



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

TRABAJO FINAL DE GRADO

TÍTULO DEL TFG: Aplicación de quedadas para realizar actividades deportivas

TITULACIÓN: Grado en Ingeniería Telemática

AUTOR: Antoni Mas Canet

DIRECTOR: Antoni Oller Arcas

FECHA: 23 de octubre del 2017

Título: Aplicación de quedadas para realizar actividades deportivas

Autor: Antoni Mas Canet

Director: Antoni Oller Arcas

Fecha: 23 de Octubre del 2017

Resumen

Tras la aparición de las redes sociales y el boom de las aplicaciones móviles, las relaciones interpersonales se están produciendo cada vez más en un entorno online. Este factor añadido al crecimiento de la inactividad física está produciendo una desconexión social en nuestra sociedad. Dado este problema, se propone realizar una aplicación móvil que ayude a motivar a la gente a practicar deporte y a conocer nuevas personas.

Esta aplicación permite la creación de actividades deportivas al aire libre y la búsqueda de estas, con el objetivo de encontrar las actividades y las personas más afines al criterio del usuario. Además, cada usuario tendrá un conjunto de comentarios y votaciones que ayudarán a valorar su comportamiento en la plataforma. Los usuarios que participen de forma conjunta en una misma actividad tendrán acceso a un chat interno para poder comunicarse de forma rápida.

Antes de realizar la aplicación, se ha realizado un estudio de la competencia actual en el sector, con el objetivo de descubrir un nicho de mercado, y un estudio de los posibles usuarios interesados en esta.

Para realizar la aplicación se ha utilizado una plataforma de desarrollo que permite generarla para los dispositivos móviles con sistema operativo Android, iOS y Windows.

Para la parte del servidor, se ha utilizado un servicio contratado que proporciona todas las opciones de desarrollo necesarias para poder realizar las funciones requeridas por la aplicación.

Como resultados de este trabajo, se ha conseguido realizar una versión beta de la aplicación y publicarla en la tienda de aplicaciones de Android, con las funcionalidades requeridas para que los usuarios puedan crear actividades o unirse a ellas.

Title: Social network application for sports activities

Author: Antoni Mas Canet

Director: Antoni Oller Arcas

Date: October 23, 2017

Overview

Ever since the emerge of the social networks and the boom of mobile applications, the interpersonal relationships are increasingly taking place in an online environment. This factor, added to the growth in physical inactivity, is causing a social disconnection within our society. To confront this issue, the creation of a mobile application that encourages people to play sports as well as meet other peers is proposed.

This application allows for the publication of outdoor sports activities, moreover, permitting the search of such activities and people closest to the chosen criteria. In addition, users can receive and leave reviews which will help to value their behavior on the platform. Users who engage in the same activity are given access to an internal chat so that they can quickly communicate.

Previous to the conception of the application, an analysis of the sector's current competition aimed to discover a niche in the market has been carried out, Furthermore, a study of potentially interested users has also been performed.

In order to create the application, it has been used a development platform that permits the generation of applications for mobile devices with operative system Android, IOS and Windows.

Regarding the server, a contracted service that provides all the development options needed to perform the application functionalities has been employed.

The main outcome of the project is the development of a beta version of the application, as well as its publication into the Android application store. This first, published version includes all of the functionalities required for users to create sports activities or search and join existing ones.

Agradecimientos

Dedico este trabajo a mi familia y amigos por apoyarme siempre que ha sido necesario

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1. CONTEXTO.....	4
1.1. Motivación personal.....	4
1.2. Motivación social.....	4
1.3. Estado actual del mercado móvil en aplicaciones deportivas	5
1.4. Objetivos de la aplicación	7
CAPÍTULO 2. DISEÑO DE LA APLICACIÓN	8
2.1. A quien va dirigida la aplicación.....	8
2.2. Personas.....	10
2.3. Flujo de proceso	12
2.4. Definición de funcionalidades.....	15
2.5. Arquitectura de la información	15
2.6. Diseño de los wireframes	16
CAPÍTULO 3. CONSTRUCCIÓN DE LA MARCA.....	17
3.1. Nombre de la aplicación	17
3.2. Eslogan de la aplicación.....	17
3.3. Gama de colores.....	18
3.4. Diseño del logotipo, el icono y la pantalla de carga	18
CAPÍTULO 4. ARQUITECTURA DE BINC	20
4.1 Esquema de la arquitectura	20
4.2 Detalle de los servicios del Baas.....	21
4.2.1 Servicio de autenticación.....	21
4.2.2 Base de datos.....	21
4.2.3 Almacenamiento	22
4.2.4 Notificaciones Push	22
4.2.5 Analíticas	22
4.2.6 Funciones	22
4.3 Comparación de la arquitectura.....	23
CAPÍTULO 5. TECNOLOGÍAS Y ENTORNO DE TRABAJO	24

5.1 Elección de la plataforma de desarrollo de la aplicación	24
5.2 Elección del Baas	26
5.3 Entorno de trabajo	27
CAPÍTULO 6. DESARROLLO DE BINC	29
6.1 Creación de las vistas.....	29
6.2 Estructuración de la base de datos.....	30
6.3 Autenticación.....	31
6.4 Actividades	34
6.5 Usuarios	37
6.6 Notificaciones.....	40
6.6 Analíticas.....	41
CAPÍTULO 7. PUESTA A PRODUCCIÓN	43
7.1 Preparación del Backend	43
7.2 Preparación de la aplicación para producción	43
7.3 Subida a la tienda de aplicaciones.....	44
CAPÍTULO 8. PLANIFICACIÓN DEL TRABAJO	46
CAPÍTULO 9. CONCLUSIONES	47
9.1 Objetivos conseguidos.....	47
9.2 Conclusiones personales.....	47
9.2 Futuras implementaciones.....	48
BIBLIOGRAFÍA	50
ANNEXO I: WIREFRAMES	54

INTRODUCCIÓN

El mundo del deporte y la tecnología cada vez están más ligados tanto a nivel profesional como amateur con dispositivos que nos permiten medir el rendimiento, sistemas de planificación de entrenamientos, gps para poder realizar rutas sin conocerlas etc.

Además de estos dispositivos específicos para la práctica del deporte y juntamente con el auge de los smartphones, también han ido apareciendo aplicaciones móviles que usando las características de nuestro teléfono nos permiten sustituir a muchos de estos haciéndolos más accesibles al gran público.

Pero no solo esto sino que le han añadido al deporte un factor social muy importante donde aunque entrenes solo cuando terminas el entrenamiento puedes subir los datos registrados a redes sociales especializadas y así poderte comparar con tus amigos o incluso con profesionales de ese mismo deporte que están usando las mismas plataformas.

Todas estas aplicaciones le dan un punto de emoción a cada entrenamiento, pero a veces pueden llegar a quitarle ese punto lúdico de diversión que generalmente busca cualquier deportista amateur, saliendo con amigos sin preocuparse por los tiempos, el kilometraje o las pulsaciones.

Teniendo en cuenta todo esto, la idea de este proyecto es aportar un punto de vista distinto a estas plataformas online, desarrollando una aplicación móvil para llevar de vuelta el mundo virtual al mundo real. De forma que nos permita descubrir nuevas zonas para poder practicar deporte de una forma lúdica, y además conociendo gente nueva a la cual le gustan las mismas aficiones para poder disfrutar de estas en compañía.

La funcionalidad principal de la aplicación es crear actividades deportivas al aire libre, sin necesidad de ningún tipo de instalación, de forma simple, dinámica y social, que a su vez permita a usuarios de la zona poder encontrarlas y apuntarse fácilmente.

En el transcurso de este texto, se explican las distintas funcionalidades de la aplicación, las tecnologías utilizadas, los objetivos y las motivaciones para realizar el proyecto.

En el primer capítulo, se explicaran un caso de uso práctico de la aplicación, la motivación personal para desarrollarla y el mercado actual existente.

En el segundo capítulo, se describirán las funcionalidades específicas, el diseño de la aplicación y los objetivos a nivel de desarrollo.

En el tercer capítulo, se desarrollará como ha sido el proceso de creación de marca de la aplicación.

En el capítulo cuarto se describe la arquitectura utilizada para desarrollar la aplicación.

En el quinto capítulo, se explicaran el conjunto de tecnologías utilizadas para el desarrollo y el entorno de trabajo.

En el sexto capítulo, se explicará el desarrollo de la aplicación de inicio a fin con detalle de las funciones más relevantes

El capítulo séptimo, cuenta los pasos realizados para una vez desarrollada la aplicación publicarla en la tienda de aplicaciones.

En el octavo capítulo, se explicará la planificación seguida para realizar el proyecto, así como las metodologías usadas.

Finalmente, en el noveno capítulo las conclusiones y objetivos realizados en el proyecto.

