



El gaming y el consumo colaborativo

Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles

Autor:

Daniel Funes Garcia-Penche

Consultor:

Francesc D'Assís Giralt Queralt



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-

SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	El gaming y el consumo colaborativo
Nombre del autor:	Daniel Funes Garcia-Penche
Nombre del consultor:	Francesc D'Assís Giralt Queralt
Fecha de entrega:	06/2017
Titulación:	<i>Máster Universitario en Desarrollo de Aplicaciones para Dispositivos Móviles</i>

Resumen del Trabajo:

Este proyecto es un caso práctico en el que he trabajado en la conceptualización, diseño e implementación de una aplicación orientada al consumo colaborativo para el sector de los videojuegos. Aunque de inicio es un sector muy concreto, la plataforma se ha diseñado con el propósito de poder escalarla a cualquier tipo de producto que pudiera ser interesante para los usuarios. En cuanto a la implementación, esta se aproxima a un producto mínimo viable cumpliendo funcionalmente con los requerimientos que se plantean en la fase de diseño.

Abstract:

This final master's work is a case study where I have worked on the conceptualization, design and implementation of a collaborative consumer-oriented application for the videogames sector. Although it is a very specific target, the platform has been designed to scale it to any product type that could be interesting for the users. About final implementation, it approximates to the minimum feasible product and accomplishes with the defined requirements during the design phase.

Palabras clave (entre 4 y 8):

consumer-oriented, ios, swift, mobile app, gaming.

Índice

1. INTRODUCCIÓN	1
1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO	1
1.2 ESTADO DEL ARTE	2
1.3 OBJETIVOS DEL TRABAJO	2
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS	3
1.4 ENFOQUE Y MÉTODO SEGUIDO	3
METODOLOGÍAS EXISTENTES	3
METODOLOGÍA APLICADA AL PROYECTO	6
1.5 PLANIFICACIÓN DEL TRABAJO	7
FASES DE DESARROLLO	7
DIAGRAMA DE GANTT	10
1.6 RIESGOS DEL PROYECTO	12
2 CONCEPTUALIZACIÓN	14
2.1 ENCUESTA A LOS JUGADORES	14
LA ENCUESTA	15
RESULTADOS	16
CONCLUSIONES	21
2.2 PERFILES DE USUARIO	23
2.3 ESCENARIOS	24
3 DEFINICIÓN FUNCIONAL	26
3.1 LISTADO DE REQUISITOS	26
3.2 CASOS DE USO	28
3.2 ÁRBOL DE NAVEGACIÓN	34
4. DISEÑO	35
4.1 WIREFRAMES	35
LISTADO	35

FICHA	35
EVALUACIONES (FICHA)	36
FAVORITOS	36
RESERVAS	37
MENSAJES	37
SALA DE CHAT	38
PERFIL	38
4.2 PROTOTIPO	39
PALETA DE COLORES	39
BOTONES	39
LISTADO	40
FICHA PÚBLICA DE PRODUCTO	40
VALORACIONES	41
RESERVA	41
CONFIRMACIÓN DE RESERVA	42
CHAT	42
LISTA DE RESERVAS	43
INFORMACIÓN DE LA RESERVA	43
LISTA DE MENSAJES	44
AÑADIR O EDITAR PRODUCTO	44
EDITAR DISPONIBILIDAD	45
FICHA PÚBLICA DE USUARIO	46
PERFIL	47
FICHA PRIVADA DE USUARIO	47
EDITAR PERFIL	48
FICHA PRIVADA DE PRODUCTO	48
LOGIN	49
REGISTRO	49
5. ARQUITECTURA TÉCNICA	50
5.1 FIREBASE	50
5.2 API-REST	51
5.3 SOLUCIÓN: FIREBASE	51
5.4 ARQUITECTURA	52

DESARROLLO NATIVO	52
BACKEND	52
DISEÑO DE ENTIDADES	52
MODELADO DE DATOS	53
6. IMPLEMENTACIÓN (IOS)	55
6.1 ANTES DE EMPEZAR...	55
6.2 ARQUITECTURA DE CLASES	55
CONTROLADORES	57
VISTAS	57
NETWORK	57
EXTENSIONES	57
RECURSOS	57
6.3 PROTECCIÓN DE VISTAS	57
6.4 ALGO MÁS QUE VISTAS: BACKEND	58
6.5 LIBRERÍAS EXTERNAS	58
7. CONCLUSIONES	60
7.1 SEGUIMIENTO DEL PROYECTO	60
7.2 LECCIONES APRENDIDAS	61
7.3 TRABAJO PENDIENTE	61
7.4 LÍNEAS FUTURAS	61
BRANDING	61
ON BOARDING:	62
CONVERSIÓN DE USUARIOS:	62
CAPTACIÓN DE PRODUCTORES:	62
DESARROLLO EN ANDROID:	62
5. BIBLIOGRAFÍA	63

1. Introducción

1.1 Contexto y justificación del Trabajo

La industria *gaming* está viviendo una nueva época dorada con un crecimiento económico constante durante los últimos años. Solo durante el 2016 se ha facturado más de 50 mil millones de dólares con la venta de videojuegos y más de 30 mil millones en hardware especializado. En cuanto a los primeros, se puede decir que algunos de estos ya son comparables a las grandes producciones cinematográficas, argumento que explica el aumento de facturación en el ámbito de hardware, ya que a más calidad más demanda de recursos.

Internet con las redes sociales y sus páginas especializadas son, en parte, el gran promotor de este movimiento. No es raro ver cómo se va generando el ya conocido *hype*¹ durante los meses anteriores al lanzamiento de un videojuego, consola, componente, etc. el cual acaba generando cierta ansia por parte de los usuarios por comprarlo. Una ansiedad que se disipa una vez adquirido el producto y que suele ser sustituido por otro que saldrá en breve. No es extraño entonces ver estas grandes obras, periféricos y demás relevados a un cajón o estante para no volver a ser usados nunca más. Artículos que, o por la gran devaluación económica que sufren al poco tiempo o por puro sentimiento de coleccionista, ni serán vendidos en el mercado de segunda mano ni usados por nadie más.

Este proyecto nace con la idea de dar respuesta a esta problemática con un concepto que está funcionando muy bien en otros sectores como es la economía colaborativa. Con esto se pretende crear una plataforma digital móvil que mueva el consumismo individual hacia un modelo peer-to-peer². Un escenario en el que se puede aprovechar el excedente de recursos multimedia y donde los

¹ Estrategia inteligente para enfatizar una cosa, idea o producto hasta el punto en que los individuos sientan la necesidad de consumir.

² Red entre iguales.

consumidores se vuelven productores y viceversa. Todo de una manera fácil, ágil y sin intermediarios.

1.2 Estado del arte

El año 2016 ha sido un año muy intenso para la economía colaborativa y se prevé que lo sea aún más durante este año 2017. Según datos de las CNMC, 1 de cada cuatro españoles ya utiliza una de estas plataformas al menos una vez al año.

De entre estas plataformas cabe destacar algunas de las cuales ya son ampliamente conocidas y otras que están en pleno crecimiento:

- **AirBNB:** Alquiler de viviendas turísticas.
- **SocialCar:** Alquiler de vehículos.
- **Film2:** Proyección de películas entre particulares. Los usuarios pagan una tasa y asisten a casa del “vecino” a ver una película.
- **Relendo:** Alquiler de todo tipo de artículos entre particulares.

En este sentido, no hay ninguna app o plataforma que haga lo propio con el sector de los videojuegos. Relendo es una plataforma generalista y por lo tanto también incluyen los videojuegos pero, al no estar especializada, es difícil encontrar algo relacionado con estos.

1.3 Objetivos del Trabajo

La economía colaborativa está bajo un paradigma joven del cual todavía no existe una gran experiencia y conocimiento, ni a nivel profesional ni a nivel académico. En este sentido se definirá un único objetivo donde focalizar los esfuerzos del cual derivará en una serie de objetivos específicos que, a día de hoy, ya se reconocen como clave para el éxito de cualquier plataforma de este tipo. Una vez puesta en marcha, la evolución de los objetivos estará basada en los errores cometidos y el aprendizaje continuo.

Objetivo general

- Presentar el consumo colaborativo al sector gaming.

Objetivos específicos

- Establecer las herramientas necesarias para que los usuarios puedan ceder sus artículos mediante alquiler.
- Mostrar un gran número de artículos disponibles por cada zona.
- Aumentar la confianza entre los usuarios.
- Componer una plataforma de comunicación abierta entre usuarios.

1.4 Enfoque y método seguido

Metodologías existentes

Existen muchas metodologías de trabajo, las cuales pueden ser el resultado del cruce entre varias de estas. No hay una solución única y perfecta por lo que la elección para este proyecto se basará en el análisis de tres de las más conocidas en el ámbito de gestión de proyectos: Metodología en cascada, Scrum y Kanban.

Metodología en cascada

Se trata de una de las metodologías clásicas más antiguas en el desarrollo de software. Su principio se basa en el orden riguroso de las etapas por la que debe pasar un proyecto de tal manera que una etapa debe esperar a la finalización de la anterior para poder ser iniciada. Estas etapas son: análisis, diseño, implementación, pruebas y mantenimiento.

Análisis

Aquí se definen los requisitos que se deben cumplir tanto a nivel de hardware como de software. En esta etapa es muy importante contar con un conocimiento absoluto de las funcionalidades de la aplicación, por lo que habrá que confeccionar una lista de requerimientos completa.

Diseño

Se descompone la aplicación en elementos independientes para poder empezar a ser desarrollados de manera independiente. Por una parte se encuentra el diseño a nivel de arquitectura, donde se identifican los módulos y sus relaciones y por otra la organización del código a implementar.

Implementación

Se codifica la aplicación en base a la arquitectura creada en la etapa anterior.

Pruebas

Se inician las pruebas del código final. Se puede hacer mediante diferentes estrategias de testeo.

Mantenimiento

En la mayoría de los casos resulta ser la etapa más costosa ya que el producto no siempre acaba cumpliendo con las expectativas del usuario final.

Scrum

Se trata de una de las metodologías ágiles más extendidas en el sector tecnológico. Está orientado a equipo multidisciplinarios e indicado para proyectos complejos donde las prioridades son cambiantes o poco definidas. Por otro lado y a diferencia de la metodología en cascada, el desarrollo del producto viene dado mediante entregas parciales, las cuales vienen priorizadas por las necesidades del gestor del proyecto, denominado *product owner*.

Las iteraciones se denominan *sprints* y su duración es entre 2 y 4 semanas, los cuales conllevan 3 etapas.

- **Arranque de sprint:** Antes de cada sprint se mantienen reuniones de estimación de esfuerzo de las tareas del *backlog* y su consiguiente planificación donde se marcarán las tareas a realizar durante el ciclo entrante. Estas reuniones tendrán una duración máxima de 4 horas cada una.

- **Durante el sprint:** Se establecen reuniones diarias de un máximo de 15 minutos donde cada miembro del equipo informa de lo que hizo el día anterior y lo que hará en el presente día. En caso de necesitar la colaboración o tener que informar sobre algún tema que afecte al equipo o a la consecución del sprint lo hará en ese momento. Se trata de reuniones rápidas por lo que se hacen de pie.
- **Finalización de sprint:** Al final de cada sprint se debe de poder entregar un producto con unas funcionalidades listas para ser usadas. Luego se establecerá una reunión de retrospectiva en equipo para analizar lo que se ha hecho bien durante el sprint y discuten sobre aquellos procesos que serían mejorables. Esta no debería de durar más de 2-3 horas.

Por último, en cada equipo de sprint se define un *scrum master*, el cual es el encargado de velar por que se cumpla la metodología y los objetivos marcados por el *product owner*.

Kanban

Se trata de otra metodología ágil aunque es más flexible que *scrum*. En esta no existen ciclos de desarrollo cerrados y se basa más bien en tarjetas, las cuales representan las tareas a realizar.

Sus principios básicos son:

- **Visión general:** Esto se consigue gracias al uso de tarjetas y columnas. Estas columnas pueden ser cualquier cosa y un ejemplo sencillo sería: pendiente, en desarrollo y entregado.
- **Comienza por lo que estés haciendo ahora:** Esto significa que los miembros del equipo deben centrarse únicamente en la tarea en curso, no pudiendo ejecutar más de una a la vez. A su vez se establece un máximo de tareas a realizar paralelamente por el equipo
- **Las prioridades son flexibles:** Se pueden modificar en cualquier momento aunque siempre deberán de respetar el punto anterior. Esto significa que puede negociarse libremente la entrada de tareas y

relocalización de estas pero deberán tomarse en el momento de acabar la tarea en curso.

Metodología aplicada al proyecto

El objetivo del proyecto es poner en práctica los conocimientos adquiridos durante el máster y se ejecuta individualmente. Se trata de algo que debe ser evaluado por el equipo docente por lo que las fechas de entrega están establecidas al inicio del semestre.

Partiendo de estas dos premisas, la metodología escogida se basará en la metodología en cascada, ya que el calendario de entregas parciales ya está definido y este no cambiará durante el transcurso del proyecto. A su vez se usará un panel kanban para la organización de las tareas a realizar entre entrega y entrega.

No obstante, una vez finalizado el periodo docente se seguirá con el desarrollo de la plataforma mediante una adaptación de Scrum para trabajo individual.

1.5 Planificación del Trabajo

La planificación está basada en un calendario de lunes a domingo. De lunes a jueves habrá una carga de trabajo de 2,5h y los viernes, sábados, domingos y festivos tendrán una carga de 6 horas. En total serán 28 horas semanales.

Fases de desarrollo

Investigación de usuarios

Tiempo estimado: 5 horas

Se seleccionará un conjunto de personas interesadas en el mundo de los videojuegos, a los cuáles se les pedirá responder a una breve encuesta con preguntas relacionadas con los objetivos de ese proyecto. De esta se espera poder obtener datos cuantitativos y cualitativos valiosos para la definición de los perfiles de usuario.

Análisis del contexto y uso de la aplicación.

Tiempo estimado: 6 horas

En base a los perfiles de usuario se analizarán los diferentes contextos en los que se usará la aplicación.

Generación de casos de uso

Tiempo estimado: 4 horas

Cada perfil tiene unos objetivos distintos con la aplicación. En esta fase se definirán las historias de usuario por cada uno de ellos.

Definición funcional

Tiempo estimado: 5 horas

Se establecerán los requerimientos funcionales de la aplicación. Estos estarán basados en el cruce entre los objetivos específicos del proyecto y las necesidades recogidas durante la definición de los perfiles, contextos y casos de uso.

Definición del flujo de la aplicación

Tiempo estimado: 5 horas

Una vez marcada la funcionalidad se generará el flujo de la aplicación mediante diagramas *uml*.

Arquitectura técnica de la aplicación

Tiempo estimado: 3 horas

Aquí se definirá el modelo de la aplicación (entidades, repositorio de datos, etc), así como la arquitectura técnica elegida, ya sea mediante api *in-house* o mediante alguna de las plataformas *cloud* especializadas.

Wireframes

Tiempo estimado: 8 horas

Se realizará en una primera instancia un prototipo de bajo nivel en la que se definirá la arquitectura de la información que posteriormente servirá como base del diseño.

Diseño

Tiempo estimado: 18 horas

Diseño de las pantallas finales de la aplicación. Este no es definitivo y podría sufrir cambios durante el desarrollo.

Prototipo de alto nivel

Tiempo estimado: 2 horas

Una vez detectados posibles errores o incongruencias entre diseño y funcionalidad se procederá con un prototipo de alto nivel de la aplicación a modo de demo.

