



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

Diseño e implementación de una APP para juego  
colaborativo. Desarrollo del Back-End

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Carlos Francisco Alba Ponce

**Tutor:** Patricio Orlando Letelier Torres

Curso 2015/2016



# Resumen

---

Este Proyecto nace por una inquietud de emprendimiento de dos estudiantes que finalizábamos la carrera. Buscábamos un reto que nos resultara interesante, además de poder conseguir un beneficio por nuestro trabajo realizado.

Decidimos ubicarlo en el ámbito de las aplicaciones móviles, concretamente en el de los juegos. Creímos que este ámbito está en un continuo crecimiento de usuarios y que podríamos sacar adelante un buen proyecto.

Se ha desarrollado en el espacio emprendedor de la ETSINF, llamado START.Inf.

# Tabla de contenidos

---

1. Introducción .....	7
1.1 Objetivos del proyecto .....	7
1.2 Organización .....	8
2. Motivación .....	9
2.1 Juegos en Apps móviles .....	9
2.2 Juegos con multijugador .....	11
2.3 Juegos de cartas .....	11
3. Juego elegido: el Truc .....	12
3.1 Un poco de historia.....	12
3.2 Breve explicación del juego .....	12
4. Estado del arte.....	14
4.1 Otras Apps similares .....	14
4.1.1 Truco Free.....	14
4.1.2 Truco Argentino.....	16
4.1.3 Truco Argentino Free .....	18
4.2 ventaja sobre los competidores.....	18
5. Estudio de la idea .....	22
5.1 Lean Canvas .....	22
5.2 Análisis DAFO .....	24
5.3 Análisis económico.....	25
6. Proceso de desarrollo.....	27
6.1 Mapa de características .....	28
7. Tecnologías utilizadas .....	29
7.1 Android.....	29
Dispositivos móviles y sus características .....	29
¿Qué es Android?.....	29
Un poco de historia .....	30
El problema de la fragmentación .....	30
Cómo es android.....	30
Kernel linux.....	30
Librerías Android .....	31

Runtime de Android.....	31
Entorno de aplicaciones.....	31
Nivel aplicativo.....	32
Inicialización del sistema.....	32
Aplicaciones.....	33
Ciclo de vida.....	33
Los procesos en Android.....	34
Tipos de procesos.....	34
Intents.....	35
7.2 Android Studio.....	36
Proyectos y estructura de archivos.....	36
Android build system.....	38
Depuración y rendimiento.....	38
Android Virtual Device (AVD) manager.....	38
Definiendo layouts.....	39
Android debug monitor.....	40
7.3 Genymotion + virtualbox.....	40
7.4 GitHub.....	41
7.5 AdMob.....	42
¿Por qué admob?.....	42
7.6 Developer Console.....	43
8. Arquitectura de la aplicación: parte del Back-End.....	44
8.1 Introducción a Google Play Game Services.....	44
8.2 Estructura de Google Play Game Services.....	45
8.3 Estructura del juego.....	46
8.4 Funcionamiento del juego.....	49
Reparto de cartas entre los jugadores.....	51
Envío de señas.....	51
Tirada de carta a la mesa y ganador de ronda.....	52
Acción de envidar.....	53
Acción de trucar.....	53
Ganador de la partida.....	54
Desconexión de un jugador.....	55
Cambio de turno y de mano.....	55
Envío de frases.....	55
Variables de la lógica del juego.....	56



8.5 Logros .....	56
8.6 Ranking de jugadores .....	60
8.7 Jugar con amigos.....	61
8.8 Motivo de la elección y evaluación .....	64
9. Publicación en play store.....	66
9.1 ¿Qué es play store? .....	66
9.2 Darse de alta como desarrollador .....	66
9.3 Cómo publicar una aplicación .....	67
9.4 Precio de la publicación .....	71
10. Resultados obtenidos .....	72
10.1 Descargas.....	72
10.2 Usuarios de la App .....	74
10.3 Opiniones de los jugadores en Play Store.....	74
11 Trabajo futuro .....	78
11.1 Tareas pendientes.....	78
11.2 Posibles mejoras.....	79
11.3 Expansión a otros países .....	79
12. Conclusiones .....	80
12.1 Resultados del proyecto.....	80
12.2 Aprendizaje .....	80
12.3 Opinión personal .....	81
13. Bibliografía.....	82

## 1. Introducción



Para este proyecto, se ha decidido implementar una aplicación para dispositivos Android que permita jugar al popular juego de cartas valenciano: El Truc.

Tal y como puede apreciarse en la imagen, el título del juego es: Retruque!

El motivo de la elección es que se buscaba algún título que llamara la atención, que no fuera simple y que pudiera resultar atractivo para el jugador. En el Truc hay una acción llamada Retrucar que, conjugada en primera persona y traducida al valenciano, adquiere el nombre de Retruque!

Hoy en día, la mayor parte de los dispositivos móviles usan el sistema operativo Android, por lo que decidimos elegirlo para llevar a cabo nuestra aplicación, ya que adaptarnos a este sistema nos iba a resultar mucho más sencillo que a cualquier otro.

Para desarrollar en Android se usa el lenguaje de programación Java, el cual nos han enseñado en gran parte durante el grado que hemos cursado y, por tanto, teníamos una gran base para empezar a trabajar.

Por último, cabe citar que este proyecto ha sido desarrollado junto a Guillermo Domingo Zaragoza, que se encargará, en mayor medida, de la parte del Front-End.

Así pues, en éste se profundizó más en el Back-End de la aplicación.

### 1.1 Objetivos del proyecto

Este proyecto era un reto muy interesante para nosotros, al que sabíamos que tendríamos que dedicar un gran esfuerzo y muchas horas de trabajo, pero que a la vez nos hacía mucha ilusión sacar adelante.

Los objetivos a cumplir se describen a continuación:

- El objetivo básico era tener la App publicada en Google Play, para que cualquier persona pudiera descargarla y divertirse jugando un par de partidas con el trabajo que habíamos realizado.

- El segundo objetivo es, obviamente, poder obtener una remuneración económica. Somos conscientes de que es difícil llegar a expandir una App sin ser conocidos ni haber hecho otros trabajos anteriores, pero siempre creímos que si hacíamos un juego divertido podría llegar a expandirse y triunfar, aunque fuera en la Comunidad Valenciana (dónde es más típico el juego).

## 1.2 Organización

Al ser dos personas, era muy importante planificar bien las tareas del proyecto y repartirlas equitativamente. Había que organizarse bien e ir al día con el trabajo del compañero, para ir avanzando paralelamente hacia los distintos objetivos del proyecto.

Con el objetivo de seguir esta organización, decidimos marcarnos un horario y trabajar juntos en un mismo sitio, de tal forma que pudiéramos comunicarnos y ayudarnos mutuamente de manera fácil y rápida.

Una cosa que nos fue muy útil es tener planificado el proyecto con bastante antelación. Es decir, cuando empezamos a desarrollar el juego, ya habíamos estructurado previamente y en acuerdo junto con nuestro profesor los que serían nuestros objetivos principales y las características que iba a tener nuestro juego.

Las tecnologías que hemos utilizado en el proyecto nos han facilitado mucho la necesidad de trabajar en paralelo, pero ese es un punto que trataremos en puntos posteriores de esta memoria.



## 2. Motivación

El Truc es un juego típico de Valencia y pensamos que había llegado el momento de digitalizarlo y hacer que cualquier persona pudiera divertirse jugando una partida desde cualquier lugar y en cualquier momento en que le apeteciera.

Montar una partida de Truc no es tan sencillo, ya que lo más típico es juntar a cuatro jugadores y si a eso se le añade el tener que quedar en algún sitio físico, hace más complicado el poder jugar cuando realmente te apetece. Con nuestra aplicación, este problema se soluciona de manera rápida, ya que es posible jugar con tus amigos o con jugadores aleatorios que estén conectados. Siempre va a haber alguien listo para disfrutar de una partida de Truc junto a ti.

Por otra parte, tras analizar el mercado de aplicaciones, nos dimos cuenta de que nadie había creado una aplicación como la que habíamos pensado: no existía una aplicación para jugar al Truc con las reglas de Valencia. Vimos una gran oportunidad de negocio al comprobar esto.

Sí que existían, sin embargo, otras aplicaciones para jugar con otras reglas, ya que también es un juego al que se juega mucho en Sudamérica y que es llamado Truco.

Además, toda la gente que escuchaba nuestra idea mostraba un gran interés y decía que le gustaría probar nuestra aplicación, por lo que esto nos dio una motivación extra: darnos a conocer entre la gente y hacer que se divirtieran con algo que habíamos creado nosotros mismos.

Como extra, y con el objetivo de seguir fomentando el origen del juego, decidimos traducirlo completamente al valenciano, por lo que si un jugador tiene configurado su dispositivo móvil en catalán (Android no tiene el idioma valenciano, sólo el catalán) podrá disfrutar del juego en su idioma original.

### 2.1 Juegos en Apps móviles

Una de las cosas que más triunfan hoy en día en las tecnologías móviles son las aplicaciones de juegos. ¿Quién no ha jugado nunca a un juego que tiene descargado en el móvil en algún momento de aburrimiento? Ya sea de camino al trabajo mientras vas en el metro o en un momento de relax en el sofá de casa. Siempre resulta divertido tener un par de juegos instalados en tu dispositivo para despejarse un poco en un momento determinado. Ahí es donde entra en escena el éxito de las aplicaciones de juegos móviles.

Generalmente, un usuario juega una partida de vez en cuando, durante un periodo de tiempo medianamente corto, por lo que existe un gran reto para el desarrollador: ganarse al jugador en un periodo de tiempo muy corto, puede que incluso sea necesario ganarlo en la primera partida. De lo contrario, si el jugador no se ve sorprendido puede decidir desinstalar la App y buscar otra que le divierta más.

Sea como sea, hay una cosa clara: las App de juegos móviles son muy populares en la actualidad y son una gran manera de poder ganar dinero y de darse a conocer entre los jugadores. Una gran App puede servir para que un jugador se baje tu próxima App, para ver si le gusta tanto como la anterior. Es importante desarrollar un producto de calidad.

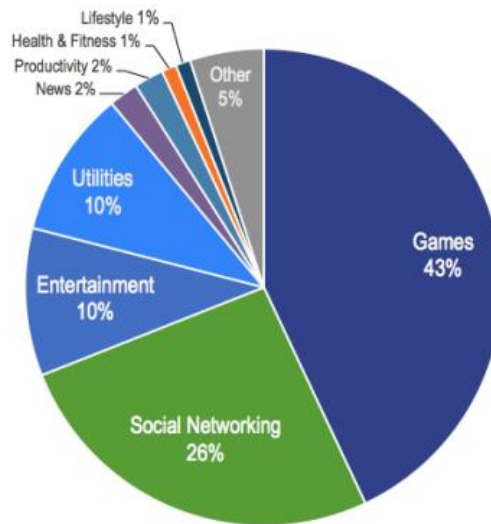
Si miramos un par de años atrás, no debemos hacer mucho esfuerzo para recordar algunas Apps de juegos como Candy Crash, Andry Birds o Flappy Bird.

Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

Apps que consiguieron millones y millones de descargas y que provocan que los jugadores descarguen los nuevos juegos de los desarrolladores.

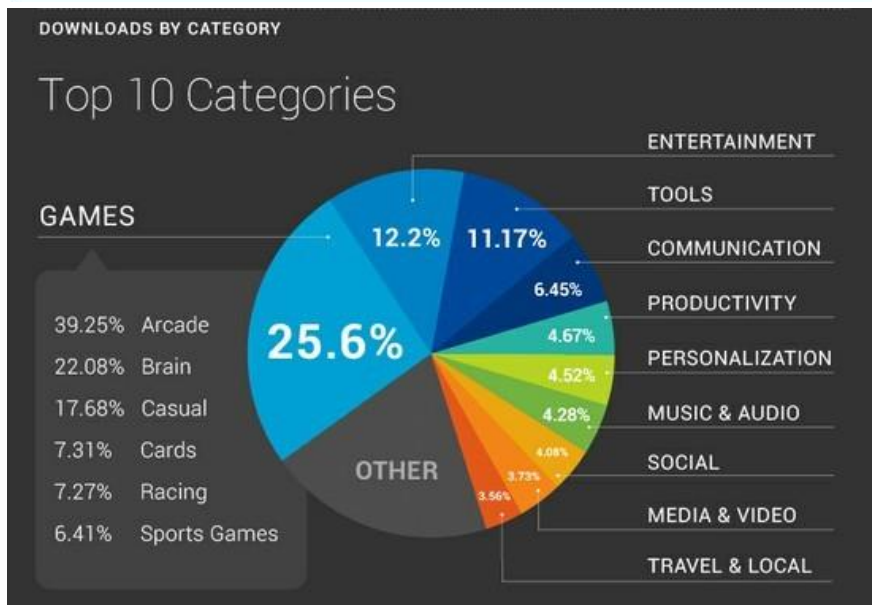
Es obvio que los anteriores son juegos mundialmente conocidos, y que es prácticamente imposible diseñar un juego que llegue a ese nivel de éxito pero, ¿por qué no? Todo es intentarlo.

WW iOS & Android Smart Device Time Spent per App Category



© FLURRY

Source: Flurry Analytics, November 2012



Como podemos ver en los dos anteriores gráficos, los juegos son las Apps más descargadas entre todos los tipos disponibles. En el primero, los datos son de todas las plataformas y, en el segundo, de dispositivos Android.

Además, de entre el 25,6% de descargas de juegos, el 7,31% son juegos de cartas, porcentaje mayor incluso que otras categorías importantes como música, redes sociales o comunicación.



## 2.2 Juegos con multijugador

Expandiendo un poco más el apartado anterior, entramos en los juegos que incluyen un modo multijugador. Si un juego resulta divertido de por sí, ¿qué pasa si incluye un modo multijugador y te permite disfrutar de una partida con tus amigos o con quién quieras, estés donde estés?

La diversión del juego aumenta notablemente, y es mucho más fácil que algún jugador desafíe a sus conocidos y les incite a descargarlo. Por decirlo de alguna manera, es una forma de propiciar la expansión del juego y atraer a un número mayor de jugadores.

Gracias a las conexiones de datos de las que disponen prácticamente la totalidad de los dispositivos móviles de hoy en día, un usuario puede jugar una partida de un juego contra un amigo que viva en la otra parte del mundo con un par de acciones y en tan sólo unos segundos. ¿Y si además se le permite al jugador buscar partidas contra jugadores aleatorios que estén conectados al juego? Siempre habrá alguien dispuesto a echar una partida. Resulta muchísimo más divertido que jugar una partida contra la máquina.

Podemos concluir tras esta breve explicación que la inclusión de un modo multijugador en un juego para dispositivos móviles es totalmente positivo, pudiendo aumentar notablemente el número de usuarios que decidan probarlo, la expansión de la App y como no, la diversión.

## 2.3 Juegos de cartas



Para concluir este apartado de la memoria hablaremos sobre una modalidad específica de los juegos móviles: los juegos de cartas.

Jugar a las cartas siempre ha sido muy popular para pasar los ratos muertos. Una baraja, una mesa y un par de amigos bastaban para tener un buen rato de diversión. Ahora, gracias a las nuevas tecnologías, podemos disfrutar de la mayoría de los juegos de cartas desde cualquier sitio, simplemente instalando una App en nuestro dispositivo móvil.

Nunca había sido tan fácil desafiar a tus amigos a una partida de Brisca, Cinquillo, etc.

Tras investigar detenidamente el mercado de aplicaciones nos dimos cuenta de que faltaba uno de los más populares, sobretodo en la Comunidad Valenciana: El Truc. Como citamos en puntos anteriores de la memoria, esta fue una de las causas que nos ayudó a decidirnos por este juego.

Con el fin de profundizar un poco más acerca de El Truc, veamos el próximo apartado, en el cual nos dedicaremos a conocer mejor nuestro juego de cartas elegido.

### 3. Juego elegido: el Truc

En este apartado de la memoria conoceremos un poco más nuestro juego elegido. Para meternos en contexto, hablaremos un poco sobre su historia, sus reglas (incluyendo las de nuestra App) y por último, profundizaremos un poco más en el motivo de nuestra elección.

#### 3.1 Un poco de historia

El juego del truco o truque, denominado “el truc” en valenciano, remonta sus orígenes al mundo árabe o musulmán, adquiriendo no obstante en la Comunidad Valenciana una enorme importancia, al estar arraigado en todas sus comarcas y poblaciones, hasta el punto de ser, posiblemente, el juego de cartas del que más uso hacen los valencianos. Tras cada celebración o comida entre amigos, en bares o en “casals” falleros, resulta casi obligado acabar con una buena partida de “truc”. Las habilidades, la forma y estrategias personales de juego se transmiten oralmente de padres a hijos, de abuelos a nietos o entre amigos. Se trata siempre de ser el mejor jugador con una estrategia de juego bien definida, en donde, además de la suerte propia de los juegos de azar y de una buena técnica en cada jugada, tiene especial importancia el uso de la mentira para generar confusión o llevar a engaño al contrario, obteniendo a cambio la correspondiente ventaja.

#### 3.2 Breve explicación del juego

Si bien existen diversas versiones del juego del “truc” en su versión valenciana, en esta aplicación hemos optado por el uso más extendido del mismo, en base a veintidós cartas, despreciando los doses, ochos, nueves, sotas, caballos y reyes de cada uno de los cuatro palos de la baraja española (Oros, copas, bastos y espadas), así como el as de oros y el de copas. La aplicación, en definitiva, se estructura en base al uso de las siguientes cartas:

Oros: Del tres al siete de oros.

Copas: Del tres a siete de copas.

Bastos: Del tres al siete de bastos y el as de bastos.

Espadas: Del tres al siete de espadas y el as de espadas.

Siendo las cartas de mayor valor en este juego, por su orden: el as de espadas, el as de bastos, el siete de espadas y el siete de oros (conocidas éstas dos cartas como “manillas”), siguiéndoles en cuanto a valor: los cuatro treses de cada uno de los palos y a continuación cada una de las cartas por su valor numérico.

Pero en el juego que examinamos no sólo se hace uso de las cartas, sino que son importantes además los gestos y la palabra, dado que cada una de las cartas importantes disponen de nombre y seña propias, siendo fundamental la comunicación entre jugadores y el traslado de las señas al compañero sin que puedan ser captadas por el rival. Bien para darle a la pareja una información de las cartas que se poseen, a efectos de efectuar una correcta estrategia de juego, o bien, para confundir al contrario, realizando señas de cartas que realmente no se poseen, induciendo con ello a confusión al mismo, a efectos de obtener, en uno y otro caso, la correspondiente ventaja en el juego. Hemos trasladado a la aplicación no sólo las cartas con sus reglas de juego, sino que además, hemos incorporado, los correspondiente iconos de señas identificativos de las cartas importantes del “truc”, tratando de hacer la versión virtual lo más próxima posible a la

real, para que todo jugador avisado pueda obtener la correspondiente ventaja sobre su contrario. Incluso en la aplicación hemos introducido un icono que permite cazar y descubrir las señas del contrario, si bien es cierto que para ello hay que tener una especial habilidad y rapidez.

A su vez se han incorporado a la aplicación de las dos formas de jugarse los puntos (también denominados piedras) denominadas “envit” o envite, con sus correspondientes alternativas: “torne el envit” o devuelvo el envite, “envide fins igualar” o envido hasta igualar y la tan temida falta. Y el “truque” o truco, con sus alternativas de “retruque” o retruco, “quatre val” o vale cuatro, y “joc fora” o juego fuera. Y todo ello con la correspondiente valoración en puntos o piedras de cada una de las jugadas y su resultado para el ganador, cuya aplicación es automática al final de cada ronda hasta completar “la cama” o juego parcial, distribuida en dos fases, la de “a males” y “a bones” , esto es, a malas y a buenas, compuesta cada una de doce puntos o piedras. El jugador o jugadores que primero completan dos camas (una sólo en nuestra App) son los ganadores de la partida.

Cabe citar que el Truc también es muy popular en otros países, sobretodo en Sudamérica, donde tienen distintas reglas según el país (se juega con toda la baraja, hay otras modalidades de jugarse los puntos, etc.) y es conocido como Truco.



## 4. Estado del arte



En este punto 4, vamos a hacer un estudio de mercado de distintas Apps de juegos de cartas, así como de las características de nuestra aplicación, y seguidamente haremos una comparación para ver los puntos a favor y en contra que tiene nuestro juego respecto a los competidores.

Como ya se ha dicho anteriormente, no hay ninguna App que permita jugar al Truco como la nuestra, pero sí que existen otros juegos para jugar al Truco (la versión Sudamericana). Estas serán sobre las que haremos un pequeño análisis.

### 4.1 Otras Apps similares

#### 4.1.1 Truco Free

El primer juego que analizaremos se llama Truco Free. Un rival sin duda muy potente que cuenta con más de 5 millones de descargas. Desarrollado por Blyts, una empresa que se dedica a desarrollar Apps de juegos, y que tiene actualmente 15 juegos disponibles para descargas en App Store para iOS.







Truco Free no especifica la forma de jugar del país que adquiere, simplemente dice que es la de América Latina. En el menú principal encontramos varias opciones:

- **Jugar.** Nos permite jugar una partida contra la máquina o buscar una partida online. Si queremos jugar online nos dice que debemos identificarnos y nos da tres opciones: mediante Facebook, mediante un email o ser un jugador Invitado. Cuando entramos en el modo multijugador nos deja buscar una partida rápida, meternos en una sala y ir, consultar las posiciones de los jugadores siguiendo un ranking (semanal o histórico), que se basa en puntos ganados o consultar nuestra lista de amigos.

Nos deja jugar una partida online contra un sólo jugador, no por parejas.

- **Perfil.** Podemos gestionar nuestro perfil del juego. Podemos introducir nuestro apodo, nuestro sexo, si queremos flor o no (debe ser una regla de países sudamericanos), los puntos con los que se gana una partida, si activamos el chat o no y si queremos que el juego vaya rápido o lento. Además, podemos elegir entre varias voces, que se reproducen durante una partida cuando realizamos acciones.
- **Ayuda.** En este apartado nos encontramos con una explicación de las reglas del Truc y de cómo ganar puntos en el juego.
- **Anotador.** Si jugamos una partida física con jugadores reales, podemos entrar aquí e ir llevando la cuenta de puntos de forma sencilla. No tiene nada que ver con el juego en sí, es una funcionalidad aparte.

Como vemos en las capturas, el juego nos muestra publicidad continuamente en la parte inferior de la pantalla mientras estamos jugando, de esta forma gana dinero.

También vemos que tiene un sistema de chat para comunicarte con el otro jugador mientras juegas una partida. Esto puede ser interesante para comunicarte con tu pareja de juego si las incluyera.

Cuando hemos entrado varias veces al juego, nos aparece una ventana que nos dice si nos gustaría valorar el juego en App Store, esto es interesante para ganar puntos a favor.

La interfaz gráfica es bastante buena y hay buenas animaciones cuando eliges una carta.

Este es un juego bastante completo y lo hemos usado para sacar ideas para nuestro propio juego.

#### 4.1.2 Truco Argentino

Seguimos analizando la competencia directa con el Truco Argentino, aunque este no alcanza el nivel del anterior. Cuenta con más de 100,000 descargas y ha sido desarrollado por un hombre llamado Ignacio Raffa, que cuenta con 4 Apps más subidas.







El Truco Argentino nos presenta una interfaz gráfica muy pobre en la que incluso se tapa su propio logo con la publicidad.

En el menú principal tenemos 3 botones en rojo que son para jugar: Jugar a 15 y Jugar a 30 para jugar una partida contra la máquina a 15 o 30 puntos y jugar online para jugar una partida en línea contra otro jugador. Además, encontramos los botones Ayuda, que nos muestra como manejar el juego (ni siquiera las instrucciones del truco) y Acerca De que nos muestra quien ha desarrollado el juego.

