

---

This is the **published version** of the bachelor thesis:

Castelo Novo, Adrián; Vergara Carreras, Enric, dir. QRHunt : una app de  
búsqueda del tesoro mediante QR. 2021. (958 Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/238429>

under the terms of the  license

# QR Hunt: Una app de búsqueda del tesoro mediante QR

Adrián Castelo Novo

## Resumen–

La llegada de los dispositivos móviles ha repercutido en la forma con la que interactuamos con el mundo real, abriendo un nuevo mundo para innovación. QRHunt es una aplicación móvil multiplataforma que pretende aprovechar el avance de las nuevas tecnologías y la globalización de los dispositivos móviles para revitalizar el tradicional juego de la búsqueda del tesoro, haciendo uso de tecnologías como los QR y aprovechando las funcionalidades incorporadas de estos dispositivos como las cámaras, micrófonos y sistema de GPS. Es una aplicación principalmente dirigida a un público joven, pero apta para cualquier grupo de edad.

**Palabras clave–** React Native, Firebase, Expo, Búsqueda del tesoro, JavaScript, QR, Multiplataforma

**Abstract–** The arrival of mobile devices has impacted the way we interact with the real world, opening up a new world for innovation. QRHunt is a cross-platform mobile application that aims to take advantage of the advancement of new technologies and the globalisation of mobile devices to revitalise the traditional treasure hunt game, making use of new technologies such as QR and taking advantage of the built-in functionalities of these devices such as cameras, microphones and GPS systems. It is an application mainly aimed at a young audience, but suitable for any age group.

**Keywords–** React Native, Firebase, Expo, Treasure Hunt, JavaScript, QR, Cross-platform



## 1 INTRODUCCIÓN

La búsqueda del tesoro es un juego en el que los participantes tratan de encontrar varios objetos escondidos en diferentes lugares, siguiendo una serie de pistas o acertijos propuestos por los organizadores. Se suele jugar en las escuelas, fiestas, actividades de empresas, actividades recreativas o también en casa. Es un juego muy popular que les encanta a los niños de todas las edades e incluso a gente más adulta, ya que además de ser un juego divertido, también es útil como evento de formación de equipos, actividad de orientación, o herramienta de participación profesional. Normalmente, los participantes trabajan en equipos pequeños. Esto ayuda a desarrollar habilidades importantes como el trabajo en equipo, la toma de decisiones, la organización, la comunicación y la creatividad. En general, es un juego muy completo que alterna tanto actividades físicas como mentales.

- E-mail de contacte: [adriancastelo@gmail.com](mailto:adriancastelo@gmail.com)
- Menció realitzada: Tecnologies de la Informació
- Treball tutoritzat per: Enric Vergara
- Curs 2020/21

Tradicionalmente, los acertijos de la búsqueda del tesoro se escribían en papel y los organizadores debían estar atentos para monitorizar el progreso de los jugadores. Sin embargo, gracias al avance de la tecnología, este proceso se puede mejorar y simplificar con el uso de soluciones digitales.

La finalidad de este proyecto es la de crear una aplicación móvil que permita al organizador crear partidas de búsqueda del tesoro con una lista de pistas variadas como imágenes, descripciones, archivos de audio o coordenadas GPS y que otras personas puedan participar en ellas. Tras crear una partida, el creador recibirá una lista de códigos QR que deberá imprimir y colocar en diferentes lugares para que puedan ser encontrados por los participantes. Los participantes podrán utilizar la App para ver las pistas y averiguar donde están escondidos los códigos QR. Desde la propia aplicación podrán escanear los QR correspondientes a cada pista para resolverlas, además de poder visualizar su progreso en la partida.

Esta es la forma más sencilla de simplificar el proceso tanto para los organizadores como para sus participantes. Una App ofrece todos los beneficios de la automatización y garantiza que todos estén informados con actualizaciones

en tiempo real. Como beneficio adicional, las funcionalidades incorporadas de los smartphone permiten aumentar la variedad de actividades sin requerir equipo adicional.

## 2 OBJETIVOS

En esta sección se definen los objetivos de la aplicación, según su importancia.

### 2.1 Objetivos principales

El objetivo principal de este proyecto es el de desarrollar e implementar un aplicación móvil multiplataforma basado en la búsqueda del tesoro. Para ello, se definen una serie de objetivos a cumplir:

- Hacer un diseño preliminar de las pantallas de la aplicación y un mock-up de estas.
- La aplicación está orientada principalmente a niños, con lo cual debe tener un diseño amigable e intuitivo.
- Dado que la aplicación será utilizada por diferentes personas, es importante que la aplicación sea cross-platform tanto para iOS como Android, de manera que cualquier persona pueda jugar independientemente del dispositivo móvil que tenga.
- Diseñar una base de datos en la que almacenar las partidas creadas, para que puedan ser compartidas entre diferentes dispositivos.
- Debe haber un menú principal con opciones para crear partidas y jugar.
- Permitir la creación de partidas con, como mínimo, 2 tipos de pistas y la posibilidad de compartirlas mediante un código.
- La aplicación debe generar códigos QR al crear la partida y poder escanearlos al jugar como jugador.
- La aplicación debe poder importar imágenes desde el carrete del smartphone o permitir sacar una foto directamente.
- La aplicación debe mostrar el progreso del jugador, indicando el número de pistas resueltas y las restantes por resolver.
- El jugador tendrá la opción de jugar partidas creadas por él mismo o introducir un código para jugar partidas creadas por otros usuarios.
- Desarrollar la modalidad de partida por defecto: "Historia". En la que el desarrollo de la partida sigue un orden definido: se empieza por la primera pista y no se podrá avanzar a la siguiente pista hasta que esta sea resuelta.

### 2.2 Objetivos secundarios

- Autenticación de los usuarios mediante Google.
- Añadir la modalidad de partida: "Desafío". En la que las pistas se muestran en una lista para poder ser resueltas en cualquier orden.

- Permitir que el creador de una partida pueda ver un ranking de los participantes y el tiempo que han tardado en completar cada partida.
- Añadir nuevos tipos de pistas.

## 3 ESTADO DEL ARTE

La búsqueda del tesoro siempre ha sido un juego bastante popular, y en esta era de digitalización, el uso de códigos QR para organizar las búsquedas se ha convertido en una tendencia ascendente. Si buscamos en la red podemos encontrar varias webs relacionadas con el tema, sin embargo, estas suelen consistir en guías o tutoriales que no proporcionan herramientas propias, ni una plataforma unificada: el organizador debe redactar las pistas mediante un procesador de texto, después utilizar una web externa para generar códigos QR e insertarlas en un documento para imprimirlas. Mientras que los jugadores necesitarán descargar una aplicación de un tercero para poder escanear estos códigos.