CAPÍTULO 1. CONTEXTO

Como punto de partida del trabajo se explicará de donde proviene la idea de la aplicación, mi motivación personal, la motivación social, el estado actual del mercado de aplicaciones deportivas y los objetivos de la aplicación.

1.1. Motivación personal

En cuanto a mí, soy una persona que practica deporte de forma continua desde hace ya bastantes años siempre de forma amateur y desde hace unos cuatro años sobre todo práctico el ciclismo y el correr siendo éste primero sin duda alguna mi favorito y del cual practico disciplinas tanto de carretera como de montaña.

Una de las mejores cosas que tiene este deporte es el compañerismo de la “grupeta” el hecho de correr con más gente te ayuda a mejorar el rendimiento ya que al ir agrupados ayuda para el viento y hacer que todo sea más sencillo. Pero no solo eso sino que además te motiva a seguir entrenando y te aporta un extra de seguridad ya que lamentablemente el estado actual de las carreteras y de las leyes de tráfico protegen muy poco al ciclista.

El problema es que muchas veces nos cuesta encontrar gente que practique deporte con un nivel parecido al nuestro y esto ocasiona que terminemos entrenando solos el 90% del tiempo. Justamente en uno de esos entrenamientos de carretera donde cada vez que sales te encuentras a 10 vehículos que pasan a menos de 30cm de ti, es donde pensé en buscar alguna página web o aplicación para encontrar a gente de mi nivel y no tener que salir sólo con lo que eso conlleva.

En ese momento fue cuando empecé a buscar por internet y no encontré ningún sitio web ni aplicación que me permitiera simplemente eso encontrar gente que le gustará practicar deporte, de una forma sencilla, y sin tener que apuntarse a un club, indagar dentro de un grupo de WhatsApp o algún subforo que alguien algún día se le ocurrió abrir.

Así es como empieza este proyecto, con el objetivo de crear una aplicación para gente a la que le gusta practicar deporte tenga más sencillo poder encontrar personas afines, y de esta forma realizar su actividad preferida de manera más divertida, amena y segura.

1.2. Motivación social

Aparte de la motivación puramente personal y basada en una necesidad como ya se ha comentado, también existe una motivación social para la realización de este proyecto, ésta no es otra que la de ayudar a fomentar el deporte en la sociedad actual.

Según estudios de la Organización Mundial de la Salud (OMS)[1] dentro del rango de edad de personas adultas (18-64 años), se recomienda que practiquen al menos un mínimo de 150 minutos semanales de actividad física aeróbica (300 minutos si se quiere obtener mayor rendimiento) de intensidad moderada o 75 minutos (150 minutos para mayor rendimiento) de actividad vigorosa, así como realizar dos o más veces por semana actividades de fortalecimiento muscular de grandes grupos musculares.

De forma inicial estas recomendaciones nos pueden parecer relativamente simples de cumplir para cualquier persona pero la realidad es que un 23% de los adultos a nivel mundial no son suficientemente activos[2], y actualmente la inactividad física es el cuarto factor de riesgo en cuanto a mortalidad mundial (6% de las muertes registradas en el mundo). De hecho es uno de los principales puntos de mira de la OMS y se ha incluido en el plan de acción global para enfermedades no transferibles[3] con la intención de reducir los niveles de inactividad física un 10% de cara al año 2025.

Dentro de las recomendaciones de la OMS[1] encontramos una que dice que los entornos favorables y el apoyo de la comunidad nos pueden ayudar a mantenernos físicamente activos. Esto hace referencia sobre todo a nivel gubernamental para que se ayude a nivel de infraestructura ya puede ser con polideportivos o parques que contribuyan a que la gente puede practicar deporte de una forma lúdica y segura, a nivel formativo y divulgativo para ayudar a concienciar, y también a nivel social motivando a la gente a fomentar la práctica del deporte.

Dentro de este punto es donde este proyecto puede aportar su pequeño granito de arena, motivando a la gente a practicar más deporte y hacerlo de forma conjunta y divertida. De hecho realizar actividades deportivas de forma colectiva nos aporta una mayor motivación, constancia, nos permite conocer gente nueva que nos puede ayudar a crecer de forma personal e incluso a practicar deporte de una forma más segura ya que si nos pasa algo tenemos a alguien al lado que nos puede ayudar y socorrer.

1.3. Estado actual del mercado móvil en aplicaciones deportivas

Como se ha comentado anteriormente actualmente no hay una aplicación realmente relevante en el mercado que ayude a solventar el problema de encontrar gente que haga deporte para quedar con ellos de una forma simple y sencilla. Aun así sí que existen aplicaciones que solventan parte de este problema y son las que se han investigado para poder ver sus puntos fuertes y sus puntos débiles y así poder aprender de estos.

Strava[4]: Esta aplicación es la red social deportiva por excelencia orientada al ciclismo, correr y la natación. Lo que nos permite Strava es compartir nuestros entrenamientos con todo detalle con nuestros seguidores, además de esto tiene una parte competitiva donde los usuarios pueden competir por tener el mejor tiempo en ciertos tramos de recorridos que hayan subido y un conjunto

de retos tipo km semanales, mensuales y demás, que ayudan a motivarte y seguir activo en la plataforma. Además de esto incluye un plan Premium que te permite hacer un seguimiento exhaustivo de tus entrenamientos para mejorar tus capacidades donde la plataforma te va dando rutinas para conseguirlo.

Como se ve, la plataforma actúa como un mundo de conexión online pero no te ayuda a conectar de forma directa con los usuarios y poder quedar con ellos para realizar estos entrenamientos.

Meetup[5]: Si se habla de aplicaciones para poder crear y realizar eventos aquí sin duda alguna la número uno es Meetup ésta nos permite realizar prácticamente cualquier tipo de evento ya sea gratuito o de pago y facilita que la gente interesada en estos se puede inscribir de forma sencilla.

El problema principal de esta aplicación es que no está orientada a la creación de actividades deportivas sino de eventos en general y eso merma mucho la experiencia de usuario y la facilidad de uso, pues aunque exista un buscador interno es realmente sencillo perderse entre una gran cantidad de eventos de todo tipo.

Runnator[6]: es la red social de corredores permite competir de forma remota entre los usuarios de la plataforma, ésta es mucho más social que Strava ya que permite de forma muy sencilla interactuar con los usuarios, de hecho no busca ser un rival de Strava sino un complemento donde poder realizar carreras de forma online con cualquier persona del mundo, e incluso con eventos con multitud de personas simultaneas corriendo y batiéndose en tiempo real online pudiendo consultar tus resultados igual que en una carrera presencial.

Como vemos tenemos una conexión social mayor pero quedándose en el mundo online aunque de forma indirecta nos puede ayudar a conocer gente cercana a la cual nos batamos en duelo, pero no es la finalidad principal de la plataforma.

Timpik[7]: nos permite organizar partidos de pádel y fútbol directamente desde la aplicación y encontrar tanto gente para poder jugar con ellos como instalaciones para la práctica deportiva. Como vemos, la plataforma sí que está orientada a encontrar gente para la práctica de actividades deportivas pero está sobretodo orientada al fútbol y al pádel, aunque también permite realizar actividades de otros deportes pero se nota mucho a nivel de funcionalidades y comunidad que los deportes principales son esos dos y eso merma la experiencia de usuario para gente que simplemente quiere quedar para correr o salir en bici que no necesita ningún local ni una cantidad mínima de participantes para poder quedar.

Decathlon sport meeting[8]: La plataforma de Decathlon nos permite exactamente lo que se está buscando, encontrar a gente cercana que practique deporte. El problema de ésta es que está olvidada y poco optimizada además de incluir un montón de funcionalidades y deportes distintos por los cuales puedes crear actividades pero siempre con las mismas opciones

independientemente del deporte escogido y eso llega a ser realmente confuso. Además de esto, el funcionamiento de la aplicación es lento y complejo haciendo que llegue a ser frustrante la mayoría de las veces.

Analizando lo argumentado en conjunto, se puede apreciar como la mayoría de aplicaciones existentes en el mercado no están orientadas a quedar con gente cercana para realizar deporte sino para interactuar online. Y que las aplicaciones que sí están orientadas a quedar con gente no están desarrolladas para deportes que se pueden practicar al aire libre y compartan similitudes, sino que son o muy específicas o demasiado abiertas incluyendo todo tipo de deporte o evento.

1.4. Objetivos de la aplicación

Como ya se ha comentado, el objetivo principal es el de conectar personas que practiquen deporte dentro de una misma zona geográfica, para ello tenemos en cuenta varios aspectos clave, que se han definido en base a las necesidades encontradas y aprendiendo de los puntos fuertes y flacos de las aplicaciones comentadas en el punto anterior, estos aspectos son:

- Que la aplicación esté orientada exclusivamente a encontrar de forma sencilla personas para la práctica del deporte.
- Acotar el tipo de deportes a aquellos que se practican al aire libre con características similares en cuanto a definición de la actividad deportiva.
- Facilitar al máximo el proceso de creación de actividades.
- Conseguir que los usuarios puedan identificar si el creador y participantes de una actividad son afines a ellos.
- Permitir una comunicación constante y fluida entre los usuarios inscritos a una misma actividad.