Diseño de la aplicación *

Tiempo estimado: 58 horas

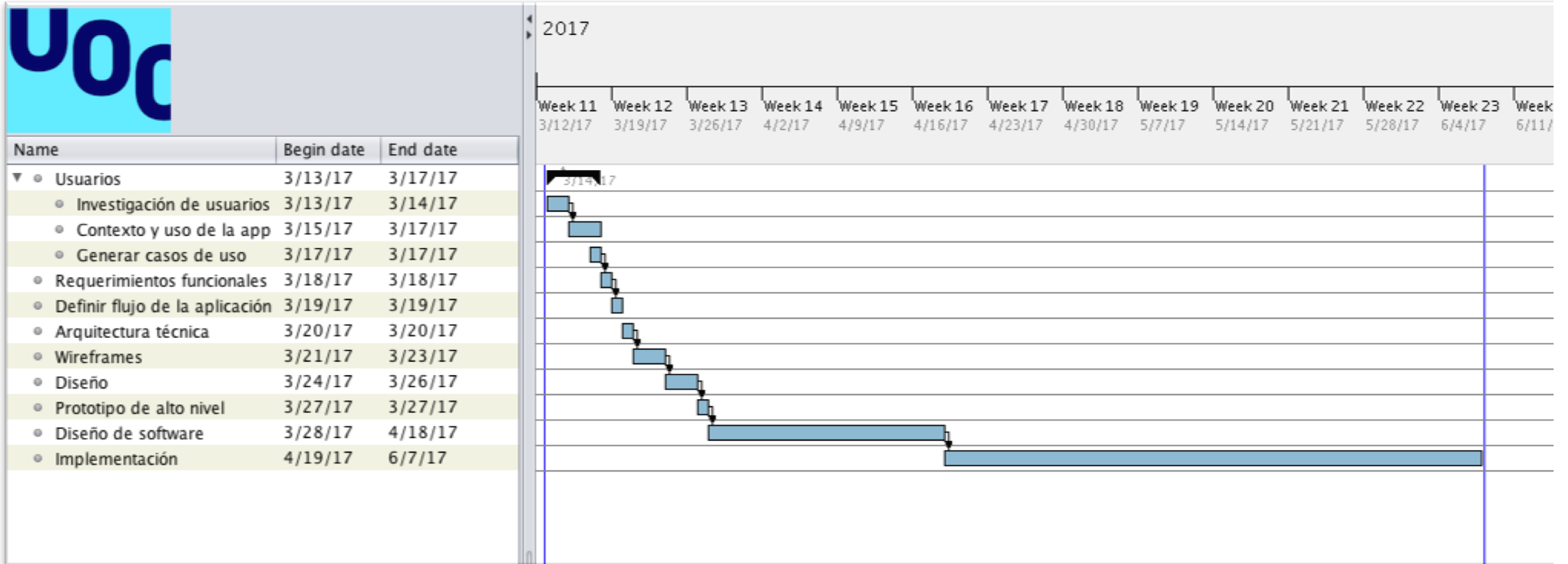
Definición, a nivel de código, de la interfaces y casos de uso de la aplicación. En esta fase no se implementarán las funcionalidades específicas a nivel de infraestructura.

Implementación *

Tiempo estimado: 95 horas

Implementación de las funcionalidades marcadas por la fase anterior.

Diagrama de Gantt



1.6 Riesgos del proyecto

Riesgo: Poco tiempo

Tipo: Requerimientos

Descripción: Debido a la naturaleza como proyecto final de máster las estimaciones no son propiamente el tiempo estimado requerido por cada tarea. En este caso nos encontramos en un escenario donde la fecha de entrega es inamovible, con muchas tareas y poco tiempo para hacerlas. Bajo este contexto se ha “estimado” el tiempo que se deberá ocupar como mucho y teniendo en cuenta la consecución de cada una de ellas. Esto supone un riesgo elevado por lo que el nivel de profundidad de cada una de ellas podría verse afectado.

Plan de acción: Llevar un control rígido de las horas invertidas en cada tarea y no sobrepasarlas sin causa muy justificada. En caso de sobrepasar, volver a reestimar las siguientes pudiendo recortar en detalle si fuera necesario.

Riesgo: Estimación sin los datos críticos

Tipo: Estimación

Descripción: Por otro lado, se ha hecho una aproximación de las horas de dedicación en una etapa muy temprana donde aún no se saben factores críticos como las funcionalidades de la aplicación y la arquitectura seleccionada. No es lo mismo diseñar 3 pantallas que 12, lo mismo que el esfuerzo de desarrollo será mayor en relación al número de casos de uso disponibles.

Plan de acción: Definir un producto mínimo viable a través de la lista de requerimientos principales que se definirá próximamente y evolucionarla a medida que el tiempo lo permita.

Riesgo: Estimación de tiempo disponible optimista

Tipo: Estimación

Descripción: También se ha tenido en cuenta una disponibilidad completa de mi tiempo libre en relación con mi horario laboral. En este sentido también hay un riesgo elevado a que este se vea reducido por requerimientos laborales o personales.

Plan de acción: Dedicar unos minutos cada domingo para evaluar el tiempo invertido de la última semana y planificar el horario de la siguiente.

Riesgo: Nuevo miembro en la familia

Tipo: Organizativo.

Descripción: Hemos adoptado una perrita que ya adoramos. Todavía no se ha separado de su madre por lo que aún no está en casa y no sabemos realmente el tiempo que nos va a ocupar. Por supuesto esto podría afectar al menos en la repartición de las horas libres.

Plan de acción: Compartido con el plan de acción anterior.

Riesgo: Boda y luna de miel

Tipo: Organizativo.

Descripción: Estamos en medio de los preparativos de boda. Me caso en junio por lo que no debería de afectar a la consecución de las tareas. No obstante la defensa del proyecto ante el tribunal será durante mi estancia en el extranjero (con una diferencia horaria de +9 horas) por lo que existe un riesgo de no poder cumplir o incluso no poder cerrar el proyecto como se espera.

Plan de acción: Planificar el viaje para poder dedicar unos minutos para consultar las cuestiones del tribunal. Ante el riesgo de no poder cerrar el proyecto, adelantar el máximo posible durante las semanas anteriores.

2 Conceptualización

2.1 Encuesta a los jugadores

Se trata de un proyecto no generalista por lo que el perfil de usuario estará centrado únicamente en aquellos que a los que le gusten los videojuegos sin contar aquellos solo lo hagan con juegos *free-to-play* (principalmente *mobile*).

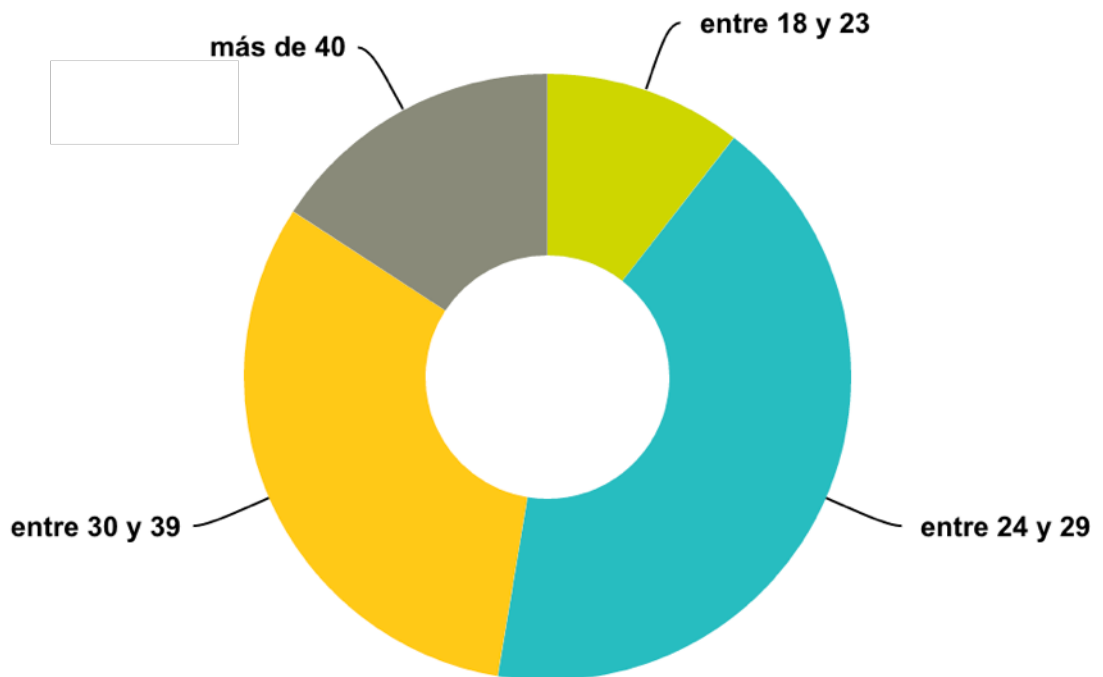
En base a esta determinación de usuarios inicial, el cual ya es un acercamiento a las dimensiones que abarcará el proyecto, se ha creado una pequeña encuesta para definir con mayor exactitud lo que serán los perfiles de usuario definitivos de la aplicación a desarrollar. Para ello se han definido una serie de preguntas con la siguientes premisas:

- Las personas que lo hagan lo harán sin obtener beneficio alguno, así que debía ser muy fácil y breve.
- Los encuestados no conocerán el motivo de esta para no influenciar en sus respuestas.

Y con los siguientes objetivos:

- Descubrir el potencial y porcentaje de productores o consumidores en el sentido del consumo colaborativo.

- Identificar el estado actual de riesgo que representan los videojuegos digitales en este tipo de proyecto.



La encuesta

- ¿Qué edad tienes?
- ¿Con qué frecuencia pruebas videojuegos que no hayas probado nunca anteriormente? Selecciona la que más se parezca a tu situación y no incluyas videojuegos free-to-play.
- ¿Cuánto tiempo dedicas de media a un videojuego antes de dejar de usarlo?
- ¿Qué haces cuando dejas de usar un juego o componente?
- ¿Qué opinas de las opciones de uso y adquisición de periféricos nuevos? (volantes, micrófonos, joysticks, etc.)
- ¿Qué opinas de las opciones de uso y adquisición de periféricos de segunda mano? (volantes, micrófonos, joysticks, etc.)
- ¿Qué opinas de las opciones de uso y adquisición de juegos nuevos (primera mano)?
- ¿Qué opinas de las opciones de uso y adquisición de juegos de segunda mano?
- ¿Has usado alguna de estas plataformas en el último año?

- ¿PC o consola?

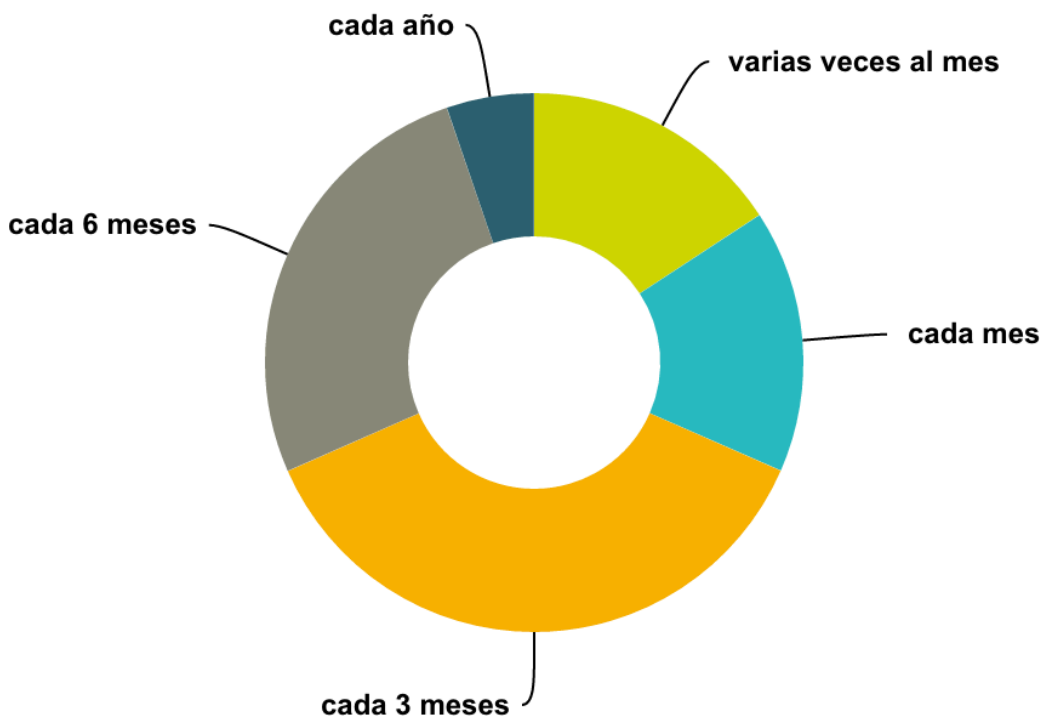
Resultados

Número de encuestados: 129

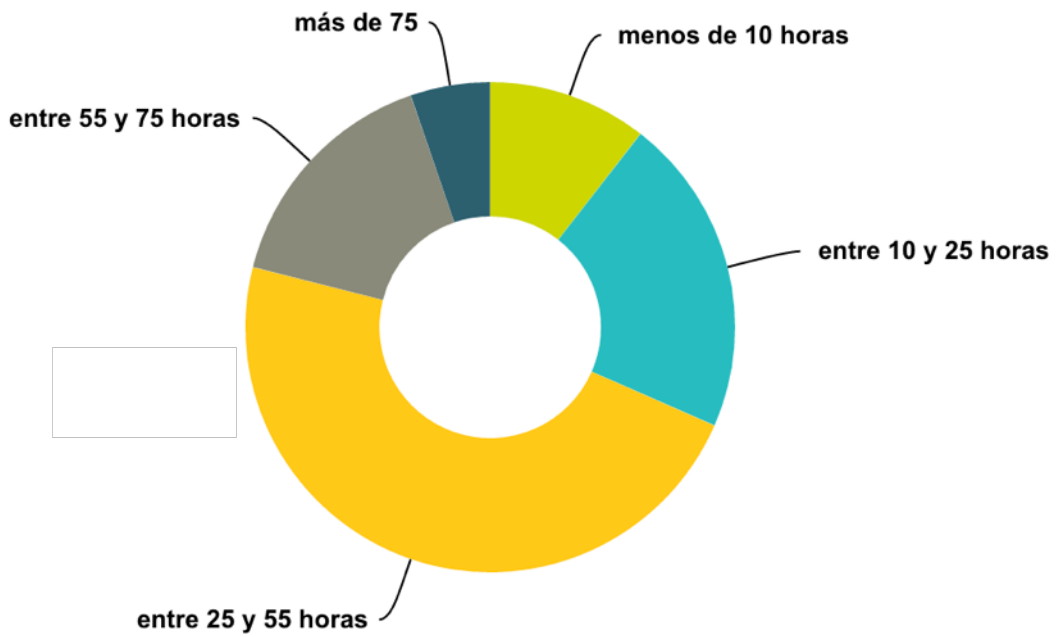
Fuente de datos: Facebook y WhatsApp.

Pregunta: ¿Qué edad tienes?