Con una investigación más exhaustiva, podemos encontrar 2 empresas que ofrecen un servicio de temática similar con apps propias: Scavify [1] y ActionBound [2]. Sus juegos consisten en una lista de desafíos varios con un sistema de puntos. Sin embargo, todas ellas están orientadas a organizaciones con muchos participantes (empresas, eventos, universidades...) y requieren de un pago por cada juego creado.

En las tiendas de aplicaciones también podemos encontrar apps con ideas similares, aunque con un funcionamiento bastante diferente:

- Geocaching [3]: Hay "tesoros" son escondidos por diferentes personas y se muestran en un mapa con GPS, no hay pistas ni desafíos.
- SnapHunt [4] y Camera Hunt [5]: Muestran nombres de objetos. El jugador debe apuntar con la cámara a dichos objetos para ser validados por un algoritmo de reconocimiento.
- Treasure Hunt [6]: Esta sí que es una app con escáner QR, pero tiene un diseño pobre y muy pocas funcionalidades. Solo permite crear juegos de 5 o 10 pistas, y todas ellas solamente admiten una descripción de texto.

En definitiva, aunque existen aplicaciones de temática similar, ninguna de ellas proporciona las funcionalidades de la aplicación propuesta. La más parecida es Treasure Hunt, pero es muy mejorable y apenas ofrece variedad en la creación de juegos. Scavify y ActionBound proporcionan más variedad, pero posee un funcionamiento muy diferente y no está enfocada para niños. El funcionamiento de estas últimas se asemejan más a una gincana de pruebas que a una búsqueda del tesoro.

## 4 METODOLOGÍA

Para la planificación del trabajo se ha optado por utilizar la metodología ágil Kanban. La decisión de escoger este método ágil viene dada por la necesidad de poder gestionar el proceso de realizar el proyecto en solitario.

A diferencia de otras metodologías como Scrum o DevOps, Kanban no se centra tanto en el trabajo en equipo ni requiere de diferentes roles específicos, lo cual lo convierte en una buena opción para un proyecto unipersonal.

Esta metodología permite visualizar de manera clara el trabajo a realizar y en qué estado se encuentra el proyecto haciendo uso de un tablero de tareas organizado por columnas. Las tareas tienen una duración estimada y se organizan en el tablero por estados según si se han hecho, se están haciendo y quedan por hacer. Kanban limita la asignación de estas tareas para aumentar la productividad, de manera que no se deben comenzar nuevas tareas hasta que se acabe alguna otra [7].

Para aplicar esta metodología, se utilizará la herramienta Asana [8], una aplicación basada en el método Kanban que sirve para gestionar tareas permitiendo organizar el trabajo en diferentes estados. Las tareas se organizarán en los estados: Tareas pendientes, En curso, Listo y Bug.

## 5 HERRAMIENTAS Y TECNOLOGÍAS

Para el desarrollo del proyecto se ha hecho una investigación de las posibles tecnologías y herramientas necesarias para llevarlo a cabo. A continuación se listan las que fueron finalmente utilizadas:

### 5.1 Herramientas

#### Asana

Como se ha comentado anteriormente, para llevar a cabo la metodología Kanban se ha decidido utilizar la herramienta de gestión de tareas Asana. La herramienta posee varias funcionalidades, como espacios de trabajo, etiquetas, notas o comentarios que facilitan el desarrollo tanto de proyectos en equipo como unipersonales. Es una herramienta similar a la popular Trello en cuanto a simplicidad de uso, pero posee una interfaz más limpia y amigable, además de un enfoque más orientado al desarrollo de metodologías ágiles.

#### Github

Para el mantenimiento del código y control de versiones se ha utilizado GitHub [9]. Se irán subiendo actualizaciones del código al repositorio a medida se desarrollen las fases del proyecto para tener un historial de las modificaciones y diferentes versiones del código. Esto permitirá hacer un rollback en caso de que se produjera algún problema crítico durante el desarrollo.

#### Visual Studio Code

Visual Studio Code ha sido el editor de código utilizado para la programación de la aplicación. Es un editor ligero y de código abierto creado por Microsoft, con una gran cantidad de plugins que facilitan la codificación.

#### Proto.io

El diseño y mock-up de la aplicación se ha realizado con Proto.io [10], una plataforma para la creación de prototipos interactivos y animados sin código. Es una herramienta muy intuitiva y ofrece un conjunto de componentes ya creados que aceleran la creación del diseño.

## 5.2 Tecnologías

A la hora de desarrollar una aplicación móvil, existen principalmente 3 tipos:

- **Nativa:** Las aplicaciones nativas se desarrollan en el lenguaje nativo de un sistema operativo específico, lo cual ofrece más posibilidades pero dificulta el desarrollo multiplataforma y aumenta su coste.
- **Web:** Estas consisten en una web embebida en una aplicación que funciona como un navegador, con lo cual se puede utilizar en cualquier sistema operativo pero ofrece un rendimiento pobre.
- **Híbrida:** Este tipo de aplicación combina las 2 anteriores, se desarrollan con lenguajes de programación típicos de una web pero haciendo uso de componentes nativos, facilitando el desarrollo multiplataforma y ofreciendo un excelente rendimiento a un coste medio.

Dado que la aplicación a desarrollar necesita ser funcionar tanto en iOS como Android y necesitaremos hacer uso de funcionalidades internas del dispositivo móvil como la cámara, los altavoces o el micrófono, la aplicación híbrida se presenta como una opción ideal.

Tras una investigación de los posibles entornos de desarrollo para plataformas híbridas se ha decidido optar por React Native junto con Expo [11] para el desarrollo del front-end, esta es una tecnología innovadora y utilizada en grandes aplicaciones como Facebook o Instagram.

Mientras que para el back-end se utilizará Firebase, una herramienta de Google que proporciona varios servicios como autenticación de usuario o bases de datos en tiempo real [12].

#### React Native

React Native es un framework de desarrollo de aplicaciones nativas multiplataforma basado en JavaScript y React.js. Este nos permite crear aplicaciones nativas de Android e iOS utilizando solamente JavaScript, mientras que React Native se encarga de convertir los elementos a código nativo, ofreciendo un buen rendimiento y optimizando costes.

React Native está basado en componentes, lo que permite reutilizar código y tener un proyecto mejor organizado. Además tiene una sintaxis sencilla de aprender e intuitiva, lo que lo convierte en un framework fácil de aprender incluso para gente con poca experiencia.

