanas y en el último mes.

Pantallas

Las pantallas en *Google Analytics* representan el contenido que los usuarios ven en la aplicación. Cuando compruebas las visitas a una pantalla, puedes descubrir el contenido que ve la mayoría de los usuarios y cómo navegan a través del mismo.

El registro de visita a una pantalla solo requiere un campo con el nombre de la pantalla que ha visitado el usuario. Creamos el *tracker* encargado de llevar la información a *Google Analytics*, a continuación indicamos el nombre de la pantalla, y enviamos esta información.

Es posible realizar esto de manera automática a través del archivo de configuración XML de esta forma:

```

<!-- Enable automatic Activity measurement -->
<bool name="ga_autoActivityTracking">true</bool>

<!-- The screen names that will appear in reports -->
<screenName
name="com.google.android.gms.analytics.samples.mobileplayground.ScreenviewFragment">
  AnalyticsSampleApp ScreenViewSampleScreen
</screenName>
<screenName
name="com.google.android.gms.analytics.samples.mobileplayground.EcommerceFragment">
  AnalyticsSampleApp EcommerceSampleScreen
</screenName>

```

Código 1.4 Código que recoge automáticamente las pantallas vistas

Como vemos, primero habilitamos la opción “ga_autoActivityTracking” y después asignamos nombres a las diferentes pantallas de las actividades.

A continuación vamos a ver un ejemplo de cómo *Google Analytics* muestra la información referidas las pantallas vistas.

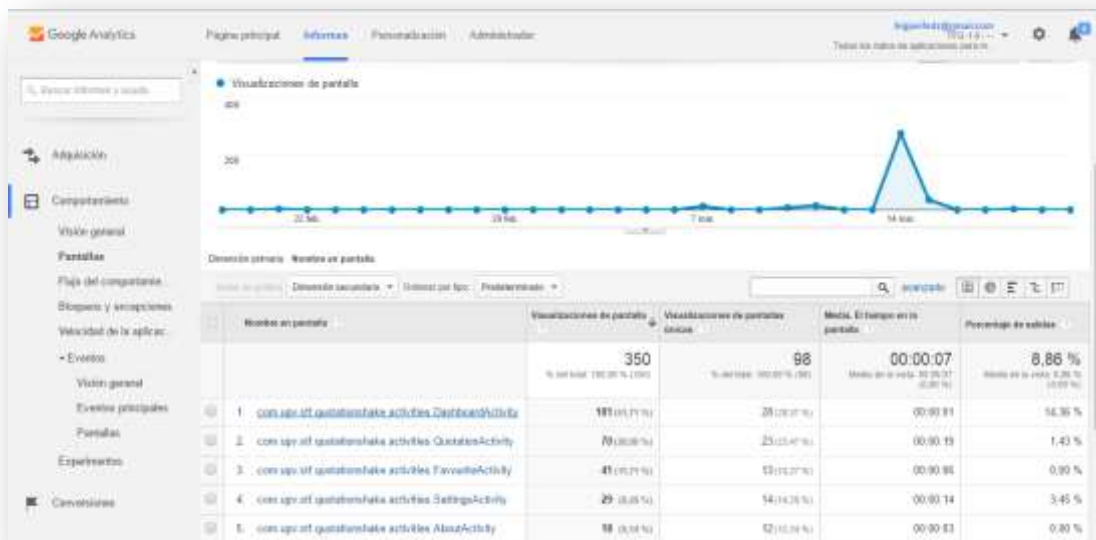


Figura 1.8 Ejemplo muestra información de pantallas vistas



Vemos que la información que muestra es primero, la pantalla vista, a continuación vemos las visualizaciones totales de esa pantalla, y luego el número de sesiones en las que se ha visto esa pantalla, al menos, una vez. También nos enseña el tiempo medio que se está en esa pantalla, y el porcentaje de veces que se ha salido de la aplicación desde ella. Puedes ver más información, como por ejemplo, los eventos o bloqueos y excepciones que se producen en esa pantalla, pero esto es lo que se muestra por defecto.

Sesiones

Una sesión representa un único periodo en el que el usuario ha interactuado con la aplicación. Las sesiones son como contenedores útiles de actividad medida de nuestra *app*, que incluye visitas a una pantalla, eventos o transacciones de comercio electrónico.

Al igual que las pantallas y otras funciones que veremos, también se puede configurar de manera manual o automática. Si optamos por la forma manual, necesitaremos incluir esto en nuestro código:

```
// Get tracker.  
Tracker t = ((AnalyticsSampleApp) getActivity().getApplication()).getTracker(  
    TrackerName.APP_TRACKER);  
  
// Start a new session with the hit.  
t.send(new HitBuilders.ScreenViewBuilder()  
    .setNewSession()  
    .build());
```

Código 1.5 Forma manual de empezar una nueva sesión

De esta manera, ya hemos iniciado una nueva sesión. Para configurar *Google Analytics* y que inicie nuevas sesiones automáticamente usaremos el archivo XML de configuración, con el parámetro `ga_sessionTimeout` y un número que indica el tiempo que tiene que pasar con la aplicación ejecutándose para iniciar la sesión.

```
<resources>  
    <integer name="ga_sessionTimeout">300</integer>  
</resources>
```

Código 1.6 Código que determina el tiempo mínimo para que empieza una sesión

A continuación vamos a ver un ejemplo de cómo *Google Analytics* muestra la información referida las sesiones.