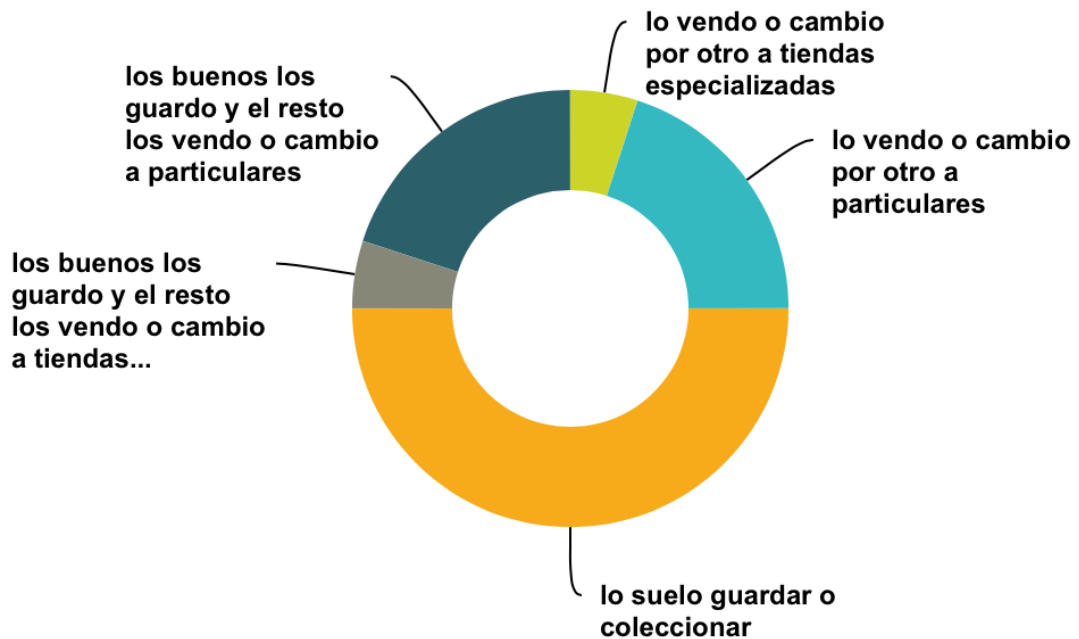
Pregunta: ¿Con qué frecuencia pruebas juegos que no hayas probado nunca antes? No incluyas videojuegos free-to-play



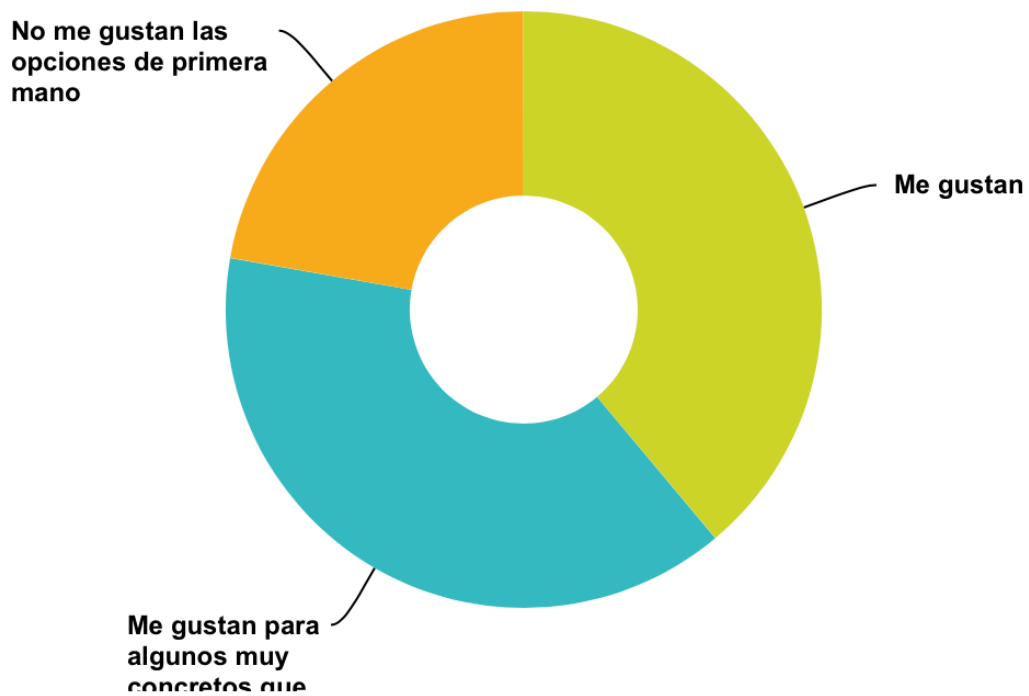
Pregunta: ¿Cuánto tiempo sueles dedicar a un juego antes de dejar de usarlo?



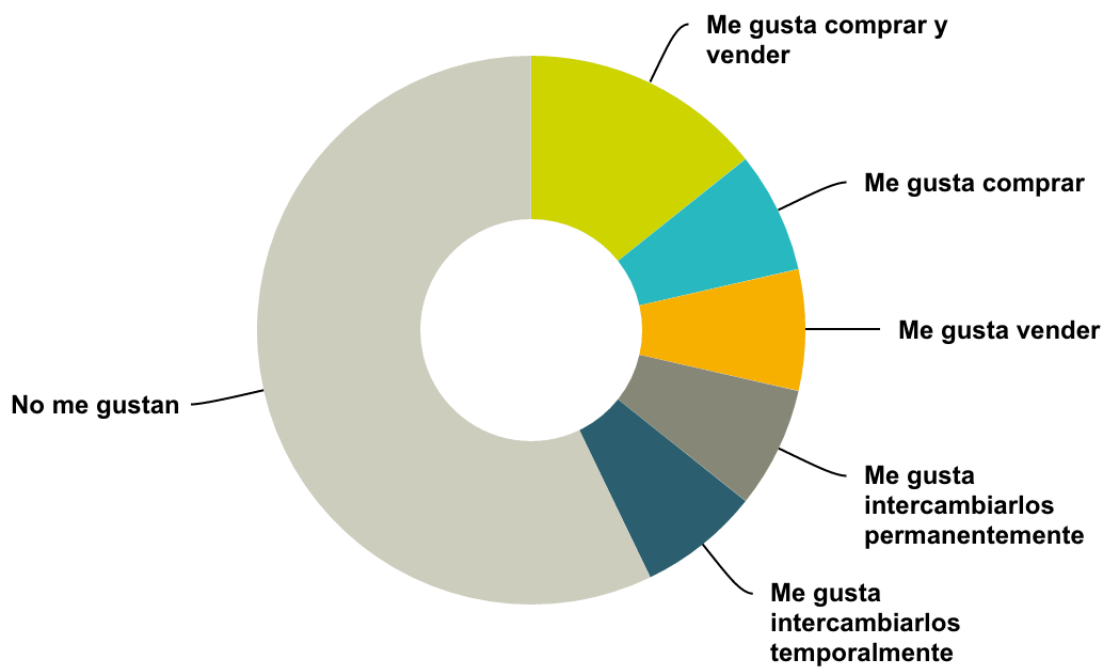
Pregunta: ¿Qué haces cuando dejas de usar un juego o componente?



Pregunta: ¿Qué opinas de las opciones de uso y adquisición de periféricos nuevos? (volantes, micrófonos, joysticks, etc.)



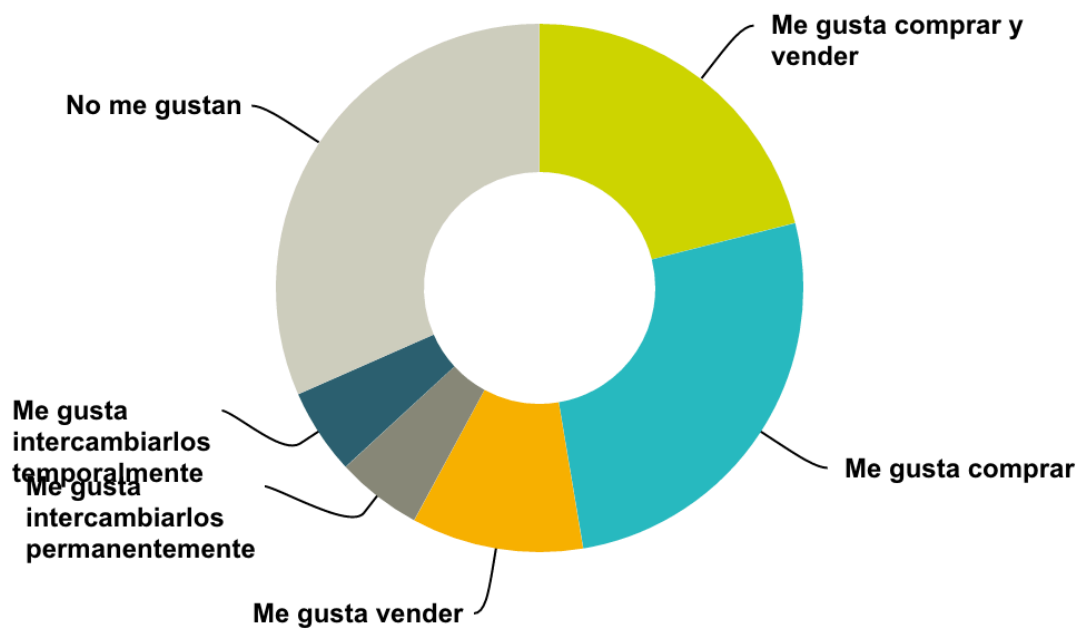
Pregunta: ¿Qué opinas de las opciones de uso y adquisición de periféricos de segunda mano? (volantes, micrófonos, joysticks, etc.)



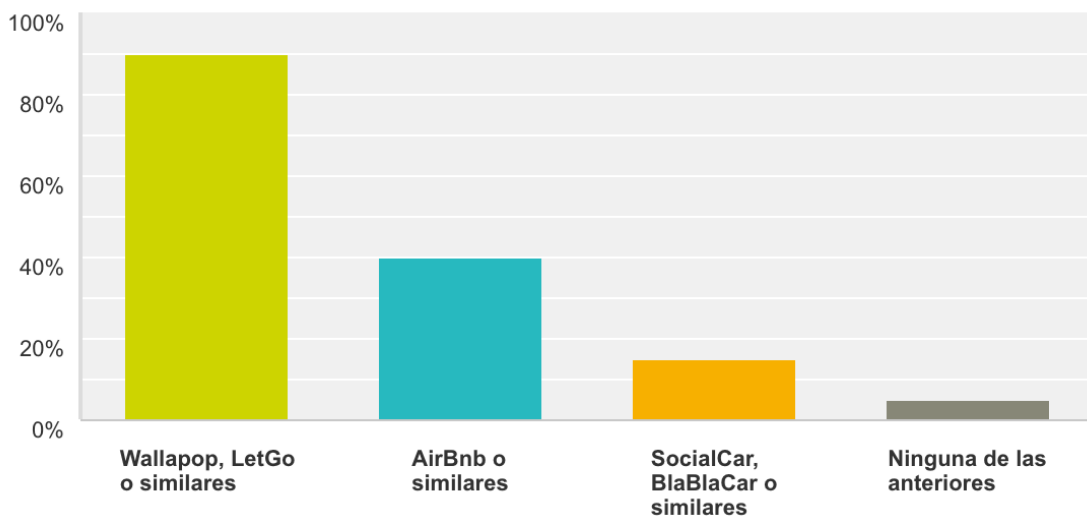
Pregunta: ¿Qué opinas de las opciones de uso y adquisición de juegos nuevos (primera mano)?



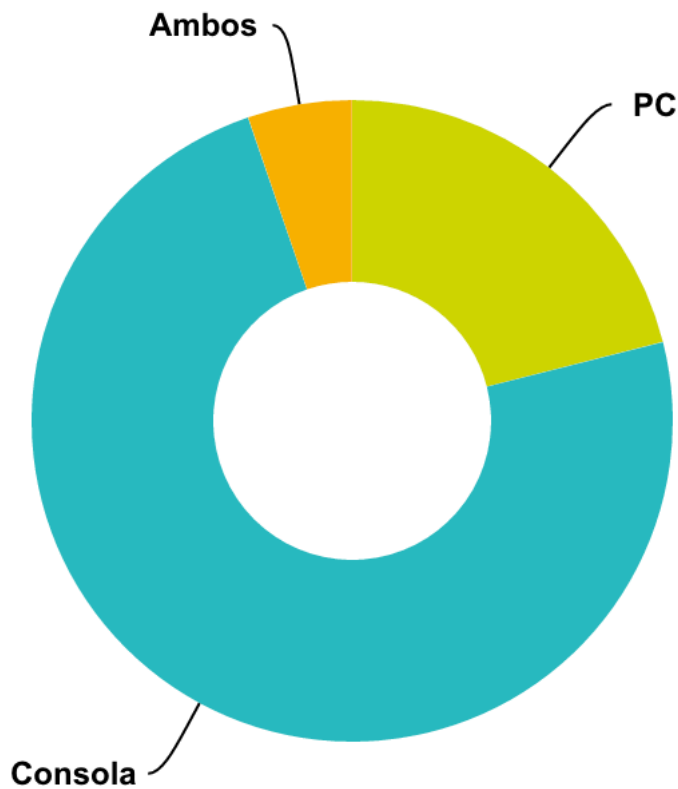
Pregunta: ¿Qué opinas de las opciones de uso y adquisición de juegos de segunda mano?



Pregunta: ¿Has usado alguna de estas plataformas en el último año?



Pregunta: ¿PC o consola?



Conclusiones

Es difícil obtener conclusiones con un número tan limitado de preguntas y encuestados pero servirán para obtener una aproximación con las que responder algunas preguntas necesarias para poder definir las funcionalidades que ayuden a cumplir los objetivos definidos anteriormente.

Objetivo: Mostrar un gran número de artículos disponibles por cada zona

Pregunta: Los productos nuevos tienen más demanda. ¿Como atraerlos?

Respuesta: Se han cruzado dos preguntas para obtener una conclusión. Por un lado se ha preguntado cada cuanto tiempo se prueban juegos “nuevos” y por otro cuantas horas en total le dedican a estos. En este sentido nos hemos encontrado con una mayoría en que los prueba cada 3 meses y les acaba dando una vida de entre 25 y 55 horas. Es decir, sería posible contar con un gran número de unidades una vez pasado este tiempo. No obstante no quita que los jugadores pudieran cederlos antes de ese plazo pero puede ser un indicador y habrá que focalizar las funcionalidades principales en promover la

publicación de juegos nuevos por parte de los productores, ya son precisamente las primeras semanas desde su publicación las que más demanda tienen por parte de los jugadores.

Objetivo: Aumentar la confianza entre los usuarios

Pregunta: ¿Cómo se va hacer?

Respuesta: Respecto a la tendencia en esta pequeña muestra de usuarios, nos encontramos con tres tipos de usuarios, todos posibles usuarios potenciales, por orden decreciente por número de respuestas:

- Los que compran juegos y los guardan.
- Los que compran y venden de segunda mano.
- Los que intercambian con otros en el mercado de segunda mano.

Por otro lado, apenas un 5% de los encuestados afirman no haber usado ninguna de las plataformas de consumo colaborativo más populares. Es decir, el 95% ya ha ejercido uno de estos dos roles durante este año por lo que la confianza en este tipo de plataformas es creciente. En este sentido habrá que seguir las buenas prácticas ya promovidas por otros servicios ya asentados.

Objetivo: Mostrar un gran número de artículos disponibles por cada zona

Pregunta: ¿Software o hardware?

Respuesta: Revisando los resultados individualmente, los usuarios que obtienen sus juegos en versión digital suelen ser jugadores de PC y parece que los de consola aún se resisten al formato físico, algo que obviamente posibilita el poder venderlos o alquilarlos. Por otro lado ha habido una mayor tendencia al mercado de segunda mano cuando se ha cruzado con los jugadores de PC. Teniendo en cuenta que han sido una minoría, parece que el producto principal en una primera fase estaría en los videojuegos.

2.2 Perfiles de usuario



Perfil: Productor

Nombre: Gunter

Edad: 31 años

Ocupación: Programador web

Comportamiento: Nunca faltan las últimas novedades en su estantería y no duda en acercarse a la tienda más cercana a comprarse el último de su saga favorita.

Motivación: Amortizar los videojuegos que compra sin necesidad de desprenderse de ellos definitivamente.



Perfil: Consumidor

Nombre: Niles

Edad: 22 años

Ocupación: Estudiante universitario

Comportamiento: Jugador hardcore. Es capaz de pasarse un juego en tan solo un fin de semana.

Motivación: Poder pasarse las últimas novedades sin necesidad de gastarse mucho dinero ni andar preocupándose en comprar y venderlos en otras plataformas de segunda mano.



Perfil: Mixto

Nombre: Elise

Edad: 27

Ocupación: Gestora de contenidos web

Comportamiento: Muy competitiva en el juego online. Su meta es siempre estar arriba en todos los enfrentamientos.

Motivación: Los juegos offline están bien pero pasan rápido para gastarse el dinero en su compra. Busca poder alquilar esos juegos durante al menos una semana sin tener que pagar tanto como se paga en las tiendas profesionales.

2.3 Escenarios

Descripción	Gunter se ha aburrido de un juego de su colección
Código	E001
Contexto	Ya hace 1 mes que Gunter se compró aquel juego de acción que tanto esperaba. Hace justo un par de días que se pasó la historia principal y desde entonces no lo ha vuelto a usar.
Objetivo	Es posible que pueda recuperar una parte del dinero invertido en el juego alquilándolo por días. Se dispone a publicar el juego de inmediato.