CAPÍTULO 2. DISEÑO DE LA APLICACIÓN

En este capítulo se explica el proceso utilizado para la definición de funcionalidades que necesitara tener la aplicación para cumplir los objetivos estipulados anteriormente y la realización de los wireframes¹.

2.1. A quien va dirigida la aplicación

Para empezar a definir qué funcionalidades deberá tener la aplicación y cómo debe interactuar el usuario con ésta, lo primero que se ha hecho es una segmentación del público que la va a utilizar.

Mediante al anuario de estadísticas deportivas del año 2017[9] se ha definido la segmentación demográfica de las personas que practican deporte en el estado español.

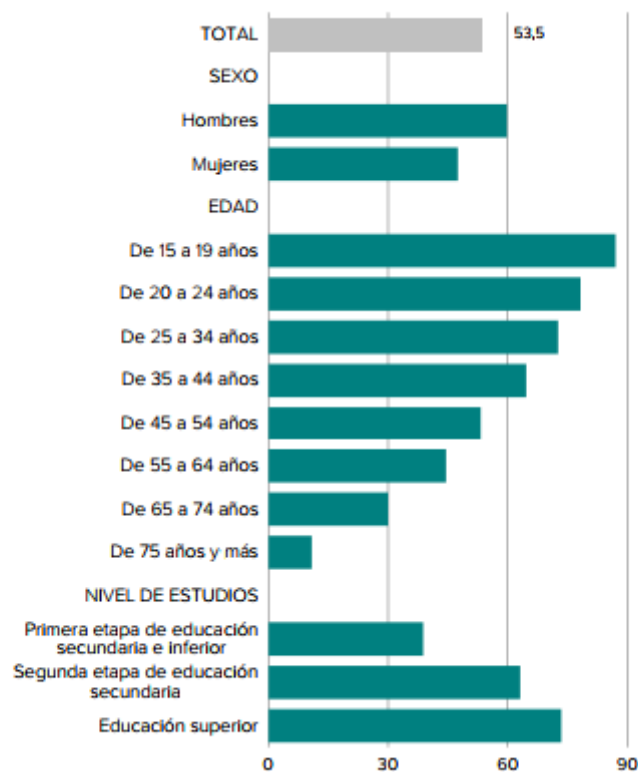


Fig. 2.1 Personas que practicaron deporte en el último año según sexo, edad y nivel de estudios. 2015 (En porcentaje de la población total investigada de cada colectivo)

¹ **Wireframe** un wireframe es un esquema de página o plano de pantalla que sirve como guía visual.

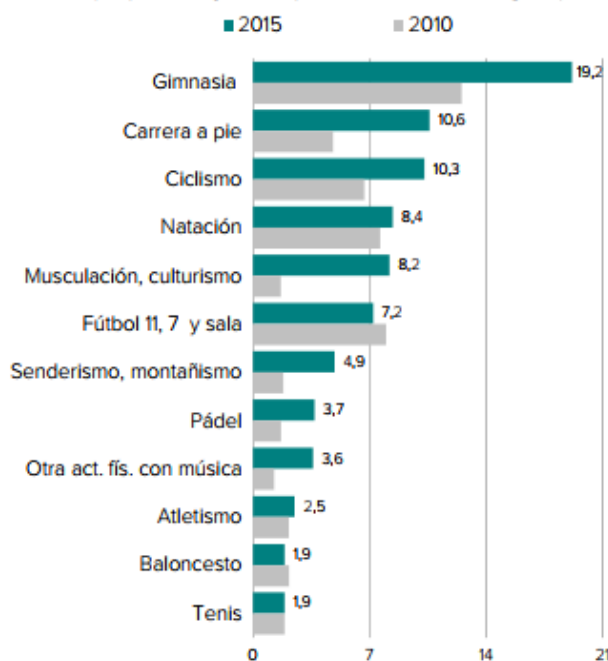


Fig. 2.2 Personas que practicaron deporte semanalmente según modalidades más frecuentes (En porcentaje de la población total investigada de cada colectivo)

En base a estos datos se puede caracterizar al español medio que practica deporte de forma más habitual y además usa el teléfono móvil de forma asidua, como hombre o mujer entre 15 y 44 años con estudios superiores al que mayoritariamente le gusta ir al gimnasio, practicar la carrera a pie, el ciclismo y la natación.

Si se cruza esta información demográfica con la del uso del móvil en España que nos aporta el informe móvil en España y el Mundo 2017 de Ditrendia[10], se observa como el sector demográfico en cuanto a edad y sexo que se comprende dentro de la práctica habitual del deporte también es muy asiduo al uso constante del dispositivo móvil y dentro de este uso de aplicaciones las más usadas son las apps sociales.

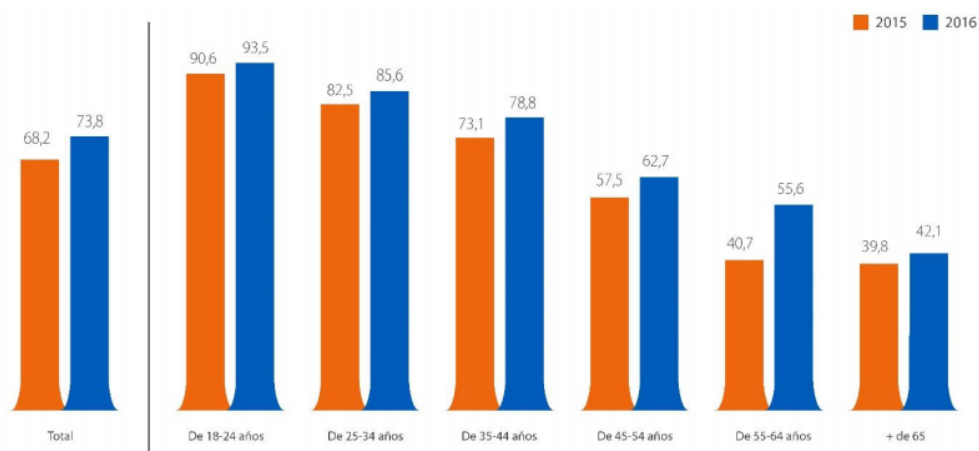


Fig. 2.3 Media de horas al mes dedicadas a las apps en dispositivos móviles por edades

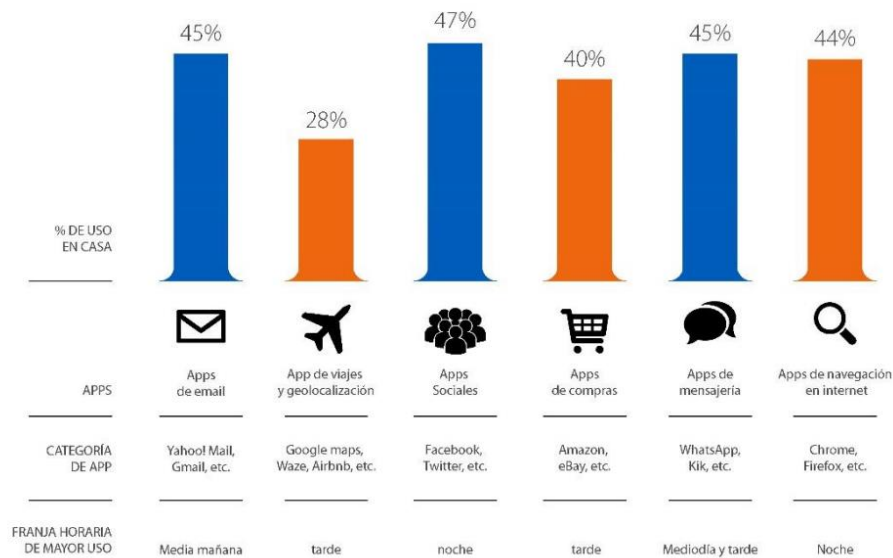


Fig. 2.4 Cuándo y dónde se utilizan las apps en el mundo

Se concluye con esta información, que el usuario medio que puede estar interesado en una aplicación para conocer gente con la que practicar deporte, será una persona hombre o mujer con un rango de edad comprendido entre 18 y 44 años que practica mayoritariamente la carrera a pie, ciclismo o la natación, con estudios superiores y que hace uso del móvil unas 80 horas semanales de las cuales la mayoría son en aplicaciones sociales.

2.2. Personas

Una vez se ha averiguado el público al que va dirigida la aplicación, se pasa a la realización de un perfil específico para cada uno de los distintos usuarios dentro del segmento demográfico de interés.