#### Expo

Además de React Native, también se ha decidido acompañar el desarrollo del front-end utilizando Expo, un conjunto de librerías y herramientas que facilitan el desarrollo en React Native y proporciona varios módulos para evitar tener que desarrollar código independiente de iOS y Android.

Expo utiliza Expo SDK, el cual es que provee acceso a funcionalidades nativas del dispositivo como a la cámara, contactos o almacenamiento local sin modificar código nativo. También proporciona un cliente para probar la aplicación fácilmente a medida que se desarrolla.

## Firestore

Firestore es una plataforma en la nube, que nos ofrece varios servicios de back-end para el desarrollo de aplicaciones móviles o web. Nos permite crear una base de datos, sincronizar datos fácilmente, generar analíticas o así como otras varias funciones. En concreto, se han utilizado los siguientes servicios para el desarrollo del proyecto:

- **Firestore authentication:** Nos permite autenticar a los usuarios de la aplicación móvil. Ofrece métodos para registrarse e iniciar sesión mediante varios servicios (Google, Twitter, Facebook, números de teléfono, etc). En nuestro caso hemos decidido hacer uso de la autenticación con Google al ser uno de los servicios más populares.
- **Firestore Cloud Firestore:** Es una base de datos flexible y escalable, mantiene los datos sincronizados entre la app y el servidor de Firestore.
- **Firestore Storage:** Permite almacenar archivos de gran tamaño que no pueden ser almacenados en la base de datos, como imágenes o audios. Posteriormente estos archivos pueden ser accedidos mediante un enlace generado por el propio Firestore.

Firestore ofrece un generoso plan gratuito para el uso de estos servicios, el cual cubre las necesidades para desarrollo este proyecto. Aunque, en caso de que fuese necesario, se podría mejorar a un plan de pago fácilmente.

## 6 PLANIFICACIÓN

La planificación de este proyecto se divide principalmente en cuatro fases:

- **Análisis del proyecto e identificación de requisitos:** En esta primera fase se estudia el proyecto a desarrollar, definiendo el alcance del proyecto y proponiendo unos objetivos y requisitos iniciales. Además de un estudio de las posibles tecnologías y herramientas que se utilizarán.
- **Diseño inicial:** Se empieza a realizar un diseño preliminar de la aplicación, siguiendo los objetivos y requisitos propuestos en la fase anterior. Esta fase incluye la creación de diagramas de casos de uso, diagramas de flujo, un diseño de la base de datos y un prototipo visual de la aplicación.
- **Desarrollo:** Es la fase con más volumen del proyecto, en la que se empieza a programar la aplicación, siguiendo los objetivos y diseños creados e implementando las funcionalidades necesarias.
- **Documentación:** Esta última fase se centraría en finalizar la documentación final del proyecto y preparar la defensa de éste.

## 7 REQUISITOS

A partir de los objetivos se derivan una serie de requisitos a cumplir, tanto funcionales como no funcionales, que definen las restricciones que el sistema debe satisfacer. A continuación se detallan los diferentes requisitos establecidos:

### 7.1 Requisitos funcionales

- RQF1. El usuario debe iniciar sesión con una cuenta de Google para utilizar las funciones de la aplicación.
- RQF1. Si es la primera vez que el usuario inicia sesión, el sistema deberá registrar sus datos en la base de datos.
- RQF2. La aplicación mantiene el inicio de sesión, hasta que el usuario cierre sesión o borre los datos de la aplicación.
- RQF3. La aplicación ha de proveer al usuario la opción de crear partidas.
- RQF4. El usuario ha de poder visualizar las partidas que ha creado.
- RQF5. El usuario creador de una partida ha de poder descargar un PDF con los códigos QR de cada pista.
- RQF6. El usuario creador de una partida debe tener la opción de crear una partida pública.
- RQF7. Cuando el usuario crea una partida pública, la aplicación debe generar un código para poder compartirlo con otros jugadores.
- RQF8. El usuario ha de poder acceder a la cámara del terminal para poder escanear realizar fotos o escanear códigos QR.
- RQF9. El usuario ha de poder importar imágenes desde el carrete del dispositivo.
- RQF10. El usuario debe tener la opción de jugar partidas que ha creado.
- RQF11. El usuario ha de tener la opción de introducir un código para jugar partidas públicas de otros jugadores.
- RQF12. El usuario ha de poder visualizar sus datos personales.
- RQF13. La aplicación debe proveer al usuario de una opción para cerrar la sesión.
- RQF14. Cuando un usuario jugador finaliza una partida online, la aplicación debe almacenar su puntuación en la base de datos.
- RQF15. Durante la creación de una partida, el usuario debe poder añadir diferentes tipos de pista.
- RQF16. Durante la creación de una partida, el usuario tener las opción de editar o borrar pistas ya creadas.
- RQF17. El usuario jugador debe poder visualizar el número de pistas totales y su progreso durante una partida.
- RQF18. El usuario creador debe poder hacer uso del micrófono del dispositivo para grabar pistas de audio.
- RQF19. El usuario jugador ha de poder reproducir pistas de audio.

## 7.2 Requisitos no funcionales

- RQNF1. La aplicación debe tener una apariencia visual amigable e intuitiva.
- RQNF2. La aplicación debe ser funcional en todo dispositivos Android e iOS.
- RQNF3. La aplicación requerirá permisos de acceso a la cámara, micrófono y archivos internos del dispositivo.
- RQNF4. El dispositivo debe poseer conexión a internet.

## 8 DISEÑO

En esta sección se define el diseño general de la aplicación, tanto de su interfaz como funcionamiento. Para ello se han realizado diagramas de caso de uso y de flujo, además de un diseño de la base de datos y de la interfaz.

### 8.1 Casos de uso

A pesar de que todos los usuarios de la aplicación tienen acceso a las mismas funcionalidades, podemos diferenciarlos en 2 roles según el uso que le quieran dar a la aplicación:

**Creador de juegos:** Representa al organizador del juego, que pretende utilizar la aplicación para crear partidas para que otros puedan jugarlas.

**Jugador:** Representa al usuario que pretende jugar partidas ya creadas.

En el diagrama de casos de uso se muestran los diferentes roles que puede tomar un usuario de la aplicación y las acciones que pueden realizar cada uno.