Escenario	Niles busca entretenimiento para el fin de semana
Código	E002
Contexto	Niles va a tener todo el fin de semana libre, algo que no se repite muy a menudo pero los juegos de los que dispone ya los tiene bastante aborrecidos.
Objetivo	Buscar un juego con el que pueda pasar el siguiente fin de semana jugando y, si es posible, el cual pueda finalizar la historia principal en ese tiempo.

Escenario	Elise busca un juego muy específico
Código	E003
Contexto	Elise sabe que ya hace un par de semanas que ha salido el último capítulo de la saga Uncharted. Es un juego al que le tiene muchas ganas pero no está dispuesta a gastarse tanto dinero por unas pocas horas de juego.

Objetivo	Buscar el uncharted más cercano a su domicilio. No tiene prisa en jugarlo por lo que puede esperar a que esté disponible.
Escenario	Gunter no se fía de cualquiera
Código	E004
Contexto	Gunter tiene una serie de juegos publicados pero no está dispuesto a entregárselo a cualquiera porque sabe que hay un riesgo a que el juego no sea devuelto.
Objetivo	Gunter quiere ver la popularidad de la persona que le está solicitando la reserva del juego. Esto incluye ver su perfil y las opiniones que ha recibido de otros usuarios en alquileres anteriores.

3 Definición funcional

3.1 Listado de requisitos

Nombre	R01 Autenticación
Caso de uso relacionado	UC01.1 UC01.2
Descripción	<p>Un usuario debe poder registrarse y autenticarse en la plataforma mediante usuario y contraseña o mediante Facebook o Google. El usuario constará de:</p> <ul style="list-style-type: none"> - Nombre público - Email - Contraseña (Si es mediante oauth2 no se conocerá) - Zona (Geolocalizado)

Nombre	R02 Publicación de videojuegos
Caso de uso relacionado	UC02.1
Descripción	<p>Un usuario debe poder publicar un juego con las siguientes características:</p> <ul style="list-style-type: none"> - Nombre - Descripción - Fotografías - Precio - Disponibilidad

Nombre	R03 Disponibilidad
Caso de uso relacionado	UC02.2

Descripción	Un usuario debe poder definir la disponibilidad un juego publicado la cual constará de un calendario semanal. Este debe ser visible en la ficha pública.
--------------------	--

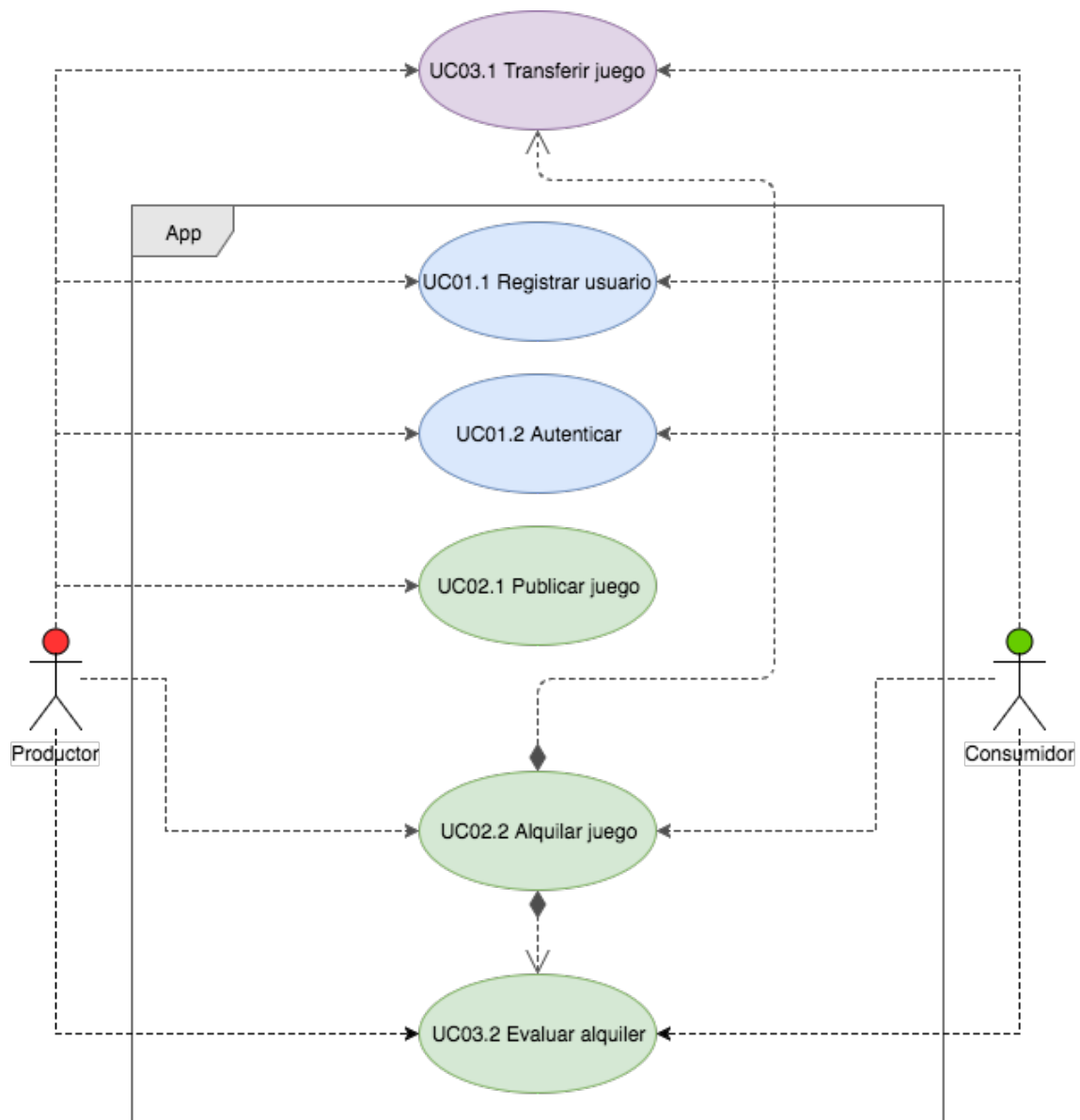
Nombre	R04 Solicitud de reserva
Caso de uso relacionado	UC02.2
Descripción	Un usuario autenticado debe poder solicitar la reserva de un juego para uno o más días a través de su ficha pública.
Restricción	Las fechas no podrán extenderse más allá de la próxima reserva. Si la reserva empieza un lunes y existe otra reserva previa el viernes siguiente, esta solo podrá extenderse hasta el jueves.

Nombre	R05 Reservar de juego
Caso de uso relacionado	UC02.2
Descripción	Un usuario debe poder acceder a las peticiones de reserva de sus juegos publicados y aceptarlas o denegarlas. En caso de ser aceptadas deberá verse reflejado en su ficha pública y quedar bloqueado para no poder ser alquilado por otro usuario en las mismas fechas.

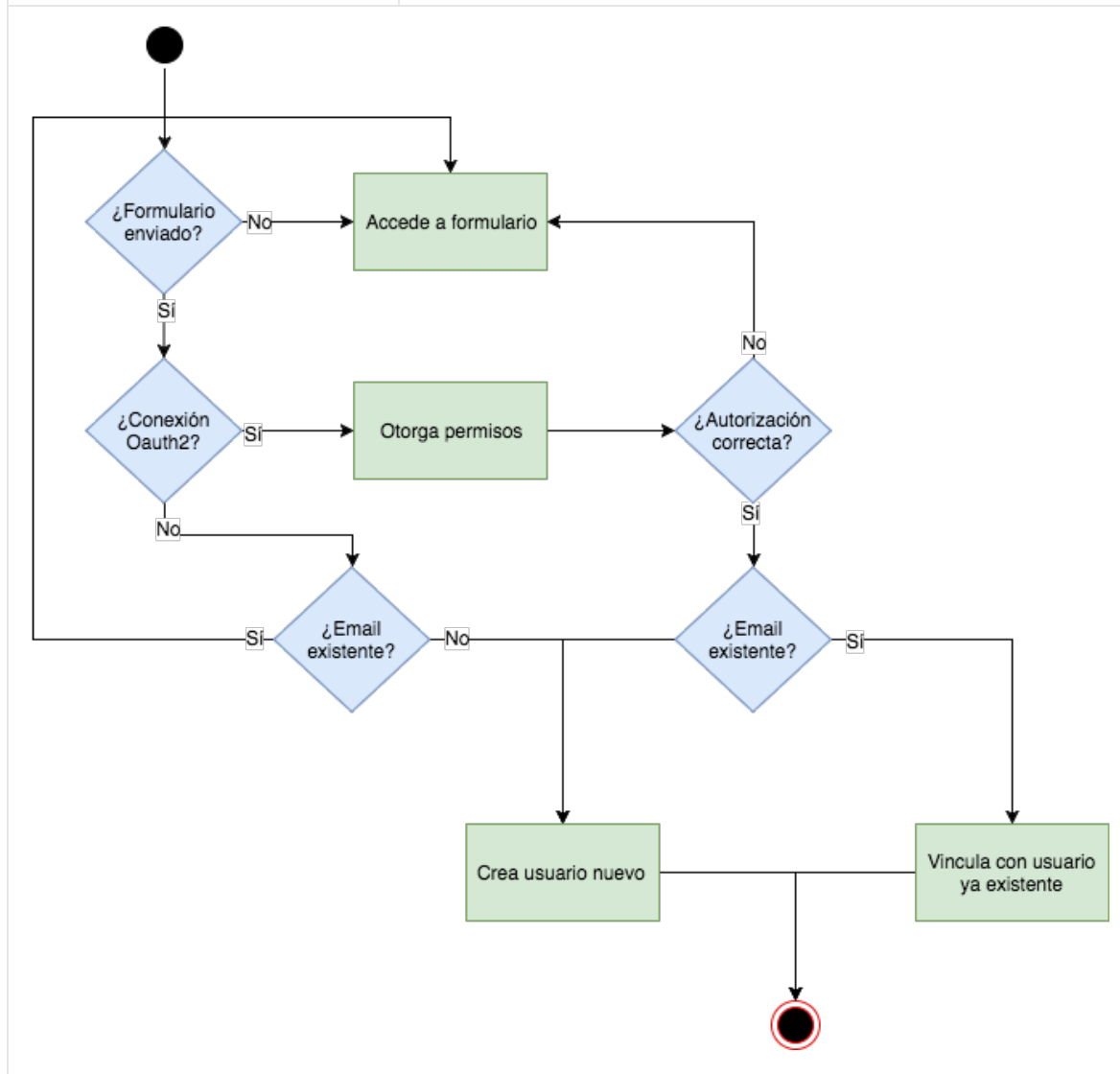
Nombre	R06 Valorar un juego
Caso de uso relacionado	UC03.2

Descripción	<p>Un usuario debe poder valorar un alquiler, tanto el producto como el consumidor. La valoración constará de una descripción y una puntuación del 1 al 5 de los siguientes factores:</p> <ul style="list-style-type: none"> - Comunicación - Estado del juego - Puntualidad
--------------------	---