Para ello se ha utilizado un concepto de experiencia de usuario denominado personas[11] donde se hace exactamente eso crear perfiles de personajes imaginarios para representar diferentes tipos de usuarios que podrían utilizar un servicio. Esto ayuda a tener una visión más profunda del usuario al que nos dirigimos y a la toma de decisiones en base a estos, desligándonos de la subjetividad personal.

A continuación, se muestran cada una de las personas:



Nombre: María Pérez

Edad: 30 años

Profesión: Enfermera

Estado civil: Soltera

Antecedentes:

María es enfermera le gusta el mundo del deporte y la nutrición, práctica desde hace tiempo la carrera a pie e intenta fomentar el deporte y la nutrición saludable entre los suyos.

Objetivos para practicar deporte:

- Mejorar su salud y su cuerpo
- Conocer gente nueva
- Relajarse de la jornada laboral

Cualidades personales:

- Es muy activa en redes sociales
- Le gusta hacer reír a la gente
- Tiene una gran capacidad comunicativa

Fig. 2.5 Persona número uno



Nombre: Pere Arcarnos

Edad: 40 años

Profesión: Trabajo administrativo

Estado civil: Casado, con dos hijos de 16 y 10 años

Antecedentes:

Pere trabaja en un trabajo de oficina en el que está sentado la mayor parte del tiempo, ha empezado hacer deporte porque está cogiendo unos kilos demás, como le gusta mucho el ciclismo se ha comprado una bicis nueva para salir hacer rutas por el monte.

Objetivos para practicar deporte:

- Mejorar su forma física
- Conocer gente nueva para salir por la montaña en bicicleta
- Olvidarse durante un rato de la rutina diaria
- Poder practicar deporte con sus hijos

Cualidades personales:

- Le gusta el fútbol y el ciclismo
- De joven había practicado mucho deporte aunque lleve un tiempo parado
- Tiene poco tiempo debido al trabajo y la familia

Fig. 2.6 Persona número dos



Nombre: Juan Hermida

Edad: 19 años

Profesión: Estudiante

Estado civil: Soltero

Antecedentes:

Juan es un estudiante que lleva toda la vida practicando deporte, el que más practica en la actualidad es el triatlón de larga distancia, suele salir a entrenar con su primo pero este no tiene el mismo nivel que él y siempre tiene que esperarlo, le cuesta mucho encontrar personas de su edad que practiquen deportes de larga distancia con la misma intensidad.

Objetivos para practicar deporte:

- Llevar sus entrenamientos al máximo nivel
- Participar en pruebas deportivas de larga distancia
- Relajarse del día a día en la universidad

Cualidades personales:

- Es introvertido
- Tiene una gran forma física
- Ha participado en pruebas deportivas que normalmente la gente de su edad ni se plantearía.

Fig. 2.7 Persona número tres

2.3. Flujo de proceso

Una vez se ha visto a qué personas se dirige la aplicación, el siguiente paso que se ha implementado es la creación de un diagrama de flujo para definir los pasos que un usuario ha de realizar en la aplicación durante los casos de uso principales de ésta. Se identifican dos de estos casos como clave, para que la experiencia de usuario en la aplicación sea sencilla y eficaz para el funcionamiento principal de la aplicación. Estos son el de crear y el de encontrar actividades deportivas cercanas a las cuales el usuario pueda inscribirse y así conocer personas con las que practicar deporte de forma conjunta.

El primer caso de uso que se ha definido es el proceso que deberá realizar un usuario desde el momento en que abre la aplicación hasta llegar a crear una actividad.

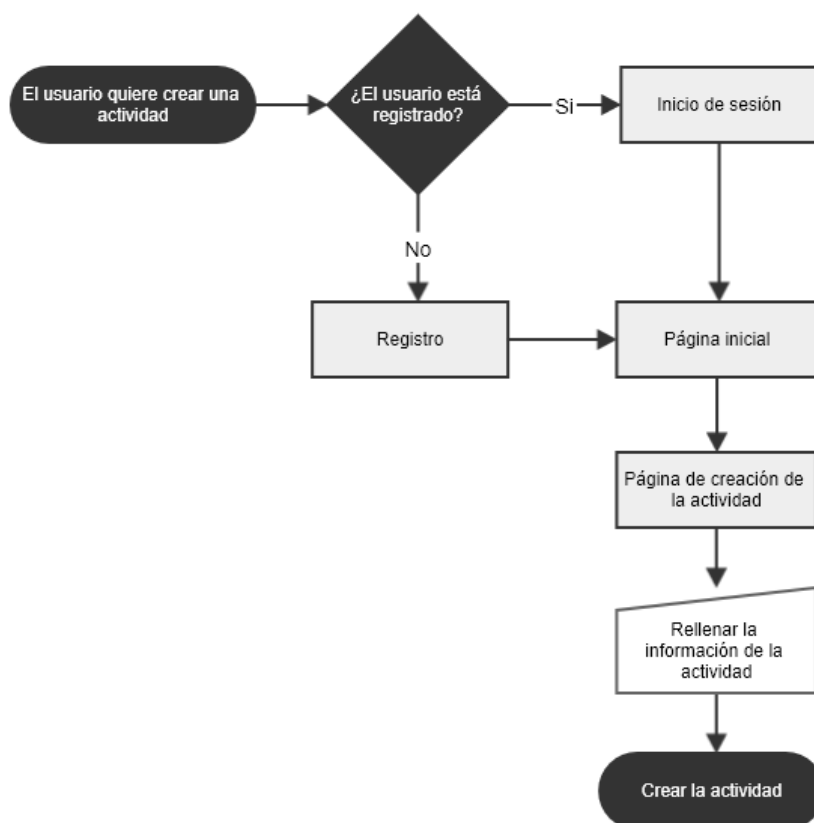


Fig. 2.8 Diagrama de flujo del usuario que quiere crear una actividad

En este flujo de usuario los puntos críticos que se han detectado han sido, por un lado, la navegación desde la página principal hasta la página de creación de una actividad, ya que deberá ser muy sencilla para evitar complicar el uso de la aplicación, esto se tendrá en cuenta en el desarrollo de la arquitectura de la información.

Y por otro lado, el momento donde el usuario rellena la información de la actividad. Al ser en una aplicación móvil, se deberá facilitar al máximo la forma en que estos datos son introducidos para que el usuario no se pierda en este punto, esto se tendrá en cuenta en el desarrollo del wireframe de la página donde se creará la actividad.

El segundo caso de uso se define como el proceso desde el momento en que el usuario abre la aplicación hasta que este encuentra una actividad afín a su criterio y se inscribe a ésta.

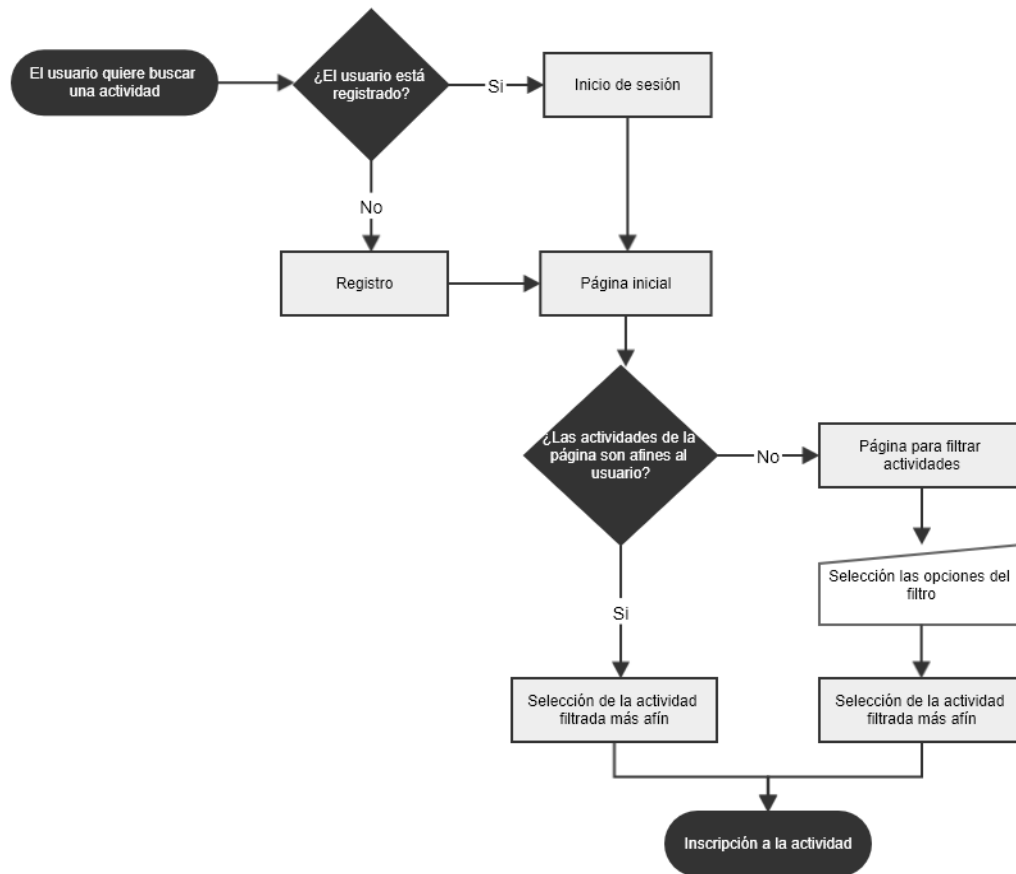


Fig. 2.9 Diagrama de flujo del usuario que se quiere inscribir a una actividad

En este segundo flujo de usuario los puntos críticos que se han detectado son:

En primer lugar el momento en que el usuario entra en la página principal y ve las actividades disponibles. Este, se identifica como un momento de decisión clave ya que los pasos que este debe dar para inscribirse a una actividad disminuyen mucho si ésta es afín o no. Por lo tanto en la fase de desarrollo se deberá tener esto en cuenta para que se muestren en pantalla las actividades más interesantes en función de cada uno de los usuarios y así facilitar la interacción del usuario con la aplicación.