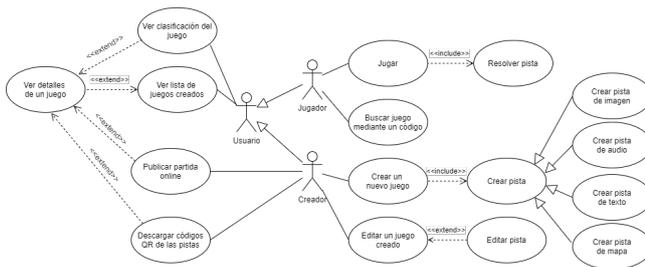


Fig. 1: Diagrama de casos de uso

### 8.2 Diseño de la base de datos

La base de datos del sistema es de tipo NoSQL y está implementado sobre Firebase Firestore. En Firestore, los datos se almacenan en documentos que contienen varios campos. A su vez, estos documentos se almacenan en colecciones, que son contenedores para los documentos. Estas colecciones también pueden contener otras subcolecciones con más documentos, lo cual permite una organización jerárquica de los datos.

Para la estructura de este proyecto, se han definido principalmente las siguientes colecciones:

- **Usuarios:** Esta colección contiene un documento por cada usuario registrado, con varios de sus datos personales como el nombre, correo electrónico y foto de perfil.
- **Juegos Creados:** Se trata de una subcolección que tiene cada usuario, cada documento contiene los datos de una partida creada, junto con todas sus pistas.
- **Partidas públicas:** Colección en la que se almacenan partidas públicas. El ID de cada documento está representado por un código alfanumérico de 5 cifras, para que puedan ser compartidas fácilmente con otros usuarios.
- **Puntuaciones Usuarios:** Aquí se almacenan los usuarios que han participado en una partida pública, junto con el tiempo que han tardado en completarla.

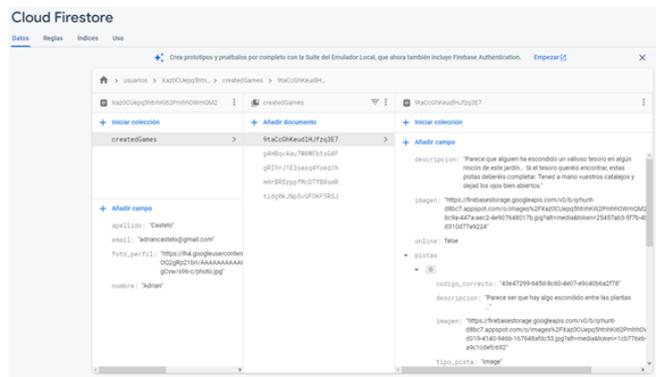


Fig. 2: Ejemplo de una colección de Firestore

### 8.3 Diseño del prototipo

Para realizar el prototipaje y un primer diseño de la aplicación se ha utilizado la herramienta web “proto.io” [10]. El principal objetivo de este prototipo es el de definir las diferentes pantallas que tendrá la aplicación y el flujo de uso entre estas. El diseño de la aplicación se ha adaptado en todo momento a los cambios de requisitos, con lo cual el diseño inicial diverge en algunos aspectos respecto a la aplicación final.

Aunque el público principal al que va dirigido sean niños, la búsqueda del tesoro es un juego para todas las edades. El diseño debe ser amigable para niños pero también apto para todo el público.

Para ello, se han sustituido las fuentes de letra por defecto por otras de un estilo “rounded” más amigable y se han creado botones que simulan un efecto 3D con una animación de pulsación correspondiente. También se han añadido algunas animaciones, como por ejemplo una pequeña explosión de confeti cuando un jugador termine una partida.

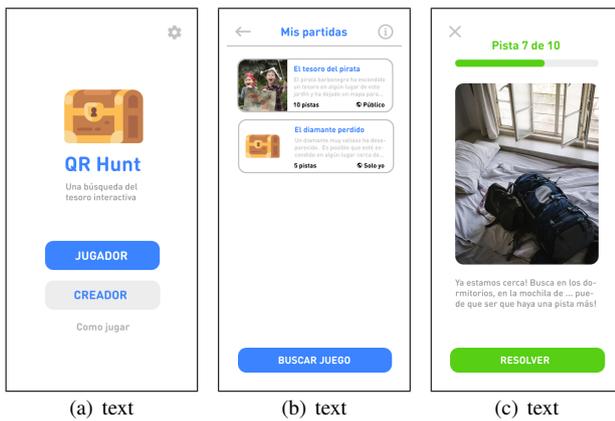


Fig. 3: Prototipo inicial realizado en Proto.io



Fig. 4: Componente del botón antes de ser pulsado



Fig. 5: Componente del botón tras ser pulsado

## 9 DESARROLLO

Como se ha mencionado anteriormente, el front-end será desarrollado con React Native y Expo, mientras que para el back-end se utilizarán los servicios de Firebase.

Las librerías de React Native y Expo proporcionan una gran cantidad de componentes y utilidades que son de gran utilidad para el desarrollo del proyecto, sin embargo, ninguna de ellas proporciona un sistema de navegación entre pantallas.

Para solucionar este problema se ha utilizado la librería React Navigation [13], la cual proporciona una navegación mediante un Stack que nos permite explorar las pantallas de la aplicación de manera fluida y eficiente. La librería se encarga de manejar el historial de pantallas y proporciona diferentes tipos de transición entre estas.

A continuación se procederá a explicar el desarrollo y funcionamiento de la aplicación a nivel interno.

### 9.1 Autenticación

Dado que existirá una opción de crear partidas online, será necesario poder identificar a los diferentes jugadores mediante nombres o algún identificador. Para ello, haremos uso de la herramienta Authentication de Firebase, la cual ofrece una gran variedad de métodos de inicio de sesión. Concretamente se ha implementado el inicio de sesión a través de Google, ya que es uno de los servicios más populares y disponible en varias plataformas.

Para implementarlo en el front-end, expo proporciona el paquete expo-google-app-auth que proporciona un método de autenticación tanto en iOS como Android [14]. Cuando un usuario abre la aplicación, se hace una llamada a la base de datos de Firebase para comprobar si existe una sesión activa. Si existe, se le redirige a la pantalla del menú princi-

pal. En caso contrario se le redirige a la pantalla de inicio de sesión. La sesión iniciada queda registrada en Firebase Authentication y, si es la primera vez que inicia sesión, también se envían los datos de usuario a Firestore para almacenarlos.

De esta manera se evita que el usuario deba iniciar sesión cada vez que abre la aplicación, ya que la sesión activa queda almacenada.

Una vez el usuario ha iniciado sesión, tiene la opción de visualizar sus partidas creadas, crear partidas nuevas o buscar partidas públicas mediante un código.

### 9.2 Comunicación entre el Front-end y Back-end

Cuando el usuario decide crear una partida, puede introducir un título, una descripción y una imagen de portada, además de crear el número de pistas que desee. Estos datos se van almacenando en un JSON de manera local. Una vez decide guardar la partida, la aplicación se prepara para enviar los datos a la base de datos de Firestore y que los almacene.