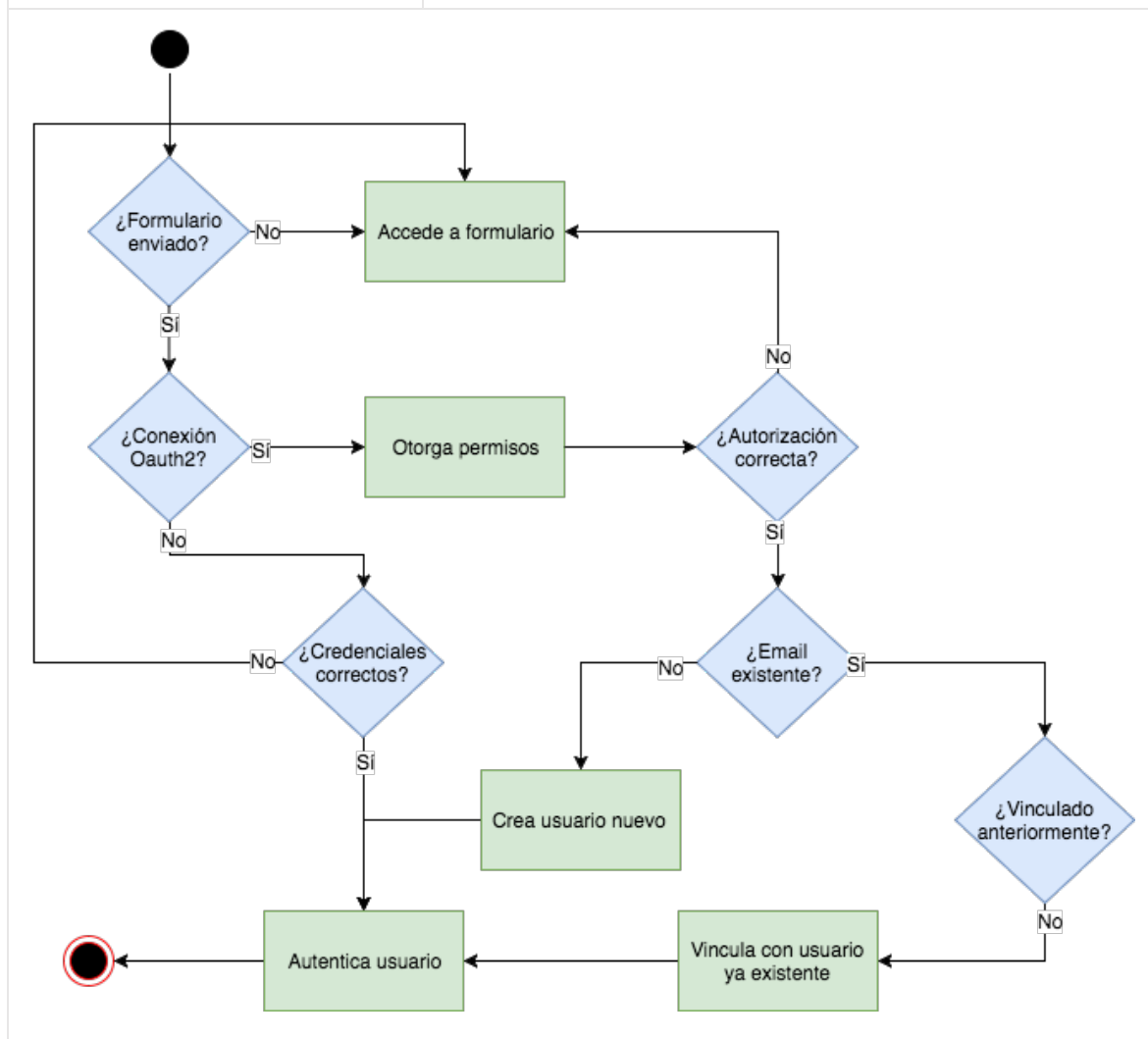
3.2 Casos de uso

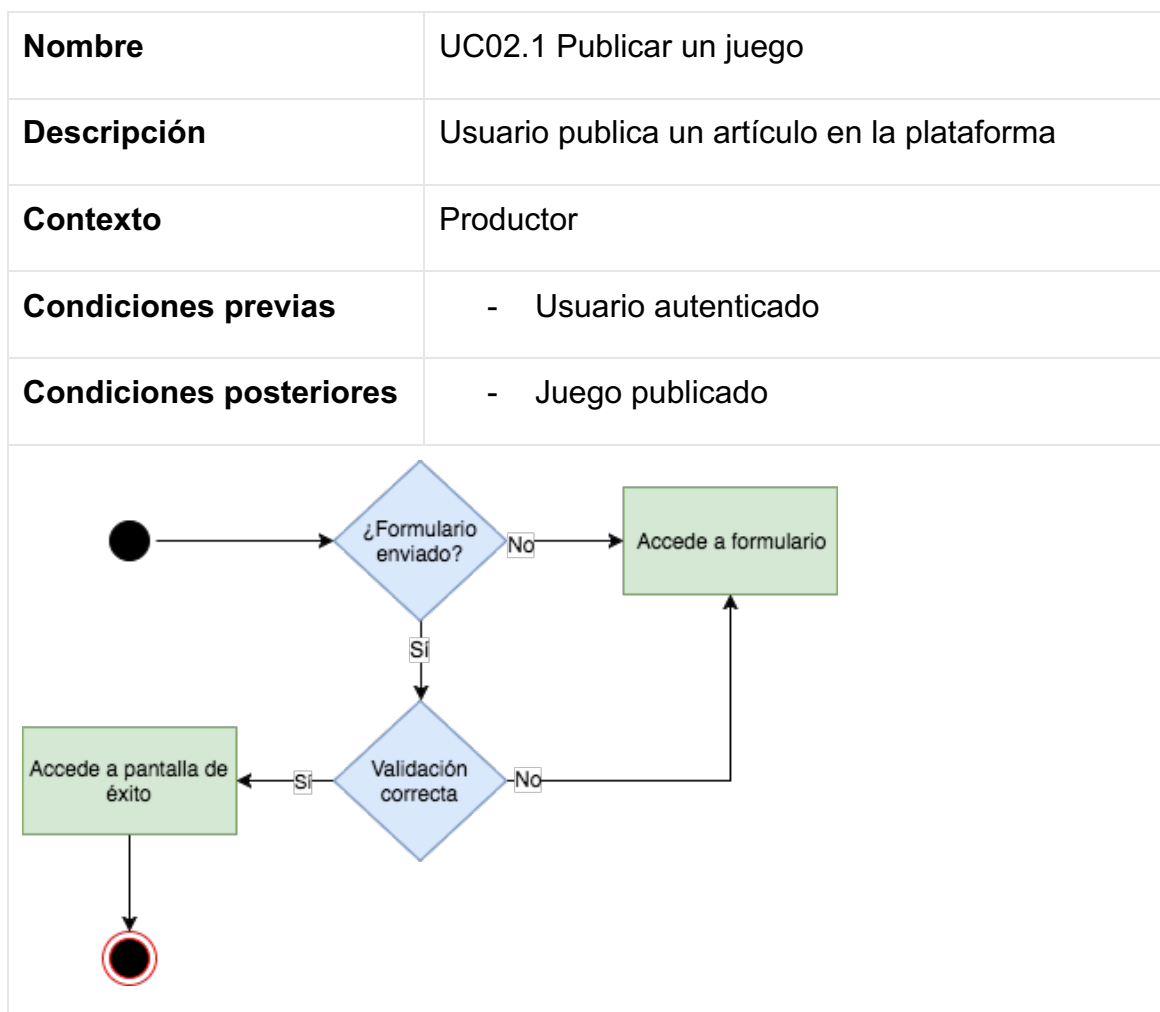


Nombre	UC01.1 Registrar usuario
Descripción	Usuario anónimo se registra en la plataforma
Contexto	Productor y consumidor
Condiciones previas	- Usuario anónimo
Condiciones posteriores	- Usuario registrado - Usuario autenticado

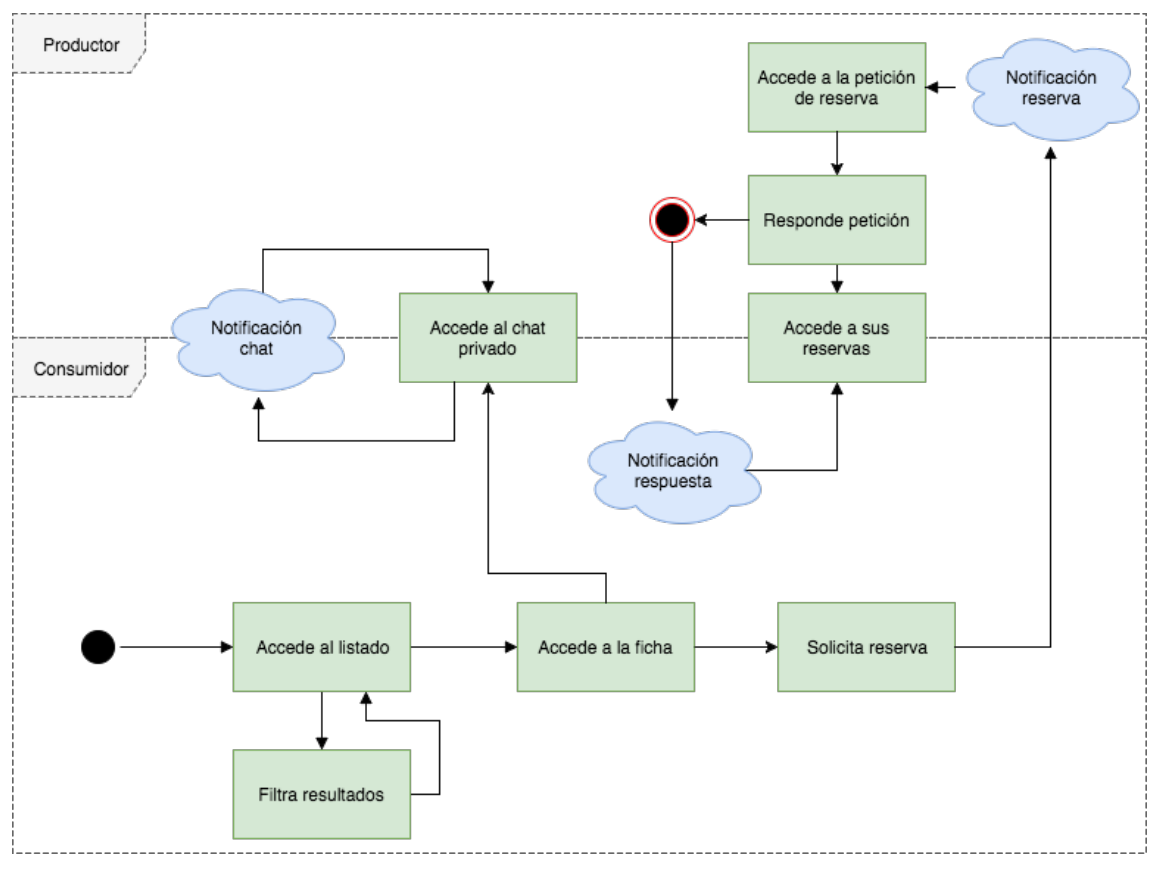


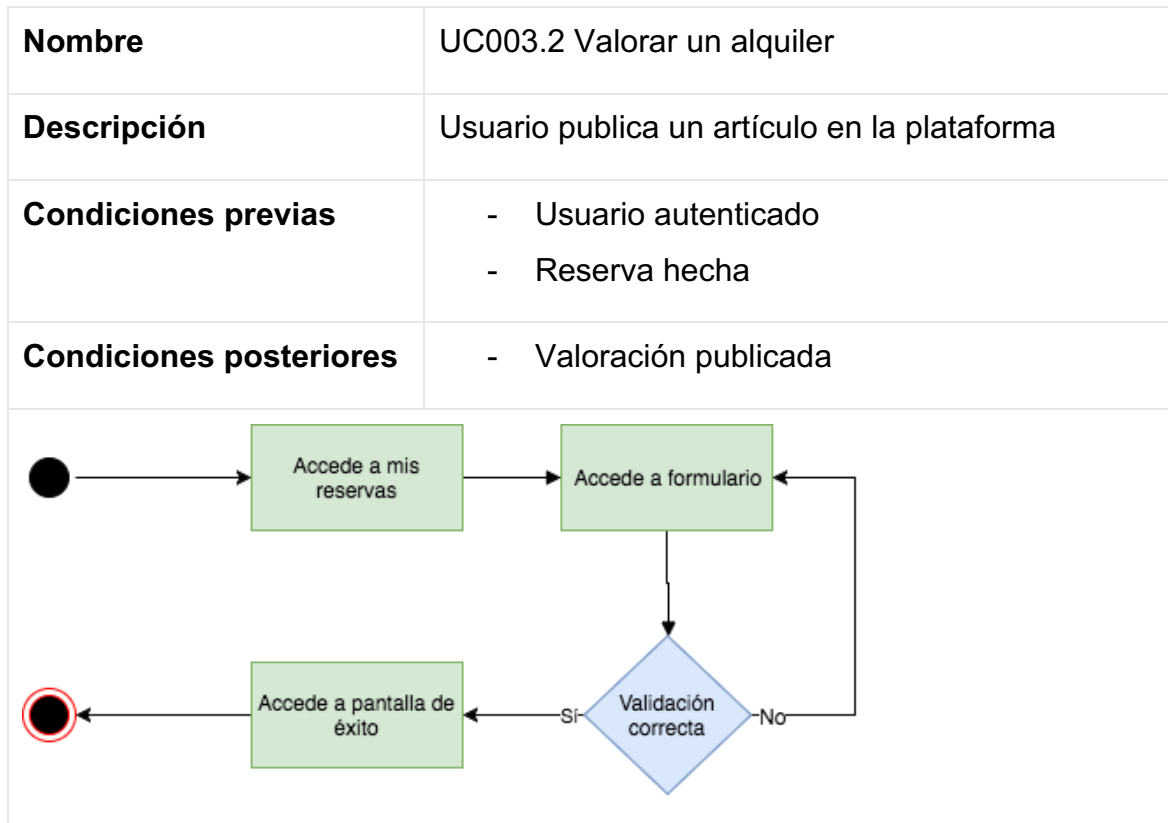
Nombre	UC01.2 Autenticar usuario
Descripción	Un usuario se identifica en la plataforma
Contexto	Productor y consumidor
Condiciones previas	- Usuario registrado
Condiciones posteriores	- Usuario autenticado





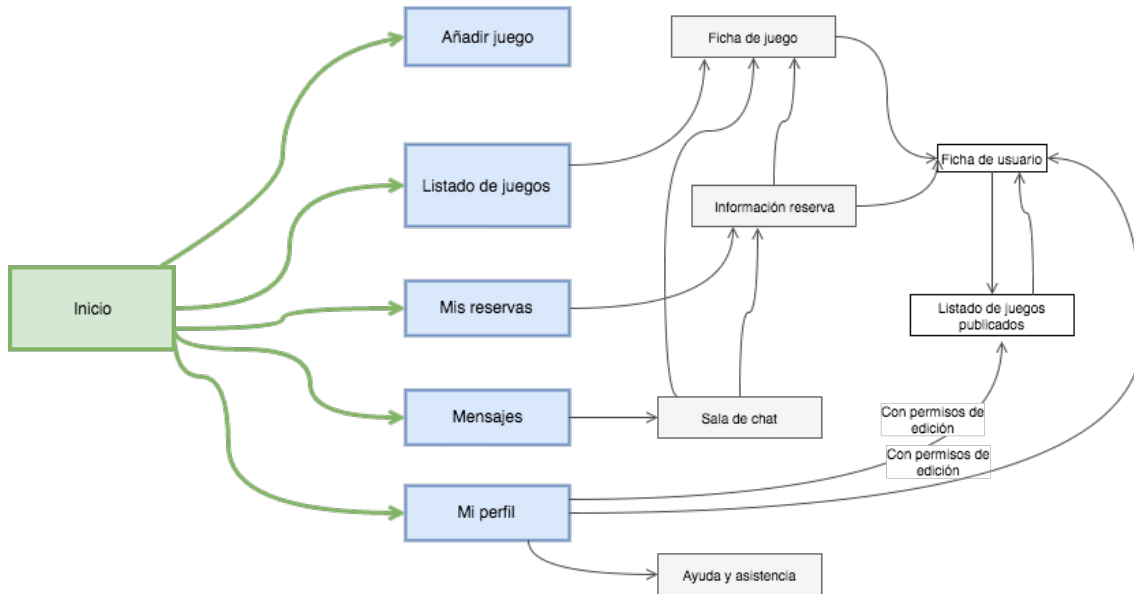
Nombre	UC02.2 Alquilar un juego
Descripción	Se alquila un juego entre dos usuarios
Condiciones previas	<ul style="list-style-type: none"> - Usuario autenticado - Juego publicado
Condiciones posteriores	<ul style="list-style-type: none"> - Reserva hecha.





3.2 Árbol de navegación

La navegación estará basada en la estrategia de “*bottom bar*”, tal y como sugieren las guías de estilo de Apple³ y Android⁴. Las pestañas principales serán las que se consideran más importantes para facilitar el cumplimiento de los objetivos de los usuarios: el explorador de productos, las reservas efectuadas, añadir un nuevo producto, las salas de chat y el perfil de usuario.



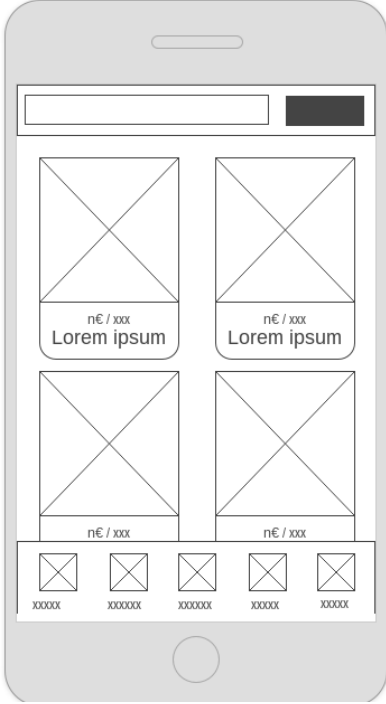
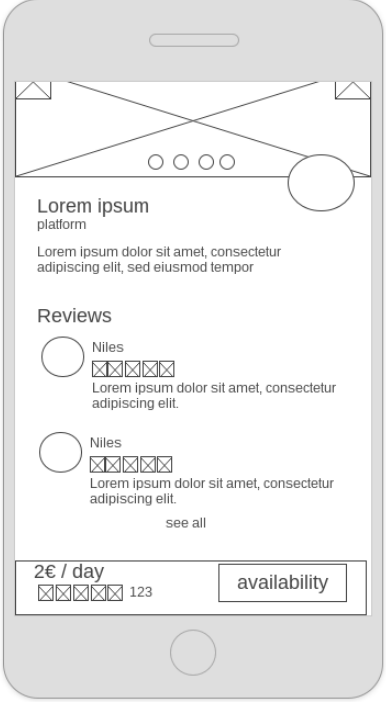
³ <https://developer.apple.com/ios/human-interface-guidelines/ui-bars/tab-bars/>

⁴ <https://material.io/guidelines/components/bottom-navigation.html>

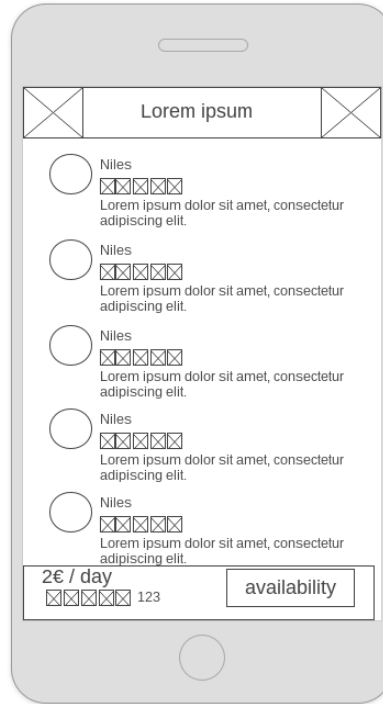
4. Diseño

4.1 Wireframes

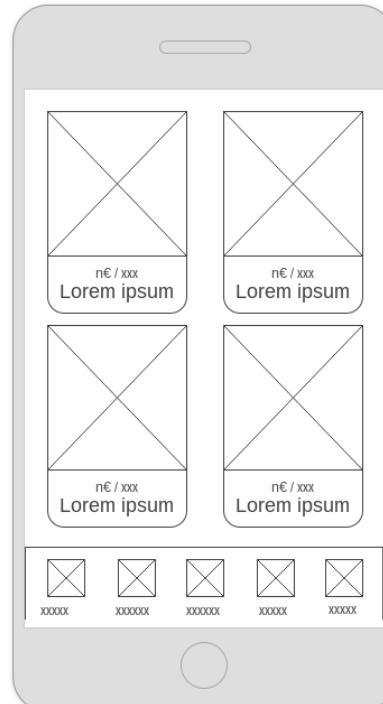
En la etapa de prototipado de bajo nivel se han diseñado las pantallas más frecuentes y con información más compleja.