El segundo punto que se ha identificado, corresponde a la navegación desde la página principal hasta la página de filtro, ésta tendrá que ser muy sencilla y directa. Se tendrá en cuenta a la hora de implementar la arquitectura de la información.

El último momento a destacar al igual que en el anterior flujo, es el momento en que el usuario introduce los datos requeridos por la aplicación, en este caso en la pantalla de filtro. Se tendrá en cuenta como ya se ha comentado antes en la parte de creación del wireframe de la pantalla.

2.4. Definición de funcionalidades

Una vez se ha visto el público y los flujos de los procesos clave en la aplicación, se define el conjunto de funcionalidades totales para que los usuarios puedan crear y encontrar de forma sencilla actividades deportivas en su zona.

Las funcionalidades son las siguientes:

- Proceso de autenticación (Registro y Login).
- Crear una actividad.
- Buscar una actividad.
- Inscribirse a una actividad.
- El creador de una actividad puede validar qué usuarios inscritos pueden participar.
- Seguir a un usuario.
- El creador de una actividad puede invitar seguidores a ésta.
- Comunicarse con los participantes de una actividad.
- Valorar las actividades en las que se ha participado.
- Comentar el comportamiento de los usuarios.
- Buscar a un usuario.
- Aceptar o declinar invitaciones recibidas.
- Salir de una actividad.
- Recibir notificaciones informativas.
- Modificar el perfil del usuario.
- Finalizar actividades.

2.5. Arquitectura de la información

Antes de empezar a realizar los wireframes y ya definidas las funcionalidades, se ha creado un diagrama de arquitectura de la información donde se muestra el conjunto de pantallas que tiene la aplicación y la navegación entre estas.

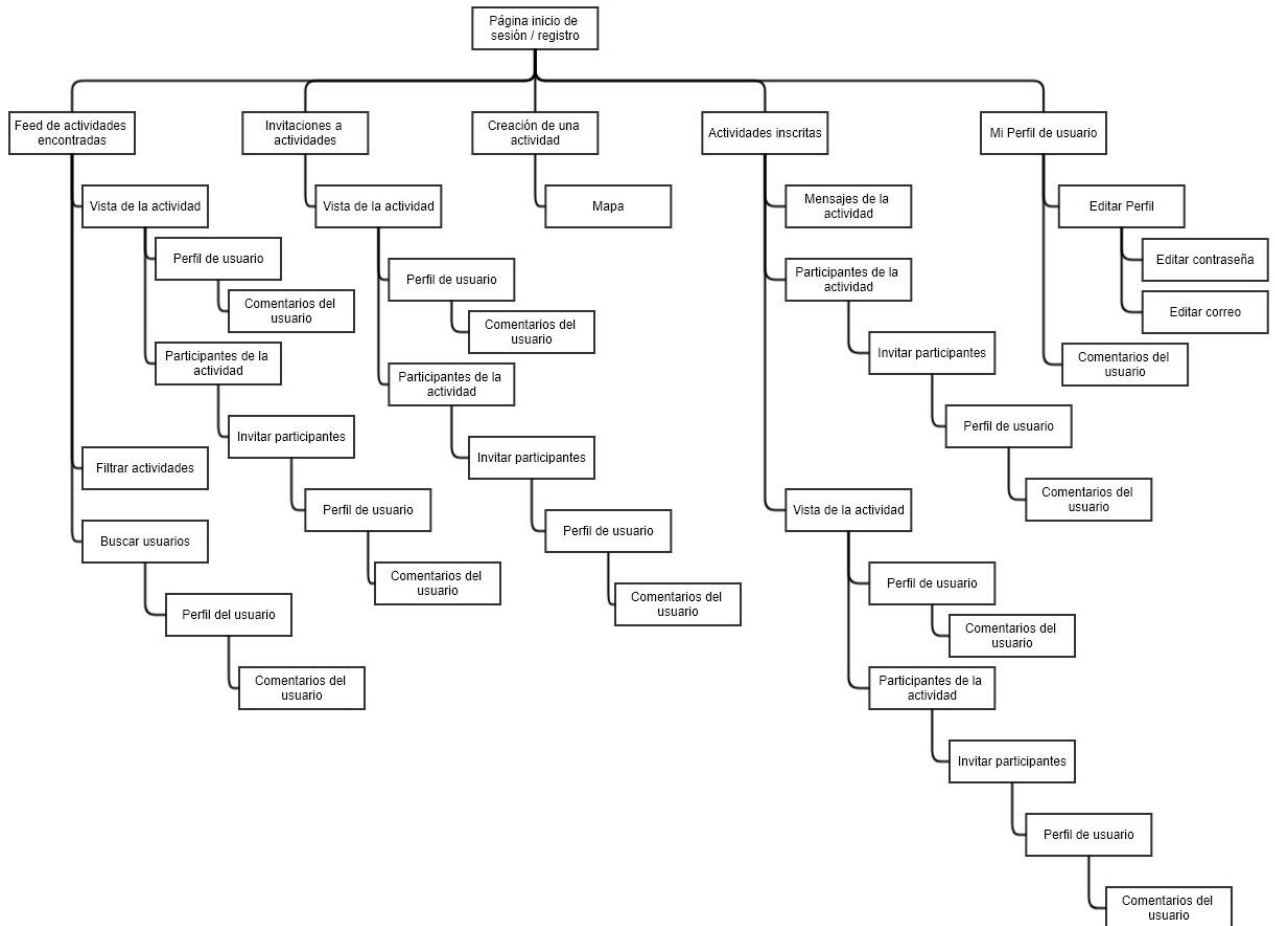


Fig. 2.10 Arquitectura de la información

2.6. Diseño de los wireframes

El paso final para este capítulo es la creación de los wireframes, un wireframe es una esquema de página o plano de pantalla que sirve como guía visual para ver los diferentes elementos que contendrán cada una de las páginas de la aplicación. Esto, es muy útil para ver una estimación de cómo será ésta antes de desarrollarla. Y en la parte de desarrollo se usaran los wireframes como base de la aplicación ayudando así a hacer la implementación de la parte visual mucho más ágil.

El conjunto de pantallas de los wireframes se adjuntan en el anexo uno de este documento.

CAPÍTULO 3. CONSTRUCCIÓN DE LA MARCA

En este capítulo se va a mostrar el proceso que se ha seguido para construir la marca de la aplicación, empezando por el origen del nombre, el eslogan, la selección de colores, el diseño final del logotipo y finalmente el icono de la aplicación.

3.1. Nombre de la aplicación

Para poder definir el nombre de la aplicación el proceso que se ha realizado ha sido el de juntar un conjunto de palabras asociadas a los valores que se quieren representar. Estos valores son tener una vida activa, seguir un estilo de vida saludable y fomentar el compañerismo.

Además de esto también se ha buscado un nombre que estuviera disponible online mediante el portal web namechek.com[12] para poder publicar la aplicación y que fuera simple y fácil de recordar.

Mobile Apps ([learn more](#))

Google PlayStore	available
App Store	available

Fig. 3.1 Disponibilidad del nombre Binc

Del conjunto de palabras inicial la que se ha escogido es “vinc”, que en catalán quiere decir vengo, pero modificándola usando un modelo de construcción evocativo haciéndola así única y completamente diferente, modificando la “v” inicial por una “b” quedando así finalmente como Binc.

De esta forma se construye un nombre nuevo completamente inventado pero con cierto significado.

3.2. Eslogan de la aplicación

Una vez establecido el nombre de la aplicación se ha buscado una frase corta, directa, fácil de memorizar y que transmitiera un mensaje claro. Para ello también se han tenido en cuenta los valores anteriormente comentados llegando así a la idea de que la aplicación debería representar un club deportivo global al cual todo el mundo puede inscribirse.