Sin embargo, dado que Firestore no es capaz de almacenar archivos de audio e imágenes de las pistas, es necesario almacenar estas previamente en Firebase Storage. Las imágenes y audios son convertidos a formato BLOB para que puedan ser enviados a Firebase Storage, éste las almacena en la nube y nos devuelve una URL para acceder al archivo. Una vez se han obtenido todas las URL, se añaden los enlaces al JSON con los datos de la partida y finalmente se envían a Firestore para que pueda almacenar la partida. Esto permite que el usuario pueda recuperar sus datos aunque desinstale la aplicación o se cambie de dispositivo.

En cuanto a la visualización de las partidas creadas, se utiliza la función *onSnapshot* que permite recuperar la lista de partidas creadas de la base de datos de Firestore. La ventaja de esta función es que recupera los datos una única vez, y solo se vuelve a comunicar con la base de datos cuando detecta que se ha modificado algún documento relevante. Esto también se ha combinado con la opción de persistencia offline de los datos, lo cual permite que las partidas puedan ser almacenadas en la caché, de manera que el usuario pueda seguir accediendo a ellas a pesar de no tener conexión a Internet.

Si el usuario selecciona una partida para ver sus detalles, se le presenta la opción de publicar su partida para el resto de jugadores o descargar la lista de códigos QR de las pistas. Si publica una partida online, se genera un código alfanumérico de 5 dígitos que se sube a la base de datos. El creador puede compartir ese código con otros jugadores para que lo introduzcan en su aplicación y puedan jugar dicha partida.

Cuando un jugador empieza una partida se guarda la hora inicial, y una vez finaliza se guarda la hora actual para obtener el tiempo que ha tardado en completar la partida. Estos datos se almacenan en la base de datos, que serán utilizados para mostrar una tabla de clasificación de los participantes.

### 9.3 Generación de PDF con lista de QR

Como se ha mencionado anteriormente, cada partida posee un número de pistas y para resolverlas se debe escanear su código QR correspondiente. Para ello es necesario generar

un archivo PDF con los códigos QR de cada pista, para que el creador de la partida pueda imprimirlos y colocarlos en los lugares que correspondan.

Para lograrlo, se han utilizado las librerías `react-native-qrcode-svg` para generar códigos QR, `expo-print` para generar un archivo en formato PDF y `expo-sharing` para abrir el documento con varias apps, compartirlo con otra persona o imprimirlo directamente. Cuando el usuario hace click en el botón de “Descargar PDF”, se llama a un componente que renderizan varios componentes `QRCode` para cada pista en HTML. Para poder generar el PDF utilizamos la función `printToFileAsync()` de `expo-print` que transforma código HTML a PDF. Sin embargo, el componente `QRCode` no es HTML real, con lo cual no podemos utilizarlo directamente. Para solucionar esto, guardamos una referencia de cada componente `QRCode` y lo transformamos a una imagen en formato BASE64.

De esta manera podemos generar un código de HTML puro y transformarlo a PDF. Finalmente se utiliza la función `Sharing.shareAsync(pdf)` para poder compartirlo. En la figura 6 podemos ver un ejemplo de un PDF generado.



Fig. 6: PDF con lista de códigos QR

## 10 TESTING

Durante la fase final del desarrollo, se han ido realizando una serie de testeos para comprobar el correcto funcionamiento de las funcionalidades añadidas y buscar posibles problemas o fallos en el comportamiento de la aplicación.

Se han realizado principalmente 2 tipos de pruebas:

### 10.1 Pruebas de caja negra

Se han realizado pruebas de caja negra en diferentes componentes de la aplicación, como la creación de partidas y pistas, comprobando las entradas y salidas de datos para intentar descubrir fallos o irregularidades. Para ello se ha hecho uso de la herramienta Jest, un framework JavaScript creado específicamente para el testeo de aplicaciones, que se caracteriza por su rapidez.

### 10.2 Exploratory testing

También se ha realizado exploratory testing durante el transcurso del proyecto, comprobando diferentes ejecuciones en la aplicación

En este tipo de pruebas no se necesita una planificación tan detallada y funciona mejor cuando quien hace el testeo ya tiene un conocimiento previo del funcionamiento de la aplicación.

Gracias a este tipo de pruebas se han podido encontrar algunos errores visuales y funcionales en poco tiempo.

## 11 RESULTADOS

La idea principal del proyecto era la de conseguir una aplicación funcional en la que se pudiesen crear y jugar juegos de búsqueda del tesoro haciendo uso de la tecnología QR. Para ello se definieron una serie de objetivos principales y secundarios. En el caso de los objetivos principales, se han conseguido todos los propuestos:

La aplicación es cross-platform y permite crear partidas con hasta 4 tipos diferentes de pistas, incluyendo la subida de imágenes, grabación/reproducción de audio y el uso de mapas. Estas pueden ser editadas y se sincronizan con la base de datos de Firebase para que puedan ser compartidas entre diferentes dispositivos.

Se ha conseguido la generación de PDFs con los códigos QR listos para imprimir.

La funcionalidad de jugar partidas también es funcional, mostrando el progreso de la partida y los 4 tipos de pistas. Además de la función “online” para compartir una partida creada con otros usuarios para que la puedan jugar.

Finalmente también se decidió implementar la autenticación de usuario mediante una cuenta de Google. La principal ventaja de esto es que asegura que las partidas creadas por los usuarios no se pierdan al borrar la aplicación o cambiar de dispositivo.

A continuación se muestran los resultados obtenidos en la aplicación:

### Inicio de sesión y menú principal

El usuario puede iniciar sesión con Google para acceder al menú principal, que ofrece opciones para jugar o crear partidas. Desde ahí también puede pulsar el icono de la esquina superior derecha para acceder a la pantalla de configuración, dónde podrá visualizar su foto de perfil, nombre, apellidos y correo. Además de tener la opción de cerrar sesión.

### Creación de partidas

Para crear una partida es obligatorio introducir un título, una descripción sobre el juego y añadir como mínimo 1 pista. De manera opcional, también se permite añadir una imagen de portada. En caso de que el usuario no añada una imagen, se mostrará la foto de un cofre como placeholder tras crear la partida.

También ofrece la opción de crear partidas en modalidad “Historia” o “Desafío”. La primera consiste es una aventura lineal, en la que las pistas deben ser resueltas en el orden en que se crearon, mientras que en la segunda se pueden ver todos a la vez y resolver en cualquier orden.