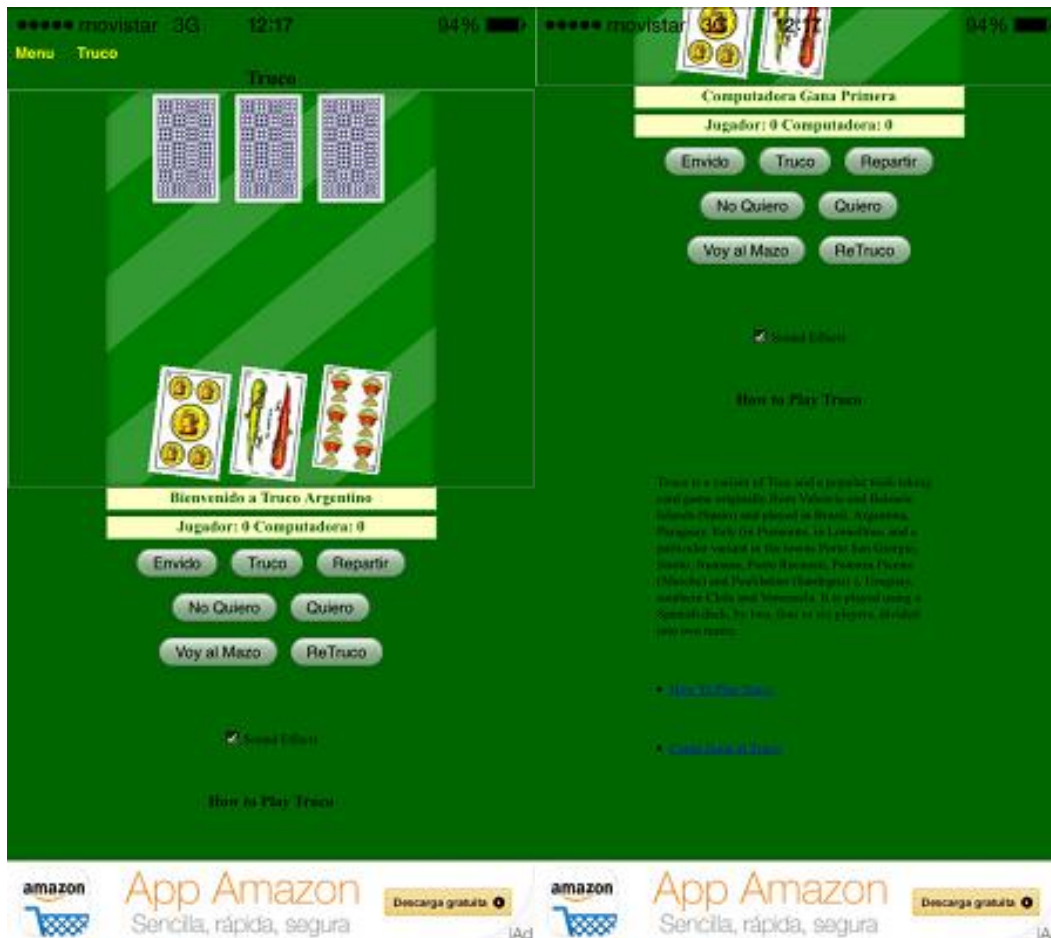
Si elegimos jugar online entramos en el menú de la última captura, donde nos asigna un número de usuario y nuestras estadísticas. Si elegimos jugar ahora nos empareja con otro usuario y empezamos una partida online. Si elegimos jugar con amigo nos pide que introduzcamos su identificador para poder conectar con él.

La publicidad sigue apareciendo por todo el juego, tapándonos parte de las cartas incluso.

En definitiva, un juego bastante pobre si lo comparamos con el anterior.

### 4.1.3 Truco Argentino Free

Por último, veremos el Truco Argentino Free, el menos trabajado de los tres. Desarrollado por Jaime García Ghirelli y sin datos de descargas. Disponible en App Store.



El Truco Argentino Free es un juego para un único jugador en el que juegas contra la máquina una partida de Truc con las reglas de Argentina. Como podemos ver en las capturas, la interfaz gráfica es muy deficiente y todos los botones están activos para ser pulsados siempre, de tal forma que nunca sabes cuando puedes pulsarlos o cuando no tiene sentido pulsarlos durante la partida.

Además, las instrucciones del juego se encuentran en la parte inferior de la pantalla, bajando con el dedo, pudiendo ocultar en la pantalla la partida y pararte a verlas.

No tiene modo online ni ranking, únicamente puedes jugar una partida contra la máquina.

## 4.2 ventaja sobre los competidores

De estos tres competidores que hemos analizado, es obvio que el más potente es el de punto 4.1.1, el Truco Free. Es un juego muy completo del que hemos sacado ideas.

A continuación vamos a hacer una comparación de nuestra aplicación y la de estos competidores. Analizaremos si tenemos sus características (tick verde o cruz roja) y la ventaja que tenemos sobre ellos. Pero cabe destacar que tenemos una ventaja exclusiva

sobre todos que nos coloca por encima: nuestro juego es el único que tiene modo multijugador online para cuatro jugadores.

### **Retruque! vs Truco Free**

Menú principal ✓	Modo multijugador en línea para 2 jugadores ✓	Ranking ✓
Amigos en el juego ✓	Contador de puntos en partida ✓	Personajes ✗
Jugar contra la máquina ✗	Señas en modo multijugador ✓	Ayuda ✗
Gráficos HD ✓	Animaciones en la partida ✓	Sonidos ✓
Anotador fuera de partida ✗	Partida rápida contra jugador aleatorio ✓	Publicidad ✓

Como podemos ver arriba, Retruque! Tiene la mayoría de características que posee nuestro competidor más potente, por lo que podemos deducir con este primer análisis que, como mínimo, estamos muy cerca de su nivel.

### **Retruque! vs Truco Argentino**

Menú principal ✓	Sonidos en la App ✓	Marcador de puntos ✓
Acciones del juego (trucar, etc.) ✓	Ayuda ✗	Publicidad ✓
Jugar con amigos ✓		

En comparación con el Truco Argentino, podemos observar que tenemos prácticamente la totalidad de sus estadísticas, exceptuando la Ayuda que también tiene Truco Free. Nuestra App también es superior a esta.

### **Retruque! vs Truco Argentino Free**

Acciones del juego (trucar, etc.) ✓	Sonidos en el juego ✓	Reglas del Truc ✗
Marcador de puntos ✓		

Por último, comparamos con el Truco Argentino Free. No debemos de mirar mucho para darnos cuenta de que es el producto menos trabajado de los tres competidores y que no tiene muchas características. Nuestro juego vuelve a tener prácticamente la totalidad de estas características.

Tras analizar estos tres competidores, vamos a ver que en nuestro juego tenemos unas características que van a hacer que los superemos:

- **Multijugador online para 4 jugadores.** Esta, sin duda, es nuestra característica más relevante, ya que no hemos encontrado ningún juego que la tenga. Nos coloca directamente por encima de cualquier competidor al ser una característica única.
- **Sistema de logros.** No hay nada mejor para enganchar al jugador que desafiarle a que consiga ciertos objetivos del juego. Nuestra aplicación contiene 15 logros (unos más fáciles de conseguir que otros) para que el jugador ponga a prueba su capacidad e intente demostrar que es el mejor.
- **Tour inicial.** Cuando entras por primera vez al juego, éste muestra un pequeño tour inicial indicando al jugador qué acciones puede realizar desde el menú principal y le invita a que envíe una invitación a un amigo, por ejemplo.



- **Búsqueda de jugadores cercanos.** ¿No sabes con quién jugar y te apetece jugar con alguien de tu barrio? Retruque! Tiene un buscador de jugadores cercanos para que puedas desafiar a tus vecinos.
- **Perfil real de usuario.** Como hemos visto en las características de Truco Free, éste tiene un personaje ficticio que te representa en la partida. Nuestra aplicación usa tu perfil real de Google+ y muestra tu nombre y tu foto en la partida. Estos datos son únicamente mostrados cuando juegas con jugadores conocidos, para los demás jugadores aleatorios tu perfil queda oculto bajo el nombre de Jugador XXXX y una foto genérica.
- **Frases en la partida.** ¿Quieres picar un poco a los demás jugadores? Tenemos más de 15 divertidas frases para añadir algo de picante a la partida. Puedes decirlas cuando sea tu turno y todos los demás jugadores verán un bocadillo saliendo de tu icono de jugador junto con tu frase. ¿Qué te contestarán los otros jugadores?
- **Publicidad Interstitial.** Es la publicidad por la que más se paga. No queríamos molestar al jugador con los típicos banners apareciendo en mitad de una partida, ya que pensamos que podría resultar incluso molesto. Nuestra estrategia de publicidad es la llamada Interstitial, que consiste en una publicidad que se muestra al usuario a pantalla completa durante unos pocos segundos (5 en nuestro caso) y que puede ser cerrada en cuanto haya transcurrido este tiempo. En nuestro juego, se muestra una vez se ha acabado la partida, en la pantalla de resultados, por lo que el jugador no se verá molestado por publicidad durante el transcurso de la partida.
- **Menú desplegable de acciones para el jugador.** Es molesto tener siempre los botones de acciones a la vista del jugador y ocupando espacio en la pantalla. Retruque! Tiene un menú desplegable con las acciones disponibles que el jugador podrá elegir si abrirlo o no conforme a sus necesidades en la partida.
- **Posibilidad de tapar la carta.** Una de las posibilidades del Truco es la de tapar tu carta durante tu turno. Nuestro juego es el único que la tiene.
- **Diálogos con material design para preguntas al jugador.** Cuando un jugador envidia o truca, los demás jugadores verán un moderno diálogo preguntando sobre qué acción desean realizar. Además, se muestra en la parte superior de la pantalla para que el jugador pueda ver sus cartas y, de esta forma, tomar mejor su decisión. En los competidores son pequeños botones sin mayor relevancia.
- **Tiempo para los turnos visible.** Cada jugador tiene 40 segundos en su turno que puede usar para decidir que carta tirar o realizar alguna de las acciones que le estén permitidas. Debajo de su icono de perfil aparecerá una barra de tiempo que permitirá a él mismo y a los demás jugadores tener la información del tiempo restante que le queda a ese jugador. En caso de que el jugador que tiene el turno no haya tirado ninguna carta durante su tiempo, el juego tirará una automáticamente por él para que la partida pueda seguir avanzando.
- **Partidas mixtas.** Es posible jugar partidas mixtas con amigos y con jugadores aleatorios. Por ejemplo, se puede invitar a un amigo y buscar a otros dos jugadores aleatorios para así jugar una partida de cuatro jugadores. Retruque! Es el único juego que permite esta acción.

- **Fondos de partida dinámicos.** Jugar una partida siempre con el mismo fondo puede llegar a resultar monótono. Nuestra aplicación elige un fondo aleatorio de entre varios disponibles para cada partida, aportando así un toque distinto y novedoso.

Éstas son las características principales que nos diferencian sobre nuestros competidores. Creemos que ha sido positivo analizarlos a fondo para así tomar ideas y ver cómo era posible superarlos.

Este análisis nos ha hecho ver que todos disponen de un apartado de Ayuda que permite que jugadores que no saben jugar al Truc puedan ver las reglas y, de ese modo, poder aprender mientras disfrutan del juego. Lo tendremos en cuenta para añadirlo en próximas versiones.

Sin lugar a dudas, nuevas ideas y características incluidas en Retruque! nos sitúan por encima de nuestros competidores de forma notable. Destacando, como ya se ha citado anteriormente, el exclusivo modo de 4 jugadores en línea que posee nuestro juego. Si a eso le añadimos que somos el único juego que contiene las reglas del Truc valenciano (todos los productos usan las reglas de la versión Sudamericana), podemos esperar un gran éxito del juego en nuestra zona geográfica.



## 5. Estudio de la idea



Llegamos al punto 5 de la memoria, en el que vamos a hacer un estudio de la idea de proyecto que tuvimos, siguiendo una serie de análisis. Empezaremos por el Lean Canvas, seguido del DAFO y analizaremos la proyección económica de la App para finalizar.

Para concretar un poco más sobre los tres análisis, vamos a definirlos brevemente:

- **Lean Canvas:** Lean Canvas es una herramienta rápida y eficaz para crear y comunicar modelos de negocio para startups. Su creador, Ash Maurya, te propone un método híbrido basado en Canvas Business Model y la metodología Lean, de la adaptación surge una herramienta específica para que las startups puedan diseñar modelos de negocio de la forma más eficaz. Lean Canvas ayuda a descomponer tu modelo de negocio en nueve partes que luego serán sistemáticamente sometidas a pruebas en orden de menor a mayor riesgo.
- **Análisis DAFO:** Es una metodología de estudio de la situación de una empresa o un proyecto, analizando sus características internas (Debilidades y Fortalezas) y su situación externa (Amenazas y Oportunidades) en una matriz cuadrada. Es una herramienta para conocer la situación real en que se encuentra una organización, empresa o proyecto, y planear una estrategia de futuro. El objetivo del análisis DAFO es determinar las ventajas competitivas de la empresa bajo análisis y la estrategia genérica a emplear por la misma que más le convenga en función de sus características propias y de las del mercado en que se mueve.
- **Proyección económica:** Un análisis de dos puntos: ingresos y tiempo. Su principal objetivo es estimar en un periodo de tiempo determinado, cuáles pueden ser los ingresos del proyecto y si puede salir rentable, así como prevenir el crecimiento o decrecimiento de los ingresos que pueda aparecer en un momento del periodo de vida.

### 5.1 Lean Canvas

A continuación, veremos la tabla del Lean Canvas.



<p><b>2</b> <b>Problema</b></p> <ul style="list-style-type: none"> <li>- En la actualidad no existe un juego para poder jugar al truco online para Android. Al menos de la versión española, si que existen versiones de países sudamericanos con distintas reglas de juego.</li> <li>- Hay muchos aficionados al truco a los que le gustaría jugar una partida en un rato libre y no pueden, ya que hace falta reunir varios jugadores.</li> <li>- A muchos jugadores o incluso parejas de jugadores les atrae la posibilidad de demostrar que son mejores que otros al truco.</li> </ul>	<p><b>4</b> <b>Solución</b></p> <ul style="list-style-type: none"> <li>- Modo online para jugar desde cualquier parte.</li> <li>- Posibilidad de invitar a amigos y mandarles notificaciones para poder jugar una partida online.</li> <li>- Un sistema de estadísticas o de clasificación según partidas ganadas o perdidas podría crear un ranking de mejores jugadores del juego.</li> </ul>	<p><b>3</b> <b>Proposición única de valor</b></p> <p>Ofrecemos la posibilidad de que los jugadores del truco online disfruten de una partida de truco con sus amigos o con otros jugadores desconocidos en el momento que quieran, con la comodidad de no tener que reunirse físicamente con los demás jugadores y desde el sitio que quieran, simplemente con una conexión a internet. También crearemos un ranking para poder comparar al jugador con sus amigos y demostrar quien es el mejor jugando al truco.</p>	<p><b>9</b> <b>Ventaja injusta</b></p> <p>Existen versiones del truco de países sudamericanos, no podríamos copiarlos el mismo juego y cambiando las reglas. Necesitamos intentar innovar en cuanto a funcionalidades para competir.</p>	<p><b>1</b> <b>Segmentos de Clientes</b></p> <p><b>Clientes objetivo:</b> Aficionados a jugar al truco. Españoles y especialmente valencianos. No hay una edad determinada, ya que los juegos de cartas tienen un rango de edad de los jugadores muy amplio (podríamos estar hablando desde los 14 hasta los 60 años aproximadamente).</p>
	<p><b>8</b> <b>Métricas clave</b></p> <p>Algunas posibles métricas:</p> <ul style="list-style-type: none"> <li>- Número de partidas semanales.</li> <li>- Número de descargas mensuales.</li> <li>- Número de jugadores semanales.</li> </ul>		<p><b>5</b> <b>Canales</b></p> <p>Nuestro canal de distribución para los clientes será la tienda de aplicaciones de Android: Google Play.</p>	<p><b>Early adopters:</b> Tenemos a nuestra disposición a un grupo de amigos a los que les gusta jugar al truco y que podrían ser nuestros primeros jugadores que prueben las distintas versiones que vayamos diseñando.</p>

<h2 style="margin: 0;">7 Estructura de costes</h2> <p>Como costes fijos tendríamos el coste de subir la aplicación a Google Play y la tarifa mensual de la contratación de un servidor donde poder alojar las partidas online.</p>	<h2 style="margin: 0;">6 Ingresos</h2> <p>Nuestra fuente de ingresos será la publicidad que pondremos en el juego. También cabe la posibilidad de ofrecer funcionalidades especiales a usuarios Premium o de ofrecer el mismo juego sin publicidad pero pagando una cantidad por la descarga inicial. Además, en el juego podemos incluir monedas que apostarás para poder jugar partidas. Esas monedas o algunas mejoras como el diseño de las cartas podrían comprarse con dinero real.</p>
--	---

## 5.2 Análisis DAFO

Aquí tenemos la primera parte del análisis DAFO, que está dividida en cuatro puntos.

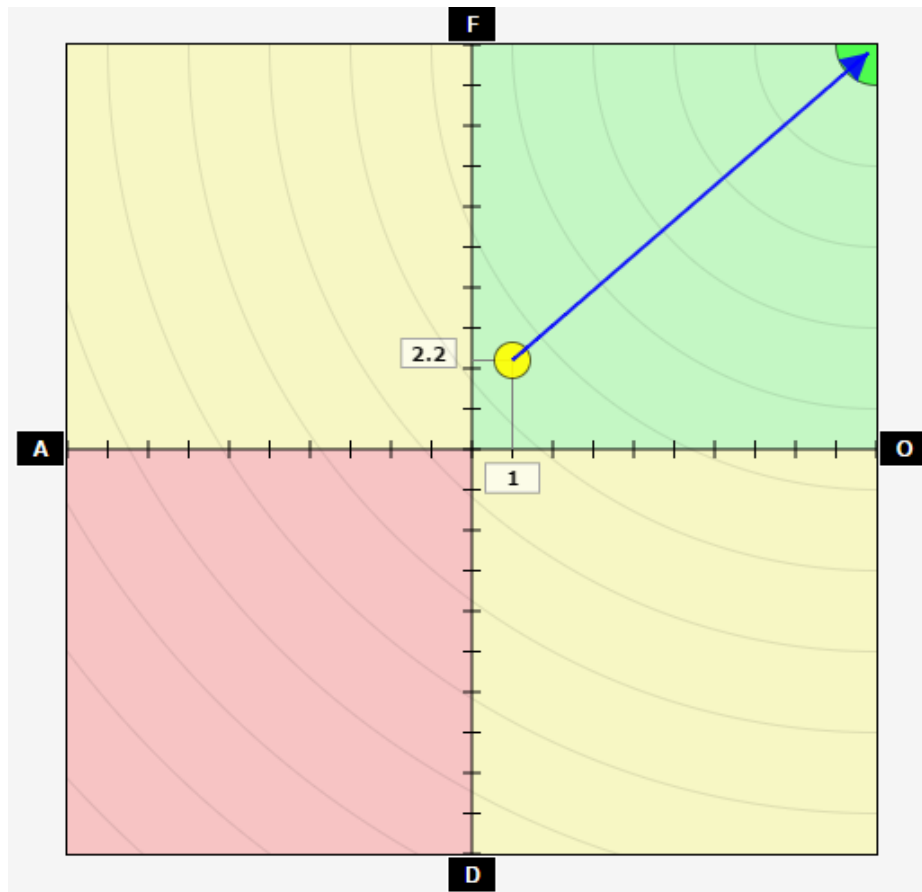
Fortalezas	Peso	Debilidades	Peso
Originalidad de Productos/Servicios	8	Marketing Efectivo	5
Precio Competitivo	10	Base de Clientes	5
Innovación de Productos/Servicios	10	Alianzas de Negocios	3
Ubicación Geográfica	9	Presencia Online	3
Liderazgo en el Mercado	7	Reputación	3
Calidad de Productos/Servicios	6	Relación con los Proveedores	2
Cuota del Mercado	6	Costos	2
Plantas y Equipamiento	4	Estado Financiero	2
Posicionamiento en el Mercado	4	Cadena de Distribución	1
Personal Experimentado	4		
Propiedad de Patentes/Tecnologías	2		
<b>Total</b>	<b>70</b>	<b>Total</b>	<b>26</b>
Oportunidades	Peso	Amenazas	Peso
Mayor utilización de Internet por los Consumidores	5	Adquisición de Empresa Proveedora de Tecnología	3
<b>Total</b>	<b>5</b>	<b>Total</b>	<b>3</b>

Como vemos, la mayoría de puntos asignados a nuestro proyecto son para las fortalezas, ya que como hemos comentado anteriormente, es un producto totalmente novedoso, sin competidores directos y con una gran esperanza de éxito.

Por otra parte, también tenemos debilidades, como no. Nuestras principales debilidades son que no somos desarrolladores conocidos ni hemos llevado otros proyectos por lo que puede costar el hecho de expandirnos y dar a conocer nuestra aplicación. Tampoco tenemos una base de clientes determinada más allá de la gente cercana con la que nos podamos comunicar. Además, cualquier coste del proyecto ha salido directamente de nuestro bolsillo, por muy pequeño que sea.

En cuanto a las oportunidades y amenazas, no tenemos ningún punto destacado que comentar.





En el grafo que hemos generado, podemos observar como nuestro proyecto se ubica completamente en el cuarto de las fortalezas, por todo lo que hemos comentado anteriormente.

Por ello, de este análisis podemos deducir que, aparentemente, nuestro proyecto tendrá éxito.

### 5.3 Análisis económico

Para acabar con el punto 5, efectuaremos un análisis económico de nuestro producto.

Comenzaremos con el análisis económico del primer año de vida de la aplicación. En este primer periodo nos encontramos con una aplicación nueva, que no obtendrá un gran número de descargas y que tendrá una inversión inicial.

Los únicos beneficios que obtendremos serán de la publicidad y de las descargas que obtenga nuestra aplicación, por lo que será esencial la expansión de esta para recuperar gastos y obtener algo de beneficio.

Cabe destacar que tanto los gastos como los beneficios serán repartidos entre dos personas, al ser un proyecto conjunto.

Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

<b>Retruque!</b>		<b>Primer año de vida de la App</b>	
<b>Costos Fijos</b>			
Descripcion	Detalle	Gasto Mensual	Gasto Anual
Alta como desarrollador Google	Darse de alta como desarrollador de Google Play	0,00	25,00
Diseñador gráfico	Servicio de diseñador gráfico para iconos de la App	0,00	50,00
Publicidad en línea	Adwords (aproximado 0,05 por clic)	20,00	240,00
<b>TOTAL</b>		<b>20,00</b>	<b>315,00</b>
<b>Ingresos</b>			
Descripcion	Detalle	Ingreso Mensual	Ingreso Anual
Descargas	0,02 por cada descarga (100 mensuales)	20,00	240,00
Publicidad	(100000 impresiones, 5% CTR, 0,12/click)	80,00	960,00
<b>TOTAL</b>		<b>80,00</b>	<b>1200,00</b>
<b>TOTAL Ingreso - Gastos</b>		<b>60,00</b>	<b>885,00</b>

En cuanto a los años de vida posteriores del juego, preveemos un aumento notable de los beneficios. En este periodo, la App debe de estar expandida sobre todo por la Comunidad Valenciana, habremos añadido mejoras y tendrá un gran número de usuarios y partidas diarias.

Nuestros ingresos seguirán siendo por publicidad y por descargas en su mayor parte pero también existirá contenido Premium en el juego, del que podemos sacar un beneficio mensual.

Por otra parte, nuestra aplicación no necesita ninguna otra inversión aparte de la publicidad que podamos contratar para darla a conocer, por lo que los gastos no variarán demasiado.

Como se ve en la tabla de abajo, los ingresos anuales aumentarán notablemente.