<p>Listado</p>	
<p>Ficha</p>	

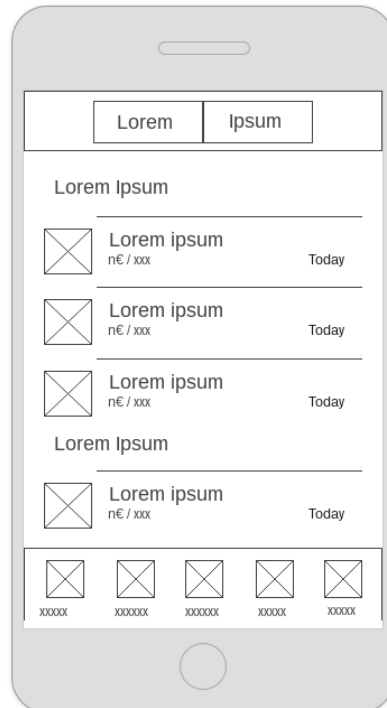
Evaluaciones (Ficha)



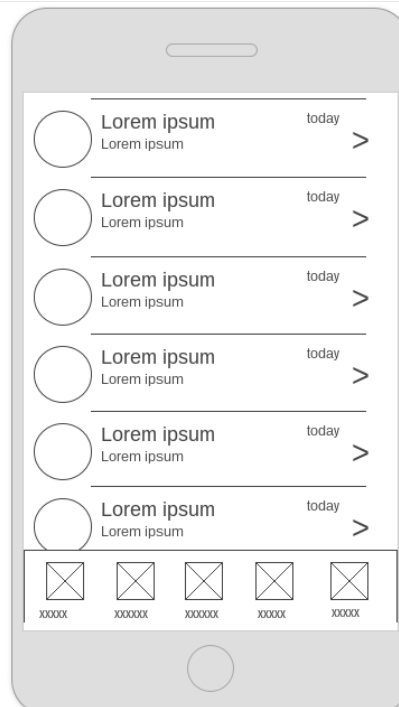
Favoritos



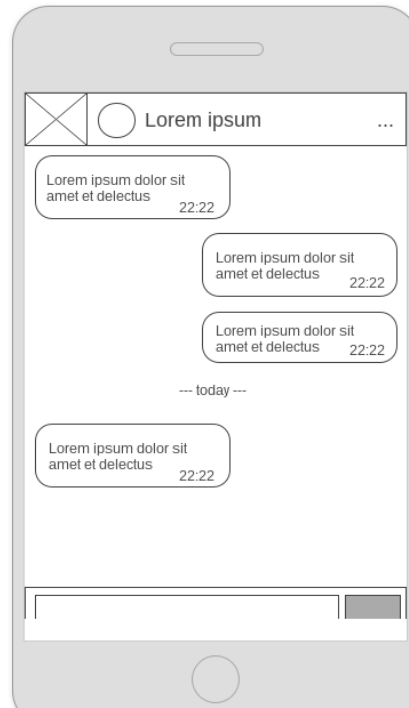
Reservas



Mensajes



Sala de chat



Perfil

(con modo edición si es el del propio usuario)



4.2 Prototipo

Tras acabar los Wireframes se ha pasado a la fase de diseño de alto nivel. En este proceso se ha pasado por tres iteraciones:

- **Primer diseño:** Consistió en una transformación casi calcada de las pantallas dibujadas en la etapa anterior.
- **Evaluación:** Se han seleccionado dos personas que están interesadas en la app (una de ellas es diseñador) y se han valorado los posibles problemas de arquitectura, diseño y usabilidad. En esta se detectaron problemas de incongruencias con el uso de los colores en los CTA⁵ y fondos, y así como algunos detalles a mejorar en cuanto a la usabilidad en cuanto al seguimiento de las reservas.
- **Segundo diseño:** Se ha tomado nota de todos los comentarios y sugerencias de los *testers* y se ha traspasado al nuevo diseño

El diseño está orientado a la usabilidad por lo que se busca que el contenido sea el principal punto de atención y que este sea algo “invisible” para el usuario.

Paleta de colores

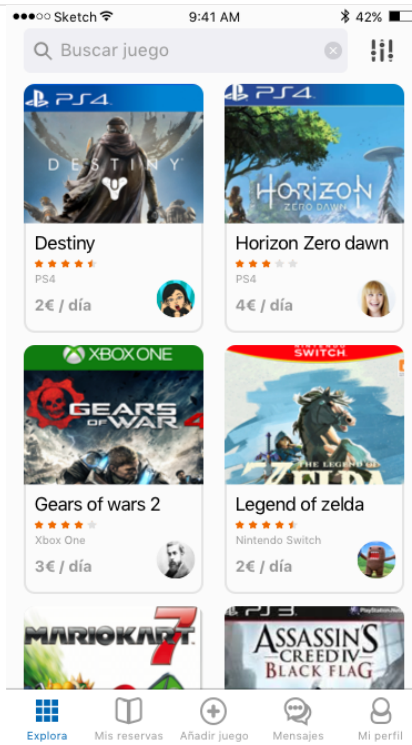


Botones

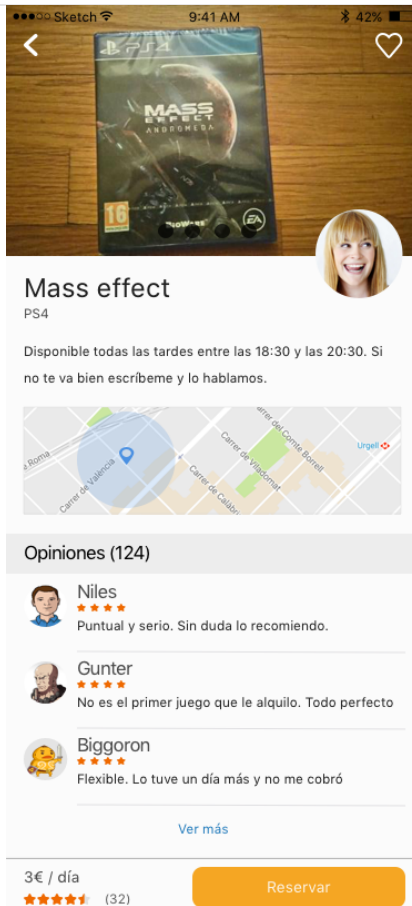
Se han dividido en dos categorías, los principales CTA de la pantalla activa y los secundarios. Estos primeros se han creado con un fondo destacado con el objetivo de llamar la atención al usuario y los segundos con un efecto de fundido entre el contenido con el objetivo de ser captados durante la lectura del contenido.

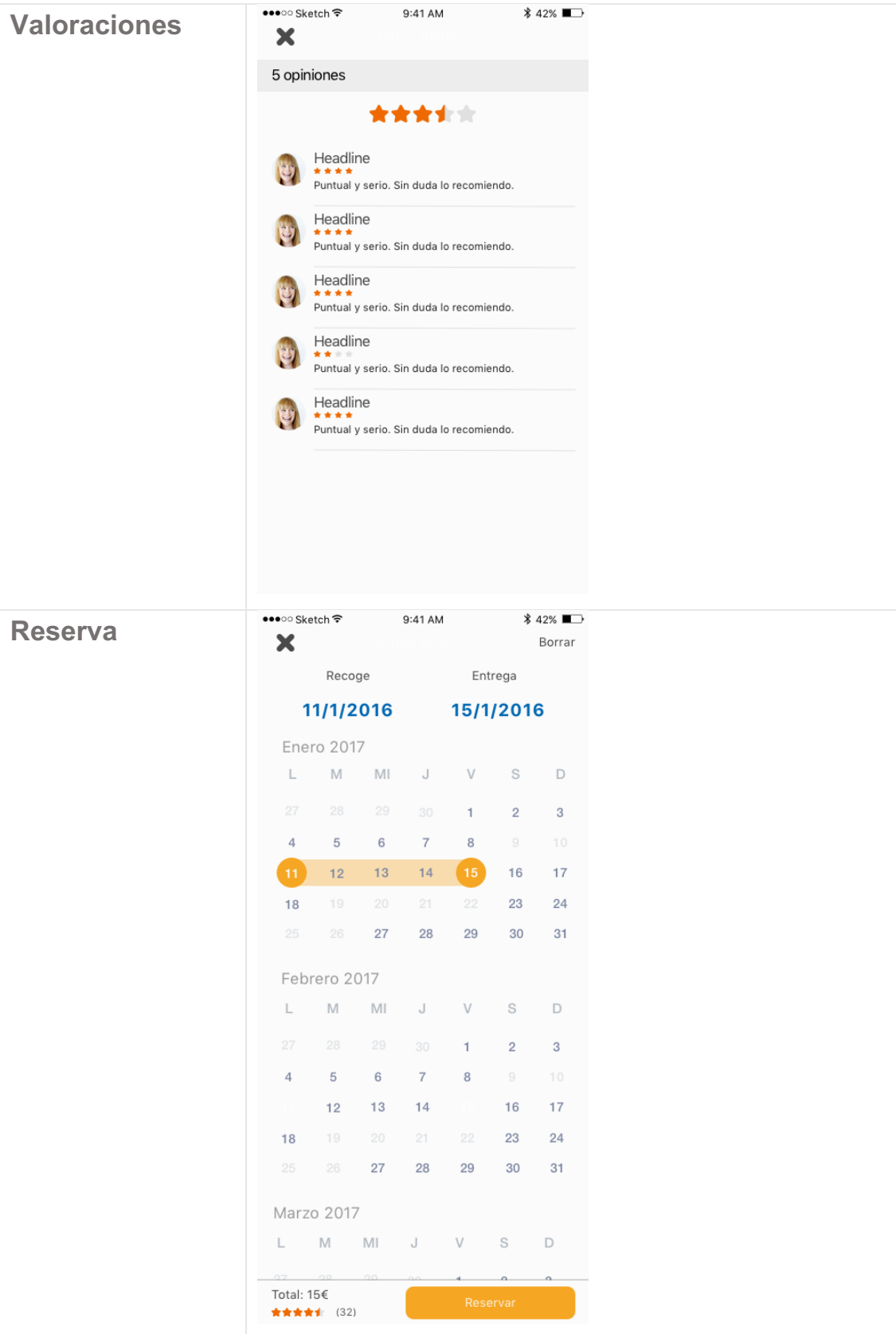
⁵ Call to action: Elemento con el que queremos que el usuario realice una acción


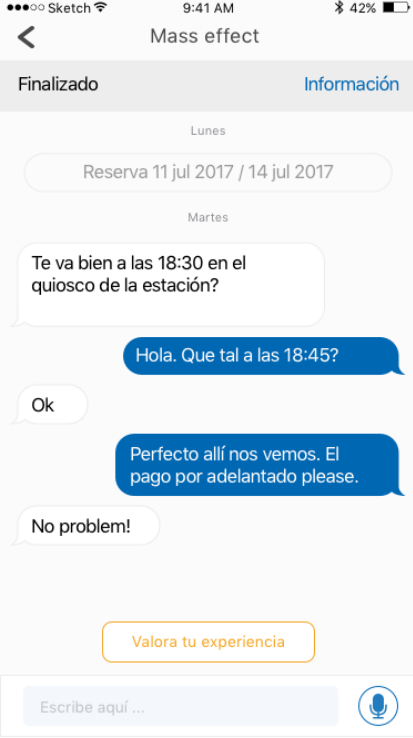
Listado



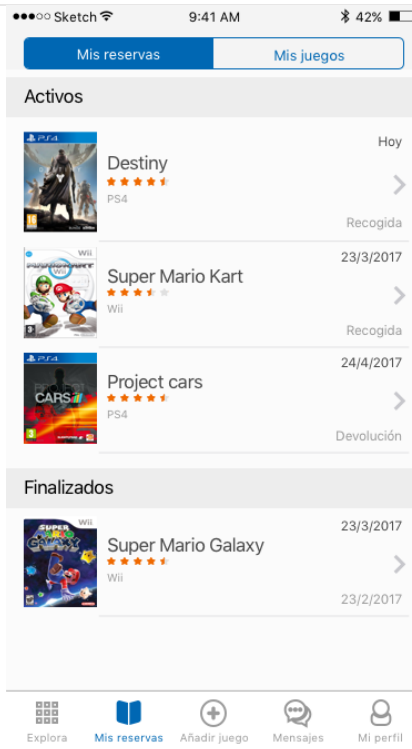
Ficha pública de producto



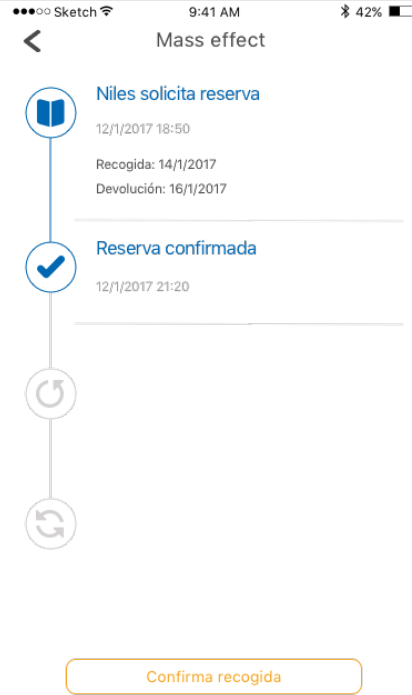


Confirmación de Reserva	 <p>¡Reserva finalizada!</p> <p>Solo falta ponerte de acuerdo con las hora de entrega.</p> <p>¡Contacta ahora!</p>
Chat	 <p>Sketch 9:41 AM 42%</p> <p>Mass effect</p> <p>Finalizado Información</p> <p>Lunes</p> <p>Reserva 11 jul 2017 / 14 jul 2017</p> <p>Martes</p> <p>Te va bien a las 18:30 en el quiosco de la estación?</p> <p>Hola. Que tal a las 18:45?</p> <p>Ok</p> <p>Perfecto allí nos vemos. El pago por adelantado please.</p> <p>No problem!</p> <p>Valora tu experiencia</p> <p>Escribe aquí ...</p>

Lista de reservas



Información de la reserva



<p>Lista de mensajes</p>	
<p>Añadir o editar producto</p>	

Editar
disponibilidad

Sketch 9:41 AM 42%
✕ Borrarr

Enero 2017

L	M	MI	J	V	S	D
27	28	29	30	1	2	3
4	5	6	7	8	9	10
	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Febrero 2017

L	M	MI	J	V	S	D
27	28	29	30	1	2	3
4	5	6	7	8	9	10
	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Marzo 2017

L	M	MI	J	V	S	D
27	28	29	30	1	2	3
4	5	6	7	8	9	10
	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

Ficha pública de usuario

The screenshot shows a mobile app interface for a user profile. At the top, there's a header with a dog's photo and the name 'Elise'. Below the name is a bio: 'Gamer porque sí. Todo el mundo sabe quién que va a caer cuando aparezco en el matchmaking.' A map shows the user's location in Valencia. The 'Opiniones' section lists three reviews from other users: Link (5 stars), Gunter (5 stars), and Infiltrada (5 stars). The 'Juegos publicados' section shows two game cards: 'Destiny' for PS4 and 'Super Mario Galaxy' for Wii, both priced at 3€ per day. At the bottom is a navigation bar with icons for 'Explora', 'Mis reservas', 'Añadir juego', 'Mensajes', and 'Mi perfil'.

Elise

Gamer porque sí. Todo el mundo sabe quién que va a caer cuando aparezco en el matchmaking.