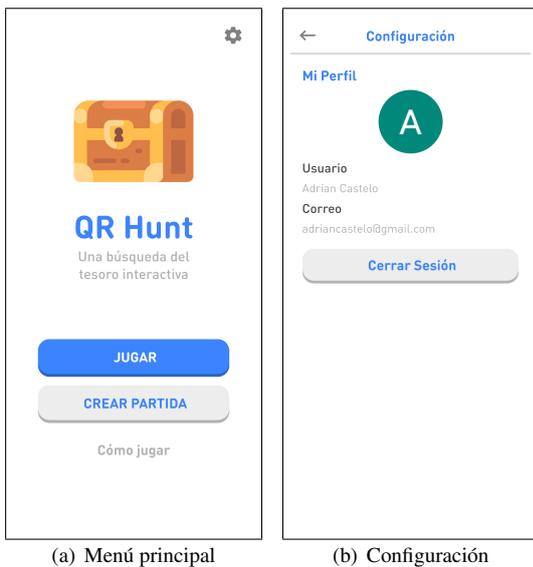


Fig. 7: Pantallas de menú principal y configuración

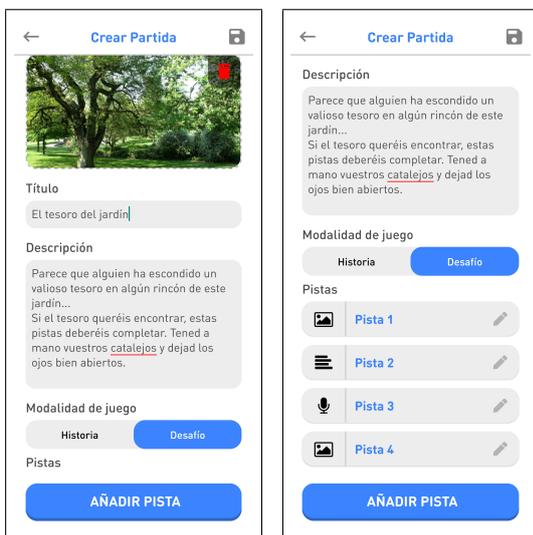


Fig. 8: Pantalla de crear partida

### Creación de pistas

Si el usuario presiona el botón “añadir pista”, se le mostrará un modal con 3 opciones a escoger: Pista de imagen: Muestra una pantalla con un campo para subir una imagen y otro para introducir una descripción opcional. Pista de texto: Muestra una pantalla con un campo para introducir una descripción. Pista de audio: Muestra un cronómetro para contar el tiempo y un botón para empezar a grabar, además de un campo para introducir una descripción opcional. Cuando el usuario pulsa el botón, se activa el micrófono del dispositivo y se empieza a grabar el audio. Si el usuario vuelve a pulsar el botón se para la grabación, se almacena localmente y se muestra un pequeño reproductor para que el usuario pueda verificar el contenido del audio.

- **Pista de imagen:** Muestra una pantalla con un campo para subir una imagen y otro para introducir una descripción opcional.
- **Pista de texto:** Muestra una pantalla con un campo para introducir una descripción.

- **Pista de audio:** Muestra un cronómetro para contar el tiempo y un botón para empezar a grabar, además de un campo para introducir una descripción opcional. Cuando el usuario pulsa el botón, se activa el micrófono del dispositivo y se empieza a grabar el audio. Si el usuario vuelve a pulsar el botón se para la grabación, se almacena localmente y se muestra un pequeño reproductor para que el usuario pueda verificar el contenido del audio.
- **Pista de mapa:** Muestra un mapa y una descripción opcional. También se ofrece un botón para obtener la localización actual y que pueda ser usada como pista.

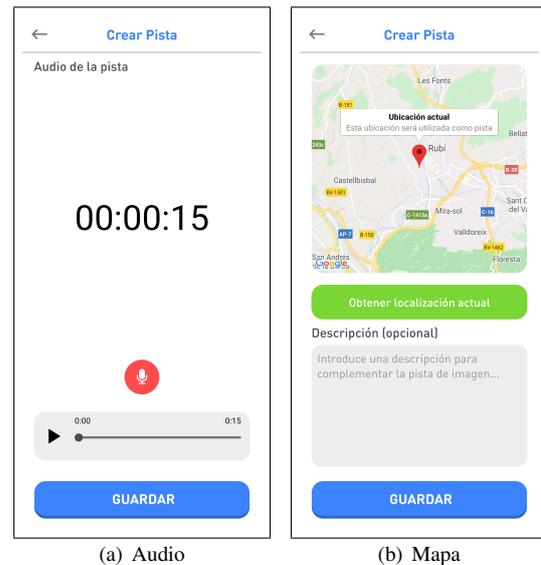


Fig. 9: Pantallas de crear pista de audio y mapa

Las pistas guardadas se muestran en una lista y se pueden editar pulsando el icono del lápiz. Esto nos lleva a una pantalla con los datos de la pista creada precargados donde tenemos la opción de editar los datos y guardar o bien eliminar la pista de la lista.

### Partidas creadas

Desde el menú principal el usuario puede pulsar en “Jugar” para visualizar la pantalla “Mis partidas” con una lista de sus partidas creadas (Fig 10. a). Al seleccionar una se puede visualizar el título, la descripción completa, el número de pistas y la modalidad de juego (Fig 10. b). Además, tenemos un botón para jugar la partida, un switch para crear una partida online y otro botón para descargar un PDF con los códigos QR. Estas dos últimas opciones solo son visibles para el usuario que ha creado el juego.

### Partidas online y tabla de clasificación

La opción de partida online permite compartir las partidas creadas de un usuario con otros jugadores. Al activar el switch, se genera un código alfanumérico de 5 dígitos que se utiliza como identificador único de la partida online (Fig 11. a). Este identificador se almacena en Firestore junto con los datos del juego. Después se mostrará un recuadro donde el creador de juego podrá visualizar el código y compartirlo con otros jugadores para que puedan jugarlo en sus dispositivos correspondientes. Esta opción también

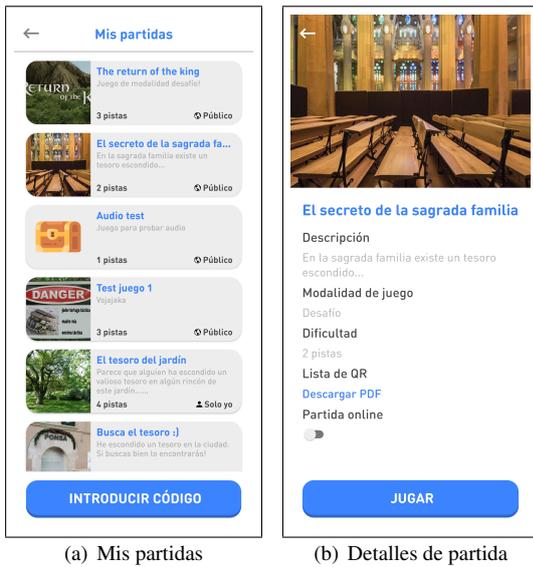


Fig. 10: Pantallas de lista de partidas creadas y detalles de una partida

activa un botón para visualizar una clasificación de los participantes que han completado las partidas y el tiempo que han tardado (Fig 11. b).