<b>Retruque!</b>		<b>Años posteriores</b>	
<b>Costos Fijos</b>			
Descripcion	Detalle	Gasto Mensual	Gasto Anual
Publicidad en línea	Adwords (aproximado 0,05 por clic)	40,00	480,00
<b>TOTAL</b>		<b>40,00</b>	<b>480,00</b>
<b>Ingresos</b>			
Descripcion	Detalle	Ingreso Mensual	Ingreso Anual
Descargas	0,02 por cada descarga (500 mensuales)	10,00	120,00
Elementos premium	Elementos premium de los que disponga el juego	50,00	600,00
Publicidad	(100000 impresiones, 5% CTR, 0,12/click)	250,00	3000,00
<b>TOTAL</b>		<b>310,00</b>	<b>3720,00</b>
<b>TOTAL Ingreso - Gastos</b>		<b>270,00</b>	<b>3240,00</b>

Es posible que el ritmo de descargas varíe o disminuya con el tiempo, ya que está acotado geográficamente y en algún momento (cuando se haya expandido en gran medida), las descargas comenzarán a reducirse. Sin embargo, se seguirá ganando dinero por los demás elementos.

## 6. Proceso de desarrollo



En esta sexta parte de la memoria vamos a describir el proceso de desarrollo de Retruque!

Nuestro proyecto nos ha llevado, aproximadamente, 8 meses desde principio hasta el final. Suena a mucho tiempo, pero vamos a explicar un poco la dedicación y tareas realizadas.

La idea nos surgió al quedar varias veces con nuestro profesor, buscando el proyecto adecuado que nos gustaría hacer. Buscábamos un producto novedoso, que nos gustara hacer y que fuera emprendedor. Tras barajar varias alternativas, decidimos hacer un Truc para jugar online.

Durante el primer mes y el segundo de reuniones con nuestro profesor, concretamos muy bien todas las características que debería tener nuestro juego, así como los distintos análisis tanto de nuestro producto como de los juegos que pudieran hacernos competencia.

Más tarde, estuvimos un par de meses apartados del proyecto, ya que debíamos acabar con las últimas asignaturas que nos quedaban y no teníamos tiempo que dedicarle al proyecto.

Los últimos 3 meses, como se ha dicho antes, los dedicamos completamente al desarrollo del juego, ya que el trabajo previo que habíamos hecho junto con el profesor nos permitió tener las ideas y los objetivos muy claros desde el primer momento. Sabíamos lo que queríamos pero no sabíamos cómo hacerlo, ya que nunca habíamos hecho un juego online y no teníamos mucha idea de cómo empezar a desarrollar.

Empezamos un trabajo de pruebas e investigación que nos iba a llevar aproximadamente 2 semanas, en las que probamos distintas tecnologías como el login con Facebook o tener nuestro propio servidor para las partidas, pero encontramos Google Play Game Services y vimos que nos facilitaría muchísimo el trabajo, de ahí nuestra elección final.

Una vez tuvimos claras las tecnologías que íbamos a usar y cómo tenía que ser nuestra aplicación, pudimos comenzar finalmente a programar en Android Studio. Este proceso dura desde esta etapa hasta la publicación de la App en Play Store y dura, como ya se ha dicho anteriormente, unos 3 meses.

Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

Decidimos empezar logeando a los jugadores con Google+ y con el modo para 2 jugadores, que nos costó un poco, ya que tuvimos que aprender cómo funcionaba Google Play Game Services y cómo podríamos comunicar a los jugadores.

Una vez entendimos el funcionamiento de esta tecnología, fuimos poco a poco avanzando hasta conseguir terminar el modo de 2 y abordar el modo de 4, que añadía bastante dificultad a la programación.

Sin embargo, nos costó bastante menos de lo que creíamos, debido a que la base que teníamos del modo de 2 jugadores nos iba a resultar muy útil.

Cuando conseguimos acabar el modo de 4, incluimos también un sistema de logros para desafiar a los jugadores y un ranking de victorias, así como una mejora notable de la interfaz gráfica con animaciones y sonidos que resultarían muy atractivas para el usuario final.

## 6.1 Mapa de características



En la imagen observamos el mapa de las características que tenía que tener nuestra aplicación. Como se ha dicho en el apartado anterior, fueron consensuadas junto con el profesor y las estructuramos en distintas etapas o Sprints, separándolas de mayor a menor importancia.

Cabe destacar que todas las características de la primera línea (las que entendíamos que eran las esenciales para subir el juego a Play Store) han sido incluidas en su totalidad en la primera versión de la App, pero también hemos incluido la mayor parte de la segunda fila, que en un principio iban a entrar en la segunda versión.

El motivo de esta inclusión de características fue que nos vimos capaces de desarrollarlas para la primera versión y quisimos hacer un juego mucho mejor del que teníamos pensado en su versión inicial. Pudimos incluir los efectos de sonido, el sistema de logros, las señas para 4 jugadores y una remodelación de la interfaz gráfica de la aplicación, haciéndola mucho más visual y divertida para el jugador.

Por tanto, nos dejamos las características restantes y la tercera fila para una próxima versión, que iremos desarrollando con el paso del tiempo.

## 7. Tecnologías utilizadas

En nuestro proyecto hemos utilizado numerosas tecnologías. Hacer una aplicación para Android no es una tarea sencilla y necesita muchos elementos para realizarse. A continuación, pasamos a describirlas.

### 7.1 Android

#### Dispositivos móviles y sus características

Un dispositivo móvil es un aparato de pequeño tamaño, con capacidades de proceso limitadas con respecto a otros dispositivos. Además nos puede proporcionar conexión a una red y poseen memoria limitada, diseñados específicamente para una función, pudiendo llevar a cabo otras funciones mas generales y variadas.

Entre sus características más destacables se encuentran las siguientes:

- Reducido tamaño, que junto con su peso reducido facilitan su movilidad.
- Equipamiento específico tanto de procesamiento, como de almacenamiento y alimentación, con la finalidad de mejorar la autonomía.
- Proporcionan servicios de conectividad inalámbrica.
- Almacenamiento local de datos, así como sincronización con otras plataformas.
- Interacción con el usuario adaptada.

Algunas de las tecnologías que posibilitan el desarrollo de este tipo de dispositivos son:

HSDPA, ARM, 3G/3.5G, Memorias Flash, Servlets, WIFI, Li-ion, OLED, Multitouch, teclados virtuales, Pantallas capacitivas, Bluetooth, Pantallas resistivas, Cloud computing, AMOLED, microSD HC, SoC, GPU, Web services, GPS/AGPS

El verdadero éxito de los dispositivos móviles se encuentra en su capacidad para ofrecer múltiples funciones y servicios a sus usuarios.

#### ¿Qué es Android?

Android es una plataforma software diseñada con el objetivo de abstraer el hardware y facilitar el desarrollo de aplicaciones para dispositivos con recursos limitados, generalmente dispositivos móviles.

- Se trata de una plataforma “open source” y se distribuye con licencia Apache 2.0.
- Define toda una pila software, incluyendo el sistema operativo a utilizar, middleware, soporte de ejecución para las aplicaciones y un conjunto mínimo de apps básicas.
- Se apoya en el uso de una máquina virtual propia, basada en el kernel de Linux.
- Proporciona un conjunto completo de APIs y herramientas de desarrollo, compilación, depuración y emulación.

Inicialmente, Android fue una propuesta del Open Handset Alliance, una organización dedicada al desarrollo de software libre, aunque su expansión ha sido propiciada por el uso masivo de los servicios propietarios de Google, afectando de esta forma a la naturaleza “open source” de la plataforma.



## Un poco de historia

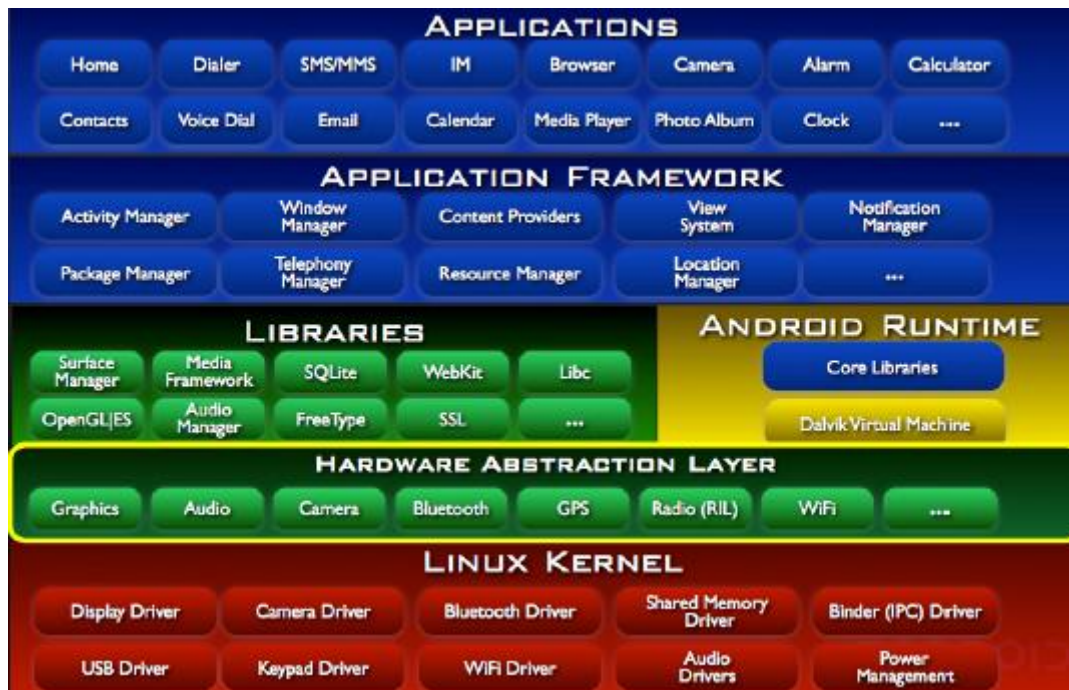


## El problema de la fragmentación

La fragmentación en Android hace referencia a la variedad de versiones y especificaciones técnicas de hardware (forma, resolución de pantalla, tamaño, prestaciones, etc.), en definitiva, hay una enorme variedad de entornos Android.

- Esta variedad de entornos dificulta que los programadores adapten sus aplicaciones al universo de dispositivos disponibles en el mercado. En ocasiones supone un problema a la hora de recibir actualizaciones por parte de los fabricantes.
- Esta variedad ha permitido que la plataforma penetre con fuerza en multitud de mercados, adaptándose tanto al poder adquisitivo como a las necesidades de los compradores existentes en cada uno.

## Cómo es android



## Kernel linux

Android está basado en un kernel probado de Linux, dotando de una gran fiabilidad al sistema. Es la única capa que depende del hardware del dispositivo y es responsabilidad del fabricante. Android se aprovecha del kernel para la mejora de la seguridad, la gestión de memoria y procesos así como la gestión de red y alimentación.

Android utiliza un modelo de controladores, que proporciona soporte a los nuevos accesorios a través de controladores. De esta forma desacopla el hardware del software a través de una capa de abstracción HAL (Hardware Abstraction Layer).



## Librerías Android

Las librerías más importantes son las siguientes:

- **Surface Manager:** Gestión de acceso a pantalla
- **Media Framework:** Reproducción de imágenes, audio y video.
- **SQLite:** Potente y ligero motor para bases de datos relacionales.
- **WebKit:** Navegador web usado por el navegador de Android y los WebView.
- **SGL:** Gráficos 2D.
- **OpenGL:** Gráficos 3D.
- **FreeType:** Renderización vectorial y fuentes en bitmap.
- **SSL:** Capa de comunicación segura por sockets.
- **Libc:** Reimplementación optimizada de las librerías en C.

## Runtime de Android

Android hace uso de dos máquinas virtuales muy optimizadas, si tenemos en cuenta las condiciones que imponen los dispositivos móviles en la actualidad. Estas máquinas virtuales ejecutan bytecode de tipo Dalvik y son Dalvik Virtual Machine (DVM) y Android Runtime (ART).

- Máquina Virtual Dalvik:
  - Uso del kernel para la ejecución de apps (1app = 1 proceso = 1DVM) → ejecución concurrente de instancias de la DVM.
  - Utiliza los servicios ofrecidos por el OS de aislamiento de procesos, gestión de memoria y soporte multihilo.
  - No es una máquina virtual de Java (JVM) → ejecuta aplicaciones en formato propio Dalvik Executable (DEX) y no soporta clases AWT o Swing.
  - Aplicación de compilación de tipo JIT (just-in-time) → conversión de ejecutables DEX a código máquina cuando dicho código se necesita y solo entonces.
- Android Runtime (ART) → opcional en KitKat, pero obligatorio en Lollipop
  - Aplicación de compilación de tipo Ahead-of-time, que convierte el ejecutable DEX a código máquina cuando se instalan las aplicaciones para una mejora de las prestaciones.
  - Mejora la recolección de basura Dalvik.
  - Introduce mejoras de depuración como informes más detallados en caso de excepción y crash de las aplicaciones.

## Entorno de aplicaciones

El entorno de aplicaciones nos proporciona una plataforma de desarrollo que permite la reutilización de componentes, en la que cada aplicación publica sus capacidades y se utilizan unas a otras en base a ellas.

A través de este framework, el desarrollador puede acceder al hardware de los dispositivos, localizarse, gestionar el ciclo de vida de las aplicaciones etc.

Las aplicaciones básicas (core applications) del sistema también utilizan las APIs ofrecidas por el framework.

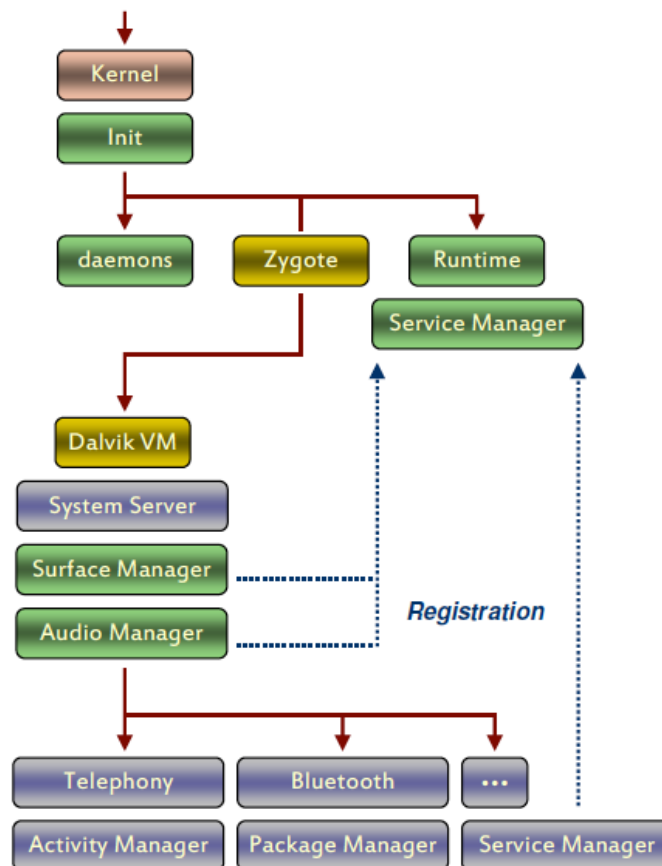


### Nivel aplicativo

En esta capa se ubican las aplicaciones preinstaladas por los fabricantes y operadores y se instalan las que deseen los usuarios. Las aplicaciones bien se desarrollan en Java utilizando el SDK (Software Development Kit) de Android o en código nativo C/C++, lo que requiere del uso del NDK (Native Development Kit) de Android.

Entre las aplicaciones incluidas por defecto destaca la aplicación de Google Play, que ofrece al resto de apps servicios esenciales para, por ejemplo, el trabajo con los mapas o la facturación a clientes.

### Inicialización del sistema



1. El cargador arranca el kernel y lanza a ejecución el Init.
2. Se lanzan los procesos demonios que gestionan los recursos hardware (USB, radio...).
3. Se crea el proceso Zygote, que no es más que una instancia de la máquina virtual.
4. El proceso Runtime solicita a Zygote que lance una nueva instancia de la máquina virtual para ejecutar el servidor del sistema,
5. Los primeros procesos que se ejecutan gestionan la pantalla y el audio.
6. A continuación se lanzan el resto de procesos.



## Aplicaciones

Centrándonos ya en el desarrollo de aplicaciones en Android debemos destacar los siguientes componentes:

- **Activity:** Componente que define la interfaz de usuarios, típicamente asociado a una pantalla de la aplicación.
- **Broadcast Receiver:** Componente que responde a notificaciones o cambios de estado. Su activación puede conllevar la creación de un proceso.
- **Service:** Componente sin interfaz que realiza una tarea en segundo plano.
- **Content Provider:** Componente que permite a las aplicaciones compartir información.

En este contexto, el desarrollo en Android es centrado en Activities.

El concepto de Actividad (Activity) viene a estar relacionado con cada una de las pantallas de nuestra aplicación, de esta forma su función se asocia a la interfaz de usuario, es decir, cada actividad se asociará a una pantalla de la aplicación y definirá su funcionalidad. Toda actividad tiene una vista asociada, generalmente un Layout, y que será lo que se le mostrará al usuario. Además toda Actividad ha de ser una subclase de la clase Activity.

Una aplicación está formada por varias actividades independientes que colaboran con el objetivo común de satisfacer las necesidades del usuario.

## Ciclo de vida

Cada aplicación Android se ejecuta en su propio proceso Linux y posee su propia instancia de la máquina virtual, de esta forma vamos a tener el siguiente procedimiento.

Cuando una petición necesita de la intervención de un componente de la aplicación, Android se asegura de que el proceso asociado a la aplicación este en ejecución, arrancándolo en caso de necesidad. También verifica que la instancia adecuada del componente este disponible y en caso de no estarlo, crea una nueva.

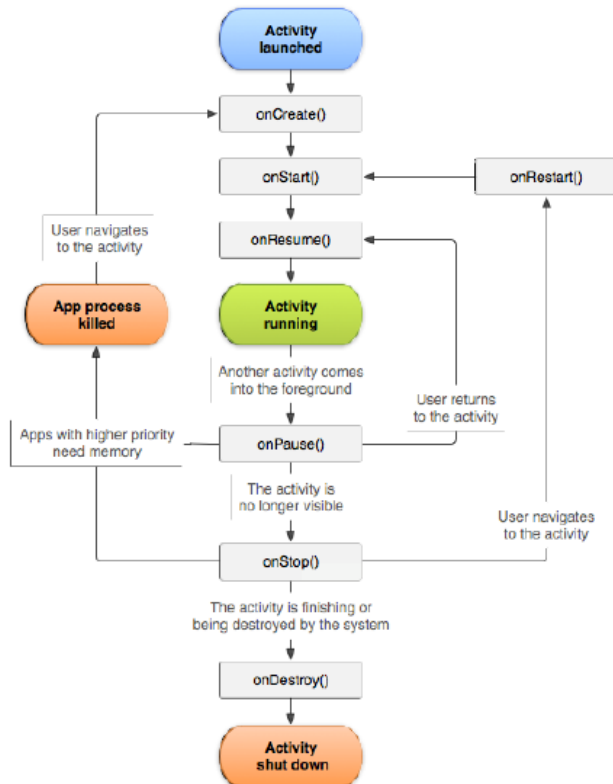
El proceso Linux que encapsula toda la aplicación se crea cuando el código de la misma debe ejecutarse, es decir, en el momento de arrancar la aplicación. Así pues permanece en ejecución hasta que, o bien no se necesita más, o bien el sistema reclama los recursos ocupados para otras aplicaciones. Como vamos a ver ahora, los usuarios no controlan el ciclo de vida de las aplicaciones, es el propio sistema el que lo hace.

El ciclo de vida de los componentes tiene los siguientes estados:

- **El inicio.** Android instancia los componentes en respuesta a un Intent. Éste se corresponde con la intención de arrancar la aplicación.
- **El fin.** Android destruye esta instancia de la aplicación.
- **Entretanto,** su vida transita entre los estados activo e inactivo, o en el caso de las Actividades, el de visible e invisible para el usuario.

El ciclo de vida completo va desde onCreate, método que inicializa la Actividad principal por primera vez, hasta onDestroy, donde se destruye tanto la aplicación como su proceso. Las Actividades son visibles entre onStart y onStop, pero tan solo entre onResume y onPause son interactivas y se ejecutan en primer plano permitiendo la interacción con el usuario.





Un control correcto del ciclo de vida nos va a permitir desarrollar una aplicación fiable y estable, y por ello se recomienda sobrescribir todos los métodos de transición entre actividades. Entre ellos los más importantes, el `onCreate` para inicializar la Actividad de la forma que deseamos, así como el método `onPause` cuando se necesario asegurar que los cambios en los datos se almacenan correctamente y se gestiona correctamente el fin de la interacción con el usuario.

## Los procesos en Android

En Android, la destrucción del proceso de una aplicación no la decide el usuario sino el sistema, de modo que su proceso continuará en memoria incluso después de que el usuarios cierra la aplicación que se está ejecutando.

Una app sólo es eliminada cuando el sistema necesita memoria para otra. En casos muy extremos en los que el sistema necesite memoria de forma urgente (ante una llamada telefónica por ejemplo), el sistema podrá eliminar de memoria incluso apps en ejecución, es decir, con componentes activos. Por este es conveniente realizar un bien seguimiento del ciclo de vida de nuestra aplicación, con el fin de no perder datos de cierta relevancia para el usuario.

Para determinar que procesos eliminar, Android los ordenará jerárquicamente según su importancia relativa.

## Tipos de procesos

1. Proceso en primer plano (Foreground process)
  - Contiene a la Actividad en ejecución con la cual el usuario interactúa, la última sobre la que se activó el método `onResume`.
  - Actualmente, sólo puede haber un proceso de este tipo en todo el sistema y nunca debería ser eliminado.
2. Proceso visible (Visible process)
  - Contiene una actividad visible en pantalla pero que no está en ejecución, por tanto se ha invocado su método `onPause`.

- Se considera importante porque suele contener una de las últimas aplicaciones que el usuario utilizó, y quizás quiera volver a relanzarla.
  - No se suele eliminar, a menos que sea necesario para mantener en ejecución al proceso en primer plano.
3. Proceso en segundo plano (Background process)
    - Alberga actividades que o son visibles, en las que se ha invocado su método `onStop`.
    - Su eliminación no tienen un impacto directo en la experiencia que el usuario tiene de la aplicación.
    - Cuando el usuario la vuelva a lanzar y navegue hacia alguno de los componentes del proceso, éste se puede instanciar nuevamente, aunque su estado de perderá y deberá restaurarse.
    - Normalmente hay muchos procesos de este tipo, por lo que el sistema intenta asegurar que el que contenga la última actividad vista por el usuario sea el último en ser eliminado.
  4. Proceso vacío (Empty process)
    - No aloja ningún componente activo.
    - Estos procesos se mantienen en memoria a modo de “caché” para mejorar los tiempos de activación de las aplicaciones cuando estas vuelven a ser utilizadas y relanzadas por los usuarios.

## Intents

Hasta ahora hemos visto que las Actividades son las pantallas de nuestra aplicación, como crearlas y destruirlas, y su ciclo de vida entre otros, pero ¿Cómo navegamos entre ellas?

Para ello se introduce el concepto de Intent. Un Intent es un mensaje asíncrono que facilita el intercambio de información entre distintos componentes de la plataforma Android, no solo entre Actividades. Usaremos los Intents cada vez que queramos:

- Lanzar una Actividad, mediante `startActivity` y `startActivityForResult`.
- Lanzar un servicio, mediante `startService`.
- Remitir una notificación, uso de `sendBroadcast`.
- Comunicarnos con un servicio con `bindService`.

Los intents permiten indicar acciones genéricas, por ejemplo, si nuestra aplicación es una agenda y queremos llamar a un contacto, el Intent lanzará la aplicación registrada por defecto a tal propósito, en este caso la aplicación del teléfono. Si se diese el caso en el que quisiésemos mandar un correo y tuviésemos más de una aplicación que nos pudiera satisfacer esta petición (Gmail e Inbox), se abriría un desplegable en el cual el usuario indicaría que aplicación arrancar.

Existen dos tipos de Intents, el Intent explícito y el Intent implícito.