Con esa idea en mente se ha decidido usar como eslogan la frase “Únete al equipo” de esta forma se refuerza el valor de la comunidad y el conjunto de usuarios que utilizan la aplicación ayudando a fomentar la confianza entre los usuarios haciéndoles ver que pertenecen a un mismo colectivo.

3.3. Gama de colores

A la hora de seleccionar el color representativo de la marca se ha utilizado el color más afín con el significado psicológico que estos tienen en base a los valores que se quieren transmitir.

En base a los sentimientos que transmiten los colores[13] los que más se adecuan a la actividad física el movimiento la felicidad y el compañerismo son el rojo y el naranja de estos dos el naranja transmite más la parte social y el rojo la parte activa y deportiva. Justamente por eso al final se ha usado una gama de rojo para poder transmitir acción energía y actividad de forma que se transmita el valor del deporte desde el momento que se ve el logotipo de la aplicación.

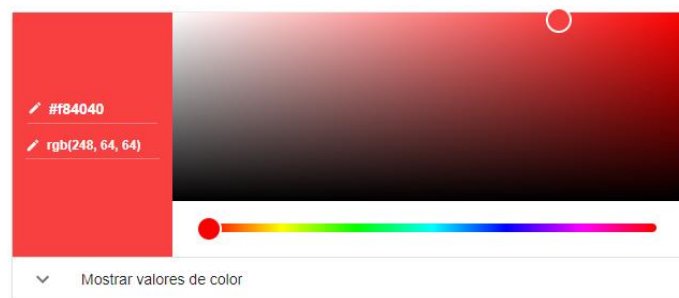


Fig. 3.2 Color utilizado.

Como se puede ver en la imagen la gama de rojo que se ha escogido ha sido un rojo suave para que al usar la aplicación este no destaque tanto que llegue a cansar al usuario pero que destaque lo suficiente como para que sea relevante.

3.4. Diseño del logotipo, el icono y la pantalla de carga

Para el diseño del logotipo se ha decidido usar un diseño basado en tipografía. La escogida para la realización de este ha sido la gobold[14] en su versión bold italic con pequeñas modificaciones, ésta es una tipografía libre de uso personal y con posibilidad de compra para un uso comercial.

A nivel estético se distingue por tener un trazo grueso, con inclinación itálica, rasgos uniformes, una estructura de líneas mixta ya que mezcla tanto rectas como curvas en sus elementos constitutivos y una interpretación ortográfica mayúscula, estos rasgos hacen que la tipografía transmita fuerza y dinamismo ligando así con los valores que se han comentado durante todo este capítulo.

A continuación se muestra el diseño final del logotipo con la tipografía mencionada y el color rojo identificativo de la marca.

The logo consists of the word "BINC" in a bold, italicized, red sans-serif font.

Fig. 3.3 Logotipo de la aplicación

Una vez diseñado el logotipo, se ha realizado el icono y la pantalla de carga de la aplicación que son dos elementos clave del diseño de esta.

Para el icono se ha escogido un icono en forma circular siguiendo la guía de estilos material[15] de Google. El color del icono es del mismo del rojo escogido para la marca con las letras "BC" en el interior con la misma tipografía que el logotipo para representarlo de forma iconográfica, esta vez de color blanco para generar un mayor contraste y así conseguir que las letras sean fáciles de leer e interpretar.



Fig. 3.4 Icono de la aplicación

Finalmente la pantalla de carga se ha realizado juntando el logotipo de la aplicación con el eslogan mencionado en el punto dos de este capítulo y usando la misma tipografía de color blanco y el color rojo de la marca de fondo.



Fig. 3.5 Pantalla de carga de la aplicación

CAPÍTULO 4. ARQUITECTURA DE BINC

En este capítulo se explicará la arquitectura utilizada para el funcionamiento de la aplicación.

4.1 Esquema de la arquitectura

Se va a utilizar el esquema que se presenta a continuación, para poner en contexto la arquitectura utilizada:

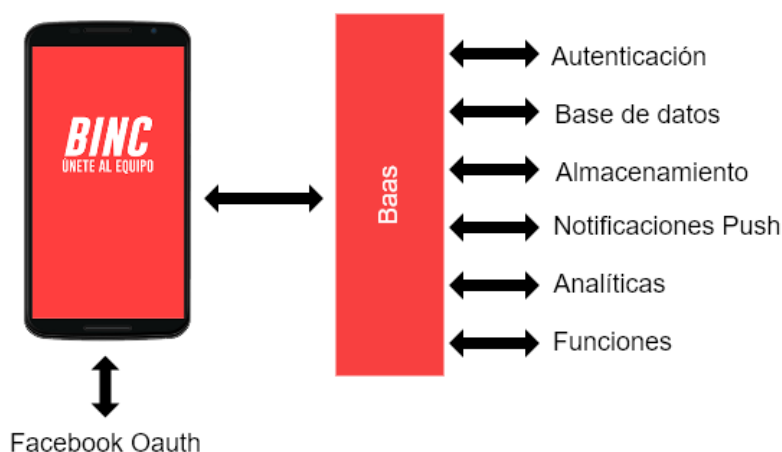


Fig. 4.1 Esquema de la aplicación

Como se puede ver en la imagen, la arquitectura de la parte del backend no es la de un servidor convencional, ya que se ha decidido utilizar un “Backend as a service” (Baas)² para gestionar esta parte de la aplicación. Esto permite agilizar el desarrollo centrándolo en la aplicación móvil.

La utilización del Baas se presenta a través de kits personalizados de desarrollo de software (SDK) que se encargan de tratar cada uno de los servicios requeridos por la aplicación.

Concretamente los servicios utilizados por la aplicación son, como se puede apreciar en la figura superior, los siguientes:

- Servicio de autenticación
- Base de datos
- Almacenamiento de imágenes
- Notificaciones push
- Analíticas
- Servicio de funciones en la nube

² **Backend as a service (BaaS)** es un modelo para los desarrolladores de aplicaciones móviles y web que permite acceder a distintos servicios alojados en internet permitiendo prescindir de una api personalizada.

Por otra parte la aplicación también utiliza la API³ de inicio de sesión de Facebook[16] para permitir el inicio de sesión y el registro de los usuarios mediante la cuenta de la red social.

En cuanto a la aplicación móvil se ha decidido crear una aplicación híbrida, eso permite que esta puede utilizarse en los distintos sistemas operativos móviles existentes actualmente con el mismo código sin necesidad de generar una aplicación propia para cada uno de estos.

Estas dos decisiones de utilizar un Baas y una aplicación móvil híbrida permiten que el desarrollo de la aplicación sea mucho más veloz y eficaz ya que se disminuyen el número de bugs y errores que puedan existir. Esto es debido a que no hay que encargarse del código y la infraestructura de un servidor, el código de la aplicación para Android, y el código de la aplicación para los, sino que solo hay un código existente que es el de la aplicación móvil híbrida.

4.2 Detalle de los servicios del Baas

A continuación se detallan cada uno de los servicios comentados en el punto anterior ofrecidos por el SDK del Baas que utiliza la aplicación.

4.2.1 Servicio de autenticación

Este servicio es el que permitirá a los usuarios el registro y el acceso a la aplicación mediante distintos métodos como pueden ser email y password, login con Facebook, twitter, google u otras plataformas en este caso los servicios utilizados son los de Facebook y los de email y password convencionales.

Además de gestionar el acceso y el registro de distintos métodos, también se utiliza este servicio para gestionar la recuperación de contraseñas si estas se han perdido, la modificación del email y la comprobación del mismo.

4.2.2 Base de datos

El servicio de base de datos es el que se encargara de almacenar toda la información de la aplicación para después poder acceder a ella en cualquier momento usando una conexión a internet.

Este conjuntamente con el servicio de autenticación también es el que se encargará de controlar el acceso a la información privada de la base de datos en función de distintos parámetros establecidos previamente.

³ **API** o *interfaz de programación de aplicaciones* se entiende como un conjunto de procedimientos o métodos que ofrecen ciertas funciones para ser utilizadas por otro software como una capa de abstracción.

4.2.3 Almacenamiento

El servicio de almacenamiento funciona como una red de entrega de contenidos (CDN)⁴, en este caso el uso que se le da es el de almacenar y distribuir todas las imágenes de la aplicación. Al formar parte del propio Baas queda mucho más integrado y eso facilita el uso con el sistema.

4.2.4 Notificaciones Push

El sistema de notificaciones push es el que permite enviar de forma remota notificaciones a todos los dispositivos que tengan instalada la aplicación y permitan que esta les envíe este tipo de notificaciones.

Este conjuntamente con el sistema de analíticas que se explicará a continuación permite enviar de forma segmentada estas notificaciones a los usuarios. Por ejemplo una notificación para actualizar la versión de la aplicación a todos aquellos usuarios que tengan una versión anterior a una que nosotros estipulamos.