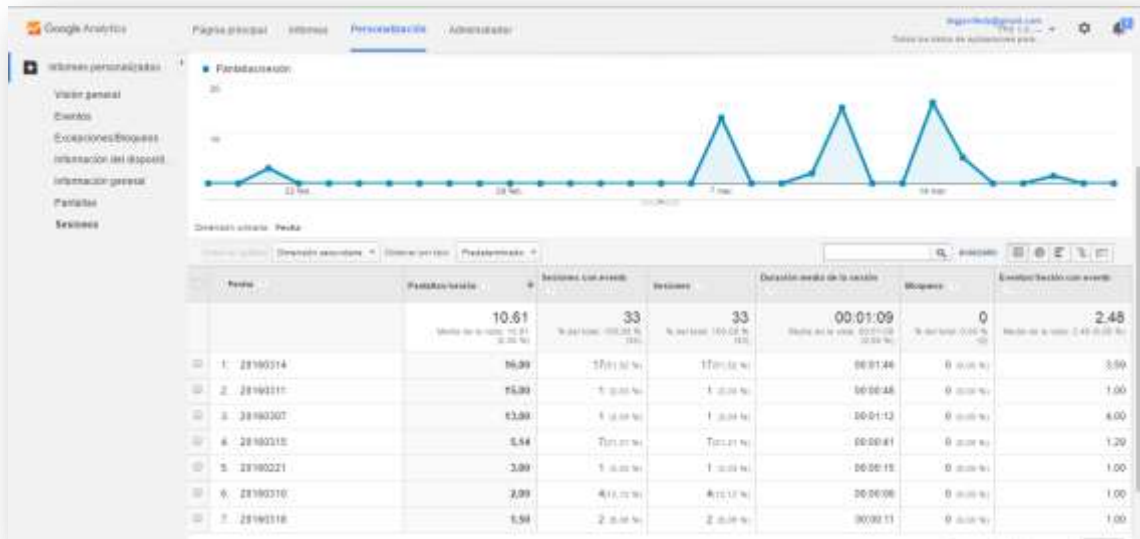


Figura 1.8 Ejemplo muestra información de las sesiones

Podemos obtener mucha información acerca de las sesiones, aquí vemos como tenemos las sesiones ordenadas por fechas. En cada fecha nos informa de las sesiones que ha habido, cuantas pantallas se han visto por sesión, y los eventos que han ocurrido en ellas. Además de los fallos que han podido haber y la duración media de las sesiones. Podemos obtener más información, pero esta es la que hemos decidido filtrar nosotros porque nos parece la más importante.

Bloqueos y excepciones

La medición de los bloqueos y de las excepciones permite medir el número y el tipo de bloqueos detectados y no detectados y las excepciones que ocurren en tu aplicación. Una excepción tiene dos campos a rellenar, la descripción, que no es obligatoria, y un boolean que dice si el error es o no falta, que si es obligatorio. Los datos de bloqueos y de excepción se encuentran en *Google Analytics* disponibles en el informe de bloqueos y de excepciones.

Las excepciones detectadas son errores en tu aplicación para los que se ha definido un código de excepción.



Las excepciones se miden definiendo los valores de los campos antes mencionados en el objeto de seguimiento y enviando el hit, tal como vemos en el ejemplo:

```
//Get tracker.  
Tracker t =  
    ((AnalyticsSampleApp)  
    getActivity().getApplication()).getTracker(TrackerName.APP_TRACKER);  
  
// Build and send exception.  
t.send(new HitBuilders.ExceptionBuilder()  
    .setDescription(getExceptionMethod() + ":" + getExceptionLocation())  
    .setFatal(getExceptionFatal())  
    .build());
```

Código 1.6 Extracto de código que muestra como enviar una excepción a Google Analytics

Las excepciones no detectadas representan condiciones inesperadas durante la ejecución, a menudo críticas, que bloquean la aplicación. Estas excepciones se pueden enviar a *Google Analytics* automáticamente estableciendo un valor booleano a `ga_reportUncaughtExceptions` en el archivo de configuración XML.

```
<bool name="ga_reportUncaughtExceptions">true</bool>
```

Código 1.7 Código para que capture excepciones no detectadas

De esta manera se envían automáticamente las excepciones no detectadas a *Google Analytics*.

Otra forma de capturar las excepciones no detectadas de una forma más avanzada es la clase `ExceptionReporter`.

`ExceptionReporter` se puede usar como controlador predeterminado de excepciones no detectadas. Una vez enviada una excepción a *Google Analytics*, la clase `ExceptionReporter` puede enviar la excepción a cualquier controlador de excepciones no detectadas que especifiques.

Con el siguiente código podemos crear un nuevo objeto `ExceptionReporter` y definirlo como nuevo controlador predeterminado de excepciones no detectadas. Por lo tanto, cada excepción no detectada se enviará a *Google Analytics* y, a continuación, se enviará al controlador de excepciones no detectadas anterior. En la mayoría de las aplicaciones, el controlador predeterminado registrará la excepción en el registro y finalizará la aplicación.

```

UncaughtExceptionHandler myHandler = new ExceptionReporter(
    myTracker, // Currently used Tracker.
    Thread.getDefaultUncaughtExceptionHandler(), // Current default uncaught exception
    handler.
    context); // Context of the application.

// Make myHandler the new default uncaught exception handler.
Thread.setDefaultUncaughtExceptionHandler(myHandler);

```

Código 1.8 Código necesario para hacer de ExceptionReporter nuestro controlador predeterminado

Cuando se usa la medición automática de excepciones, hay que tener en cuenta que:

- Todas las excepciones que se envían usando la medición automática de excepciones aparecen como críticas en los informes de *Google Analytics*.
- De forma predeterminada, el campo de descripción se establece automáticamente usando el tipo de excepción, el nombre de la clase, el nombre del método y el nombre de subproceso.