- Intent explícito: Se indica exactamente el componente a lanzar, en este caso siempre haremos referencia a una Actividad de nuestra aplicación.
- Intent implícito: Son solicitudes de tareas abstractas, en las que el destinatario del Intent se determina en tiempo de ejecución y como hemos visto antes, puede que se requiera de la intervención del usuario si el sistema encuentra más de una



Actividad que satisfaga la petición. Por ejemplo, “quiero tomar una foto” o “quiero compartir algo” (Facebook, Twitter, Whatsapp).

## 7.2 Android Studio

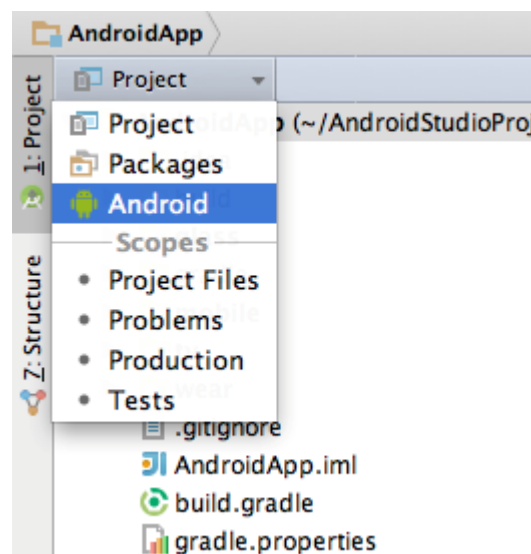
Android Studio es el IDE oficial para el desarrollo de aplicaciones en Android. Aunque se puede desarrollar en otras plataformas como Eclipse, nosotros escogimos el IDE por excelencia de Google. Android Studio está basado en IntelliJ IDEA. De las capacidades más interesantes que esperamos de IntelliJ, Android Studio nos ofrece:

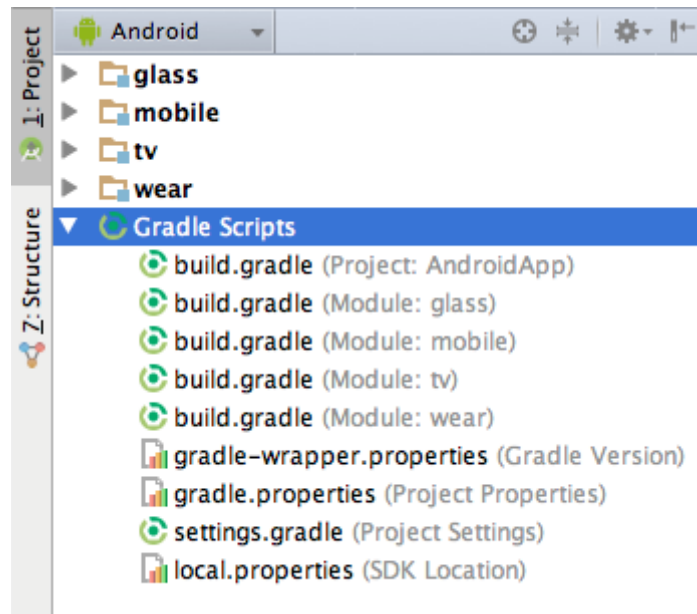
- Gradle-based build system: Sistema para la compilación y construcción de dependencias llamado Gradle.
- Otros sistemas de construcción como Maven, además de muchas opciones para la generación de APK.
- Plantillas de código que ayudan a desarrollar características frecuentes de aplicaciones.
- Editor de layouts muy avanzado, con soporte drag and drop para la edición de los temas de las aplicaciones. También soporte XML del mismo.
- Herramientas para capturar el rendimiento, la usabilidad, la compatibilidad de versiones y otros problemas.
- Capacidad de firmas para las aplicaciones.
- Soporte para la plataforma Google Cloud, facilitando la integración con Google Cloud Messaging y App Engine.

A continuación veremos una introducción a algunas características clave de Android Studio.

### Proyectos y estructura de archivos

Por defecto, Android Studio muestra los archivos del proyecto en la vista del proyecto. Esta vista nos proporciona un acceso rápido a los archivos fuente más importantes además de mostrarnos la estructura de nuestro proyecto. También nos ayuda a trabajar con Gradle.

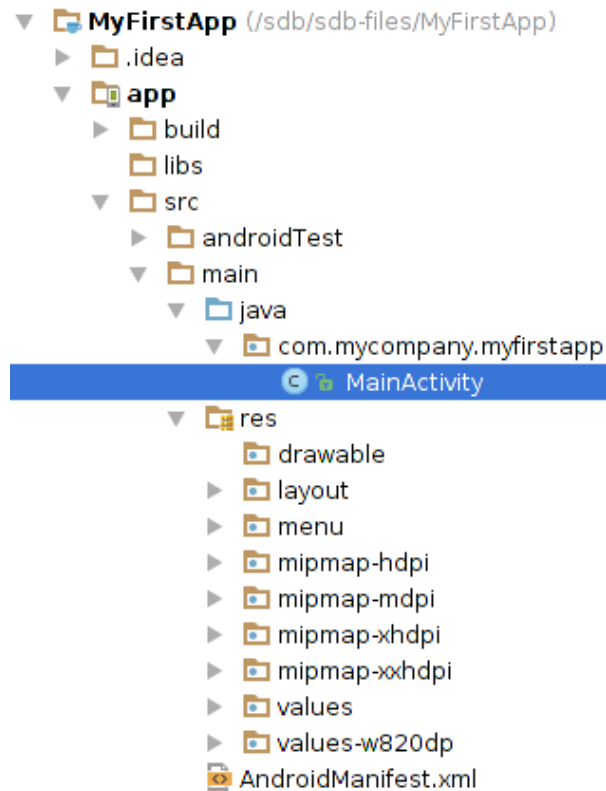




La vista del proyecto contiene todos los archivos en el nivel más alto de la jerarquía, bajo los scripts de Gradle. Cada módulo del proyecto es una carpeta (glass, mobile, tv y wear en la figura 2) y dentro de cada módulo contiene los siguientes cuatro elementos:

- `java/` - Código fuente Java para cada módulo.
- `manifests/` - Manifest para cada módulo.
- `res/` - Recursos del módulo.
- `Gradle Scripts/` - Ficheros Gradle y propiedades.

**Ejemplo:** La vista del proyecto en Android Studio agrupa todas las instancias de una imagen para diferentes densidades de pantalla, bajo el mismo directorio en `res/ejemplo.png`.



### Android build system

El sistema de construcción de Android es un kit de herramientas que nos permite compilar, probar, ejecutar y empaquetar las aplicaciones. Este sistema sustituye al sistema Ant que hace uso del ADT Eclipse.

Con este sistema podemos:

- Personalizar, configurar y extender los procesos de construcción.
- Crear múltiples APKs para una aplicación, con características diferentes usando los mismos módulos y el mismo proyecto. Gracias a esto podemos reutilizar y reusar código.

### Depuración y rendimiento

Android Studio nos proporciona un gran número de mejoras para asistirnos en la depuración y ayudar a mejorar el rendimiento de nuestro código, incluyendo un asistente para la emulación de dispositivos móviles, depuración línea a línea y herramientas de análisis del rendimiento.

### Android Virtual Device (AVD) manager



El AVD de Android Studio pone a nuestro servicio una máquina virtual que simula un dispositivo Android. El AVD manager nos viene con toda la gama Nexus de Google, incluyendo las tablets pero además podremos configurar nuestro dispositivo como

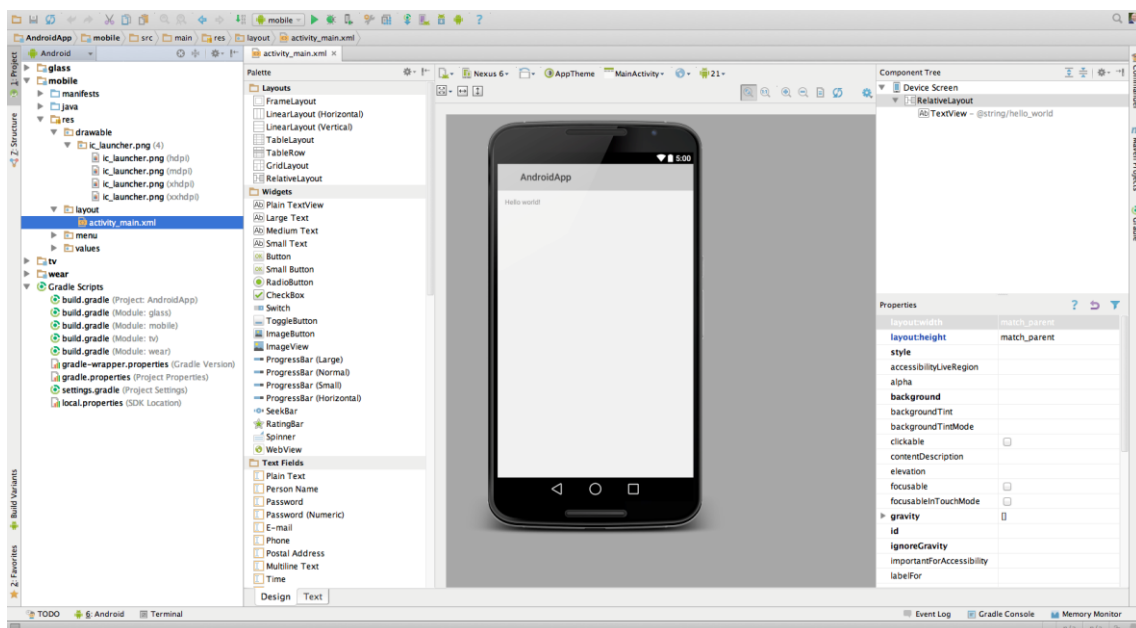
nosotros queramos especificando sus propiedades. Podremos elegir cuanta memoria RAM tendrá, establecer las dimensiones del dispositivo y su resolución y muchas más configuraciones.

El único requerimiento está en el procesador de nuestro ordenador. Deberá ser capaz de soportar la aceleración hardware de Intel, puesto que Android Studio instala el Intel® x86 Hardware Accelerated Execution Manager (HAXM). Sin él no podremos ejecutar los emuladores.

Por último, haremos uso del SDK de Android para descargar e instalar las diferentes APIs de Google así como todas las versiones de Adroid que necesitemos.

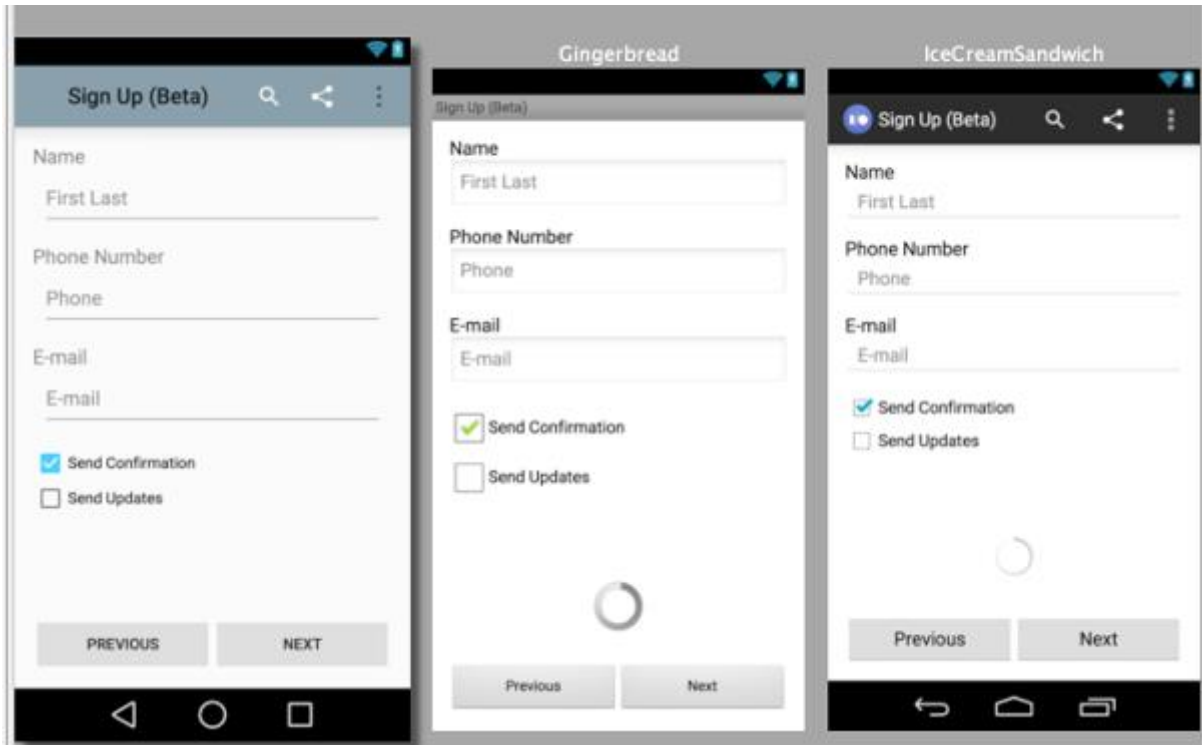
## Definiendo layouts

Android Studio nos permite trabajar con los layout de nuestra aplicación en modo diseño como vamos a ver, o en modo texto, editando directamente el XML del layout.




El modo diseño de Android Studio nos permite realizar cambios en la interfaz arrastrando los componentes de la lista a la misma pantalla. Además Android Studio nos provee de una ventana “preview”, que nos permite visualizar como quedaría nuestro layout en diferentes tamaños, desde de pixeles incluso en diferentes versiones de Android.





### Android debug monitor

Si queremos depurar nuestra aplicación Android Studio nos provee del Android Debug Monitor  .

Tendremos a nuestra disposición una serie de herramientas para controlar los comportamientos de nuestros emuladores, así como ejecutar nuestra aplicación en modo debug, haciendo uso de breakpoints permitiéndonos hacer trazas de comportamiento.

Podremos mostrar mensajes “Log” en el logcat, una “consola” donde también podremos ver las excepciones que da nuestra aplicación.

### 7.3 Genymotion + virtualbox

Genymotion es un emulador de Android que aprovecha la arquitectura x86 para ejecutar de forma fluida y rápida distintos dispositivos Android. Olvidando la lentitud del emulador nativo de Android (AVD que hemos visto anteriormente) podemos ejecutar todo tipo de aplicaciones y juegos en las plataformas Windows, Mac o Linux.

Uno de los principales usos de Genymotion es facilitar el desarrollo de aplicaciones Android. Se integran perfectamente con el Android Studio, Eclipse y todas las herramientas de depuración que hemos visto. En el caso de Android Studio, tenemos un plugin de Genymotion que nos permite lanzar en apenas un par de segundos las aplicaciones desde el IDE para testearlas. Esto nos ahorra mucho tiempo.

Genymotion está basado en el uso de máquinas virtuales x86 optimizadas para correr sobre Virtualbox. En nuestra opinión la diferencia entre Genymotion y el AVD es notable.



Genymotion, como AVD, nos permite crear máquinas virtuales con diferentes características, resoluciones de pantalla y configuraciones Android (2.3, 4.1, 4.2, 4.3, 4.4 y 5), además viene por defecto con una serie de dispositivos, Nexus 4, Nexus 5, Nexus 7, Galaxy Nexus, HTC One, Moto X, Galaxy S4, Xperia Z, entre otros.

En cuanto a lo que nos ofrece Genymotion, se encuentra el uso de nuestra conexión a internet, la simulación ubicaciones GPS con el widgets que facilita la búsqueda y el posicionamiento sobre un mapa, simulación de la cámara, estado de la batería, rotación del dispositivo, etc.

El problema más significativo de Genymotion es que no incluye por defecto las aplicaciones de Google, por motivos de licencia. Actualmente se está trabajando para conseguir un acuerdo con Google. Esto limita bastante su funcionalidad, puesto que no podremos trabajar ni con Google Maps, ni con Gmail, y tampoco con los Google Play Services.

## 7.4 GitHub

GitHub es una plataforma de desarrollo colaborativo de software para alojar proyectos utilizando el sistema de control de versiones Git. El código se almacena de forma pública, aunque también se puede hacer de forma privada, creando una cuenta de pago.



GitHub aloja tu repositorio de código y te brinda herramientas muy útiles para el trabajo en equipo, dentro de un proyecto.

Además de eso, puedes contribuir a mejorar el software de los demás. Para poder alcanzar esta meta, GitHub provee de funcionalidades para hacer un fork y solicitar pulls.

Realizar un fork es simplemente clonar un repositorio ajeno (genera una copia en tu cuenta), para eliminar algún bug o modificar cosas de él. Una vez realizadas tus modificaciones puedes enviar un pull al dueño del proyecto. Éste podrá analizar los cambios que has realizado fácilmente, y si considera interesante tu contribución, adjuntarlo con el repositorio original.

Es un controlador de versiones muy potente y muy útil para los equipos de trabajo. Además Android Studio también nos facilita la sincronización de nuestros proyectos con GitHub. Podremos bajarnos nuestro repositorio con el comando “update”, que actualizará nuestro proyecto local con el código del repositorio en la nube y en caso de existir problemas (dos usuarios han cambiado las mismas líneas de código), nos dará la posibilidad de hacer un “merge” y fusionar ambos códigos como queramos. Una vez hemos realizado modificaciones, podremos subir al repositorio de GitHub nuestros cambios mediante los comandos “commit” y “push”.

Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

GitHub también nos da la posibilidad de crear “branches” (ramas), de esta forma podremos alojar diferentes módulos de nuestro proyecto en diferentes ramas, facilitando así tanto el desarrollo como el mantenimiento y el testing de cada módulo por separado.

Podremos subir a GitHub tantos proyectos como queramos, eso sí, serán públicos y visibles para todos los usuarios. Si queremos tener proyectos privados, deberemos adquirir una cuenta premium.

## 7.5 AdMob



Puesto que hemos desarrollado este proyecto dentro del contexto emprendedor, queremos sacarle un rendimiento económico. En principio nuestra aplicación seguirá un modelo Free, el cual si tiene la suficiente adopción evolucionará hacia un modelo Freemium, en el cual los usuarios pagarán por ciertas funcionalidades. Con el modelo Free la descarga es totalmente gratuita, de esta forma utilizaremos AdMob para sacarle rentabilidad a nuestro proyecto.

AdMob es la plataforma publicitaria más usada y reconocida por los programadores de aplicaciones. Más de 650.000 aplicaciones usan AdMob que ha pagado más de un millón de dólares.

AdMob reúne la tecnología más avanzada en una sola plataforma para que puedas obtener datos sobre tus usuarios, generar más compras en la aplicación y aumentar tus ingresos publicitarios. Con estas herramientas que nos proporciona AdMob nos olvidamos de utilizar recursos de programación costosos o aplicaciones externas para monitorizar y ver que gusta y que disgusta a nuestros usuarios.

### ¿Por qué admob?

Google lleva más de una década ayudando a millones de desarrolladores a mejorar sus negocios digitales. Entre las características más importantes:

- **Compatibilidad con varias plataformas:** AdMob puede usarse con las principales plataformas, Android y iOS y, además, admite los motores de juego más importantes, como Unity y Cocos2d.
- **Actualizaciones automáticas en Google Play:** La integración de AdMob con los servicios de Google Play incorpora mejoras automáticas para las aplicaciones de Android sin cambios adicionales en el SDK.
- **Recibe pagos rápidamente:** “Recibe pagos en tu moneda local rápidamente y de manera fiable. En AdMob no aplicamos tarifas de transferencia”, es lo que nos dicen desde la página de AdMob aunque todavía no lo hemos comprobado.
- **Fácil y gratuito:** El SDK puede instalarse muy rápidamente y totalmente gratuito.

En la búsqueda de entender a nuestros usuarios, AdMob nos proporciona:

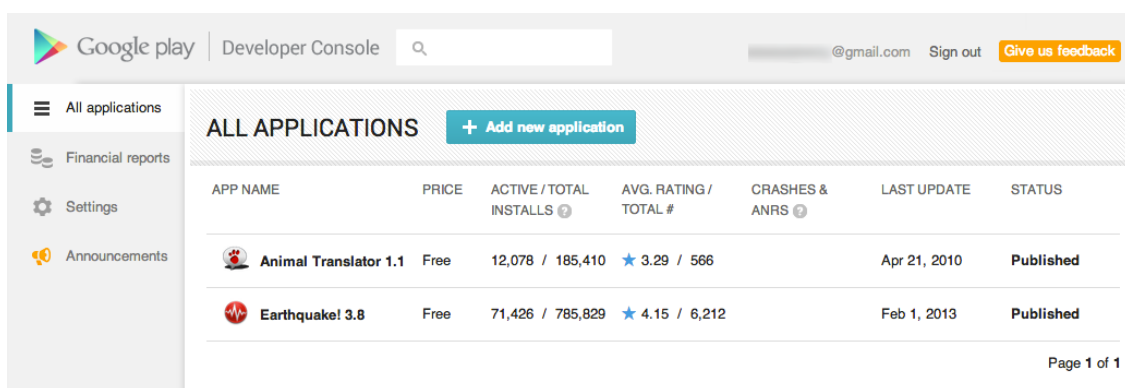
- **Analytics para aplicaciones:** Nos permite analizar el rendimiento de las aplicaciones desde AdMob con Google Analytics. Nos permite conocer en tiempo real, desde dónde se descarga la aplicación, y que funciones usan los usuarios con más frecuencia.
- **Informes de visualización del flujo de tráfico:** Con Analytics, comprobamos cuántos usuarios navegan por la aplicación con los informes gráficos de flujo de tráfico. Podemos ver qué ruta siguen los usuarios para realizar una compra y en qué momento salen de la aplicación, entre otros datos.

## 7.6 Developer Console

La consola de desarrollo de Google nos permite manejar todo lo relativo a operaciones y herramientas para la publicación de nuestra aplicación. Desde ella podremos subir aplicaciones a Play Store, configurar los precios y la distribución desde cualquier navegador.

Para ello primero deberemos registrarnos como desarrolladores de Google, y una vez nos manden el correo de verificación, podremos acceder a la Google Play Developer Console. Para registrarnos como desarrolladores deberemos abonar una cantidad anual de 25 dólares.

En la ventana All Applications encontraremos una lista con nuestras aplicaciones, tanto si están como si no están subidas a Google Play. Haciendo click sobre una aplicación podremos ver detalles del producto.



The screenshot shows the Google Play Developer Console interface. At the top, there's a navigation bar with the Google Play logo, 'Developer Console', a search bar, and user information like '@gmail.com', 'Sign out', and 'Give us feedback'. Below this is a sidebar with menu items: 'All applications', 'Financial reports', 'Settings', and 'Announcements'. The main content area is titled 'ALL APPLICATIONS' and includes a '+ Add new application' button. It displays a table with the following data:

APP NAME	PRICE	ACTIVE / TOTAL INSTALLS	AVG. RATING / TOTAL #	CRASHES & ANRS	LAST UPDATE	STATUS
Animal Translator 1.1	Free	12,078 / 185,410	★ 3.29 / 566		Apr 21, 2010	Published
Earthquake! 3.8	Free	71,426 / 785,829	★ 4.15 / 6,212		Feb 1, 2013	Published

Page 1 of 1

Desde la consola podremos desarrollar también la página principal que veremos en Play Store. Esta será la página que vean los usuarios en su dispositivo móvil y desde donde podrán obtener la información de la aplicación y descargarla.

Subir capturas de pantalla, videos o alguna decoración especial puede resultar atractivo para los usuarios. Es conveniente dar una buena descripción, comentar los cambios de las nuevas versiones y mantener informado al usuario de todo. Todos estos datos pueden ser actualizados en cualquier momento.

## 8. Arquitectura de la aplicación: parte del Back-End

Aquí llegamos a la parte dividida de la memoria, en la que mi compañero Guillermo hará la parte del Front-End y dónde yo me dedicaré al Back-End de la aplicación.