Opiniones

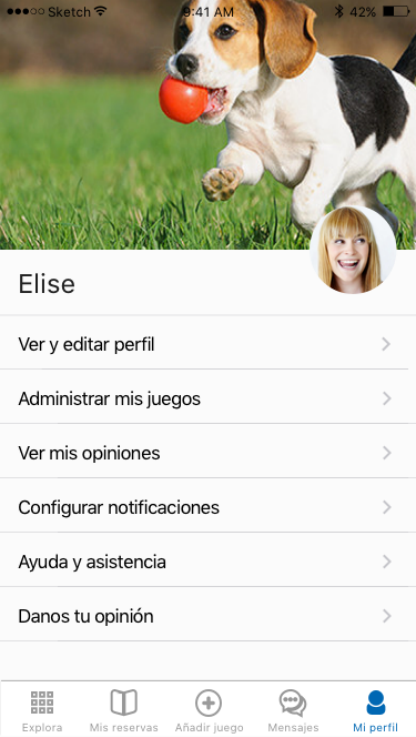
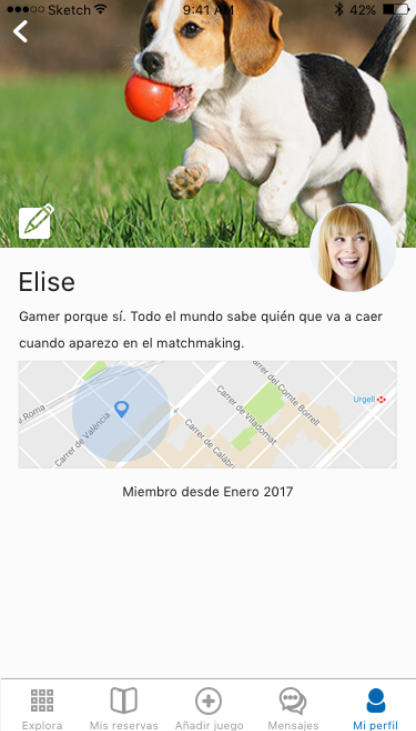
- Link ★★★★★
Puntual y serio. Sin duda lo recomiendo.
- Gunter ★★★★★
No es el primer juego que le alquilo. Todo perfecto
- Infiltrada ★★★★★
Puntual y serio. Sin duda lo recomiendo.

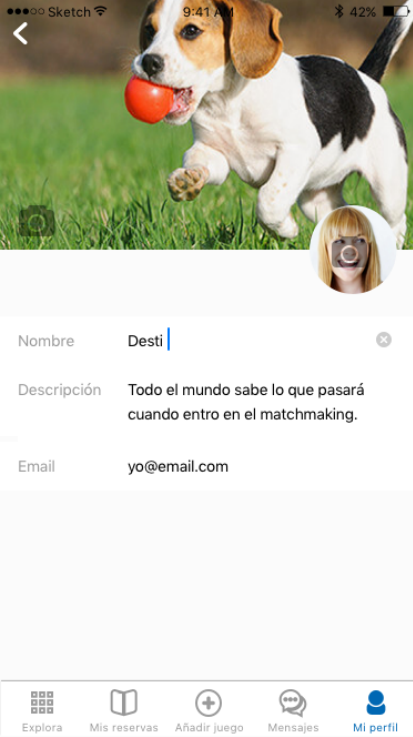

Ver más

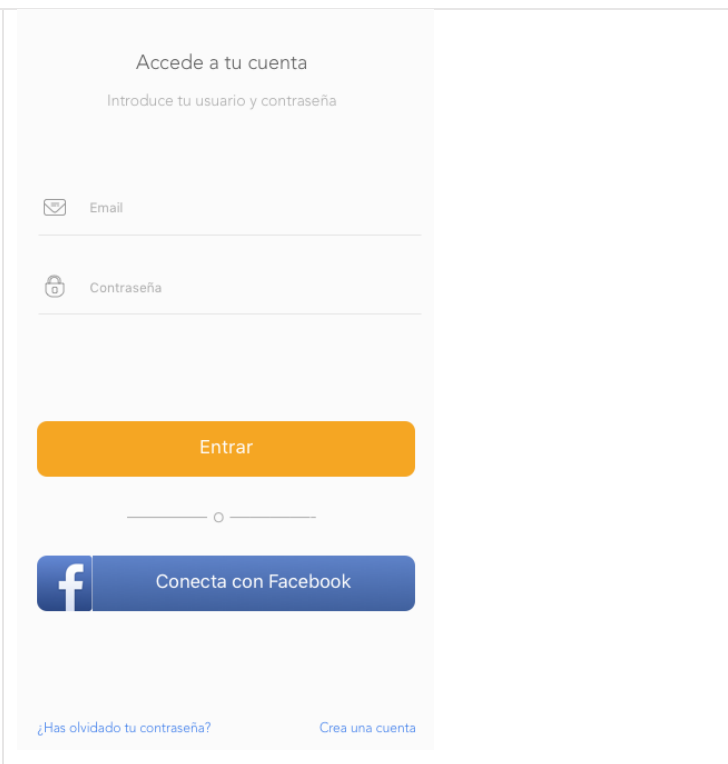
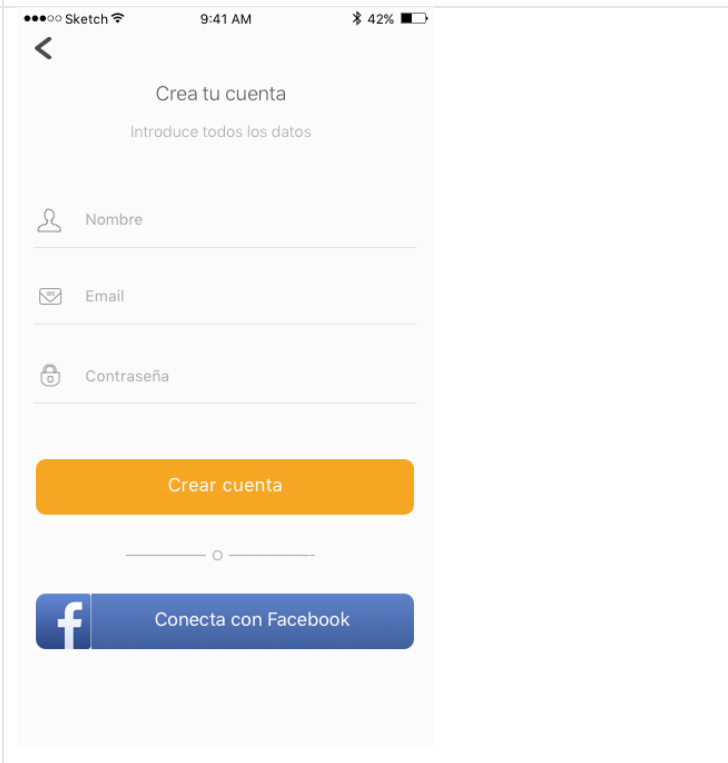
Juegos publicados

- Destiny ★★★★★
PS4
3€ / día
- Super Mario Galaxy ★★★★★
Wii
3€ / día

Explora Mis reservas Añadir juego Mensajes Mi perfil

<p>Perfil</p>	
<p>Ficha privada de usuario</p>	

<h3>Editar perfil</h3>	
<h3>Ficha privada de producto</h3>	

<h3>Login</h3>	 <p>Accede a tu cuenta Introduce tu usuario y contraseña</p> <p>Email</p> <p>Contraseña</p> <p>Entrar</p> <p>o</p> <p>Conecta con Facebook</p> <p>¿Has olvidado tu contraseña? Crea una cuenta</p>
<h3>Registro</h3>	 <p>Sketch 9:41 AM 42%</p> <p><</p> <p>Crea tu cuenta Introduce todos los datos</p> <p>Nombre</p> <p>Email</p> <p>Contraseña</p> <p>Crear cuenta</p> <p>o</p> <p>Conecta con Facebook</p>

5. Arquitectura técnica

En la definición de la arquitectura se han estudiado dos escenarios diferentes. Por un lado estaría la implementación rápida de Firebase y por otro el desarrollo personalizado de una api-rest.

5.1 Firebase

Firebase cuenta con un gran número de servicios ya integrados en la plataforma y que facilitan el arranque casi inmediato (omitiendo la curva de aprendizaje). Entre sus servicios se pueden destacar algunos muy útiles para el desarrollo de una app móvil:

- **Autenticación de usuarios:** Da soporte a los principales proveedores de oauth2, acceso por usuario y contraseña o adaptador a un sistema propio.
- **Base de datos:** Está basado en NoSQL, es en tiempo real y tiene soporte offline (la SDK guarda los datos en el dispositivo como prevención ante cualquier corte de conexión)
- **Almacenamiento:** Permite almacenar ficheros de los usuarios (video, audio, etc.)
- **Campañas:** Proporciona herramientas para envío de campañas, programación de envíos, segmentación de usuarios, etc.
- **Notificaciones:** Las notificaciones se envían mediante FCM, el sucesor natural de GCM, y es compatible con las plataformas a las que se le dará soporte: iOS y Android.
- **Analítica:** Provee una amplia gama de estadísticas de uso de las aplicaciones.
- **Funciones Lambda:** Es posible crear funciones reactivas a eventos de otros productos de Firebase como por ejemplo Analytics o Database, así como endpoints directos sin necesidad de contar con una infraestructura propia de servidores.

5.2 Api-Rest

Api rest es el método más tradicional, no por ello peor. Entre sus características principales que afectan al desarrollo estarían:

- **Flexibilidad:** No contar con un paquete “todo-en-uno” representa un problema en un inicio pero te habilita el poder implementar las tecnologías o servicios de cada funcionalidad por separado y de manera algo más desacoplada.
- **Centralización:** Uno de los puntos fuertes de implementar una api-rest propia es el hecho de centralizar tu lógica de negocio en el backend. En este sentido nos encontraríamos con un escenario donde los clientes solo se dedicarían a pedir y proporcionar datos evitando así posibles incongruencias de comportamiento entre los diferentes sistemas.
- **Reducción de código en cliente:** Está relacionado con el punto anterior. El hecho de que nos proporcionen un backend no significa que haya menos código para el desarrollador. No obstante es un problema que se está subsanando con las funciones lambda como servicio.
- **Integración óptima con micro-servicios:** La integración con micro-servicios es óptima aunque, nuevamente, es algo que se puede resolver con las funciones de Firebase.

5.3 Solución: Firebase

Firebase es un servicio muy popular entre los desarrolladores móvil porque evita tener que invertir tiempo en la implementación y gestión de bases de datos, seguridad, api. etc. Además cuenta con una capa gratuita que te proporciona un margen de captación de tráfico con el cual ya puedes empezar a plantear modelos de negocio que te permitan cubrir los gastos y obtener beneficios.

5.4 Arquitectura

Desarrollo nativo

El desarrollo de los clientes será nativo tanto para Android como para iOS, aunque durante este trabajo final de máster solo se desarrollará una parte de la plataforma de iOS.

Se ha escogido la opción nativa frente a otras soluciones híbridas por una razón de performance y escalabilidad. Además, las actualizaciones de los diferentes sistemas operativos son cada vez más frecuentes y plataformas como Cordova no siempre pueden dar la misma experiencia que un sistema nativo.

Backend

Los servicios que se integrarán durante el transcurso del TFM serán:

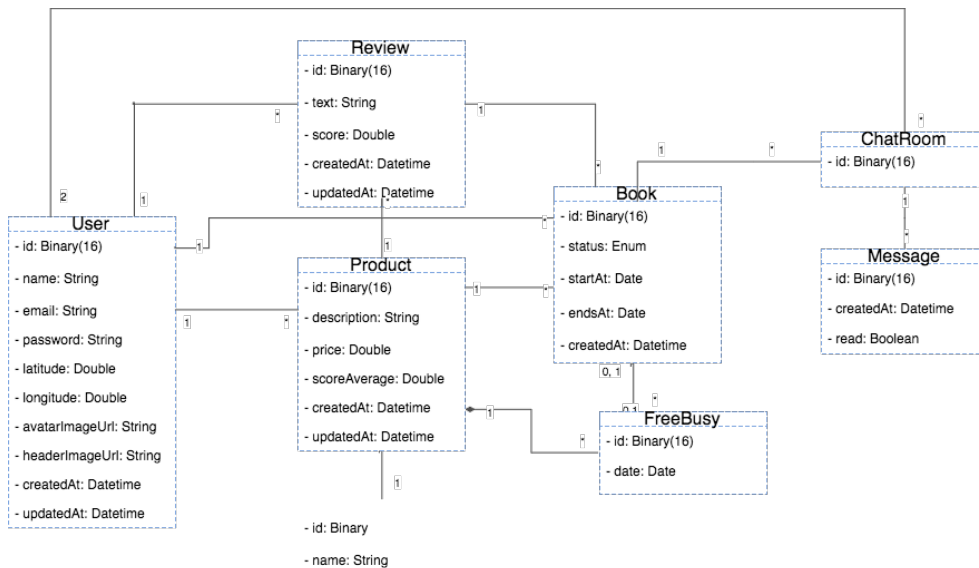
Database	Persistencia de datos
Storage	Para las imágenes de usuarios y productos
Functions	<ul style="list-style-type: none">- Funciones reactivas a los nodos relacionados con las reservas y mensajes de chat (de momento no se usará Socket.IO)- Gestión subida imágenes

Diseño de entidades

El diseño de entidades se ha definido en base a las siguientes restricciones:

- Un producto debe estar relacionado con un usuario registrado.
- Las reservas seguirán existiendo pese a una eventual baja de los usuarios.
- La disponibilidad de un producto depende de las reservas o de lo que haya definido el propietario en su ficha. Por ello se ha generado una

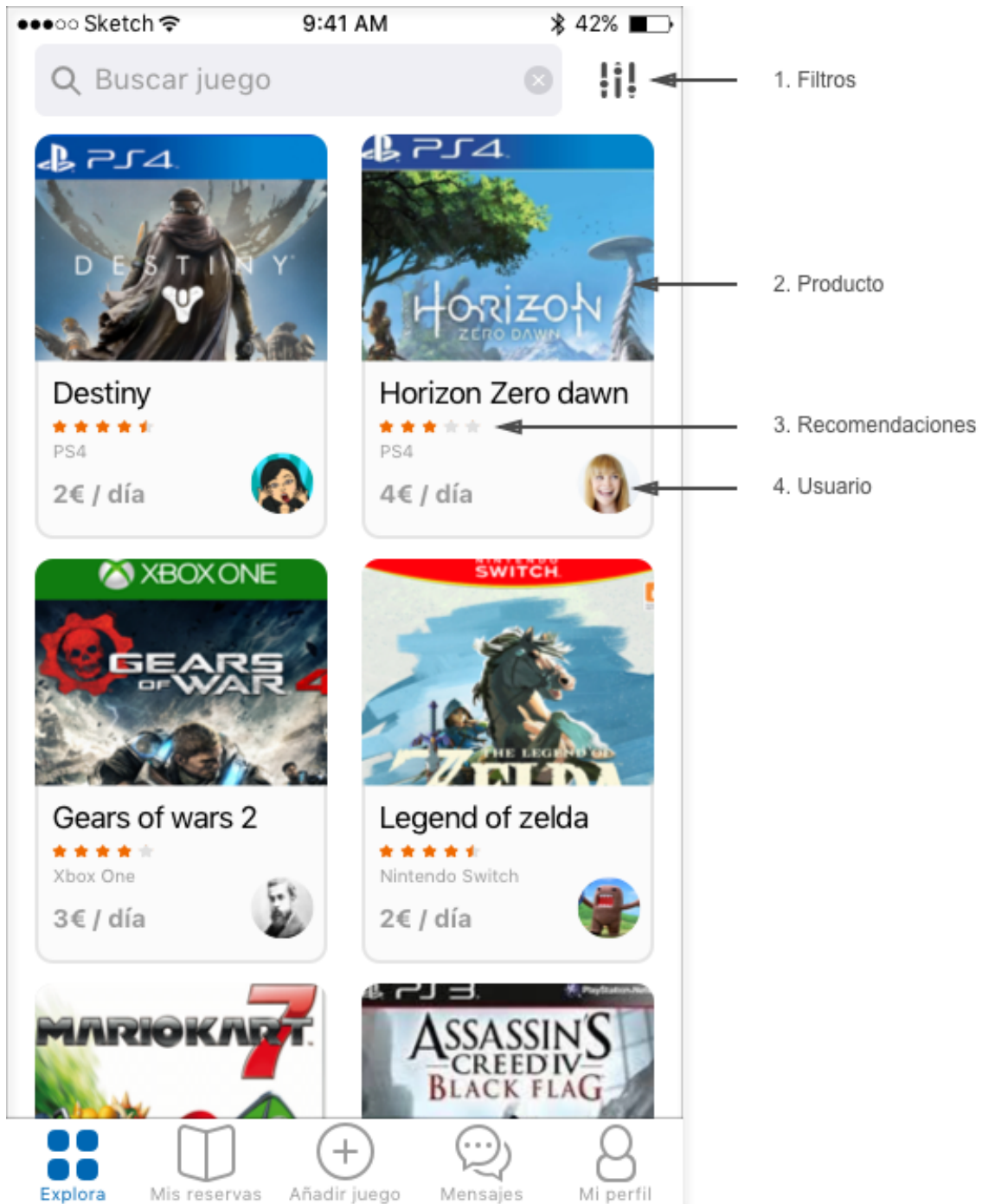
relación de composición entre la reserva y la disponibilidad, las cuales se generarán al confirmar la reserva.