Google Analytics a la hora de mostrar los fallos de la aplicación, primero nos filtra por versión de la aplicación y nos muestra las excepciones y los bloqueos que ha habido. Al entrar en una versión en concreto ya nos detalla más el error encontrado con una pequeña descripción que nosotros podemos modificar desde el código.

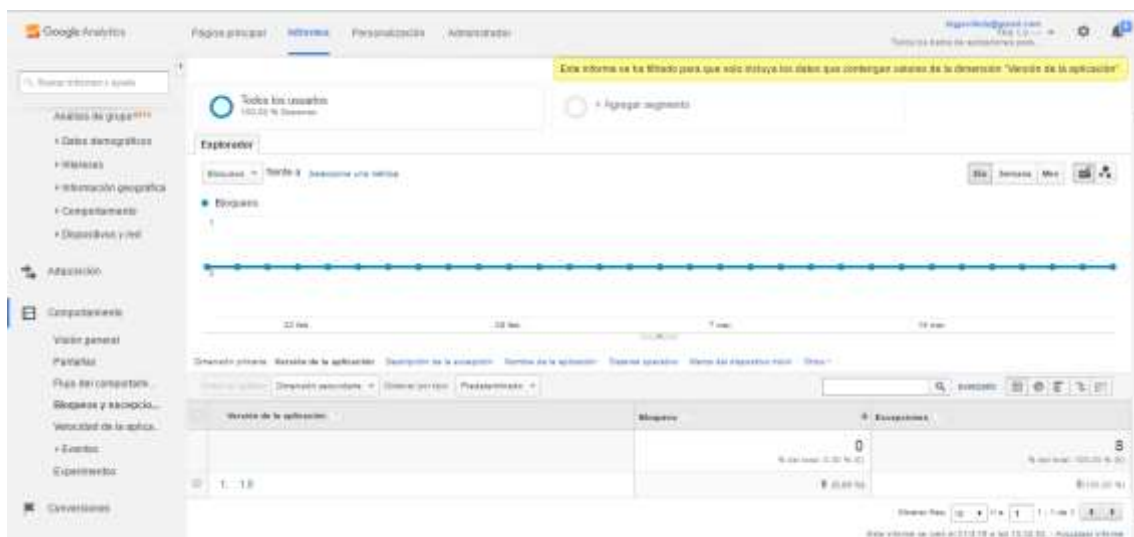


Figura 1.8 Ejemplo muestra información de las bloqueos y excepciones

Eventos

Los eventos son una forma útil de recopilar datos sobre la interacción de un usuario con componentes interactivos de la aplicación. Estos eventos pueden ser casi cualquier cosa, como la pulsación de un botón o el uso de un elemento determinado.

Un evento consta de cuatro campos que puedes usar para describir la interacción de un usuario con el contenido de tu aplicación. Estos campos son la categoría, la acción que se ha llevado a cabo, y dos campos opcionales, una etiqueta y un valor a dicho evento. Este valor es a elección del usuario, se puede dar cualquier valor si se considera necesario.

```

tiempo = System.currentTimeMillis();

Tracker t = AnalyticsTrackers.getInstance().get(AnalyticsTrackers.Target.APP);
t.send(new HitBuilders.EventBuilder()
    .setCategory("Pulsado")
    .setAction("Cita añadida a favoritos")
    .setLabel("Favorito")
    .build());
    
```

Código 1.9 Extracto de código que muestra como recoger eventos en nuestra app

Vemos como creamos primero el *tracker*, y luego rellenamos los campos que necesitamos. En este caso se detecta cuando añadimos a favoritos una cita.

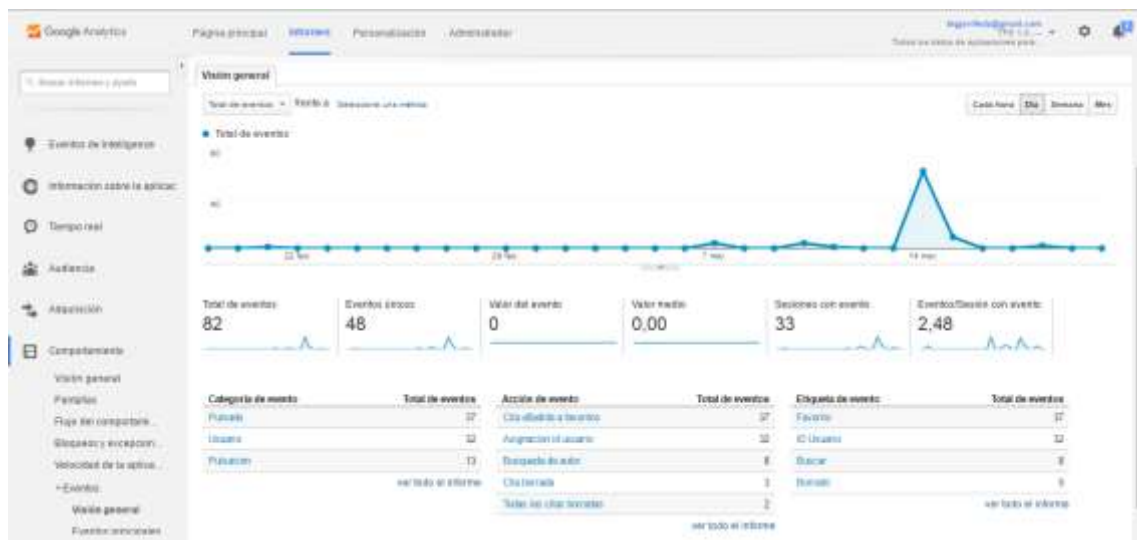


Figura 1.9 Ejemplo muestra información de eventos producidos

Aquí vemos como *Google Analytics* nos muestra la información sobre los eventos de nuestra *app*. Por defecto muestra el número total de eventos, su valor medio, en este caso no hemos dado valor a los eventos, las sesiones en las que ha habido eventos, y la media de eventos por