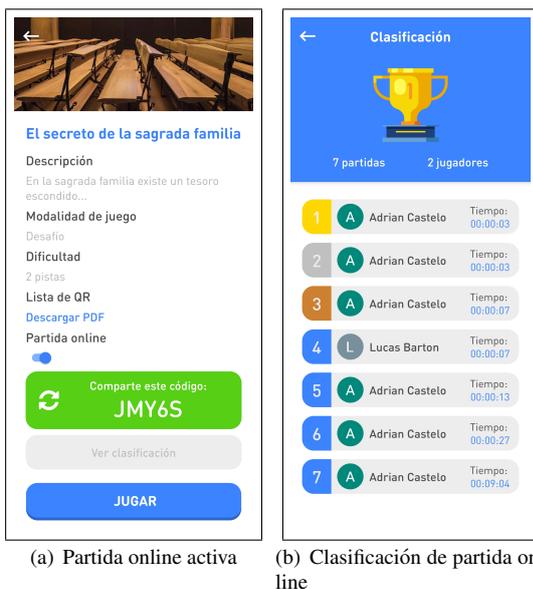


Fig. 11: Pantallas de partida online y tabla de clasificación

**Jugar partidas**

Desde el menú principal, los usuarios podrán pulsar la opción “Jugar” para visualizar sus partidas creadas y jugarlas, o bien pulsar el botón “Introducir código” para introducir el código de un partida online y jugarla.

Al jugar una partida en modalidad Historia, aparecerá una pantalla que mostrará la primera pista del juego. También se mostrará una barra de progreso junto con el número de la pista actual y el total, además de un botón para resolver la pista.

En la modalidad Desafío, se muestra una lista con todas las pistas, que se podrán resolver en cualquier orden.

Tipos de pista:

- **Imagen:** Imagen: Se muestra una imagen con zoom y una descripción si existe. El usuario puede pulsar sobre la imagen para verla en tamaño completo.
- **Texto:** Se muestra la descripción de la pista.
- **Audio:** Se muestra un reproductor con un slider táctil y botones para reproducir, pausar o adelantar/retroceder el audio.
- **Mapa:** Muestra un mapa con una localización marcada y una descripción si existe. La localización también se puede abrir en Google Maps para utilizar el GPS.

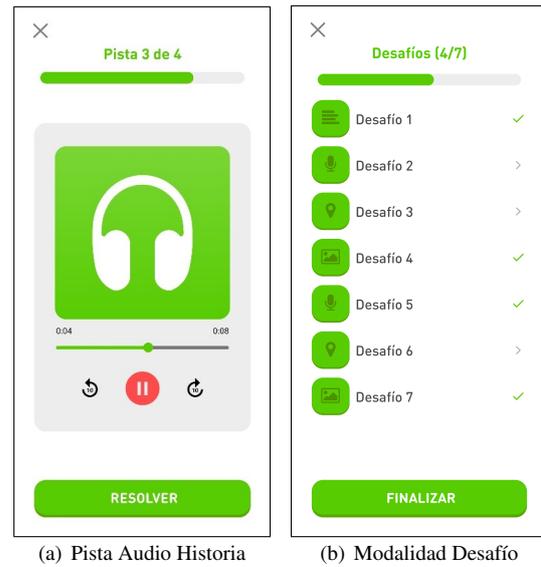


Fig. 12: Pantallas de pista de audio en modalidad Historia y partida de modalidad Desafío

El usuario deberá utilizar la información de la pista para encontrar el lugar donde el creador de la partida ha escondido el código QR correspondiente a esa pista. Una vez lo encuentre podrá pulsar el botón “Resolver” para abrir el escáner QR (Fig 13. a). Este incluye un botón para activar la linterna en caso de que sea necesario. Si el código escaneado es correcto, se mostrará una animación con un “tick” y se avanzará a la siguiente pista. En caso contrario se mostrará una animación con una cruz y seguirá escaneando hasta que el usuario introduzca el código correcto o cierre el escáner. Una vez el usuario haya completado todas las pistas se mostrará una última pantalla de ganador con una animación de confetti y se le redigirá al menú principal (Fig 13. b).

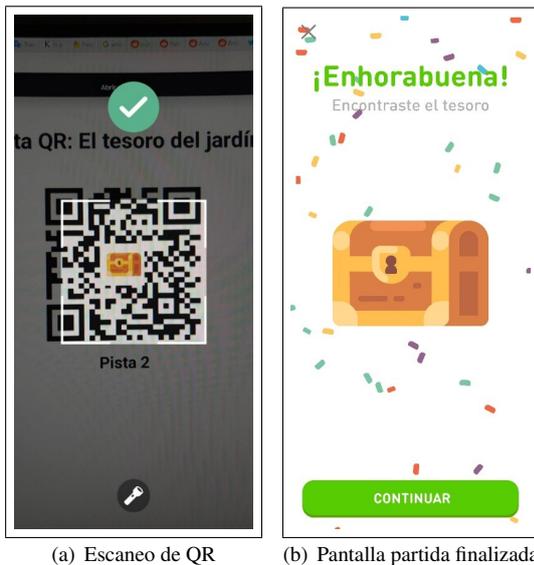


Fig. 13: Pantallas escáner QR y de partida finalizada

## 12 CONCLUSIONES

El objetivo principal de este trabajo de fin de grado era el de conseguir una aplicación funcional en la que se pueden crear y jugar juegos de búsqueda del tesoro haciendo uso de la tecnología QR y aprovechando las funcionalidades de los móviles.

En general, se han seguido los diseños del diagrama de casos de uso y flujo durante el desarrollo de la aplicación, además del diseño inicial de la interfaz, aplicando pequeños cambios y matices cuando era necesario. Esto ha permitido crear una aplicación fiel a la planteada al inicio del proyecto.

También se ha seguido la planificación propuesta inicialmente, a excepción un retraso la fase inicial del desarrollo de la aplicación, pero que se compensó con más trabajo durante las siguientes semanas. La metodología elegida fue la correcta y permitió tener organizadas las tareas pendientes y tener un control sobre las cosas que quedaban por hacer.