Antes de entrar más profundamente en la materia, veamos que es el Front-End y el Back-End de una aplicación. Según Wikipedia:

*En diseño de software el front-end es la parte del software que interactúa con el o los usuarios y el back-end es la parte que procesa la entrada desde el front-end. La separación del sistema en front-ends y back-ends es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas. La idea general es que el front-end sea el responsable de recolectar los datos de entrada del usuario, que pueden ser de muchas y variadas formas, y los transforma ajustándolos a las especificaciones que demanda el back-end para poder procesarlos, devolviendo generalmente una respuesta que el front-end recibe y expone al usuario de una forma entendible para este. La conexión del front-end y el back-end es un tipo de interfaz.*

Por tanto, entendemos que el Back-End es la parte que procesa los datos y se comunica con el servidor.

Una vez que tenemos los términos claros, podemos avanzar en este punto de la memoria.

### 8.1 Introducción a Google Play Game Services



Google Play Game Services será una parte fundamental en el desarrollo de una plataforma de juegos para la próxima generación.

Sus servicios ayudan a los juegos a ser más sociales, con logros, rankings y modo multijugador, así como más potentes, permitiendo guardar partidas y ajustes en la nube.

Están disponibles para dispositivos Android, pero también para iOS o cualquier otro dispositivo conectado.

Gracias a ellos, un programador puede centrarse en su tarea principal: la programación del juego en sí, dejando de lado las conexiones entre jugadores y tareas que podrían costarle su tiempo.

De esto se ocupa Google Play Game Services.

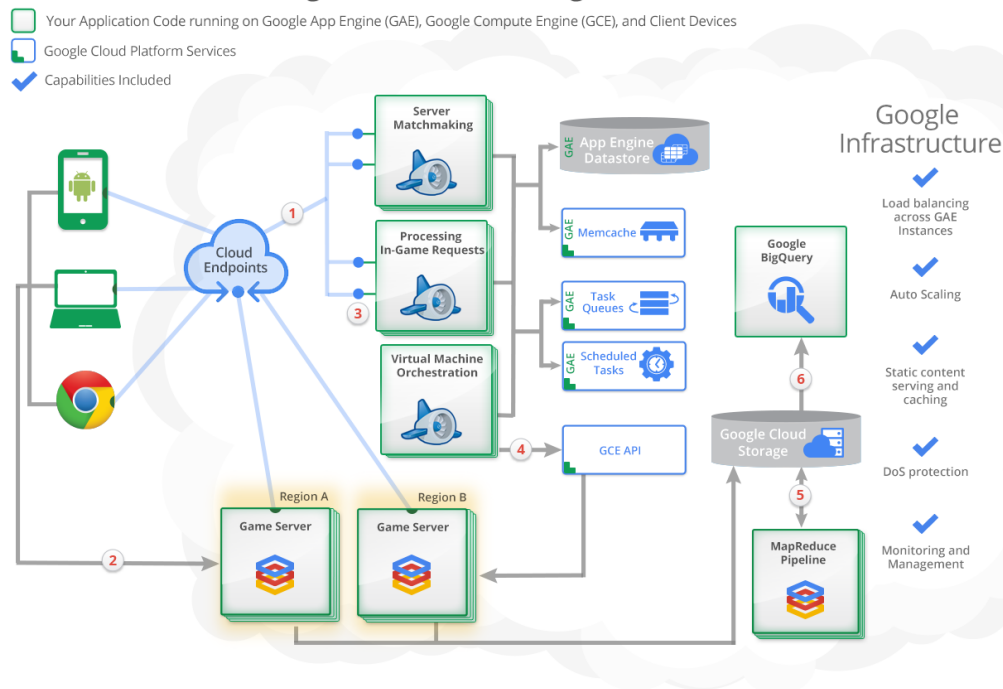
Con Game Services, tienes la posibilidad de añadirle a tu juego:

- **Logros**, que incrementan la adicción del jugador y le desafían a conseguir distintos objetivos marcados por ti.
- **Rankings sociales y públicos**, que usan tus círculos de Google+ para obtener las puntuaciones de tus amigos y de todo el mundo.
- **Almacenamiento en la nube**, que permite guardar fácilmente tus partidas y ajustes del juego. Ahora los jugadores ya no tienen que volver a empezar desde el primer nivel como antiguamente.
- **Multijugador en tiempo real**. Sin duda, el plato fuerte de Game Services. Permite añadir un modo multijugador en línea cooperativo o competitivo en tus juegos. Usando los círculos de Google+, un jugador puede invitar a sus amigos a una partida o buscar partidas con jugadores desconocidos de todo el mundo. Lo mejor es que su API gestiona las conexiones de los jugadores de forma automática, por lo que el desarrollador solo tiene que centrarse en la lógica de su juego.

## 8.2 Estructura de Google Play Game Services

Veamos cómo funciona una partida en Game Services. Para ello, nos ayudaremos de la ilustración inferior a este texto.

### Dedicated Server Gaming Solution on the Google Cloud Platform



Cuando un dispositivo quiere entrar a una partida multijugador (en nuestro caso un Smartphone), lo primero que hace es hacer una búsqueda de servidores disponibles (**Punto 1**).

Una vez se encuentra un servidor disponible, se procede a conectarse a él, para que pueda empezar la partida (**Punto 2**).

Durante ésta, los jugadores estarán mandando mensajes y obteniendo respuestas del servidor, para comunicarse con los demás jugadores y recibir los datos que éstos vayan

mandando (**Punto 3**). De esta forma, la partida podrá ir avanzando. Este proceso se realiza un número indeterminado de veces, que podrá concretarse más dependiendo de lo que pueda durar una partida.

Los **puntos 4, 5 y 6** se encargarán de almacenar nuestras puntuaciones, partidas o logros que vayamos obteniendo durante el proceso del juego. De ahí la conexión con Google Cloud Storage.

Este sería el procedimiento que un dispositivo seguiría para jugar una partida con Google Play Game Services.

### 8.3 Estructura del juego

La estructura de Retruque! es relativamente sencilla, lo que no quiere decir que haya sido fácil el desarrollo del juego.

Como se ha dicho en puntos anteriores de esta memoria, el juego se basa en el modo multijugador. Los jugadores pueden elegir jugar una partida de 2 jugadores (uno frente al otro) o una partida de 4 jugadores, en las que se forman dos equipos de dos jugadores y compiten conjuntamente por ganar la partida. Además tiene extras para aumentar la diversión del jugador como logros, ranking, etc. Pero esto será comentado más detenidamente en el punto siguiente.

Obviamente, para que todo esto funcione, el juego lleva detrás un código escrito en lenguaje Java.

En nuestro código podemos distinguir claramente **cinco** bloques distintos:

- **Variables del juego:** es el primer bloque de código que encontramos. Aquí definimos todas las variables que utilizamos en el código, separadas por tipos y por el modo de juego (utilizadas en el modo de 2 o de 4 jugadores). De esta forma, nos es muy fácil encontrar una variable para modificarla o añadir una nueva del mismo tipo de una forma estructurada y limpia. A continuación, veremos un pequeño ejemplo para el modo de 2 jugadores con dos tipos distintos de variables.

```
//Strings
String idJugador1 = null;
String idJugador2 = null;
String turno = null;
String remoteId;
String mano;
String ganadorRonda1 = null;
String ganadorRonda2 = null;
String sCartasJ2 = "";
String ganadorEnvid = null;
String desconectado = "";

//TextViews
TextView textoAccion1;
TextView textoAccion2;
TextView nombreJugador1;
TextView nombreJugador2;
TextView fraseAleatoria;
```

- **Conexiones:** En este segundo bloque de código se gestionan las conexiones del juego. Encontramos tanto las conexiones para logear al jugador con Google+ como para buscar partida, invitar amigos, entrar a una sala de juego, desconectarse de una sala, logout, etc. En definitiva, todas las conexiones que el jugador tenga que hacer en referencia al servidor del juego.



Cabe destacar que Game Services ya trae una interfaz definida que hace que todas estas conexiones funcionen perfectamente sin necesidad de que el desarrollador haga ningún cambio. Por otra parte, nosotros hemos modificado algunos de estos puntos para satisfacer las necesidades de nuestro juego. Por ejemplo, cuando un jugador sale de una partida, hemos añadido el envío de un mensaje para que los demás jugadores vean que la partida ha acabado debido a una desconexión. Se puede deducir fácilmente de estos párrafos que Game Services permite adecuar su interfaz a las necesidades del desarrollador. Un punto a su favor sin duda.

```
void startQuickGame4jugadores() {
    // quick-start a game with 3 randomly selected opponent
    final int MIN_OPPONENTS = 3, MAX_OPPONENTS = 3;
    Bundle autoMatchCriteria = RoomConfig.createAutoMatchCriteria(MIN_OPPONENTS,
        MAX_OPPONENTS, 0);
    RoomConfig.Builder rtmConfigBuilder = RoomConfig.builder(this);
    rtmConfigBuilder.setMessageReceivedListener(this);
    rtmConfigBuilder.setRoomStatusUpdateListener(this);
    rtmConfigBuilder.setAutoMatchCriteria(autoMatchCriteria);
    switchToScreen(R.id.screen_wait);
    keepScreenOn();
    desbloquearCartas();
    resetAll();
    //resetTurno();
    Games.RealTimeMultiplayer.create(mGoogleApiClient, rtmConfigBuilder.build());
}
```

En la imagen podemos ver un ejemplo de este bloque de código. Se trata de la función que nos permite buscar una partida rápida de 4 jugadores, con jugadores aleatorios.

La mayoría de esta función estaba definida por la interfaz, pero nosotros añadimos unas pequeñas modificaciones para adecuarlo a nuestra funcionalidad deseada. Fijamos un número de oponentes que debía de buscar el jugador para entrar en la partida y, además, reseteamos las variables del juego para que en la nueva partida todas las variables adquirieran su valor inicial por defecto.

- **Envío de mensajes a otros jugadores:** Para que el juego avance, el jugador que realiza una acción debe de mandar mensajes a los demás jugadores de la partida (en el siguiente punto veremos el por qué). Game Services también nos proporciona ejemplos de cómo mandar un mensaje a los participantes deseados. Nuestra aplicación hace un gran uso de los mensajes entre jugadores durante una partida, por lo que esta funcionalidad es vital para nosotros. Veamos algunos ejemplos de envíos de mensaje:

```
public void enviarMensajeRepartir() {
    byte[] messageRepartir = ("S" + " " + sCartasJ2).getBytes();
    for (Participant p : mParticipants) {
        if (!p.getParticipantId().equals(mMyId)) {
            Games.RealTimeMultiplayer.sendReliableMessage(mGoogleApiClient, null, messageRepartir,
                mRoomId, p.getParticipantId());
        }
    }
}

public void enviarPuntuacionCompañero(int puntosMios, int puntosRival) {
    byte[] messageGanadorRonda4J = ("A " + puntosMios + " "+puntosRival).getBytes();
    for (Participant p : mParticipants) {
        if (!p.getParticipantId().equals(mMyId) && esDeMiEquipo(p.getParticipantId())) {
            Games.RealTimeMultiplayer.sendReliableMessage(mGoogleApiClient, null, messageGanadorRonda4J,
                mRoomId, p.getParticipantId());
        }
    }
}
```





Las imágenes son dos ejemplos de envíos de mensajes. La estructura de un mensaje es sencilla. Simplemente hay que declarar un mensaje a enviar (con un identificador, que en nuestro caso son letras en mayúsculas y los parámetros que se quieren enviar) y el o los destinatarios que los recibirán.

En el primer mensaje, los destinatarios son todos los jugadores de la partida que no sean el que envía, mientras que en el segundo hemos ampliado más el filtro: todos los jugadores que, además, sean de mi equipo.

De esta forma, vemos que también es muy sencillo adaptar los mensajes a nuestras necesidades una vez más.

- **Recepción de mensajes:** En este cuarto bloque, está ubicado el código que se encarga de recibir los mensajes que envíen los demás jugadores. Una vez que un jugador recibe un mensaje, lo clasifica dependiendo del identificador que lleva el mensaje (en nuestro caso, la letra mayúscula) y trata los datos o realiza unas acciones para corresponder el objetivo del mensaje. Veamos un ejemplo.

```
case 'A':  
    String recibido = new String(buf, "UTF-8");  
    String[] datos = recibido.split(" ");  
    puntosTotalesMios = Integer.parseInt(datos[1]);  
    puntosTotalesJugador2 = Integer.parseInt(datos[2]);  
    break;
```

En este caso, se está tratando la recepción del mensaje A. Una vez recibido, se tratan los datos del mensaje utilizando un Split y se almacenan en un array con los datos. Después, simplemente tenemos que utilizar los datos en función de nuestras necesidades. En este ejemplo, el mensaje A nos informa de los puntos que lleva el propio jugador y de los puntos del rival.

- **Lógica del juego:** El último de los bloques, y el que más trabajo nos ha llevado. Aquí es dónde el desarrollador debe “crear” su juego. Es esencial actualizar las variables del juego con el valor que deben tener, por ejemplo, con los puntos del juego. Para ello hacemos uso de los mensajes que hemos explicado anteriormente.

Hay numerosas funciones y uso de las variables definidas en el primer bloque. Aquí es dónde tratamos los datos y hacemos que el juego avance.

En este bloque también se incluye la interfaz gráfica, pero esa parte está explicada detalladamente en la memoria del Front-End.

Para esta parte del juego, la interfaz de Game Services nos indica que incluyamos el código con la lógica pero, como es obvio, la interfaz no nos proporciona ninguna función ya que el juego es totalmente personalizado y sería imposible.

Por último, veremos un ejemplo de una función de este bloque, la de cambiar el turno entre los jugadores. Es una función muy sencilla.

Simplemente se encarga de comprobar el turno actual y pasarlo al siguiente.

```

void cambiarTurno() {
    if (numeroJugadores == 2) {
        cambiarBarraProgreso();
        animarDesaparecerMenu();
        if (mMyId.equals(idJugador1) && turno.equals(idJugador1)) turno = idJugador2;
        if (mMyId.equals(idJugador2) && turno.equals(idJugador2)) turno = idJugador1;
    } else if (numeroJugadores == 4) {
        if (mMyId.equals(idJugador1) && turno.equals(idJugador1)) turno = idJugador2;
        if (mMyId.equals(idJugador2) && turno.equals(idJugador2)) turno = idJugador3;
        if (mMyId.equals(idJugador3) && turno.equals(idJugador3)) turno = idJugador4;
        if (mMyId.equals(idJugador4) && turno.equals(idJugador4)) turno = idJugador1;

        animarDesaparecerMenu();
        cambiarBarraProgreso();
    }
}

```

## 8.4 Funcionamiento del juego

En esta parte vamos a profundizar más todavía en cómo funciona el juego.

Lo primero a tener en cuenta son las variables de Game Services, que son esenciales en el funcionamiento de la partida, ya que contienen la información de las conexiones, el ID del jugador, la sala de juego, etc.

Hay cuatro que son las más importantes, pasemos a analizarlas.

```

// Cliente usado para interactuar con Google APIs.
private GoogleApiClient mGoogleApiClient;

// Creamos el nuevo cliente de Google con acceso a Plus y Games
mGoogleApiClient = new GoogleApiClient.Builder(this)
    .addConnectionCallbacks(this)
    .addOnConnectionFailedListener(this)
    .addApi(Plus.API).addScope(Plus.SCOPE_PLUS_LOGIN)
    .addApi(Games.API).addScope(Games.SCOPE_GAMES)
    .build();

```

La variable **mGoogleApiClient** contiene el código único de cliente que se usará para conectar con los servidores de Game Services. Cada cliente que se conecte a la aplicación tendrá un código distinto.

De esta forma, el servidor podrá identificar a cada uno de los jugadores, recuperando sus puntuaciones, datos, conexiones, etc.

```

//Id de mi participante
String mMyId = null;

//Contiene el ID del jugador del dispositivo
mMyId = room.getParticipantId(Games.Players.getCurrentPlayerId(mGoogleApiClient));

```

mMyId contiene el ID del jugador del dispositivo. Es una de las variables que más se usan en todo el juego, ya que hay que hacer numerosas comparaciones con distintas variables.



Un ejemplo muy claro lo tenemos al comprobar el turno o qué jugador va de mano. ¿Cómo comprobamos si es mi turno o si voy de mano? Hay que hacer una comparación entre la variable turno o mano y `mMyId`. También puede servir para comprobar cuál de los 2 o 4 jugadores eres. Repetimos, es una variable esencial.

```
//Lista de participantes en el juego activo
ArrayList<Participant> mParticipants = null;

//Contiene la lista de los participantes de la sala
mParticipants = room.getParticipants();
```

La variable `mParticipants` tiene una función muy importante cuando empieza la partida. A partir de ahí, ya no tiene demasiada relevancia en el juego.

En una partida de 2 o 4 jugadores, es importante asignar a cada uno de los dispositivos un ID de jugador. En nuestro código, éstos son conocidos como `idJugador1`, `idJugador2`, etc. En el momento en que empieza una nueva partida, es necesario asignarlos. ¿Cómo lo hacemos? Gracias a la variables `mParticipants`.

Es imprescindible que en cada dispositivo se sepa que ID de jugador corresponde a cada uno de los restantes dispositivos, para tener coherencia en el juego. Para ello, y gracias a `mParticipants`, recorreremos la lista de jugadores de la partida y los ordenamos por orden gracias a un `compareTo` (los ID son cadenas de caracteres) y, una vez ordenados, asignamos uno a cada jugador, de forma que en todos los dispositivos los IDs de jugadores son equivalentes.

La segunda función de `mParticipants` en *Retruque!* es obtener los datos de cada jugador. En la pantalla de juego, los jugadores tienen un nombre y una imagen de perfil. Pues bien, obteniendo los jugadores desde `mParticipants`, podemos extraer estos datos de cada uno de los jugadores para luego mostrarlos en su icono de jugador. ¿Útil, verdad?

```
// ID de la sala activa, null si no estamos jugando
String mRoomId = null;

//Contiene el ID de la sala de juego
mRoomId = room.getRoomId();
```

Por último, tenemos la variable `mRoomId`, que se encarga de obtener el ID de la sala de juego. Su principal función es, obviamente, obtener el ID de la sala del juego. Todos los jugadores de la misma partida tendrán el mismo valor de `mRoomId`.

Además, cuando se envía un mensaje a los demás jugadores hay que pasar como parámetro el ID de la sala de juego, cosa que es lógica. No tendría sentido que un jugador recibiera un mensaje de un jugador ajeno a su propia partida.

Por último, cuando una partida acaba o un jugador sale, su valor cambia a `null` esperando para la próxima conexión a otra sala.

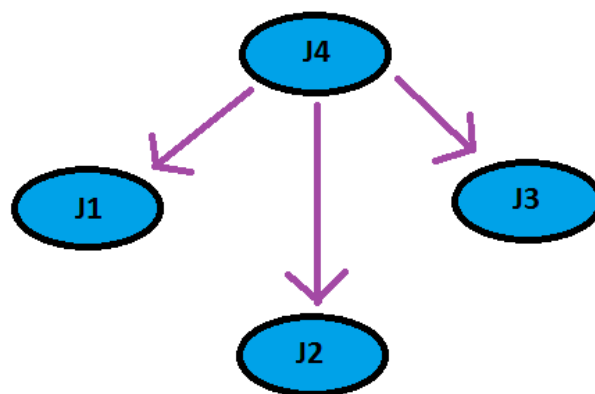
Una vez que ya conocemos las variables necesarias para que *Game Services* funciones correctamente y nos permita conectar a los jugadores, podemos pasar a explicar cómo avanza el juego.

El Truc es un juego en el que, aparte de tirar las cartas sobre la mesa como en cualquier juego de cartas, es posible proponerle al rival distintas “apuestas” para que una mano sume más puntos al que la gane. Hablamos de trucar y envidar. Para recordar lo que era, revisar el punto 3.2 de esta memoria.

Pero vayamos paso a paso. Nos centraremos en explicar una partida de 4 jugadores, ya que tiene la misma base que la de 2, pero con mayor complejidad.

### Reparto de cartas entre los jugadores

Lo primero que ocurre cuando entramos a una partida es que en nuestra mano debe de haber 3 cartas, con las que jugaremos la ronda. Nuestro código hace que el jugador al que se le asigna la mano, calcule 3 cartas para él mismo y para cada uno de los demás jugadores y se lo notifique. Siguiendo este esquema:



En este ejemplo, el jugador 4 es la mano, por lo que calculará 12 cartas en total y enviará 3 mensajes, uno para cada uno de los demás participantes. Así, cada uno tendrá sus 3 cartas.

### Envío de señas

Una vez que tenemos las cartas, empieza lo que llamamos el “Tiempo de señas”, donde cada jugador puede mandar una o dos señas a su compañero, bien para decirle las cartas que tiene o la estrategia a seguir o bien para engañar al equipo rival. Cada jugador es libre de elegir las señas que quiere mandar, y éstas son enviadas a todos los demás jugadores, siguiendo el mismo patrón que la imagen anterior.

A todos los jugadores... ¿para qué? En el juego real, es posible mirar a un jugador contrario y ver las señas que hace con la cara, ¿por qué no iba a ser posible en Retruque!? Una vez mandadas las señas, el compañero las verá sin problemas durante un par de segundos, mientras que los rivales verán un globo con dos interrogantes dentro y, si son capaces de pulsarlo a tiempo (tarea muy complicada), podrán pillar la seña como en la realidad.

En la siguiente imagen podemos ver una captura del tiempo de señas para entenderlo mejor:



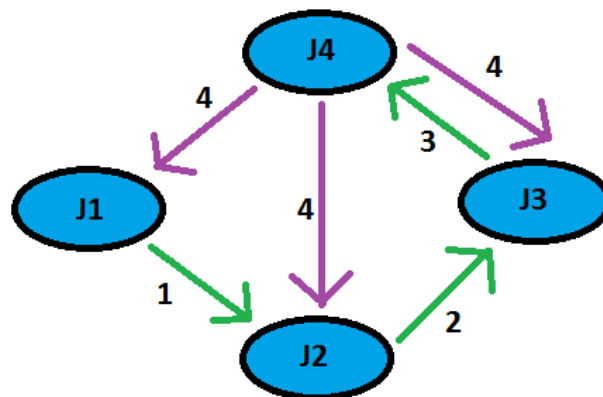
Los jugadores tienen un tiempo de 20 segundos para decidir que señas quieren mandar. Si todos los jugadores mandan las señas antes de los 20 segundos, se cancelan los callbacks para que nadie siga esperando y pueda iniciarse la partida.

Una vez que se han mandado las señas, tiene el turno el jugador que va de mano y éste irá pasando de jugador en jugador al tirar una carta sobre la mesa. El jugador puede realizar distintas acciones como tirar una carta, envidar, trugar, tirar la falta (tipo de envid) y algunas más que será más interesante contar en la memoria del Front-End.

Para cada una de las acciones que realice el jugador, habrá un distinto tipo de intercambio de mensajes. Pasemos a analizarlos.

#### Tirada de carta a la mesa y ganador de ronda

Cuando el jugador tira la carta sobre la mesa, simplemente mandará un mensaje a todos los participantes como en la primera imagen que hemos visto, notificando así de la carta que sus dispositivos deben de mostrar sobre la mesa. Pero cuando todos han tirado, deben de saber quién ha ganado la ronda, ya que el equipo que gana 2 de las 3 rondas de una mano, la gana y suma un punto a su favor. Para ello, decidimos hacerlo de la siguiente manera:



Como vemos en la imagen, los jugadores mandan otro mensaje aparte que van avanzando hasta llegar al último. Éste, que tiene todos los datos de los demás jugadores, realiza unas funciones y calcula quién ha ganado la ronda. Tras ello, notifica a todos los demás jugadores quién es el ganador de la ronda.