sesión. Nos muestra la información dividida en las categorías, acciones y etiquetas que nosotros hemos creado en el código. En nuestro caso, por ejemplo, dentro de la categoría “pulsación” tenemos las acciones “búsqueda de autor”, “cita borrada” y “todas las citas borradas”.

Tiempos

La medición de los tiempos de usuario ofrece una forma de poder mostrar y controlar los tiempos de tu aplicación. Puede ser útil, por ejemplo, para medir los tiempos de carga de los recursos o para saber el tiempo que un usuario está en una determinada pantalla. Pero *Google Analytics* no mide el tiempo, eso tienes que hacerlo tú, solo recibe el tiempo y lo muestra en sus informes. Ahora veremos cómo mandamos a *Google Analytics* este tiempo.

```
public void resetQuotation() {
    newQuotation = true;
    addQuotation = false;
    supportInvalidateOptionsMenu();

    Tracker t = AnalyticsTrackers.getInstance().get(AnalyticsTrackers.Target.APP);
    t.send(new HitBuilders.TimingBuilder()
        .setCategory("Tiempo")
        .setValue(System.currentTimeMillis() - tiempo)
        .setVariable("Tiempo en obtener cita")
        .setLabel("Tiempo Reset")
        .build());

    progressBar.setVisibility(ProgressBar.INVISIBLE);
}
}
```

Código 1.10 Extracto de código que muestra como enviar tiempos a Google Analytics

Como vemos, en variable pasamos nuestro tiempo ya medido, y luego se mostrará después en sus informes clasificado con las etiquetas y categorías que podemos definir.

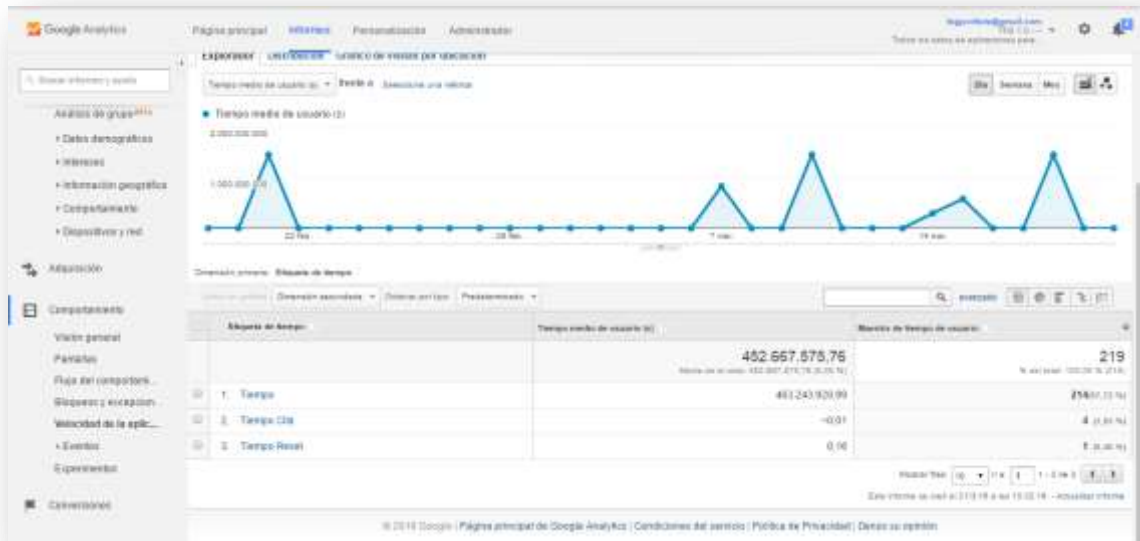


Figura 1.10 Ejemplo muestra información de los tiempos recogidos

Aquí vemos como nos muestra el tiempo de las diferentes etiquetas que hemos implementado. Las dos primeras eran pruebas y no ofrecen datos reales, la última si ofrece un tiempo real de cuánto tarda la aplicación de media en ofrecerte una nueva cita.

4. Publicar QuotationShake

Ya tenemos nuestra aplicación lista para que nos envíen datos que nos ayudaran a mejorar nuestra app, pero aún queda un paso más, publicar nuestra aplicación y ver cómo se comporta con el uso de los usuarios.

Antes de publicar una aplicaciones en *Google Play* y distribuirla a los usuarios, se debe tener lista la aplicaciones, probarla y preparar el material promocional.

Se tiene que comprender el proceso de publicación y prepararte para el lanzamiento de tu producto en *Google Play*. Algunas de las tareas que deben realizarse antes de publicar tu aplicación son, crear un paquete de la aplicación (APK) firmado y listo para el lanzamiento o comprender los requisitos de la aplicación. Las tareas de preparación y publicación son varias y aquí vamos a enumerarlas. No obstante, algunas tareas que se pueden omitir, aunque de todas formas las explicaremos en este apartado.