A nivel personal, el desarrollo de este proyecto me ha permitido profundizar más sobre tecnologías emergentes como React Native, Expo y Firebase. Ha sido una gran experiencia de la cual estoy convencido de que me será de utilidad de cara al futuro.

## 13 VÍAS DE CONTINUACIÓN

En cuanto a posibles mejoras o vías de continuación en un futuro, podría ser interesante investigar sobre las tecnologías de realidad aumentada y si se les puede dar algún uso en la aplicación, ya que es una tecnología interesante que probablemente llamaría la atención de los niños.

Otra posible línea de continuación sería la expansión hacia otros públicos, como por ejemplo el uso de la aplicación como guía turística aprovechando el grabador y reproductor de audio ya creados.

## 14 AGRADECIMIENTOS

Antes que nada, me gustaría agradecer el apoyo que me han ofrecido por mis padres, no solo durante este proyecto sino durante todos estos años de formación en esta carrera.

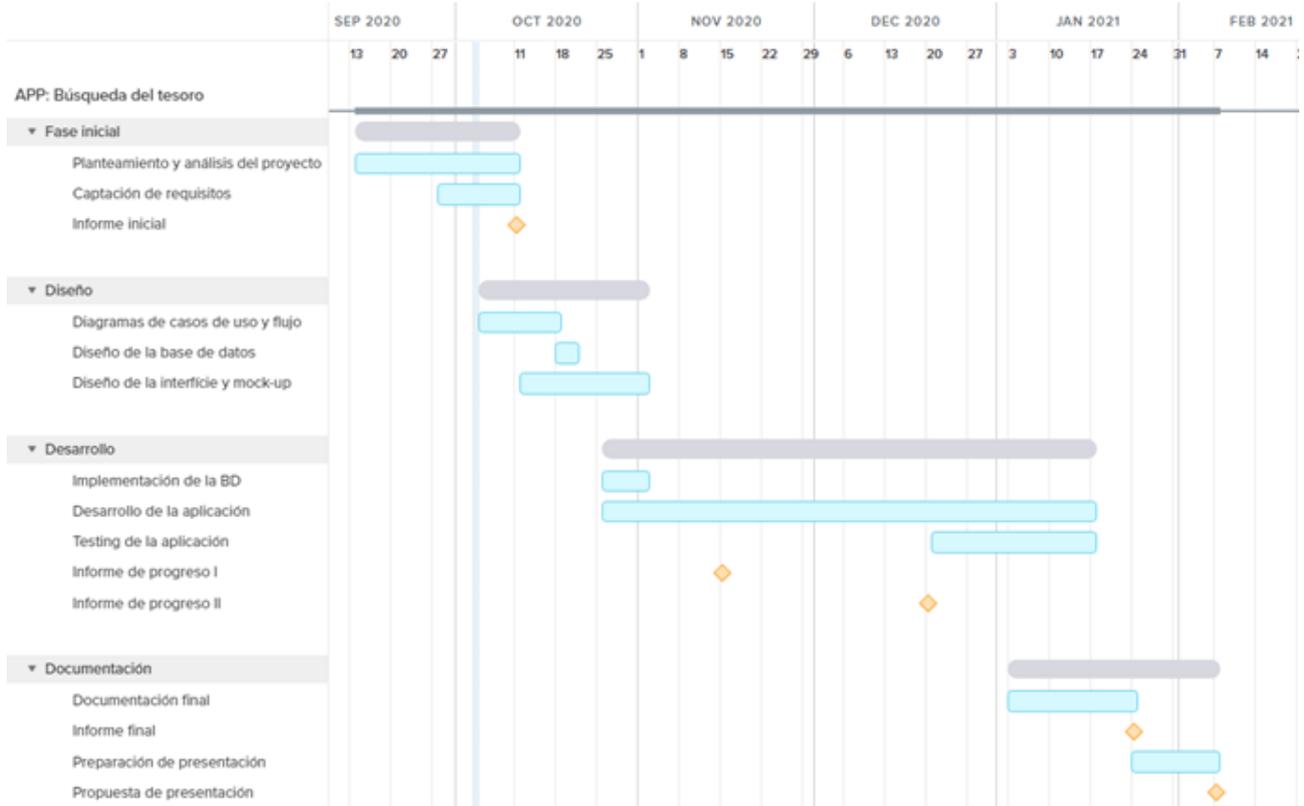
También quiero agradecer a mi tutor del proyecto, Enric Vergara, que propuso la idea de crear esta aplicación y me ha aconsejado cuando me ha hecho falta.

Y finalmente me gustaría dar las gracias a todos aquellos que me han ayudado en el desarrollo de este proyecto con sus opiniones, ideas o comentarios.

## 15 BIBLIOGRAFÍA

- [1] Scavify. Create powerful engagement for your group with our customizable scavenger hunt app. Disponible en: <https://www.scavify.com/howitworks>
- [2] ActionBound. Create mobile adventures and interactive guides for smartphones and tablets. Disponible en: <https://en.actionbound.com/>
- [3] Groundspeak Inc. Geocaching. Disponible en: <https://play.google.com/store/apps/details?id=com.groundspeak.geocaching.intro>
- [4] Boscaf Software. SnapHunt - A Scavenger Hunt Game. Disponible en: <https://play.google.com/store/apps/details?id=com.em2.snaphunt>
- [5] Robbie Elias. Camera Hunt - Scavenger Game. Disponible en: <https://play.google.com/store/apps/details?id=com.camerahunt.camerahunt>
- [6] Kid apps. Treasure Hunt — Gps mode — QR codes. Disponible en: <https://play.google.com/store/apps/details?id=com.kidapps.treasurehunt>
- [7] Kanbantool.com. ¿Por qué utilizar la metodología Kanban? Disponible en: <https://kanbantool.com/es/metodologia-kanban>
- [8] Asana. Disponible en: <https://asana.com/es/product>
- [9] Github. Disponible en: <https://github.com/features>
- [10] Proto.io. Disponible en: <https://proto.io/>
- [11] Fernando Ontiveros. ¿Por qué desarrollar apps de React Native en Expo? Disponible en: <https://medium.com/leopark-lab/por-qué-desarrollar-apps-de-react-native-en-expo-2e1b83e4d00a>
- [12] Sara López. Firebase: qué es, para qué sirve, funcionalidades y ventajas. Disponible en: <https://www.digital55.com/desarrollo-tecnologia/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- [13] Reactnavigation.org. Moving between screens. Disponible en: <https://reactnavigation.org/docs/navigating>

### A.1 Diagrama de Gantt



### A.2 Diagrama de flujo