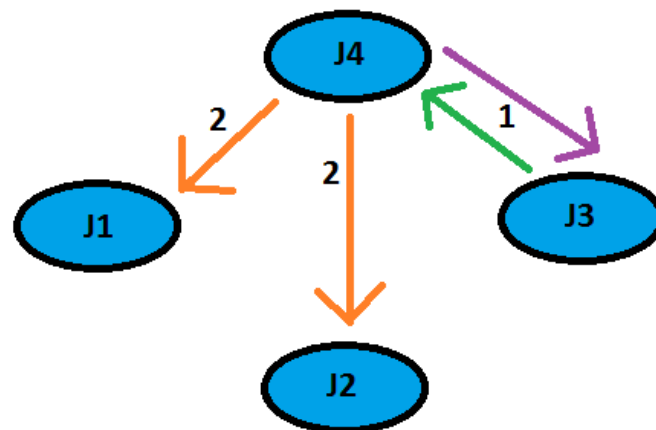
Este proceso se repetirá hasta que alguno de los dos equipos gane 2 de las 3 rondas. Entonces, el último jugador que tenga el turno mandará un mensaje complementario a todos los jugadores, indicándoles que actualicen los puntos de sus marcadores (el mensaje incluye los puntos de los dos equipos) y que toca repartir y empezar una nueva mano.

### Acción de envidar

Pasemos ahora a ver que ocurre cuando un jugador envida. El envid únicamente pueden realizarlo los dos últimos jugadores que tengan el turno de cada equipo en la misma ronda. Es decir, si empezó el jugador 1, podrán envidar los jugadores 3 y 4.

Si uno de estos dos envida, el otro será el que conteste y el que pueda subir la apuesta.

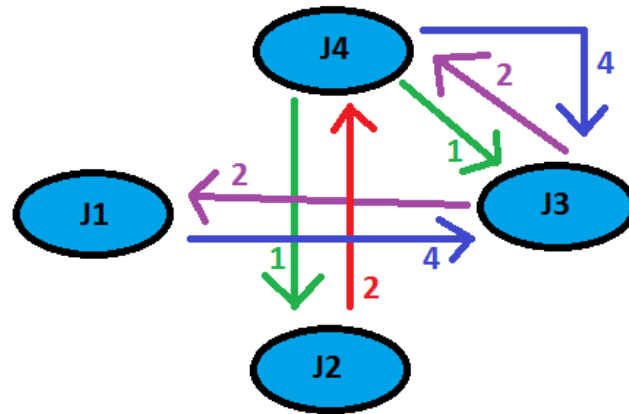
La estructura de mensajes para una acción de envidar es la siguiente:



En este ejemplo, envida el jugador 4. Por tanto, se envía una serie de mensajes con el jugador 3 y, una vez haya acabado el flujo de mensajes, bien porque uno de los dos haya querido el envid o bien porque no lo haya querido, la respuesta se notifica a los jugadores restantes. Aceptar el envid causará que los jugadores que ganen la mano ganen una mayor cantidad de puntos.

### Acción de trucar

Por último, hablemos de la acción de trucar, la más compleja del juego debido al gran intercambio de mensajes que hay entre los jugadores.



En este ejemplo, el jugador que truca es el J4. A diferencia que con el envid, el truc sí que se pregunta a los dos jugadores del equipo contrario, por lo que debe mandar dos mensajes (uno a cada uno de los rivales). El J3 retrucará (subirá la apuesta) y el J2 no querrá el truc.

Como los dos jugadores podían optar por opciones distintas como en este ejemplo, nuestra solución fue aceptar la respuesta afirmativa más alta. Es decir, si un jugador quiere y otro retruca, aceptar el retruc o si un jugador quiere pero otro no, aceptar el quiero, etc.

En este caso, el J3 retruca y, por tanto, ahora son los dos jugadores del equipo contrario los que deben elegir qué hacer, por lo que el J3 manda un mensaje a cada uno de ellos y espera respuesta.

El J4 responde con un quiero y el J1 con un no quiero. Aquí acaba el flujo de mensajes de la acción de trucar.

Como ya hemos adelantado previamente, es la acción más compleja del juego, ya que requiere un gran número de mensajes y de respuestas entre los dos jugadores de los dos equipos hasta acordar una respuesta final.

Este proceso se puede repetir hasta cuatro veces en una misma acción de trucar, ya que el truc tiene distintos niveles: trucar, retrucar, quatre val y joc fora. Puede llegar a hacerse muy complejo, por lo que teníamos que tener muy clara la idea.

### Ganador de la partida

Cuando alguno de los dos equipos llega a los 24 puntos acumulados, gana la partida. Para que los demás jugadores sepan que ha acabado, es necesario notificar el resultado de la partida.

Siempre que el último jugador al que le toca el turno en la ronda calcula los puntos para mandárselos a los demás, también comprueba si alguno de los dos equipos ha ganado la partida (comprobando si alguno de los dos marcadores llega a 24).

Cuando alguno de los dos equipos ha ganado, en sus pantallas se mostrará un icono de color verde, mientras que a los perdedores se les mostrará un icono de color rojo.





Junto a ellos, aparecerá un mensaje de enhorabuena o de consuelo y un sonido de victoria o de derrota.

Como último detalle, si la partida finaliza por una desconexión de un jugador, también aparecerá este mensaje para que los jugadores sepan el motivo de la finalización de la partida.

### **Desconexión de un jugador**

Como ya se ha comentado en varias ocasiones, cuando un jugador abandona la partida, ésta debe finalizar. No tendría ningún sentido que continuara con 3 jugadores en vez de los 4 necesarios.

Aquí es cuando se ejecuta un método de la interfaz proporcionada por Game Services: `leaveRoom`. Como su propio nombre indica, la función de `leaveRoom` es que el jugador que sale deje la sala de juego. Como necesitábamos que los demás jugadores supieran que la partida había acabado, modificamos esta función añadiendo el envío de un mensaje de desconexión a todos los jugadores de la partida, para que ellos también salieran.

Por último, la variable `mRoomId` adquiere el valor null, para que cuando se busque una nueva partida pueda volver a tener un nuevo código de sala.

### **Cambio de turno y de mano**

Son dos de las acciones más simples del juego. Cuando un jugador tira una carta sobre la mesa, manda un mensaje a los demás jugadores actualizando su variable `turno`. Cada jugador comprueba si la variable `turno` equivale a su ID de jugador y, por tanto, sabe si le toca o no le toca.

Lo mismo ocurre con el cambio de mano. El jugador mano envía un mensaje a todos actualizando su variable `mano` y estos comprobarán si su ID de jugador equivale al valor de mano.

### **Envío de frases**

Para aumentar la diversión y la competitividad entre los jugadores de la partida, Retruque! posee un sistema de envío de frases. Cuando un jugador tiene el turno, en su menú podrá seleccionar un botón de frases y le aparecerá una lista con las que puede mandar. Si selecciona una, un mensaje será mandado a todos los jugadores, y en su icono de jugador aparecerá un bocadillo con la frase que seleccionó previamente.

Actualmente, hay 16 frases distintas en el juego.

### Variables de la lógica del juego

Una vez que hemos explicado las principales acciones del juego, finalizaremos este punto 8.4 de la memoria analizando algunas de las variables más importantes de la lógica del juego.


- **ronda:** Variable de tipo int que indica el número de ronda actual. Sus valores oscilan entre el 1 y el 3. Se actualiza mediante un mensaje que manda el último jugador que tiene el turno de la ronda.
- **mano:** String que contiene el ID de uno de los jugadores, que será el que sea el mano.
- **numeroJugadores:** una de las variables más importantes de la lógica. Es un entero que tiene como valor 2 o 4 e indica el número de jugadores de la partida. Algunas funciones ejecutan unas acciones u otras dependiendo de si la partida es de 2 o de 4, por lo que esta variable es esencial para el juego.
- **idJugador1, idJugador2, idJugador3, idJugador4:** son cadenas que contienen los IDs de los jugadores de la partida. Se asignan en cuanto empieza y no cambian su valor hasta que ésta finaliza.
- **puntosTruc:** Entero que indica en cada nueva mano el valor de los puntos que se sumarán al jugador que la gane. Si no se hace la acción de trucar, valdrá 1. Si se hace, cambiará su valor dependiendo del nivel del truc al que se llegue a apostar entre los jugadores.
- **puntosEnvid:** tiene la misma función que puntosTruc, pero hay una diferencia: si ningún jugador envida, su valor será 0. En caso de que sí que haya envid, variará dependiendo de la apuesta.
- **ganadorRonda1:** String que almacenará el ID del jugador que gane la primera ronda de la mano. Es importante almacenarlo, ya que en caso de empate en las rondas posteriores, el jugador o el equipo que haya ganado la primera será el que gane la mano.
- **hayEmpate:** variable booleana que indica si hay empate en una ronda, para consultar el ganador de la primera ronda y decidir qué equipo gana.
- **hayEnvid:** indica si alguno de los jugadores ha envidado, para sumar al cúmulo de puntos de la mano los puntos del envid o no.
- **hayTruc:** tiene la misma función que hayEnvid, pero para los puntos del truc.
- **carta1, carta2, carta3:** son los objetos de tipo Carta que tiene cada jugador en su mano. Variarán cada vez que se cambie de mano.
- **baraja:** Es un ArrayList de objetos Carta que contiene todas las cartas de la baraja. Su principal función es a la hora de repartir las cartas a los jugadores.

Obviamente, el juego contiene muchas más variables, pero aquí hemos destacado las que resultan más importantes para la partida.

En los siguientes puntos, pasaremos a analizar otras de las funciones que nos proporciona Google Play Game Services y que también hemos incluido en el juego.

### 8.5 Logros

Los logros son unos objetivos del juego a los que se reta al usuario a conseguirlos y demostrar que es un gran jugador. Su dificultad puede variar, siendo posible conseguir algunos en la primera partida y otros en la número 70.



**Retruque!**  
648419090213

Publicado ▾

Estadísticas de jugadores		LOGROS <span style="float: right; background-color: #007bff; color: white; padding: 2px 5px; border-radius: 3px;">Añadir nuevo logro</span>					
Descripción general		#	NOMBRE	ID <small>?</small>	PUNTOS	% DESBLOQUEADO: N.º JUGADORES EN TOTAL / TIEMPO <small>?</small>	ESTADO
Detalles de ingresos		1	Primera partida	Cgklpb6oxu8SEAIQCC	5	8% 5 / 1,1 hora	✓ Publicada
Datos demográficos		2	¡Marea azul!	Cgklpb6oxu8SEAIQCg	5	—	✓ Publicada
Estadísticas de funciones		3	Mirón principiante	Cgklpb6oxu8SEAIQCw	5	—	✓ Publicada
Descripción general		4	Mirón profesional	Cgklpb6oxu8SEAIQDA	15	2% 1 / 0 minutos	✓ Publicada
Detalles de interacción		5	Mirón experto	Cgklpb6oxu8SEAIQDQ	30	—	✓ Publicada
Misiones		6	Gana con 33	Cgklpb6oxu8SEAIQDg	10	5% 3 / 10 minutos	✓ Publicada
Información del juego		7	La casa por la ventana	Cgklpb6oxu8SEAIQDw	10	5% 3 / 6 minutos	✓ Publicada
Aplicaciones vinculadas		8	Triunfador principiante	Cgklpb6oxu8SEAIQEA	15	3% 2 / 0 minutos	✓ Publicada
Eventos		9	Triunfador avanzado	Cgklpb6oxu8SEAIQEQ	30	—	✓ Publicada
Logros							
Marcadores							
Testing							
Publicar							

Añadir un logro es una tarea muy sencilla. Desde la Developer Console, nos vamos al apartado Logros y, posteriormente, pulsamos sobre el botón Añadir nuevo logro. Se nos abrirá una nueva pantalla como la siguiente:

**Nombre**

Inglés (Estados Unidos) – en-US

19 de 100 caracteres

---

**Descripción**

Inglés (Estados Unidos) – en-US  
(opcional para prueba)

Gana 15 partidas de ¡Retruque! jugando la partida completa

58 de 500 caracteres

---

**Icono ?**

512x512  
png o jpg  
(opcional para prueba)

---

**Logros incrementales**

15 pasos para desbloquear el logro ?

---

**Estado inicial**

Visible ?

---

**Puntos ?**

345 de 1.000 puntos de logros distribuidos  
El valor de los puntos debe estar comprendido entre 5 y 200 y debe ser múltiplo de 5.

En ella, debemos de rellenar los datos que nos piden. Tenemos también que asignarle un número de puntos que ganará el jugador al desbloquearlo. Incluso se le puede adjuntar una imagen de icono.

Actualmente, Retruque! posee 15 logros, que vamos a detallar a continuación:

- **Primera partida:** Gana tu primera partida.
- **¡Marea azul!:** Has tenido la marea azul en tus manos.



- Mirón principiante: Has pillado una seña a alguno de tus rivales.
- Mirón profesional: Has pillado 5 señas a alguno de tus rivales. (Incremental)
- Mirón experto: Has pillado 10 señas a alguno de tus rivales. (Incremental)
- Gana con 33: Gana un envid con 33 de puntuación.
- La casa por la ventana: manda un joc fora a tu rival.
- Triunfador principiante: Gana 5 partidas de Retruque! jugando la partida completa.
- Triunfador avanzado: Gana 15 partidas de Retruque! jugando la partida completa. (Incremental)
- Triunfador experto: Gana 30 partidas de Retruque! jugando la partida completa. (Incremental)
- Leyenda del truc: Gana 50 partidas de Retruque! jugando la partida completa. (Incremental)
- Sobrado: Gana por más de 10 puntos de diferencia.
- Farolero: Envida sin tener dos cartas del mismo palo y gana el envid.
- Al límite: Tira la falta a tu rival.
- Estratega: Manda 50 señas a tu compañero. (Incremental)

Obviamente, para desbloquearlos hay que conseguir el objetivo de su descripción.

Como se puede observar, existen dos tipos de logros: los incrementales y los no incrementales.

Los logros incrementales son aquellos en los que hay que seguir una serie de pasos para desbloquearlos, por ejemplo, el logro de Triunfador avanzado, en el que tenemos que ganar 15 partidas para obtenerlo.

Los logros no incrementales son los que podemos desbloquear con una sola acción, como el logro Al límite.

Una vez que hemos creado los logros en la Developer Console, ahora debemos introducirlos en el código de nuestro juego, para así decirle cuando deben ser desbloqueados o incrementados. De la siguiente manera:

```
/****** LOGROS *****/
final static String PRIMERA_PARTIDA = "CgkIpb6oxu8SEAIQCQ";
final static String MAREA_AZUL = "CgkIpb6oxu8SEAIQCg";
final static String MIRON_PRINCIPIANTE = "CgkIpb6oxu8SEAIQCw";
final static String MIRON_PROFESIONAL = "CgkIpb6oxu8SEAIQDA";
final static String MIRON_EXPERTO = "CgkIpb6oxu8SEAIQDQ";
final static String ENVID_33 = "CgkIpb6oxu8SEAIQDg";
final static String CASA_POR_LA_VENTANA = "CgkIpb6oxu8SEAIQDw";
final static String TRIUNFADOR_PRINCIPIANTE = "CgkIpb6oxu8SEAIQEA";
final static String TRIUNFADOR_AVANZADO = "CgkIpb6oxu8SEAIQEQ";
final static String TRIUNFADOR_EXPERTO = "CgkIpb6oxu8SEAIQEG";
final static String LEYENDA = "CgkIpb6oxu8SEAIQEW";
final static String SOBRADO = "CgkIpb6oxu8SEAIQFA";
final static String FAROLERO = "CgkIpb6oxu8SEAIQFQ";
final static String AL_LIMITE = "CgkIpb6oxu8SEAIQFG";
final static String ESTRATEGA = "CgkIpb6oxu8SEAIQGA";
```

Cuando creamos un logro, a éste se le asigna un código único. Debemos de definirlo en nuestro código para poder trabajar con ese logro. En nuestro caso, nos definimos 16 variables final para ellos.

Una vez definidas estas variables, llega el momento de colocar en el código dónde debemos desbloquear el logro o dónde incrementarlo. Veamos un ejemplo:

```
Games.Achievements.unlock(mGoogleApiClient, PRIMERA_PARTIDA);
Games.Achievements.increment(mGoogleApiClient, TRIUNFADOR_PRINCIPIANTE, 1);
Games.Achievements.increment(mGoogleApiClient, TRIUNFADOR_AVANZADO, 1);
Games.Achievements.increment(mGoogleApiClient, TRIUNFADOR_EXPERTO, 1);
Games.Achievements.increment(mGoogleApiClient, LEYENDA, 1);
```

En la imagen vemos la estructura de los logros. Tenemos tanto incrementales como no incrementales. En este caso, cuando un jugador gana la partida, le decimos que desbloquee el logro Primera Partida (sólo será desbloqueado si no lo ha obtenido todavía) y que incremente en uno el contador de los logros Triunfador Principiante, Triunfador Avanzado, Triunfador Experto y Leyenda del truc.

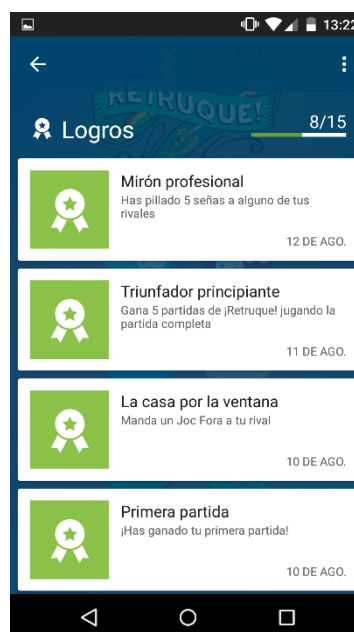
Los logros incrementales serán desbloqueados automáticamente cuando alcancen el número fijado en la Developer Console.

Para concluir este apartado, veremos cómo añadir la lista de logros a nuestro juego:

```
startActivityForResult(Games.Achievements.getAchievementsIntent(mGoogleApiClient),
    REQUEST_ACHIEVEMENTS);
```

Simplemente debemos iniciar una nueva actividad, a la que le pasaremos nuestro cliente y un código identificativo para el Intent, que no tiene que tener ningún valor específico.

La interfaz de Game Services buscará en el apartado de Logros definidos en nuestro juego desde la Developer Console y nos mostrará una lista con los logros, indicando los que hemos desbloqueado y el porcentaje que llevamos acumulado para los incrementales.



## 8.6 Ranking de jugadores

Otra posibilidad muy interesante que nos ofrece Game Services es la de añadir un ranking de jugadores a nuestro juego. Desafiar a los jugadores a llegar a las primeras posiciones de un ranking es un punto muy a favor de un juego, creando adicción y ganas de superar a los demás entre ellos.

En nuestro caso, añadimos un ranking de victorias, pero Game Services permite añadir varios rankings para el mismo juego. Veamos cómo lo hicimos.

El primer paso es crear un ranking desde la Developer Console, de forma parecida a como hicimos con los logros.

MARCADORES [Añadir nuevo marcador](#)

Página 1 de 1

N.º	NOMBRE	ID	PUNTUACIONES	ESTADO
1	<input type="checkbox"/> Ranking	Cgklpb6oxu8SEAIQCA	14	✓ Publicada

[Obtener recursos](#)

Página 1 de 1

**Nombre**  
Inglés (Estados Unidos) – en-US

Ranking  
7 de 100 caracteres

**Formato de puntuación**

Numérico  
Número de decimales: 0

Vista previa: 123,450,000

[Añadir unidad personalizada](#)

**Icono**  
512x512  
png o jpg  
(opcional)

Añadir icono  
Arrastra una imagen hasta aquí

Si no se proporciona un icono, los usuarios verán un icono de marcador estándar.

**Orden**  
Más es mejor

**Habilitar protección contra manipulaciones**  
SÍ NO

**Límites**  
(opcional)

No permitir puntuaciones inferiores a este valor: 0  
No permitir puntuaciones superiores a este valor:

Para no establecer límites, debes dejar los campos en blanco.

Como vemos en las imágenes, existen distintas posibilidades para personalizar un ranking. Hay que asignarle un nombre, decir el tipo de valores que va a coger (en nuestro caso las puntuaciones son numéricas y sin ningún decimal), un icono opcional, indicar si es mejor más o menos puntuación (hay juegos en los que podría ser mejor una puntuación baja, por ejemplo, el tiempo en pasarse un nivel) y un intervalo límite de puntuaciones. Rellenando este sencillo formulario ya tenemos el ranking creado y listo para ser añadido a nuestro juego.

En este caso, sí que es importante coger el ID del ranking (ver la primera imagen) ya que, como existe la posibilidad de crear varios, hay que indicarle al código cuál queremos mostrar. Veamos como introducirlo en el juego:

```
/****** RANKING *****/  
final static String LEADERBOARD_ID = "CgkIpb6oxu8SEAIQCA";
```

Igual que con los logros, nos declaramos una variable final con el ID de nuestro ranking.

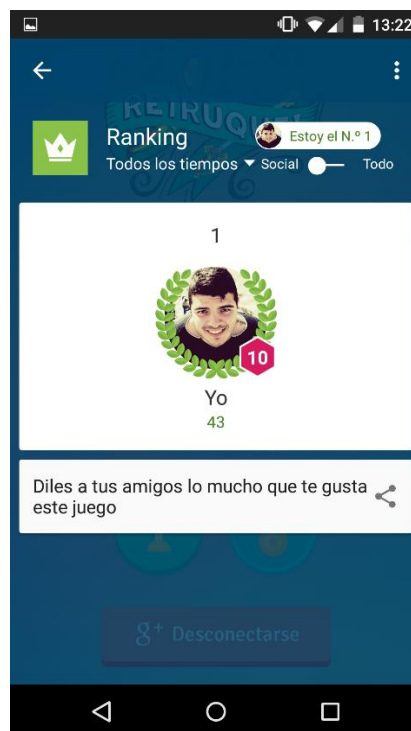
```
startActivityForResult (Games.Leadersboards.getLeaderboardIntent (mGoogleApiClient,  
LEADERBOARD_ID), REQUEST_LEADERBOARD);
```

Para mostrar el ranking, tenemos que crear una nueva actividad, pero en este caso también hay que pasarle en ID identificador el ranking. En nuestro caso, la variable LEADERBOARD\_ID.

```
//Actualiza el ranking sumando una victoria  
updateLeaderboards (mGoogleApiClient, LEADERBOARD_ID);
```

Para actualizar el ranking, usamos el método de la interfaz de Game Services: updateLeaderboards, al que le pasamos nuestro cliente y el ID del ranking que queremos actualizar.

Si todo ha ido bien, podremos ver el ranking desde nuestro dispositivo móvil como en la siguiente imagen:



Por último, hay que añadir que el ranking nos deja compararnos con nuestros amigos (gracias a los círculos de Google+) o con todos los jugadores del juego, cambiando a la pestaña Todo.

## 8.7 Jugar con amigos

Game Services nos permite invitar a nuestros amigos a una partida multijugador de nuestro juego. También podemos unirnos a una partida si nos invitan ellos. Y lo mejor



de todo: su interfaz lo trae todo ya preparado para funcionar desde un principio, sin necesidad de desarrollarlo. Vamos a ver cómo funciona.

En nuestro menú principal tenemos dos botones para jugar con amigos: el botón de invitar a jugadores y el botón de ver las invitaciones que has recibido. Aquí vemos qué código ejecuta cada uno de ellos:

```
case R.id.button_invite_players:
    switchToScreen(R.id.screen_wait);

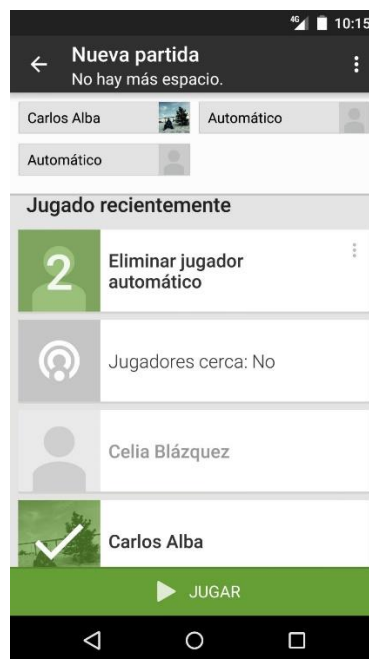
    // Muestra la lista de jugadores que pueden ser invitados
    intent = Games.RealTimeMultiplayer.getSelectOpponentsIntent(mGoogleApiClient, 1, 3);
    startActivityForResult(intent, RC_SELECT_PLAYERS);
    break;

case R.id.button_see_invitations:
    switchToScreen(R.id.screen_wait);

    // Muestra la lista de invitaciones pendientes
    intent = Games.Invitations.getInvitationInboxIntent(mGoogleApiClient);
    startActivityForResult(intent, RC_INVITATION_INBOX);
    break;

case R.id.button_accept_popup_invitation:
    // Aceptar la invitaciones desde la barra popup
    acceptInviteToRoom(mIncomingInvitationId);
    mIncomingInvitationId = null;
    break;
```

Cuando pulsamos el botón de invitar jugadores, se inicia una nueva actividad, que muestra una lista de nuestros amigos de Google+, permitiendo mandarles una invitación de juego.