Para empezar hay que asegurarse de comprender y cumplir las políticas de *Google Play* ya que las implementa activamente. No vamos a hablar de todas las políticas que se deben aceptar, pero si vamos a ver algunas de ellas, las que consideremos más importantes. Empezamos con las políticas de contenido, que juegan un papel importante a la hora de ofrecer una experiencia positiva para todos los usuarios que utilizan *Google Play*. Los términos utilizados en estas políticas tienen el mismo significado que en el acuerdo de distribución para desarrolladores. Estas políticas se deben consultar periódicamente, ya que pueden sufrir modificaciones. En cuanto a las condiciones relativas a la distribución de aplicaciones, estas incluyen precios y pagos, asistencia a usuarios, publicidad, información relacionada con retirada de productos, renuncia a garantías e indemnizaciones, entre otras cosas.

También se han de hacer algunas comprobaciones en favor de los usuarios, como comprobar que la aplicación no incluya contenido no deseado, engañoso, repetitivo o no relacionado con su función principal y comprobar que los anuncios que se muestran en tu aplicación, en caso de que aparezcan, cumplan la política de contenidos y sean adecuados para el nivel de madurez de los usuarios que hayas seleccionado para la aplicación.

Por último decir que las violaciones de las políticas pueden dar lugar a que se pueda suspender tus aplicaciones y las violaciones reiteradas pueden generar la cancelación de tu cuenta de desarrollador. Dicho esto cabe destacar que Google no está obligado a enviar ninguna advertencia antes de suspender o cancelar tu cuenta.



Seguimos con estos pasos previos a publicar nuestra app diciendo que antes de publicarla es importante asegurarse de que se cumplen las expectativas de calidad básicas para todas las aplicaciones de Android en todos los dispositivos que admiten tu aplicación, determinar el nivel de madurez de tu app y fijar los países de distribución. Este último es importante ya que afecta a diversos factores, cómo requisitos legales de países concretos, fijar el precio en su moneda o dar soporte a los distintos idiomas.



CATEGORIZACIÓN

Tipo de aplicación * Aplicaciones

Categoría * Educación

Clasificación de contenido * Nivel de madurez bajo

[Más información sobre la clasificación de contenido](#)

Figura 1.11 Madurez, categoría y tipo de aplicación que se ha de rellenar para publicar en Google Play



Esta aplicación es **Gratuita**

Para evitar infracciones comerciales, consulta las políticas de comercio y distribución de ingresos. No puedes asignar un precio a una aplicación que se ha publicado como gratuita. [Más información](#)

DISTRIBUIR EN ESTOS PAÍSES

Distribuir en 141 países y en el resto del mundo.

SELECCIONAR TODOS LOS PAÍSES

<input checked="" type="checkbox"/> Estados Unidos	Mostrar opciones
<input checked="" type="checkbox"/> Reino Unido	Mostrar opciones
<input checked="" type="checkbox"/> Francia	Mostrar opciones
<input checked="" type="checkbox"/> España	Mostrar opciones

Figura 1.12 Países de distribución y precio de la app

Pasamos ahora a definir los precios de nuestra app. En nuestro caso va a ser gratuita, pero obviamente se le puede fijar un precio. Una vez fijamos que una aplicación se gratuita, no puede volverse de pago, aunque puedes vender productos o suscripciones a ésta. Otra cosa que puedes seleccionar es la facturación integrada en la aplicación , que te permite vender contenido digital en la aplicación. Gracias a este servicio se puede vender una amplia variedad de contenido, incluido contenido descargable o contenido virtual, como niveles o partes de juegos. El servicio de facturación integrada en la aplicación te permite realizar ventas únicas y ofrecer suscripciones dentro de la aplicación. Esto puede ayudarte a monetizar la aplicación mientras el usuario la tenga instalada.

A la hora de publicar en *Google Play*, es necesario proporcionar una variedad de recursos gráficos de alta calidad para presentar tu aplicación. En nuestro caso hemos presentado varias capturas de pantalla, un icono, una imagen destacada otra promocional y un banner de TV. Una vez publicada, estos aparecerán los resultados de búsqueda y en otros lugares. Las capturas de pantalla y los videos también son muy importantes, ya que muestran cómo se ve la *app*, cómo se usa o se juega.