Por supuesto esta arquitectura es relacional por lo que la implementación no es directa y tendrá que pasar por una desnormalización para adaptarla a nosql.

Modelado de datos

Como he dicho anteriormente, el diseño de entidades anterior está creado como un diagrama relacional pero esto no es posible hacerlo en Firebase y tampoco interesa aplicarlo. En este sentido se ha decidido, tal y como recomiendan en la misma documentación de Firebase, no abusar de las relaciones embebidas y optar mayoritariamente por documentos referenciados. Esto tiene una ventaja importante en cuanto al peso en la transferencia de datos pero es necesario que el cliente haga peticiones a medida que vaya necesítándolo. Un ejemplo:



Como puede apreciarse en la pantalla, cada uno de los elementos está formado por varios atributos de diferentes entidades. Si optamos por la estrategia de relacionarlas por referencia, nos encontramos con la necesidad de hacer una petición global de productos (2) para luego hacer dos consultas más por cada una de ellas (recomendaciones (3) y datos de usuario (4)), sumando un total 90 consultas a BBDD para un listado de 30 productos. Aunque esto pueda parecer

un lastre en cuanto a performance, el acceso directo a través de índices hace que no afecte negativamente en la velocidad de carga global.

6. Implementación (IOS)

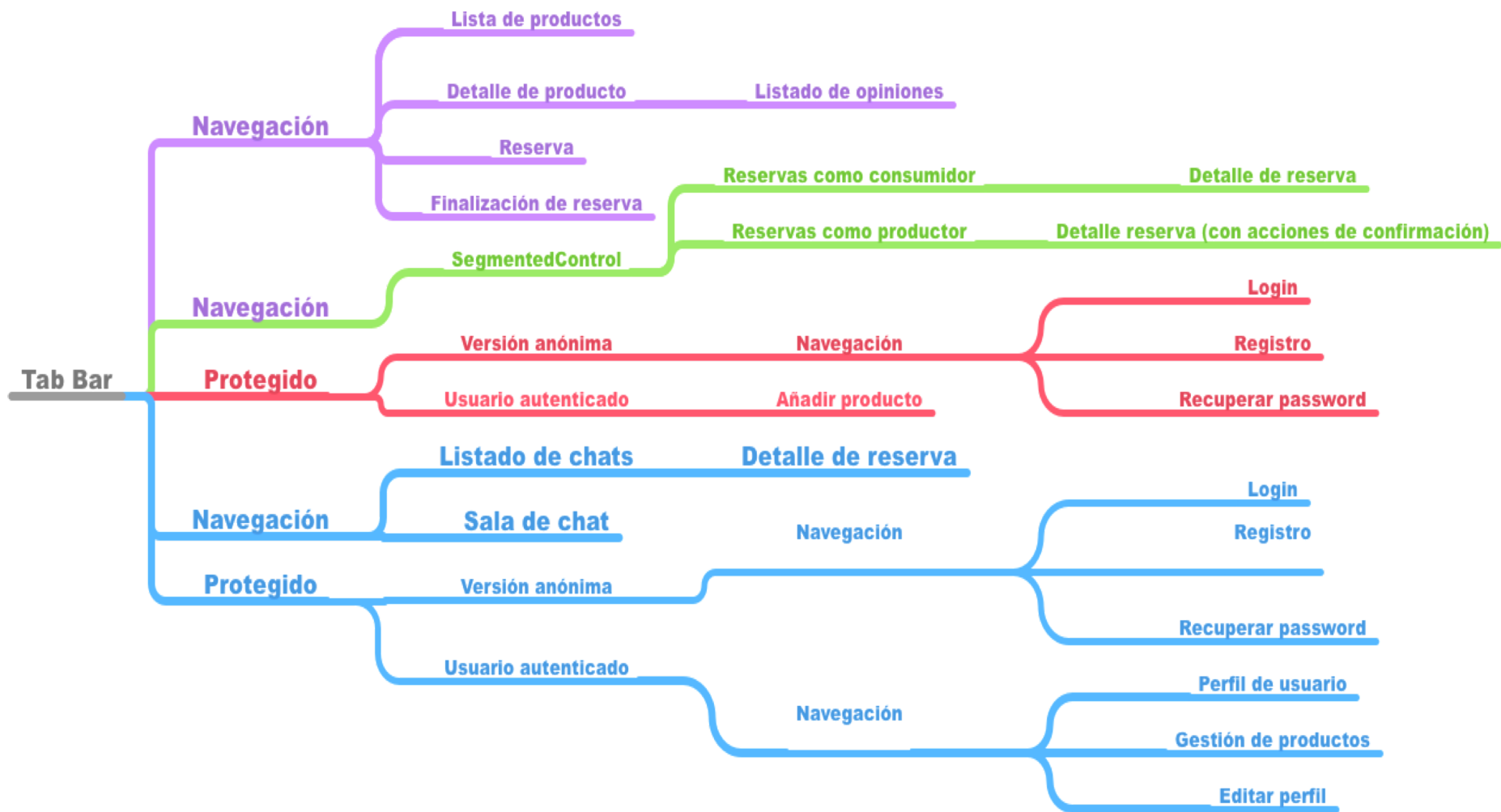
6.1 Antes de empezar...

El lenguaje por el que se ha optado ha sido Swift. La elección ha sido esta en lugar de Objective-C por ser un lenguaje mucho más moderno y “fácil” que Objective-C. Entrecomillo esto último porque esto es variable en dependencia a quien lo esté escribiendo.

Por otro lado tenía que decidir si optar por el desarrollo a través de storyboards o por el contrario olvidarme de estos y usar XIB independientes. Ambas son buenas opciones y mientras que con los storyboards es posible disponer de una visualización muy clara del ciclo de vida de los casos de uso, con los Xib se gana en escalabilidad y trabajo en equipo. En este caso podría haber optado por ambas soluciones aunque finalmente decidí no usarlos. La intención principal que hay tras esta elección es el igualar al máximo posible el código entre esta versión de iOS y la futura de Android.

6.2 Arquitectura de clases

Las clases se han separado en diferentes grupos: Controladores, vistas, extensiones... típicas de cualquier proyecto de este tipo. Hare un resumen general pero no entraré en detalle ya que entrego el código fuente junto a esta memoria.



Controladores

Aquí estarían todos los ViewControllers con sus respectivos xib. Cada uno de estos se ha dividido por tantas extensiones como delegados tenga.

Vistas

Los UITableViewCell, UICollectionViewCell, etc. se han añadido de manera que puedan ser reutilizables en el caso que se precisen. Por ejemplo, el explorador de productos y el listado de productos favoritos de un usuario comparten tipo de celda.

Network

Las conexiones a Firebase. Todas las llamadas a la BBDD están encapsuladas en métodos con callbacks, ya que estas son asíncronas.

Extensiones

Cualquier método extra que se haya añadido a las clases nativas de Swift. Por ejemplo, métodos de localización de cadenas de texto.

Recursos

Aquí van los ficheros de traducción, imágenes y otros recursos de la app.

6.3 Protección de vistas

En la app hay pantallas que solo pueden ser visualizadas por usuarios autenticados. Ejemplos son: Subir un producto, reservar o visualizar perfil. Además, estas deben identificar en todo momento cuando un usuario entre o sale para volver a ocultarlas o no.

Para solucionar esto se ha aprovechado la potencia de la librería nativa NotificationCenter y creado un wrapper con el que se elimina la complejidad por todo el código quedando de esta manera:

```
let vc = ProtectedViewController(ProfileViewController())
```

Este wrapper recibe las notificaciones de login y logout lanzadas por la clase UserNetwork mostrando así la pantalla que toque en todo momento.

6.4 Algo más que vistas: Backend

Cualquier acción que se lleve a cabo durante los diferentes casos de uso de la aplicación tiene una implicación a nivel de backend. Dada la naturaleza de las bases de datos noSql es necesario pasar por un proceso de desnormalización y para ello es necesario que los datos sean consistentes. Para ello se ha usado Firebase Functions, ya que está totalmente integrado en el sandbox ofrecido por Google. Algunas de las acciones que se realizan son:

- Insertar referencias en documentos relacionados. Por ejemplo, al publicar un producto se añade una referencia al usuario.
- Marcas en tiempo real. Se usa principalmente en las salas de chat para mostrar cuando está escribiendo un usuario. Esto sustituye la necesidad de websockets.
- Notificaciones. Hay funciones lambda que escuchan inserciones de reservas o mensajes de chat que se encargan de enviar una notificación push al otro usuario. Es algo que no se ha acabado de implementar durante este proyecto pero se ha preparado.

6.5 Librerías externas

Hay partes del Proyecto que contienen elementos bastante frecuentes en aplicaciones de este tipo y que se han solucionado con librerías de código libre. Aunque hay muchas en repositorios como github⁶, las librerías debían cumplir un mínimo de condiciones para ser implementadas en el proyecto:

- Su implementación debía estar justificada por un ahorro significativo de desarrollo y mantenimiento del código.

⁶ github.com es una plataforma para alojar proyectos con el CV git.

- La librería debía ser estable.
- El número de estrellas recibidas son un plus.
- Que el soporte ante issues sea correcto.
- Que las licencias de uso sean compatibles con la licencia de este proyecto.

Firestore:

Se han usado las librerías oficiales de Google Firebase para la interacción con el backend.

JTAppleCalendar:

Se ha implementado JTAppleCalendar para la gestión del calendario de reserva, una librería escrita totalmente en Swift y con un soporte excepcional ante problemas o dudas que se presentan en su repositorio.

JSQMessagesViewController:

Librería para la pantalla de chat. No hay mucho que decir al respecto más que las horas de desarrollo que me he ahorrado gracias a ella.

ObjectMapper:

Para el mapeo de los json llegados de Firebase se ha optado por ObjectMapper.

Kingfisher:

Kingfisher es una librería bastante conocida y usada para la carga de imágenes. Cuenta con carga asíncrona, procesadores de imágenes para recortar, redimensionar, etc.

ImageSlideShow:

ImageSlideShow es una librería que te página las imágenes por ti. La he usado en la ficha de producto y me ha ahorrado bastante tiempo de desarrollo.

SkyFloatingLabelTextField:

No es algo realmente necesario pero con SkyFloatingLabelTextField, librería desarrollada por el equipo de SkyScanner, he podido añadir algún que otro efecto.

Cosmos:

Una vista simple y práctica de puntuación mediante estrellas.

6.6 Testing

Debido al tiempo disponible para la realización del proyecto no ha habido una fase específica de test, aunque sí se ha hecho de manera continua durante todo el desarrollo. Las pruebas se han basado en las pruebas funcionales y visualización en diferentes tamaños de pantallas: iPhone5 y iPhone 7 plus.

7. Conclusiones**7.1 Seguimiento del proyecto**

Durante la planificación de este proyecto se definieron diferentes riesgos que efectivamente han ido apareciendo durante este poco tiempo.

Riesgo: [Poco tiempo](#)

Realidad: He intentado llevar un seguimiento de las horas invertidas por cada una de las funcionalidades con tal de redimensionar las funcionalidades siguientes pero la realidad ha sido que he acabado invirtiendo muchas más horas de las estimadas, perdiendo así un poco el control de las tareas.

Riesgo: [Estimación sin datos críticos](#)

Realidad: No se aleja demasiado de lo previsto. La estimación se realizó sobre un producto mínimo viable y eso ha facilitado la consecución del proyecto.

Riesgo: [Estimación de tiempo optimista](#)

Realidad: En este caso he podido invertir muchísimas más horas de las previstas, aunque esto es un indicador que con el tiempo "normal" que implica una asignatura de 12 créditos como es esta asignatura de TFM, no hubiera sido suficiente para acabar.

7.2 Lecciones aprendidas

Dada la naturaleza de esta asignatura no es posible realizar una planificación más detallada de la que he podido hacer. Esto ha derivado en una inversión de tiempo mayor durante el desarrollo ya que empecé a desarrollar justo al finalizar la fase de diseño y conceptualización por lo que ha faltado algo más de planificación técnica.

Por otro lado, encuentro dos maneras de enfocar el desarrollo. Por un lado podría optar por focalizar los esfuerzos en las funcionalidades y posponer la maquetación de las vistas para el final o trabajar en profundidad en cada una de las pantallas por las que se pasa. En esta ocasión me he decantado por esta última pero esto ha provocado "atascos" en detalles que podría haber dejado para el final. Claramente optaré por la primera para el próximo proyecto.

DCU me ha demostrado que es una de las mejores técnicas para enfocar un proyecto de este tipo. El diseño planteado en este proyecto partió con un mínimo de funcionalidades y, aunque este no ha evolucionado de momento, hay varios puntos que se han analizado y que detallaré a continuación.

7.3 Trabajo pendiente

Pese a que se han desarrollado las principales funcionalidades de la app se han quedado algunas por implementar:

- Gestión de productos, favoritos y datos del usuario
- Perfil público de usuario
- Añadir seguridad en el backend.
- Notificaciones
- La búsqueda y los filtros del listado no están implementados.

7.4 Líneas futuras

Branding

El diseño inicial no tiene identidad propia. Se trabajará próximamente en los colores usados, la manera en que presentamos la app, etc.

On Boarding:

La app abre directamente en el listado por lo que el usuario no recibe ningún tipo de guía o información sobre las posibilidades de la plataforma. Se añadirá una presentación paginada donde se mostrará de manera más amigables las opciones que tiene el usuario.

Conversión de usuarios:

La plataforma nace como una aplicación móvil por lo que la captación desde aquí es clave. En este sentido se ha dado un peso demasiado alto a la autenticación, cuando realmente ninguno de los testers que han probado la app disponía de una cuenta. Con esto se replanteará la manera en que el usuario se registra y autentica en la app, algo que seguramente aparecerá tras el on boarding.

Captación de productores:

Para que una plataforma de consumo colaborativo tenga éxito es crítico disponer de una masa de usuarios muy amplia. Para ello es necesario que los usuarios dispongan de un abanico amplio de productos por lo que hay que trabajar la parte orientada a los productores. Mostrar estadísticas, KPI y otra información útil para el productor podría ser buena idea. Por otro lado, el alquiler entre particulares cobra sentido para productos que tienen un precio relativamente alto por lo que hay que trabajar en como aportar las herramientas necesarias para que los productos se sientan seguros con los préstamos.

Desarrollo en Android:

Pretendo publicar la app en las stores y por el momento de manera localizada en España. Esto significa que mi principal público objetivo es Android, ya que tiene una cuota de mercado mucho mayor que Apple, por lo que será crítico desarrollar esta versión por delante de la de iOS.

5. Bibliografía

Ray Wenderlich (2017), <https://raywenderlich.com/>

Google (2017), "Firebase docs", <https://firebase.google.com/>

Apple (2017), "Swift Standard Library",
<https://developer.apple.com/reference/swift>

Apple (2017), "Apple resources", <https://developer.apple.com/swift/resources>

StackOverflow (2017), <https://www.stackoverflow.com/>

Maguire, M. (2001). "Context of use within usability activities". *International Journal of Human-Computer Studies*, 55, 453–483