Si nos fijamos en el código, vemos un 1 y un 3, que hacen referencia al mínimo y al máximo de jugadores que podremos invitar a la partida, esta es la única modificación que hemos añadido. El objetivo es que se puede invitar tanto a un solo jugador para una partida de 2 o a 3 jugadores para jugar una partida por parejas.

Si queremos jugar una partida de 4 y sólo tenemos a un amigo o dos disponibles, es posible añadir jugadores automáticos, que el servidor de Game Services buscará para llenar el cupo de jugadores permitidos.

Por último, es posible hacer una búsqueda de jugadores cercanos e invitarles a una partida. ¡Podemos jugar incluso con nuestros vecinos!

Cuando recibimos una invitación de algún jugador, se nos mostrará un popup en la parte superior de la pantalla con un mensaje que nos lo indique, así como un botón de aceptar que nos permitirá unirnos directamente a la partida de ese jugador.

```
@Override
public void onInvitationReceived(Invitation invitation) {
    // We got an invitation to play a game! So, store it in
    // mIncomingInvitationId
    // and show the popup on the screen.
    mIncomingInvitationId = invitation.getInvitationId();
    ((TextView) findViewById(R.id.incoming_invitation_text)).setText(
        invitation.getInviter().getDisplayName() + " " +
        "te ha invitado a una partida");
    switchToScreen(mCurScreen); // This will show the invitation popup
}
```

De eso se encarga el método `onInvitationReceived`, que hemos personalizado para que la estructura de nuestro mensaje sea: “Nombre” te ha invitado a una partida.

Si no aceptamos la invitación desde el popup, ésta estará disponible en la pantalla de invitaciones recibidas que, como vemos en el primer bloque de código de este punto, es otra actividad distinta.



```
@Override
public void onConnected(Bundle connectionHint) {

    // register listener so we are notified if we receive an invitation to play
    // while we are in the game
    Games.Invitations.registerInvitationListener(mGoogleApiClient, this);

    if (connectionHint != null) {
        Invitation inv = connectionHint
            .getParcelable(Multiplayer.EXTRA_INVITATION);
        if (inv != null && inv.getInvitationId() != null) {
            // retrieve and cache the invitation ID
            acceptInviteToRoom(inv.getInvitationId());
            return;
        }
    }
    switchToMainScreen();
}
```

Si abrimos la pantalla de invitaciones recibidas y aceptamos una, ejecutaremos el método `onConnected` que, a partir de un código de invitación, nos permite unirnos a la misma sala en la que está el jugador que nos invita.

Si nos fijamos, dentro de este método, tenemos la función `acceptInviteToRoom`, que también es llamada desde el botón aceptar del popup. Esta función es la que nos hace aceptar la invitación y unirnos a la partida.

```
// Accept the given invitation.
void acceptInviteToRoom(String invId) {
    // accept the invitation
    Log.d(TAG, "Accepting invitation: " + invId);
    idCompanero = invId;
    RoomConfig.Builder roomConfigBuilder = RoomConfig.builder(this);
    roomConfigBuilder.setInvitationIdToAccept(invId)
        .setMessageReceivedListener(this)
        .setRoomStatusUpdateListener(this);
    switchToScreen(R.id.screen_wait);
    keepScreenOn();
    Games.RealTimeMultiplayer.join(mGoogleApiClient, roomConfigBuilder.build());
}
```

Como vemos, la interfaz de Game Services nos proporciona este trabajo de conexiones prácticamente hecho, únicamente tenemos que editarlo un poco para adaptarlo a las necesidades de nuestro juego.

## 8.8 Motivo de la elección y evaluación

Una vez que ya conocemos lo que es Google Play Game Services y cómo funciona en nuestro juego, explicaremos por qué decidimos utilizar esta tecnología.

Como comentamos al principio de esta memoria, nunca habíamos tomado las riendas de un proyecto de este tipo, haciendo un juego con multijugador en línea. No sabíamos cómo hacer las conexiones entre jugadores, cómo meterles en una misma partida, cómo

almacenar sus puntuaciones, etc. Nos iba a resultar complicado empezar el desarrollo, ya que la estructura y las características que debía de tener sí que las teníamos claras.

Tras mucho investigar, contratar un servidor propio e incluso intentar empezar el desarrollo de las conexiones, las salas, etc. Vimos que se nos iba de las manos, que no sabíamos hacia dónde ir ni cómo hacerlo.

Decidimos seguir investigando para encontrar alguna tecnología que nos facilitara la faena, o que por lo menos nos ayudara en el proceso.

Nos encontramos una API que había creado Google llamada Google Play Game Services, que nos ayudaba o incluso nos gestionaba automáticamente los puntos que no sabíamos gestionar bien y además nos permitía añadir logros, un ranking y muchas más funciones que en principio no han sido incluidas pero que en próximas versiones podría pensarse.

Al principio nos costó un poco arrancar y entender su funcionamiento, pero al poco tiempo nos dimos cuenta que nos estaba ayudando muchísimo y que, prácticamente, podíamos centrarnos en la lógica del juego y en conseguir una interfaz trabajada, sin perder demasiado tiempo en las conexiones entre jugadores y este tipo de cosas.

Por tanto, y como ya se ha visto en todo este punto 8 de la memoria, Game Services nos ha sido muy útil y de gran ayuda. Google se ha tomado muy en serio esta tecnología y ha conseguido una API muy potente y que cada vez más, este trabajo se verá reflejado con su creciente uso en los juegos móviles.

Si en el futuro hacemos otro juego, es muy probable que volvamos a utilizar esta tecnología. Tomamos la decisión correcta al elegirla.



## 9. Publicación en play store

Llegados a este punto, en el que la aplicación ya está acabada, debemos empezar a pensar cómo compartirla con el mundo para que los usuarios de los dispositivos móviles Android puedan descargarla y disfrutar de ella. Para ello, la subiremos a Google Play Store. Veamos qué es y cuál es el procedimiento a seguir para subir nuestra App.

### 9.1 ¿Qué es play store?



Google Play Store (anteriormente Android Market) es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google. Esta plataforma permite a los usuarios navegar y descargar aplicaciones (desarrolladas mediante Android SDK), juegos, música, libros, revistas y películas. También se pueden adquirir dispositivos móviles como ordenadores Chromebook, teléfonos inteligentes Nexus, Google Chromecast, entre otros.

Las aplicaciones se encuentran disponibles de forma gratuita, así como también con costo. Pueden ser descargadas directamente desde un dispositivo con Android a través de la aplicación móvil Play Store.

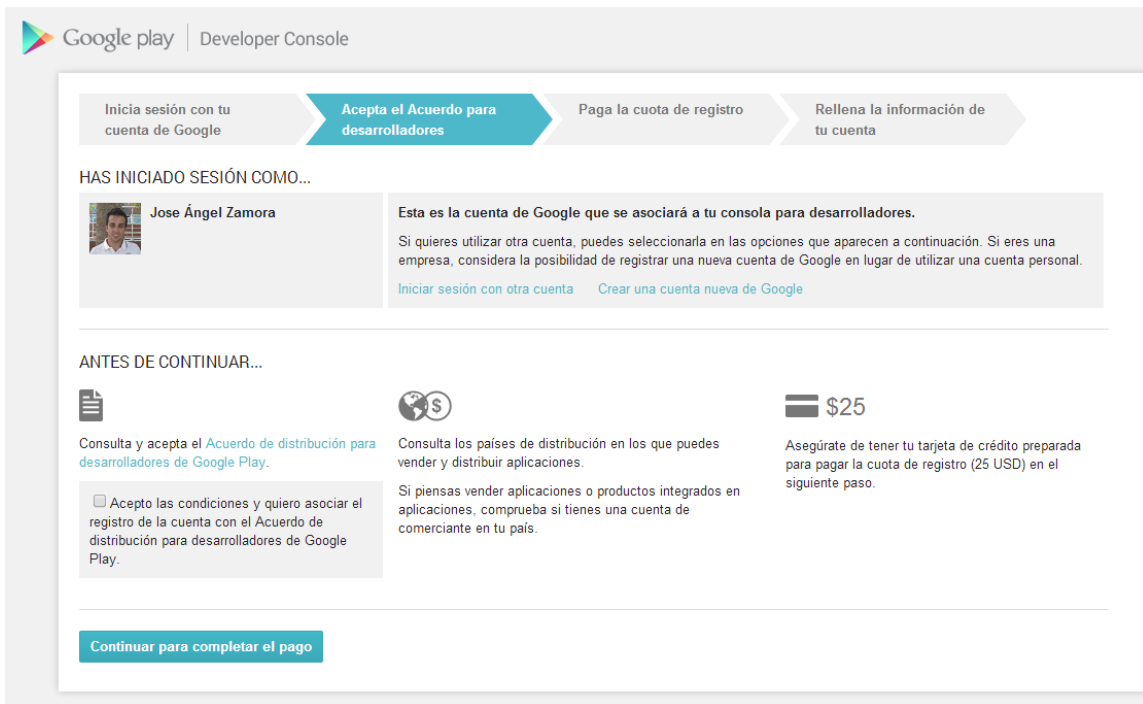
En marzo de 2012, con la fusión de Android Market con Google Music, el servicio fue renombrado a Google Play, como resultado de la nueva estrategia de distribución digital de Google. En julio de 2013, se anunció que Google Play había sobrepasado 1 millón de aplicaciones publicadas y se habían registrado más de 50 mil millones de descargas.

### 9.2 Darse de alta como desarrollador

Ahora que ya sabemos lo que es Play Store, veremos cómo podemos subir nuestra aplicación.

Lo primero que tendremos que hacer será darnos de alta como desarrolladores, para que Google nos permita subir nuestras Apps a Play Store. Este registro no es gratuito, el alta como desarrollador cuesta 25 dólares, pero este pago lo realizaremos una única vez y ya seremos desarrolladores.

Para darnos de alta como desarrolladores accederemos a Google Play Developer Console y nos logearemos con nuestra cuenta de Google. Ahí es donde deberemos realizar el pago único de los 25 dólares.

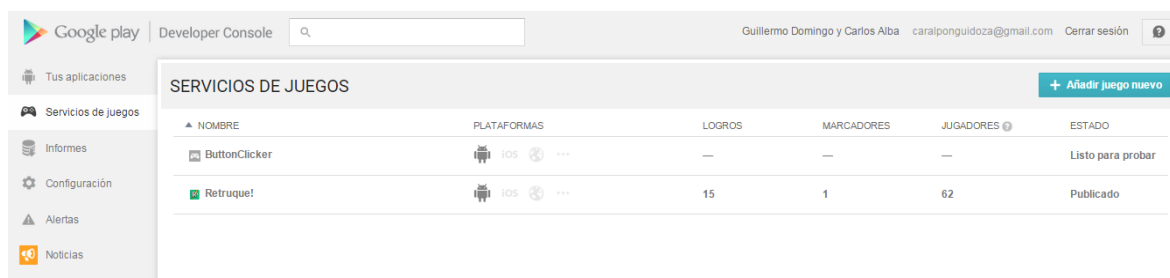


Una vez pagado, ya somos oficialmente desarrolladores de Apps y tenemos los permisos para subirlas a Play Store.

### 9.3 Cómo publicar una aplicación

Ahora que ya podemos subir nuestras aplicaciones, veamos qué debemos hacer para incorporarlas correctamente a Play Store. Subir una aplicación sin Game Services es más sencillo, pero en este apartado explicaremos cómo hacerlo incluyendo estos servicios. Tal y como hicimos para subir Retruque!



Lo primero que tenemos que hacer es crear un Servicio de juegos, que será lo que le proporcione a nuestra App el uso de los Game Services y todos sus elementos.



Una vez creado, podemos rellenar una ficha para el servicio, aunque esta no se verá en Play Store.



## Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

Estadísticas de jugadores	<b>Nombre para mostrar *</b> Inglés (Estados Unidos) – en-US	Retruque! 9 de 30 caracteres
Descripción general	<b>Descripción</b> Inglés (Estados Unidos) – en-US	Por fin el popular juego de cartas valenciano del Truc en tu móvil, como nunca antes lo habías visto. Nunca había sido tan fácil jugar una partida de Truc, estés donde estés. ¡Desafía a tus amigos con el modo multijugador en línea y demuestra que eres el mejor!  Características del juego: - Multijugador en línea 1 vs 1. - Multijugador en línea 2 vs 2. - Juega con amigos ingresando con tu cuenta de Google+. - Juega con gente desconocida buscando una partida. - Juega con gente cercana a tu localización. - Sistema de logros. - Ranking de victorias. - Sistema de señas para el modo 2 vs 2. <small>Fin de la descripción</small> 1250 de 4000 caracteres
Detalles de ingresos	<b>Categoría</b>	Cartas
Datos demográficos	<b>Juegos guardados</b>	<input type="checkbox"/> SÍ <input checked="" type="checkbox"/> NO
Estadísticas de funciones	<b>ARCHIVOS DE IMÁGENES</b>	Añade todos los recursos gráficos que se describen a continuación o utiliza recursos de imágenes de una de tus aplicaciones de Android.
Descripción general	<b>Icono de alta resolución</b> 512x512 Archivo PNG de 32 bits (alfa)	<b>Imagen destacada</b> 1024x500 Archivo JPG o PNG de 24 bits (no alfa)
Detalles de interacción		
Misiones		
<b>Información del juego</b>		
Aplicaciones vinculadas		
Eventos		
Logros		
Marcadores		
Testing		
Publicar		

Cuando tengamos la ficha rellenada, tenemos que acceder a la pestaña llamada Aplicaciones vinculadas, en la que crearemos una nueva aplicación que hará referencia a nuestro juego.

APLICACIONES VINCULADAS	<input type="button" value="Vincular otra aplicación"/>	o	<input type="button" value="Ir al siguiente paso"/>
NOMBRE	DETALLES	ID DE CLIENTE DE OAUTH 2.0	ESTADO
 Retruque!	com.game.guillermo.truc	✓	✓ Publicada

Este es el punto más importante de la creación del servicio, ya que será donde lo vinculemos con nuestro juego y si lo hacemos mal no proporcionará los servicios deseados.

Tenemos que darle un nombre (sólo es una referencia, no hace falta que sea el nombre del juego) y el nombre del paquete. Éste sí que debe ser el mismo que el paquete que tengamos en nuestro código, de lo contrario el juego no se vinculará correctamente. Además, marcaremos con un sí la opción de multijugador en tiempo real.

Una vez guardado, se nos proporcionará un código llamado ID de aplicación, que tendremos que añadir en nuestro código.



**Nombre de la aplicación**  
Inglés (Estados Unidos) – en-US

Retruque! 9 de 30 caracteres

---

**Nombre del paquete**

com.game.guillermo.truc

---

**Preferida para nuevos jugadores** ?  Para nuevas instalaciones en teléfonos Android.

---

**AJUSTES DE MULTIJUGADOR**

**Multijugador por turnos**

---

**Multijugador en tiempo real**

---

**ANTIPIRATERÍA**

**Habilitar antipiratería** ?

---

**AUTORIZACIÓN**

**ID de aplicación** 648419090213

---

**ID de cliente de OAuth2** 648419090213-temqaactrpc1ehhnpb9smeb72mqj944d.apps.googleusercontent.com

---

Solo tienes que incluir el ID de la aplicación (648419090213) en tu aplicación Android.  
Para obtener más información al respecto, consulta la [documentación para desarrolladores](#).

```
ids.xml x
<?xml version="1.0" encoding="utf-8" ?>
<resources>
  <string name="app_id">648419090213</string>
</resources>
```

Si hemos hecho el proceso correctamente, nuestro servicio ya estará creado y sólo nos faltará subir nuestra APK. Pasemos a ver cómo se hace esto.

Google play | Developer Console

Guillermo Domingo y Carlos Alba caralponguidoza@gmail.com Cerrar sesión

Tus aplicaciones

SERVICIOS DE JUEGOS

Informes

Configuración

Alertas

Noticias

**TUS APLICACIONES** + Añadir nueva aplicación

Filtrar

Página 1 de 1

NOMBRE DE LA APLICACIÓN	PRECIO	INSTALACIONES ACTUALES/TOTALES	VALORACIÓN MEDIA / TOTAL	ERRORES Y ANRS	ÚLTIMA ACTUALIZACIÓN	ESTADO
Retroque! 1.03	Gratuita	—	★ 5,00 / 23	—	21/8/2015	Publicada

Página 1 de 1

Tenemos que ir a la pestaña llamada Tus aplicaciones y dar clic en el botón verde de la derecha: Añadir nueva aplicación.

## Diseño e implementación de una APP para juego colaborativo. Desarrollo del Back-End

The screenshot shows the Google Play Console interface for the application 'Retruque!'. The app is published on August 21, 2015. The left sidebar contains navigation options: Estadísticas, Valoraciones y opiniones, Errores y ANRs, Sugerencias de optimización, Cloud Test Lab, and APK. The main content area is titled 'APK' and includes a 'Subir nuevo APK de producción' button. Below this, there is a 'CONFIGURACIÓN DE PRODUCCIÓN' section with a table showing the current production version (4) and its release date (21/8/2015 8:57:42). The table also displays the number of compatible devices (8246) and excluded devices (0). A table below lists the production version details.

▼ VERSIÓN	FECHA DE SUBIDA	ESTADO
4 (1.03)	21/8/2015	en producción

Primero, nos iremos a la pestaña APK, en la que tenemos que subir el APK firmado de nuestra aplicación pulsando sobre Subir nuevo APK de producción. Este APK lo podemos exportar fácilmente utilizando Android Studio. Tardará unos minutos en subir y ya tendremos subida nuestra aplicación al servidor. Pero el trabajo aún no está hecho, todavía tenemos que configurar algunas cosas para llamar la atención de los usuarios.

The screenshot shows the 'Descripción' page in the Google Play Console. It includes fields for the app title, a brief description, and a full description. The title is 'Retruque!' and the brief description is '¡Desafía a tus amigos a una partida de Truc con Retruque y su modo multijugador!'. The full description provides details about the game, including its characteristics and a note about a bug on some mobile devices.

**Título\***  
Español (Latinoamérica) – es-419  
Retruque!  
9 de 30 caracteres

**Descripción breve\***  
Español (Latinoamérica) – es-419  
¡Desafía a tus amigos a una partida de Truc con Retruque y su modo multijugador!  
80 de 80 caracteres

**Descripción completa\***  
Español (Latinoamérica) – es-419  
Por fin el popular juego de cartas valenciano del Truc en tu móvil, como nunca antes lo habías visto. Nunca había sido tan fácil jugar una partida de Truc, estés donde estés. ¡Desafía a tus amigos con el modo multijugador en línea y demuestra que eres el mejor!

Actualmente, hay algunos dispositivos móviles en los que el juego no funciona correctamente. Estamos trabajando para solucionar pronto este problema.

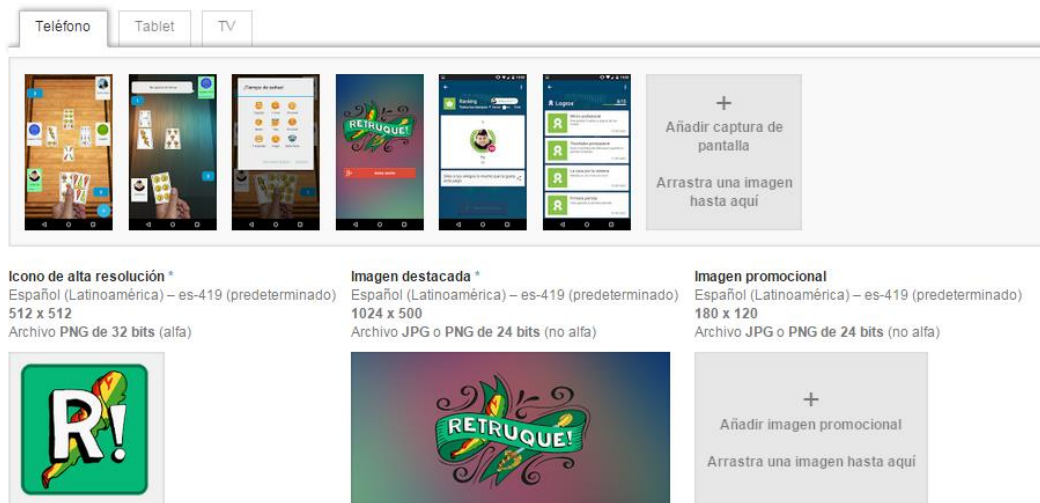
Características del juego:

- Multijugador en línea 1 vs 1.
- Multijugador en línea 2 vs 2.
- Juega con amigos ingresando con tu cuenta de Google+.
- Juega con gente desconocida buscando una partida.
- Juega con gente cercana a tu localización.

Sistema de juegos.  
1480 de 4000 caracteres

Consulta estas [sugerencias sobre cómo crear descripciones de aplicaciones acordes a la política](#) para evitar que la aplicación se suspenda por motivos comunes.

Seguidamente, tenemos que hacer clic en la pestaña Ficha de Play Store, en la que tendremos que rellenar todos los datos que nos pidan. Estos sí que serán los que vea el usuario final en Play Store, cuando abra la ficha del juego, por lo que es conveniente rellenarlos detenidamente.



Más hacia abajo, tenemos que añadir obligatoriamente un icono para la aplicación, una imagen destacada que será un fondo para la ficha en Play Store y, al menos, una captura de pantalla de la aplicación, aunque es muy recomendable subir más para llamar la atención de los jugadores.

CATEGORIZACIÓN


Tipo de aplicación \*



Categoría \*

Clasificación de contenido \*

Algunas opciones de valoración se han inhabilitado en función del contenido de tu aplicación. [Más información sobre la clasificación de contenido](#)

Nueva clasificación de contenido \*

 **CLASIFICACIÓN APLICADA**  
 ID de certificado de la IARC:  
 ffd7490a-709c-43fb-a724-bdbd07ecb7ad  
 Enviada: 20 de ago. 20:16  
[Ver detalles](#) [Más información](#)

Para finalizar, solamente nos quedará rellenar un pequeño cuestionario para la clasificación de contenido de nuestro juego. Una vez hecho, veremos como se muestra una clasificación que ha sido asignada a nuestra aplicación. También tenemos que fijar el precio que costará la descarga de la aplicación (en nuestro caso gratuita).

Si hemos hecho bien todo el proceso, podemos proceder a publicar nuestra aplicación y esperar un par de horas para verla y descargarla en Google Play Store.

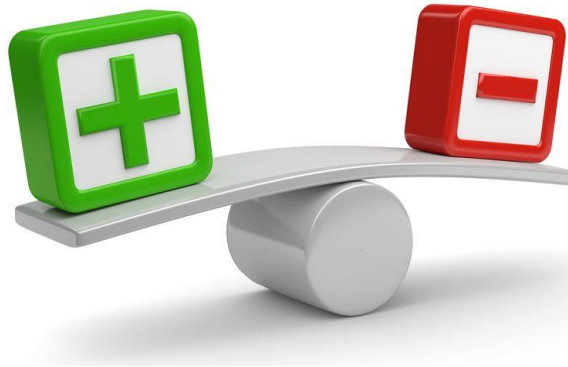
## 9.4 Precio de la publicación

Sabemos que hay que pagar para darse de alta como desarrollador de Apps y poder subir las, pero... ¿cuánto tengo que pagar por cada App que subo? La respuesta es nada.