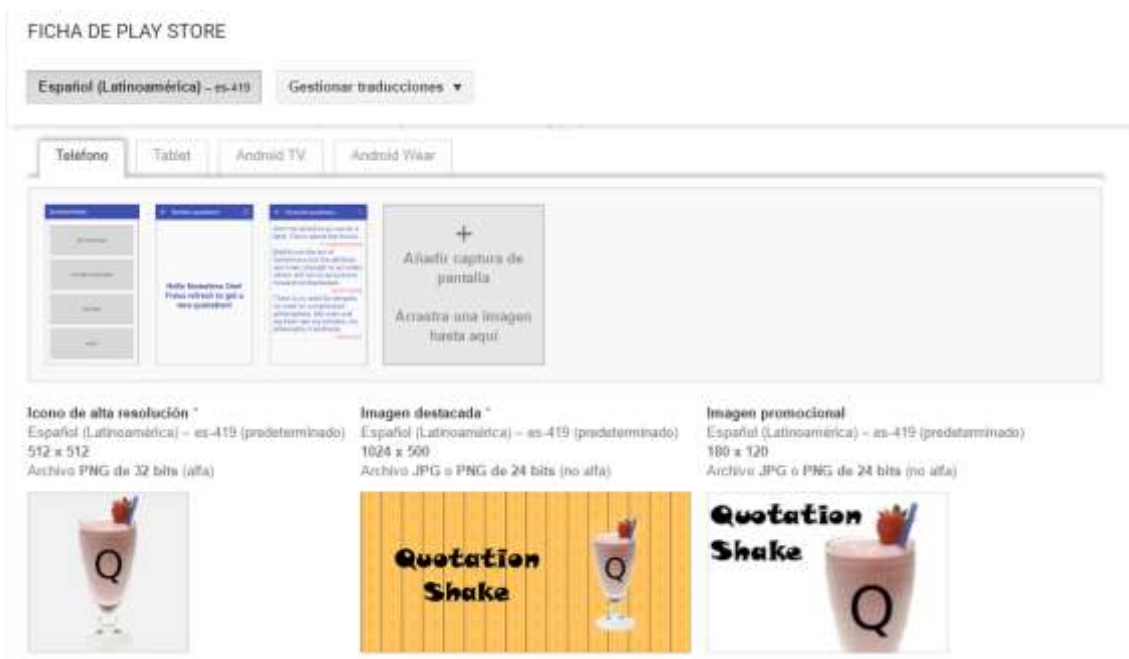


Figura 1.13 Recursos gráficos necesarios para subir la app

Todos estos recursos gráficos deben estar diseñados de forma tal que sean fáciles de visualizar y resalten la aplicación. Los recursos deben hacer referencia al mismo logotipo e icono que los usuarios encontrarán en el selector de todas las aplicaciones una vez que descarguen la aplicación. Además de estos recursos gráficos, es necesario proporcionar un título una descripción para que los usuario sepan de que trata la app.

Título* Español (Latinoamérica) – es-419	QuotationShake 14 de 30 caracteres
Descripción breve* Español (Latinoamérica) – es-419	Citas a golpe de muleca. Agita tu dispositivo y obtén citas bibliográficas. 75 de 80 caracteres
Descripción completa* Español (Latinoamérica) – es-419	QuotationShake es una aplicación sencilla que te permitirá obtener citas de personajes célebres de una manera muy divertida, agitando tu muleca. Agita tu teléfono tantas veces como necesites hasta encontrar una cita que te guste. Con un sólo clic podrás guardar la cita y saber más de la persona que la dijo. Sencillo, eficaz y divertido, así es QuotationShake. ¡ Esperemos que te guste !! 391 de 4000 caracteres

Consulta la [Práctica de Metadatos](#), para evitar infracciones comunes relacionadas con los metadatos de la aplicación. No olvides consultar el resto de [políticas del programa](#) antes de enviar tus aplicaciones.
Si tu aplicación o ficha de Play Store cumple los requisitos para enviar información por adelantado al equipo de revisión de aplicaciones de Google Play, ponte en contacto con nosotros antes de publicar contenido.

Figura 1.14 Título y descripción de la app

Pasamos ahora a hablar del más en profundidad del APK. El proceso para preparar el APK para el lanzamiento es el mismo que se seguiría para otras aplicaciones, independientemente de cómo se distribuyan. Esto incluye, la limpieza y optimización de código, la creación y firma con una clave de lanzamiento y pruebas finales. La aplicación ha de estar firmada con un certificado de publicación lo cual haremos en gracias a Android Studio cuando construyamos nuestro proyecto en modo *release*. Es importante fijar el tamaño de la app. Actualmente, el tamaño máximo permitido de un APK en *Google Play* es de 50 MB. Si la aplicación supera ese tamaño, o si se quiere ofrecer una descarga secundaria, se pueden usar archivos de expansión del APK, que *Google Play* alojará de forma gratuita en sus servidores y administrará automáticamente la descarga a los dispositivos.

- El tamaño máximo permitido para un APK publicado en Google Play es de 50 MB.
- Para cada APK, puedes usar hasta dos (2) archivos de expansión de APK, cada uno de un tamaño máximo de 2 GB.

Además de esto, es importante también asegurarse de que la aplicación esté diseñada para ejecutarse correctamente en las diferentes versiones de la plataforma Android y en los tamaños de pantalla de los dispositivos para los que se quiere distribuir. Se debe confirmar la versión mínima que admite la aplicación, ya que eso afectará a su distribución a dispositivos Android una vez que se publique.

Para los tamaños de pantalla, se tiene que confirmar que la aplicación se ejecuta correctamente y se ve bien en el rango de tamaños de pantalla y densidades de píxeles que se quiere admitir.

Una vez que tengas los APK listos para el lanzamiento a mano, puedes cargarlos en la consola para desarrolladores. Si fuera necesario, se puede reemplazar por una versión más reciente antes de la publicación.

5. Resultados

Ya tenemos nuestra app publicado y lista para que el público la use. Decidimos dejar entorno a una semana para que los usuarios interactúen con la aplicación y así poder luego interpretar y analizar los datos recogidos en *Google Analytics*.

Para empezar veremos qué resultados hemos conseguido en cuanto a pantallas vistas, los eventos detectados y las sesiones que han habido. Además de esto, haremos un análisis sobre la fidelización de nuestros usuarios.

Comenzamos este repaso a los resultados comprobando las sesiones que ha habido durante esta semana que hemos dejado nuestra *app* publicada.