4.2.5 Analíticas

El sistema de analíticas permite tener una idea del estado de la aplicación y de como los usuarios la están usando. Por ejemplo se puede saber el número de usuarios activos en un período, el tiempo de uso de estos, el conjunto de acciones realizadas en la aplicación por el mismo, etc.

Además, si tenemos este sistema integrado en el Baas nos permitirá hacer uso de esta información en otras partes, como por ejemplo el sistema de notificaciones push comentado en el punto anterior.

4.2.6 Funciones

Finalmente el servicio de funciones permite ejecutar ciertas funciones de código que necesiten de una mayor computación o funciones que necesiten un nivel de seguridad extra y de esta forma no ejecutarlas directamente en el dispositivo móvil del usuario.

Estas funciones además también se usan en conjunción con el resto de servicios y disponen de distintos activadores para que se ejecuten. Estos van desde una petición http desde la aplicación hasta que cuando detecte que se ha cargado una nueva imagen en el servicio de almacenamiento se ejecute una función para realizar cierta acción, como podría ser modificar el tamaño de esta.

⁴ **CDN** es una red superpuesta de ordenadores que contienen copias de datos colocados en varios puntos de una red. Con el fin de maximizar el ancho de banda para el acceso a los datos de los clientes de esta. Puede tener contenidos de varios tipos como archivos multimedia, software, documentos, etc.

4.3 Comparación de la arquitectura

En este apartado se presenta el esquema de una arquitectura con las mismas funcionalidades que se requieren en el Baas pero implementándola mediante un servidor propio y servicios externos independientes que se encarguen de gestionar lo que no se puede gestionar directamente desde el servidor.

Poniendo este esquema en perspectiva con el anterior podremos ver realmente las ganancias comentadas al inicio del capítulo.

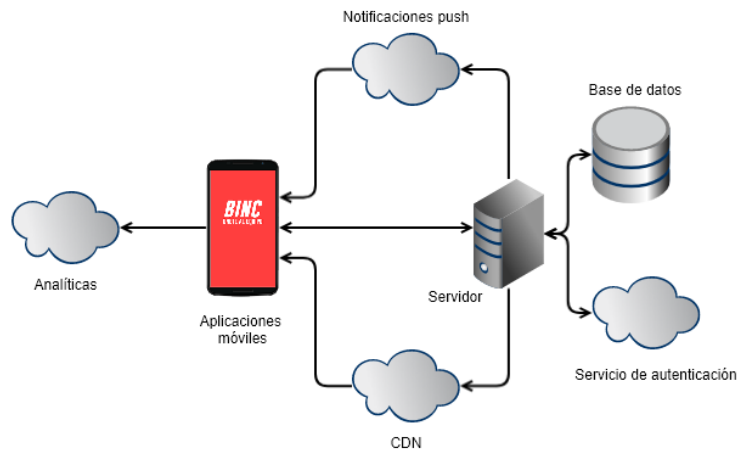


Fig. 4.2 Arquitectura con servidor propio

Como se puede apreciar en la figura 4.2 con esta arquitectura disponemos de un servidor que se conecta con nuestra base de datos y con un grupo de servicios independientes para poder tener una solución similar a la del Baas presentado en este mismo capítulo.

Esto implica que nos tendremos que hacer cargo no solo del desarrollo del servidor sino también del despliegue, mantenimiento y escalado del mismo una vez este en producción. Así como de un sistema para las copias de seguridad de la base de datos.

A esto le tenemos que añadir la contratación de cada uno de los sistemas independientes y el mantenimiento individual de estos como son el sistema de notificaciones push, la CDN y las analíticas de la aplicación.

En cuanto al servicio de autenticación es cierto que podríamos desarrollarlo en el mismo servidor pero no es tan seguro ni tan eficiente como tenerlo por separado ya sea contratado externamente o montándolo nosotros mismos aparte.

Como se puede ver tener todo esto de forma independiente es difícil de mantener y escalar, y además hace que el desarrollo sea mucho más lento ya que hay que tener en cuenta muchas más cosas. Por lo tanto, dadas las características de este proyecto y teniendo en cuenta que una persona sola es la que tiene que desarrollarlo todo, la opción del Baas cobra mucho sentido.

CAPÍTULO 5. TECNOLOGÍAS Y ENTORNO DE TRABAJO

En este capítulo se presentara la plataforma escogida para desarrollar la aplicación móvil híbrida, así como el Baas seleccionado y el entorno de desarrollo utilizado para llevar a cabo el proyecto.

5.1 Elección de la plataforma de desarrollo de la aplicación

Como se ha comentado en el capítulo cuatro se ha decidido utilizar una aplicación móvil híbrida para hacer el desarrollo más ágil. Dentro de las aplicaciones híbridas existen distintas categorías, para ello se ha hecho una comparación de las plataformas más utilizadas y estables del momento para ver cuál se adecuaba más al proyecto.

A continuación se presentan las distintas plataformas investigadas con las ventajas y desventajas correspondientes de cada una:

Xamarin[17]: Esta es la plataforma de desarrollo de aplicaciones de Microsoft. Según la compañía las aplicaciones son nativas, pero la realidad es que lo son a medias, ya que se pueden desarrollar las aplicaciones de dos formas. Normalmente se comparte el código de las acciones en los distintos sistemas operativos donde se quiere desarrollar la aplicación, este se programa mediante el lenguaje de programación C# y con acceso a los distintos elementos físicos del dispositivo móvil mediante las APIs nativas, después se crean vistas únicas para Android, ios y Windows phone.

También hay la opción de generar las vistas mediante lo que llaman Xamarin forms el problema de estos es que son bastante limitados por el momento. De esta forma el código sería 100% compartido entre las aplicaciones de los distintos sistemas operativos.

Pero como se ha comentado, normalmente se terminan desarrollando vistas separadas por sistema operativo compartiendo así un 75% del código según su documentación, aunque esto obviamente dependerá de cada aplicación.

Una de las grandes ventajas de esta plataforma es que tiene el soporte de Microsoft, lo cual le da un añadido de fiabilidad y además cuenta un conjunto de herramientas de desarrollo, puesta a producción y testeo muy interesantes aunque estas dos últimas son relativamente nuevas y están evolucionando.

React Native[18]: React native más que una plataforma es un framework⁵, que proviene de React utilizado en web y ha sido desarrollado por Facebook.

⁵ **Framework** Estructura conceptual y tecnológica definida normalmente mediante artefactos o módulos concretos de software que sirven de base para la organización y el desarrollo.

De hecho, la aplicación de Facebook e Instagram actualmente están desarrolladas con este framework.

Este es parecido a Xamarin en cuanto a que la aplicación no es del todo híbrida aunque en este caso es distinto, ya que directamente se puede combinar el código JavaScript del framework con código Swift, Java o Objective-C y juntarlo. Este framework solo nos permite desarrollar para Android y IOS.

Esto hace que al final en funciones que requieran de determinadas acciones tengamos que hacer esta mezcla de código nativo con JavaScript, y se terminen desarrollando dos aplicaciones aunque, eso sí, compartiendo parte de la lógica. Facilitando el desarrollo para la gente que tenga experiencia en web con el framework de React, ya que el funcionamiento y la lógica son parecidos.

Flutter[19]: Flutter es el SDK de desarrollo móvil que nos permite crear aplicaciones tanto para Android como para IOS creado por Google recientemente. De hecho, en estos momentos está en alfa y este es su mayor inconveniente.

El lenguaje de programación utilizado es Dart también creado por Google y básicamente permite crear aplicaciones compatibles con las dos plataformas utilizando el mismo código al completo.

Este es muy interesante y además cuenta con un entorno de desarrollo y distribución muy completo. Pero debido a su estado en alfa aún no está preparado para producción y además la información que se puede encontrar es bastante reducida pese a que tiene una documentación muy buena.

Ionic[20]: Ionic igual que React es un framework de JavaScript aunque este está basado en el framework Angular creado por Google. Ionic es híbrido del todo, de hecho, una misma aplicación con el mismo código sirve para Android, los, Windows pone e incluso para PWA's⁶.

Con este framework usando Typescript, html y css podremos crear como ya se ha comentado una aplicación única para los distintos sistemas operativos, para acceder a las capacidades nativas del dispositivo existe una API la cual podemos utilizar. Además, la misma no permite integración con determinados servicios de terceros como por ejemplo Google Analytics.

El último factor interesante es la gran comunidad que existe detrás de este framework y todo el ecosistema que se ha desarrollado con esta.

Una vez vistas las distintas opciones se ha decidido utilizar el framework Ionic para desarrollar la aplicación. Los motivos de esta decisión son poder realizar con un mismo código una aplicación para los distintos sistemas operativos siguiendo la misma lógica por la cual también se ha decidido utilizar un Baas.