Google nos cobrará lo mismo tengamos una aplicación o 30, así que podemos subir las que queramos. Pero cuidado, no conviene subir cualquier aplicación. Si nos queremos ganar una fama como desarrolladores debemos de programar Apps de calidad y sólo subir las que nos puedan servir para obtener un beneficio. Subir muchas Apps nos puede hacer ganar algo más de dinero pero a la larga la calidad pesa más que el dinero que pudiéramos sacar de subir un gran número de Apps sin mucha calidad.



## 10. Resultados obtenidos



Nuestra aplicación fue subida a Play Store en su primera versión (1.0) el día 20 de agosto del 2015. Sin embargo, distintas versiones han sido subidas posteriormente debido a que algunos dispositivos móviles experimentaban problemas a la hora de jugar una partida. Esto ha sido arreglado y el juego funciona correctamente.

Actualmente, la versión disponible para descarga es la 1.1. Veamos los resultados que hemos obtenido hasta la fecha de publicación de esta memoria, el día 2 de septiembre del 2015.

### 10.1 Descargas

En los primeros 10 días tras la publicación de la App obtuvimos 125 descargas. Un dato que no está mal, ya que es una aplicación que nadie conocía y que debe ser expandida mediante las redes sociales y la ayuda de los propios jugadores.

Tras solamente 14 días (fecha límite hasta la publicación de esta memoria), conseguimos triplicar las descargas y los ingresos, así como los usuarios que juegan a la App. A 2 de septiembre, hemos superado las 300 descargas.

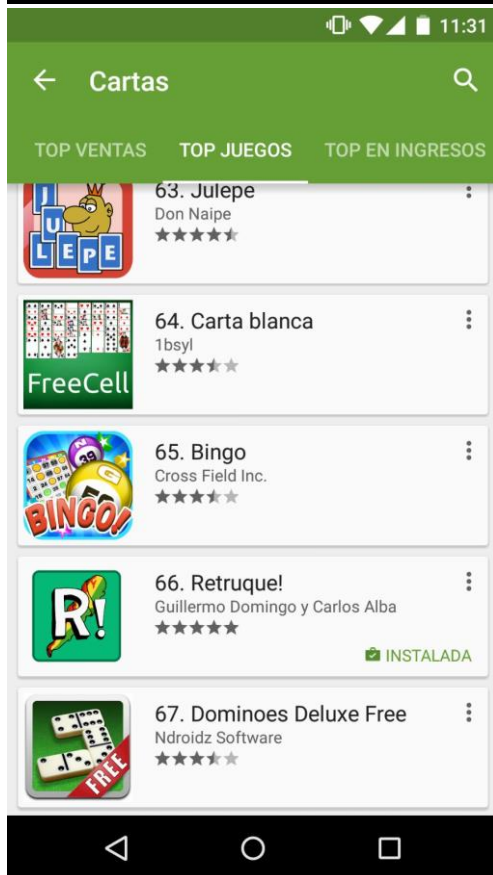
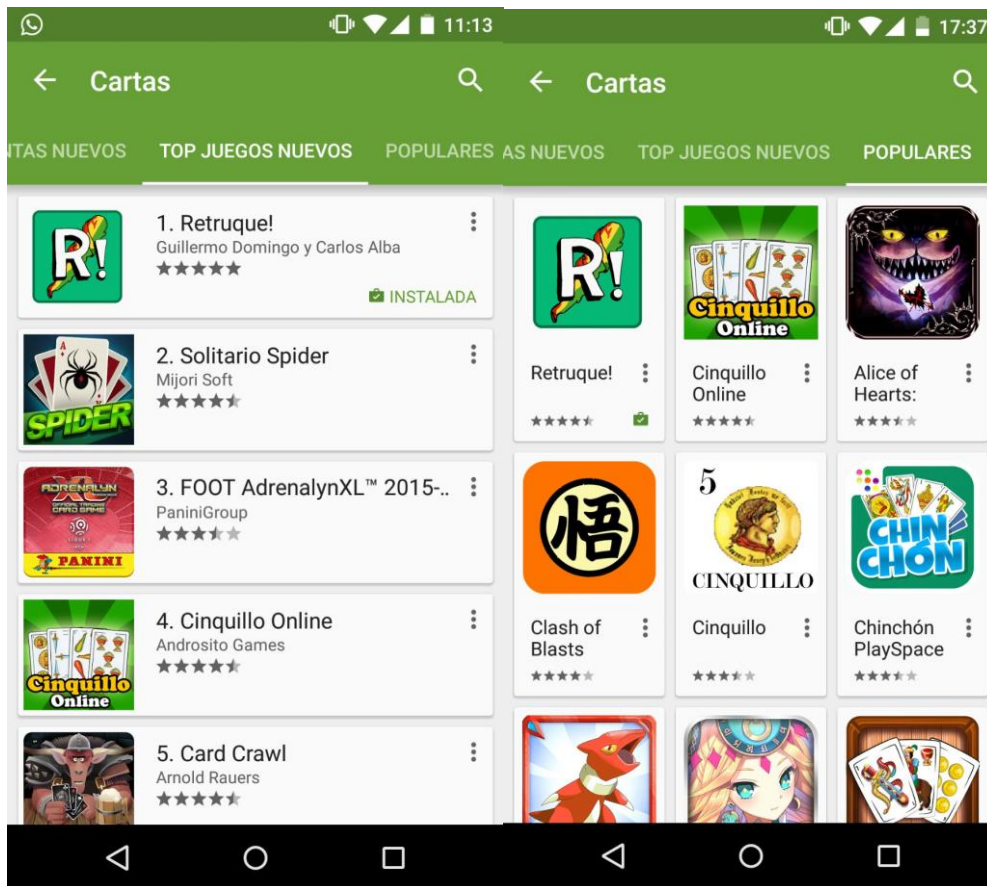


Pero este aumento no lo es todo. En dos semanas hemos conseguido ser el Top 1 en aplicaciones nuevas en toda España, en la categoría de cartas de Play Store.

También somos el juego más popular de cartas del momento, superando a juegos que tienen miles de descargas.

Como último dato, aparecemos en la posición 66 en el ranking de mejores juegos de España de cartas de todos los tiempos.

Teniendo en cuenta el cortísimo tiempo de vida de la aplicación, podemos asegurar que esto es todo un éxito.



Este trabajado de expansión está en proceso y esperamos aumentar las descargas considerablemente en poco tiempo.



## 10.2 Usuarios de la App

Otro término distinto que también conviene analizar son los usuarios activos del juego. Sigamos con el caso de los 10 primeros días. De las 125 descargas que hemos obtenido, tenemos 62 usuarios activos, lo que quiere decir que prácticamente la mitad de los jugadores juega alguna partida habitualmente, mientras que los demás juegan con menos asiduidad.

Cuatro días más tarde, esta cifra se ha visto aumentada como se aprecia en la imagen anterior y más del 60% de los jugadores siguen activos en el juego.

JUGADORES		LOGROS
Jugadores nuevos ?	Jugadores activos ?	Desbloqueado ?
Último día <b>108</b> +280,0%	Último día <b>149</b> +188,1%	Último día <b>179</b> +842,1%
Últimos 7 días <b>189</b> +155,4%	Últimos 7 días <b>224</b> +190,9%	Últimos 7 días <b>198</b> +2.100,0%
Últimos 30 días <b>263</b>	Últimos 30 días <b>266</b>	Últimos 30 días <b>223</b>
MARCADORES		PARTIDAS EN TIEMPO REAL
Puntuaciones publicadas ?	Recuento de partidas iniciado ?	
Último día <b>66</b> +725,0%	Último día <b>277</b> +1.438,9%	
Últimos 7 días <b>75</b> +400,0%	Últimos 7 días <b>308</b> +150,4%	
Últimos 30 días <b>87</b> +1.640,0%	Últimos 30 días <b>586</b> +112,3%	

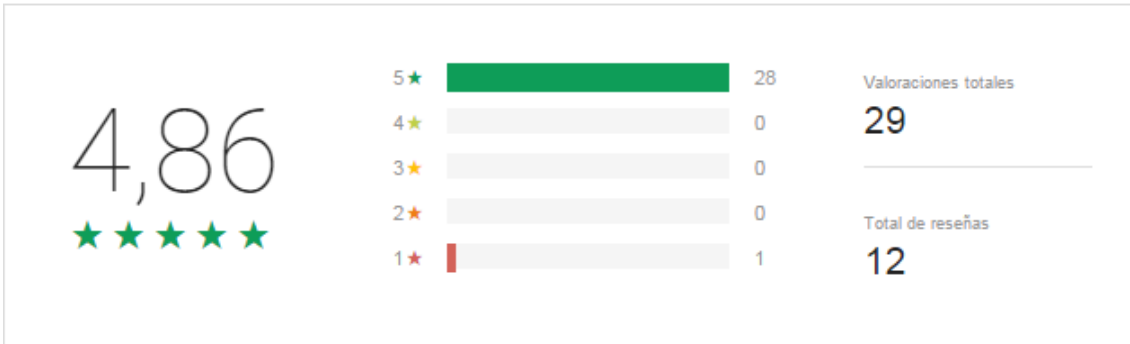
Como podemos ver en las imágenes anteriores, los usuarios han crecido en los últimos días con unos porcentajes enormes. Podemos observar algunos como 2.100% o 1.640%, lo que nos indica que se está produciendo un crecimiento altísimo de usuarios.

Podemos destacar las 586 partidas que ya se han jugado hasta la fecha de hoy y, obviamente, este número va creciendo con el paso de las horas.

## 10.3 Opiniones de los jugadores en Play Store

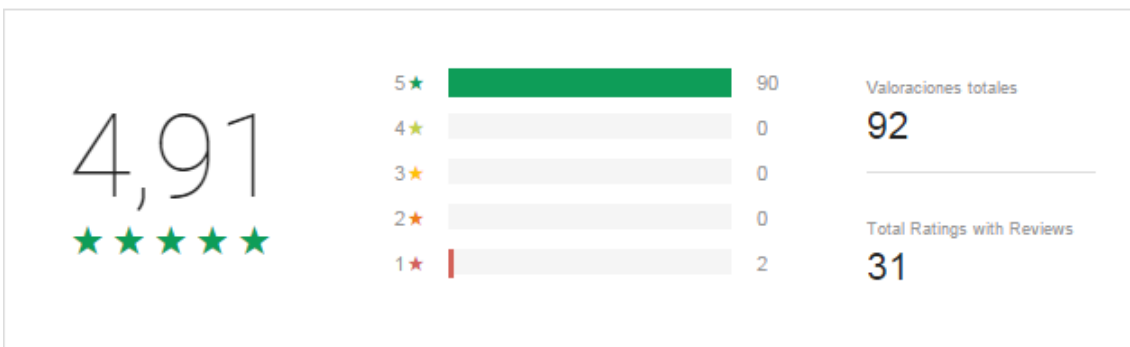
Una de las cosas que más debemos y queremos tener en cuenta son las opiniones de nuestros jugadores, ya que ellos son los que nos proporcionaran beneficios y nuestro trabajo es mantenerles enganchados a nuestro juego.

Las votaciones en Play Store tienen un rango del 1 al 5, donde el 1 es muy malo y el 5 es muy bueno. En los primeros 10 días, recibimos 29 votaciones, y en 12 de ellas los usuarios nos han dejado una reseña escrita.



De estas 29 votaciones, 28 usuarios votaron con un 5 y 1 usuario votó con un 1. Lo que nos da una media de 4,86 sobre 5. Las críticas fueron muy positivas.

Veamos ahora el cambio que se produjo tras 14 días.



Hemos obtenido un total de 92 valoraciones y 90 de ellas nos han otorgado 5 puntos. Las otras 2 restantes, nos han puntuado con un punto. La media ha subido a 4,91 y el número de reseñas escritas por los usuarios es de 31.

Aquí tenemos algunas de las reseñas que hemos leído en la ficha del juego en Play Store:

**Carmen Almela** 25/8/2015 a las 13:29 👍 0 🗨️ 0

★★★★★

Traducción automática del Catalán Mostrar opinión original

**El único truco valenciano** Pero falta gente.

Responder a esta opinión

Dispositivo: HUAWEI MediaPad M1 8.0 (hws8301l)    Versión de la aplicación: 4 (1.03)    Sistema operativo: Android 4.2

**Jorge Gallego** 25/8/2015 a las 1:00 👍 0 🗨️ 0

★★★★★

**Un juego de 10!** Excelente juego en el que pasar las horas muertas, joc fora.

Responder a esta opinión

Dispositivo: LG L80 Single (w6)    Versión de la aplicación: 4 (1.03)    Sistema operativo: Android 4.4



**Chulviyomuss** 21/8/2015 a las 15:03 1 0

★★★★★

Traducción automática del Catalán    Mostrar opinión original

Un juego por tierra Muy logrado, lo mejor las frases en valenciano. 🐼

Responder a esta opinión

Dispositivo: Galaxy S5 (kite)    Versión de la aplicación: 2 (1.01)    Sistema operativo: Android 5.0

---

**Marcos Blazquez** 21/8/2015 a las 14:16 0 0

★★★★★

Mítico juego tremendamente recreado

Responder a esta opinión

Dispositivo: Aquaris E5 (Aquaris\_E5)    Versión de la aplicación: —    Sistema operativo: Android 4.4

---

**Salva Escorihuela Sanfeliu** 31/8/2015 a las 14:45 0 0

★★★★★

Superentretenido .... la verdad q ya hacia tiempo q no encontraba un juego de cartas tan divertido

Responder a esta opinión

Dispositivo: Galaxy S4 Mini (serranotte)    Versión de la aplicación: 5 (1.1)    Sistema operativo: Android 4.4

---

**Jose Mas Pons** 31/8/2015 a las 14:38 0 0

★★★★★

Traducción automática del Catalán    Mostrar opinión original

Ya tocaba el juego valenciano por exelencia

Responder a esta opinión

Dispositivo: bird82\_wet\_a\_jb5    Versión de la aplicación: 5 (1.1)    Sistema operativo: Android 4.2

---

**Raul Valero** 21/8/2015 a las 14:07 2 0

★★★★★

Por fin! Sinceramente, ya era hora que sacaran un juego del famoso truc y además estuviera en muy buenas condiciones de jugabilidad! Un 10!

Responder a esta opinión

Dispositivo: Aquaris E5 (Aquaris\_E5)    Versión de la aplicación: 2 (1.01)    Sistema operativo: Android 4.4

Estas son algunas de las reseñas más relevantes que hemos encontrado. Todas son positivas y nos ayudan a seguir trabajando en el juego, ya que se le ve a los jugadores contentos y con ganas de jugar partidas.

Analizando las respuestas, vemos que existe una gran aceptación entre los jugadores pero, una de ellos nos dice que de momento faltan jugadores. Esto se irá solucionando con el paso del tiempo y el crecimiento de las descargas.

La única pega que podemos sacar son los dos votos negativo que no son asignado con un 1. Sin embargo, la mayoría absoluta nos ha puntuado con la máxima nota.

## 11 Trabajo futuro

En el siguiente epígrafe vamos a establecer el trabajo futuro para nuestra aplicación. El desarrollo de una aplicación móvil, de la misma manera que empieza mucho antes de escribir la primera línea de código, tampoco acaba al escribir la última.

Deberemos por tanto, centrarnos en labores de mantenimiento, mejora, soluciones a problemas y “bugs”, además de acabar con todas las tareas que se nos quedaron en el backlog.

También nos propondremos internacionalizar el juego, y extenderlo a otras regiones donde también se juega al “Truc”, como por ejemplo Argentina.

### 11.1 Tareas pendientes

Si le echamos un vistazo al mapa de características de nuestro proyecto, veremos que no solo logramos implementar todas las características del primer sprint, sino que también gran parte de las del segundo, quedándonos tan solo “Apuestas con monedas” e “Instrucciones”.

- Apuestas con monedas: la idea es utilizar un modelo basado en pagar unas monedas para poder jugar una partida. Todos los jugadores pagarían unas monedas y el o los vencedores se quedarían con las del rival, creando de esta forma un sistema de apuestas. Con esto lo que pretendemos es hacer evolucionar la aplicación a un modelo “Freemium”, en el que se paga por ciertos servicios extra. El usuario podría comprar monedas cuando se le acabasen. De este modo, nosotros conseguiríamos un plus económico.
- Instrucciones: tan solo se juega al “Truc” en Valencia o regiones cercanas. En este contexto, se pretende extender el rango de usuarios, mediante la incorporación de un sistema de instrucciones. Todavía no se ha decidido de qué forma, pero contemplamos varios escenarios. La primera opción es, en el propio juego, crear una pantalla con las instrucciones. Por el momento es la que más fuerza coge. En un segundo plano tenemos la opción de crear un portal web.

En cuanto al tercer sprint tenemos pendientes todas las tareas. Con estas tareas el juego estaría finalizado a nivel de módulos. Muchas de estas características son propias de los juegos más exitosos disponibles.

La primera de ellas sería la inclusión de “login con Facebook”. Sabemos que Google+ no es la principal red social del momento, así que queremos aumentar nuestro número de usuarios permitiendo acceder a la aplicación con una cuenta de Facebook. Hemos comprobado que, aunque el 90% de gente cuenta con un perfil de Google+, la gran mayoría no ha agregado a sus conocidos a sus círculos. Con Facebook desaparecería este problema.

La siguiente tarea tiene que ver con el proceso de transformación a un modelo “Freemium” y es la de establecer una tienda o market para la compra de funcionalidades adicionales. Entre ellas queremos incluir personajes para jugadores, tipos de barajas, además de paquetes de monedas etc.

Con la intención de mejorar la interacción entre jugadores durante la partida, propusimos la inclusión de un chat. Así pues los jugadores podrán comunicarse cuanto quieran, pasando el sistema de frases a formar parte de este nuevo sistema de chat.

En última instancia, posiblemente la creación de torneos sea la novedad más importante. Los jugadores podrán participar en diferentes torneos basándonos en su situación en el ranking. Los ganadores sumarán monedas y experiencia a partes iguales. Creemos que estos torneos servirán de motivación y animarán a los jugadores a demostrar quién es el mejor.

## **11.2 Posibles mejoras**

A continuación veremos unas posibles mejoras.

Se ha pensado en sacar una versión tanto de iOS como web. Con la versión para el sistema operativo de Apple completaremos la cuota de mercado de los smartphones y con la versión web podríamos competir con el juego de “Truc” con el que cuenta Facebook.

Introduciremos la compatibilidad con tablets, de modo que ya cubriremos todos los dispositivos tanto Android como iOS. Evidentemente, cuanto mayor sea el número de dispositivos compatibles, mayor será el número de jugadores.

## **11.3 Expansión a otros países**

Durante nuestro estudio del arte, descubrimos que no solo se jugaba al “Truc” aquí en Valencia. En algunos países del sur de América existe una variante llamada “Truco”, entre ellos Argentina, Colombia o Uruguay.

Nuestro propósito es el de dar a conocer la variante valenciana del juego allí y ver que aceptación tendría.

## 12. Conclusiones



Llegamos al último apartado de la memoria del proyecto, en el que hablaremos de las conclusiones que hemos sacado tras finalizar el juego y de todo el proceso y trabajo que esto conlleva.

### 12.1 Resultados del proyecto

Tras muchos meses de trabajo, nuestro objetivo principal era conseguir desarrollar un juego divertido, atractivo para el usuario y que pudiera tener éxito. Ha resultado complicado pero hemos conseguido una primera versión muy trabajada, incluyendo incluso más características de las inicialmente planificadas y llegando a un nivel que no pensábamos que pudiésemos alcanzar.

El código ha alcanzado una longitud aproximada de 10.000 líneas de código, métrica con la que podemos medir la complejidad del proyecto.

Con el paso de los días, estamos obteniendo un número mayor de descargas y consiguiendo poco a poco expandir el juego entre los usuarios de dispositivos Android, por lo que creemos que en poco tiempo podemos alcanzar un número considerable de instalaciones del juego.

Analizando todas estas cosas, podemos concluir diciendo que el proyecto ha sido un éxito y que puede llegar a crecer mucho.

### 12.2 Aprendizaje

Para poder alcanzar este éxito del que hemos hablado en el anterior apartado, hemos tenido que aprender muchísimo de las distintas tecnologías que han intervenido en el desarrollo del juego.

Sabíamos programar en Android, pero nunca habíamos pensado en hacer algo parecido, algo tan ambicioso. Hemos tenido que adquirir muchos más conocimientos de los que teníamos para entender mejor el sistema Android. Pero esto es sólo el principio.

También hemos tenido que investigar mucho acerca de la tecnología que íbamos a usar para conectar a los jugadores, crear partidas, etc. Hasta que descubrimos Google Play Game Services. Aquí también tuvimos que hacer un gran trabajo de investigación y pruebas para conseguir avanzar en la programación.

Como no, también hemos utilizado muchas tecnologías complementarias a Game Services y Android, como Git con GitHub, AdMob, Developer Console, aprender a subir una App a Play Store... Muchas cosas.

Por tanto podemos decir que, aparte de hacer un TFG, hemos aprendido muchísimo y esto nos puede servir de mucha ayuda de cara al futuro. Si tuviéramos que hacer una aplicación parecida, nos costaría mucho menos sin lugar a dudas. Todo el desarrollo del proyecto nos ha sido muy positivo como informáticos.

### **12.3 Opinión personal**

Para acabar con la memoria, escribiré una opinión personal sobre el proyecto realizado.

Cuando decidimos hacer este proyecto, veía que podríamos hacerlo pero que sería difícil. Por lo visto, acerté con mi pensamiento. Sin embargo, hemos conseguido llevarlo adelante y, personalmente, he aprendido muchísimo sobre las tecnologías que hemos utilizado.

Me veo capaz de desarrollar más aplicaciones para Android, gracias a los nuevos conocimientos que he adquirido y ahora ya sé cómo subir una aplicación y todo el proceso que debería seguir para la próxima que quiera desarrollar. Si a esto le sumamos que parece que va a tener éxito y que a la gente que juega nos dice que le está gustando mucho, puedo decir que tengo una gran satisfacción personal con el resultado del proyecto.

Todo me ha resultado muy positivo, lo conseguimos.



### 13. Bibliografía

- Transparencias de la asignatura “Soluciones para dispositivos móviles”, impartida por la Universidad Politécnica de Valencia durante el año 2015.
- Google Play Store, subcategories: Card Games. Recuperado en mayo de 2015, de:  
[https://play.google.com/store/apps/category/GAME\\_CARD](https://play.google.com/store/apps/category/GAME_CARD)
- Lean Canvas. Recuperado en mayo de 2015, de:  
<http://www.leanstart.es/lean-canvas/>
- Android Studio Overview, Android Developers. Recuperado en mayo de 2015, de:  
<http://developer.android.com/tools/studio/index.html>
- Genymotion. Recuperado en junio de 2015, de:  
<https://www.genymotion.com>
- Conociendo GitHub (© Copyright 2012, Luciano Castillo.). Recuperado en junio de 2015, de:  
<https://conociendogithub.readthedocs.org/en/latest/>
- Google Play Developer Console, Android Developers. Recuperado en junio de 2015, de:  
<https://developer.android.com/distribute/googleplay/developer-console.html>
- Publicando en Google Play Store, Android Developers. Recuperado en agosto de 2015, de:  
<http://developer.android.com/distribute/googleplay/start.html>
- AdMob, monetiza tus aplicaciones. Recuperado en julio de 2015, de:  
<https://www.google.es/admob/>
- Google Play Games Services, Google Developers. Recuperado en junio de 2015, de:  
<https://developers.google.com/games/services/>
- Real-Time Multiplayer, Google Developers. Recuperado en junio de 2015, de:  
<https://developers.google.com/games/services/common/concepts/realtimeMultiplayer>
- StackOverFlow. Página de consulta de dudas.  
<http://stackoverflow.com/>