Figura 1.15 Datos de sesiones recogidos gracias a Google Analytics

Vemos como hemos tenido un total de 36 sesiones (recordemos que una sesión para *Google Analytics* es cada vez que un usuario, sea nuevo o no, abre la aplicación). Diez de estas sesiones han durado entre 0 y 10 segundos, y de media tres, la mayoría de estas sesiones seguramente hayan sido porque el usuario ha entrado sin querer, eso explicaría lo poco que duran, por lo que estas no aportan demasiado. Las sesiones valiosas y que si podemos decir que han conseguido enganchar de alguna forma a los usuarios son las que han durado más de 30 segundos. De estas sesiones tenemos un total de 20 con una duración media de alrededor de dos minutos usando la *app*. Por último vemos que se han visitado una media de unas 15 pantallas, cabe decir que se cuanta cada vez que se visualiza una pantalla aunque esta ya se haya visto.

Seguimos este repaso a los resultados comprobando las pantallas vistas por los usuarios.

Nombre en pantalla	Visualizaciones de pantalla	Visualizaciones de pantalla únicas	Medio. El tiempo en la pantalla	Porcentaje de salidas
	377 <small>% del total: 100,00 % (377)</small>	110 <small>% del total: 100,00 % (110)</small>	00:00:07 <small>Medio de la meta: 00:00:07 (0,00 %)</small>	9,02 % <small>Medio de la meta: 0,00 % (0,00 %)</small>
1. com.gpv.sf.quotaforshake.activities.MainActivity	195 (51,72 %)	21 (20,19 %)	00:00:01	14,87 %
2. com.gpv.sf.quotaforshake.activities.OutletActivity	74 (19,63 %)	26 (23,64 %)	00:00:20	1,35 %
3. com.gpv.sf.quotaforshake.activities.FavouriteActivity	43 (11,41 %)	14 (12,73 %)	00:00:06	0,00 %
4. com.gpv.sf.quotaforshake.activities.SettingsActivity	32 (8,49 %)	16 (14,55 %)	00:00:13	3,12 %
5. com.gpv.sf.quotaforshake.activities.AboutActivity	22 (5,84 %)	15 (13,64 %)	00:00:03	0,00 %

Figura 1.16 Información de las pantallas vistas

Como era de esperar la pantalla principal (DashboardActivity) es la más visitada, esto se debe a que es la pantalla que aparece nada más abrir la aplicación por lo que siempre se pasa por ella y su tiempo medio es muy bajo ya que solo están en ella el tiempo que tardan en pulsar para ir a la siguiente pantalla. Y las menos vistas son AboutActivity y SettingsActivity, también algo esperado ya que en estas pantallas no hay apenas interacción con el usuario y solo se suele entrar una vez. En cuanto al resto de pantallas son las que realmente tienen más interacción con el usuario. Lo que resulta más útil de aquí es la columna donde pone “Visualización de pantallas únicas”, aquí se muestran las sesiones en las que al menos una vez, se ha visto esta pantalla. Si cogemos los datos anteriores y vemos que el número de sesiones han sido 26 (sin contar esas 10 sesiones que duraban menos de 10 segundos), tenemos que en todas las sesiones que podemos decir que son valiosas para nosotros, han visto citas de autores. Y que en más o menos la mitad de estas los usuarios han ido a ver las que tenían en favoritos, por lo que realmente es una pantalla y una función utilizada por los usuarios.

Vamos ahora con los eventos.



Figura 1.17 Datos de eventos recogidos gracias a Google Analytics

Hemos recogido un total de 89 eventos y en 36 sesiones se ha detectado al menos uno, habiendo de media unos 2 eventos en cada sesión. Por lo que en todas las sesiones ha habido eventos que nosotros hemos marcado como relevantes. Estos eventos se trataban de, añadir a favoritos una cita, borrar todas o solo una de ellas, y buscar información del autor.

El evento o función más utilizado ha sido el de añadir a favoritos una frase, en total, 40 veces se ha hecho uso de esta función. El resto de eventos no han sido casi utilizados. Igual en aplicaciones tan pequeñas como esta no es tan importante, pero en general, esto puede plantear en un futuro si seguir implementando estas funciones.

Para acabar vamos a ver los usuarios totales que tenemos y con que frecuencia vuelven a nuestra *app*.



Figura 1.18 Información de los usuarios que han usado la *app*

De aquí vamos a centrarnos en dos cosas, la primera es el número total de usuarios. En esta semana hemos conseguido captar un total de 17 usuarios. La segunda es la estadística que aparece a la derecha donde pone que el 58,3% son usuarios recurrentes y el 41,7% son usuarios nuevos ¿Esto qué quiere decir? Que en el total de esas sesiones, algo menos de la mitad son la primera vez que alguien entró en la *app*, el resto son sesiones de usuarios que han vuelto entrar.



Figura 1.19 Datos fidelización de los usuarios

La instancia de sesión hace referencia a las veces que se ha usado la aplicación, para que se entienda mejor, la primera fila representa las sesiones que han supuesto entrar por primera vez a la *app*, la segunda las sesiones que representan entrar por segunda vez a la *app* y así sucesivamente. Como se ve, no se ha conseguido enganchar a los usuarios. El número de sesiones, y por lo tanto, de usuarios que vuelven, se va reduciendo periódicamente.



6. Conclusiones

Al principio de este trabajo nos marcamos una serie de objetivos que pretendíamos alcanzar al finalizar él mismo. Lo que queríamos desde un principio era ir más allá del desarrollo de una aplicación, y esto pasaba por saber cómo funcionan las herramientas de análisis de aplicaciones, queríamos conocerlas más, saber si resultan útiles y que pueden aportar en el ciclo de vida de una *app*, además de conocer cómo es el proceso de su publicación. Para lograr esto teníamos que saber cómo funcionan estas herramientas, cómo usarlas, que información aportan y cómo interpretarla y también que alternativas hay a la hora de publicar una *app*. Para ello, era necesario saber con qué herramienta de análisis íbamos a trabajar y en que *market* iba a estar publicada nuestra aplicación, y lo que hicimos fue una pequeña investigación de las alternativas que existían dentro de ambos, finalmente nos decantamos por *Google Analytics*, de todas las alternativas nos pareció la mejor, sobre todo por sencillez a la hora de trabajar con ella y de implementarla en nuestro código y en cuanto a donde subir la aplicación decidimos hacerlo en la *Google Play* ya que al ser el *market* más usado podíamos llegar a más personas. Una vez decidido todo esto, solo faltaba subir nuestra aplicación y recuperar datos de como los usuarios la usan.

Con todo, y una vez vistos los resultados a la hora de ver el comportamiento de la *app* con usuarios, podemos decir que hemos satisfecho nuestros objetivos. No sólo hemos hecho un repaso a las alternativas a la hora de subir una aplicación y cómo hacerlo, sino que hemos comprendido mejor como funciona *Google Analytics* y qué datos aporta a la hora de valorar si nuestra *app* está funcionando como queremos. Creemos que hemos hecho ver cómo de importante es el proceso que viene después de desarrollar nuestra aplicación, no todo es desarrollo, también es necesario saber que el ciclo de vida de la aplicación no acaba ahí, después viene una parte muy importante que te ayudará a hacer que crezca más y más.

Para ir acabando, comentar que existen formas de ampliar este trabajo. La más interesante que hemos visto es usando y complementando *Google Analytics* con *google AdWords*.

Google AdWords es un programa de publicidad online de Google que permite llegar a clientes nuevos y hacer crecer en este caso, tu aplicación móvil. Con *AdWords* puedes elegir dónde publicar los anuncios, definir un presupuesto adecuado y medir el impacto de estos. Su funcionamiento es el siguiente: Google cobra al cliente por cada clic que se haga sobre su anuncio. En cuanto a su formato, no se admiten imágenes en los anuncios que, como mucho, deben ocupar 4 líneas (25 caracteres en el título, 70 en el texto y 35 en la URL). También eres tu quien especifica el importe máximo que está dispuesto a pagar por cada clic de una determinada palabra, ya que no todas las palabras tienen el mismo precio.

Los anuncios de *AdWords* se muestran junto con los resultados de búsqueda cuando alguien busca en Google utilizando alguna de las palabras clave que has decidido usar. Los anuncios aparecen en la columna lateral de la página de búsqueda, pero también pueden aparecer en posiciones diferentes. De esta forma, nuestros anuncios serán vistos por un público que está interesado en lo que ofrecemos.

Finalmente, Google nos proporcionará una cuenta personal para que podamos hacer un seguimiento de nuestra campaña controlando y gestionando los gastos al mismo tiempo. Además, en nuestro caso podemos enlazar *Google Analytics* con nuestra cuenta de *AdWords*, para poder ver cuantos usuarios hemos captado gracias a esta herramienta.

Por último, como conclusión algo más personal, me gustaría añadir lo importante que es no olvidar los pasos que vienen después del desarrollo, pues son fundamentales y muchas veces se tienden a olvidar o no darles más importancia, cuando realmente resultan tan importantes como el hecho de crear la aplicación. Saber cómo publicar tu aplicación y qué hace falta para hacerlo o tomar decisiones importantes como, ¿en qué países va a estar disponible? o si será de pago o no puede ser clave para que tenga éxito. Pero no solo esto, hacer uso de una herramienta de análisis como *Google Analytics* te aporta información valiosa que bien interpretada, te puede dar las claves para que tu aplicación este constantemente mejorando, y lo más importante, captando usuarios.



Referencias

- [1] Google developers (2016) Google Analytics Academy:
<https://analyticsacademy.withgoogle.com/>

- [2] Google developers (2015) Analytics para Android, Guía:
<https://developers.google.com/analytics/devguides/collection/android/v4/start>

- [3] Oscar G. Peinado (7 de marzo de 2012) 10 herramientas de analítica para apps:
<http://lametrica.com/10-herramientas-de-analitica-para-apps/>

- [4] Eva Gonzales (26 de noviembre de 2013) Un paseo por algunas herramientas de Analytics:
<http://www.analiticaweb.es/herramientas-mobile-analytics-1/>

- [5] Natzir Turrado (9 de enero de 2013) Comparativa de las mejores herramientas para medir apps móviles:
<http://www.doctormetrics.com/2013/01/09/mobile-appanalytics-herramientas/#.VvekkeKLTIV>

- [6] Oriol Claramunt (12 de diciembre de 2014) Principales Markets de Apps y algunas alternativas:
<http://asostats.com/principales-markets-de-apps/>

- [7] Statista (2015) Number of apps available in leading app stores as of July 2015:
<http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/>

- [8] Dinesh Vernekar (6 de febrero de 2015) How to measure retention in mobile apps using:
<https://www.linkedin.com/pulse/how-measure-retention-using-google-analytics-mobile-apps-vernekar>

- [9] Google Analytics:
<https://www.google.es/intl/es/analytics/>