⁶ **PWA** o *Progressive web App* es un término utilizado para describir webs que al verse desde un dispositivo móvil parecen aplicaciones de este e incluso puede utilizar ciertas funciones del hardware del dispositivo.

El hecho que use Angular también es un añadido interesante ya que es un framework que ya conozco en su primera versión y por lo tanto no parto de cero para empezar a desarrollar. Aunque en la versión dos que es la utilizada actualmente por Ionic cambia bastante, pero ciertas lógicas siguen siendo parecidas. Además, dispone de una documentación muy buena y de una comunidad muy grande y activa con la que poder apoyarse en cualquier momento.

5.2 Elección del Baas

Una vez decidido con qué se va a desarrollar la aplicación, falta decidir con qué empresa se contrata el servicio Baas. Para ello se ha hecho una investigación de los sistemas Baas orientados a aplicaciones móviles existentes en el mercado y se ha escogido el más afín al proyecto.

A continuación, se describen los Baas investigados más interesantes y las ventajas y desventajas de cada uno de estos:

Firestore[21]: es el Baas de Google. La intención de este servicio es dar a los desarrolladores la oportunidad de tener en un mismo sitio todo el conjunto de herramientas necesarias para la parte del backend y la gestión de la aplicación.

Así pues, dispone de muchas funcionalidades como una base de datos en tiempo real, un servicio de reporte de errores, servicio de autenticación, servicio de funciones en la nube, servicio de almacenamiento, servicio de analíticas, notificaciones push, alojamiento, configuraciones en remoto, integración con otras plataformas de Google como Adwords, generación de enlaces dinámicos a la aplicación, un emulador de dispositivos aunque solo para aplicaciones Android nativas y una última función aún en versión alfa llamada Cloud Firestore que actúa como una base de datos no relacional.

Además cuenta también con una muy buena integración con el framework de desarrollo móvil seleccionado, ya que al estar basado en Angular y este ser de Google hay mucha documentación y librerías oficiales para poder integrarlo de forma eficiente con la aplicación.

Finalmente, lo mejor de este Baas es la gran cantidad de opciones para el desarrollo móvil y la fiabilidad de tener una empresa como Google detrás. Además, en cuanto a precios se paga en función del uso, para una fase inicial de la aplicación sería gratuito, y para una fase ya más avanzada con un volumen de usuarios mayor estaríamos hablando de 25 dólares al mes.

Back4App[22]: Lo interesante de este servicio es que el sistema de backend que ofrece está basado en un proyecto de código abierto llamado Parse Server[23] y básicamente lo que ofrece es el almacenamiento, la escalabilidad y la autenticación de este sin tener que gestionarlo nosotros de forma independiente.

Parse Server básicamente es un backend realizado con node.js y express usando una base de datos mongodb, esto es interesante ya que a diferencia de

Firestore aquí aun ser un Baas disponemos de control sobre el código del servidor y podemos hacer modificaciones a este, pero con la agilidad de despliegue y desarrollo que nos ofrece un Baas.

A parte de esto, el servicio de Back4App también ofrece la capacidad de enviar notificaciones push, la capacidad de alojar una web en su servicio y un conjunto de add-ons de gestión.

En cuanto al precio también dispone de una versión gratuita pensada para desarrollo y después distintos planes de precios que van de los 4,99 dólares a los 99,99 en función del uso de la aplicación.

Kinevy[24]: El servicio Bass de Kinevy ofrece un conjunto de funcionalidades para aplicaciones móviles muy completo. Incluye autenticación, almacenamiento en base de datos, almacenamiento CDN, notificaciones push, servicio de localización, soporte de beacons y analíticas de uso.

Otra de las ventajas de este Bass es que su sdk está disponible en la mayoría de plataformas de desarrollo móvil algunas de las cuales se ha hablado anteriormente como Xamarin.

Las grandes desventajas de este servicio son el precio, ya que tiene un precio base de 200 dolares (con una versión gratuita para desarrollo), y la documentación y comunidad que no son tan activas como en los dos servicios anteriores.

En resumen, el servicio es muy completo ya que cuenta con muchas plataformas, una gran cantidad de funcionalidades y cumple con unas políticas de compliance⁷ muy rigurosas.

Una vez analizados los distintos Bass se ha decidido escoger Firebase para este proyecto. Los motivos de esta elección son: la integración con Ionic (el framework escogido para desarrollar la aplicación), su flexibilidad en los planes de precio y tarificación, el gran abanico de herramientas que ofrece para el desarrollo móvil y la comunidad que tiene detrás que al igual que la de Ionic es muy activa.

5.3 Entorno de trabajo

Para el entorno de trabajo se ha utilizado un ordenador con sistema operativo Windows 10. Esto es un inconveniente a la hora de empaquetar la aplicación ya que solo nos permitirá hacerlo para Android aunque esta esté preparada para IOS. Ya que para poder empaquetar la aplicación para IOS es necesario el uso de un ordenador con sistema operativo macOS. Esto se trata con más detenimiento en el capítulo siete.

⁷ **Compliance** se entiende como a un conjunto de políticas establecidas por una empresa, con el objetivo de asegurar que el conjunto de sus trabajadores conocen y cumplen con determinadas normativas y regulaciones que afectan a la misma.

Por la parte de desarrollo se ha utilizado todo el software y conjunto de paquetes necesarios para que Ionic funcione correctamente según los requerimientos de la documentación. Estos son los siguientes:

- Node[25]
- Ionic
- Cordova[26]
- Typescript[27]
- Android

En cuanto a entorno de desarrollo se ha utilizado el IDE de Microsoft Visual Studio Code. Este es totalmente gratuito y de código abierto, además se integra de maravilla con el lenguaje de programación Typescript que es el utilizado por Ionic en la parte de la lógica de la aplicación. Cuenta con gran cantidad de extensiones desarrolladas por la comunidad. En este caso no se ha utilizado ninguna de estas.

Finalmente, también se ha utilizado un sistema de control de versiones para poder llevar un registro del histórico de versiones de la aplicación y tener el código de ésta almacenado en un servicio externo en la nube de forma segura. Concretamente se ha utilizado GitLab, este está basado en el sistema Git y nos permite alojar el código en su plataforma de forma privada y gratuita además también dispone de sistema de integración continua y test.

CAPÍTULO 6. DESARROLLO DE BINC

En este capítulo se describen las distintas fases de desarrollo que se han realizado para desarrollar la aplicación con todas sus funcionalidades. Concretamente, se detallarán la fase de desarrollo de las vistas de la aplicación, la estructuración de la base de datos, la implementación de la autenticación, el conjunto de funciones relacionadas con las actividades y los usuarios, y finalmente la agregación de notificaciones y analíticas.

6.1 Creación de las vistas

La creación de las vistas ha comportado un desarrollo bastante continuo de las mismas, inicialmente se generaron un conjunto de vistas puramente funcionales basada en los wireframes mostrados en el apartado 2.6 de este documento con el objetivo de poder empezar cuanto antes el desarrollo de las funciones de la aplicación.

Una vez desarrolladas las funciones principales y con las vistas básicas generadas, se realizó una segunda fase donde el objetivo era el de conseguir un diseño más completo y que encajara con la identidad de marca en el capítulo tres de este documento. A continuación se muestra una captura de pantalla de esta fase:

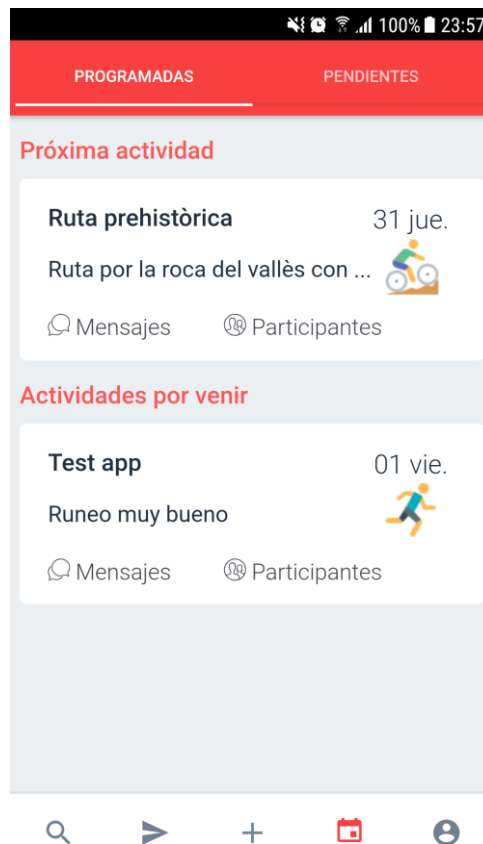


Fig. 6.1 Pantalla de actividades planificadas

Cuando todas las funciones de la aplicación ya estaban desarrolladas se llevó a cabo una fase final con el objetivo de unificar el diseño de la aplicación y prepararla para subirla a la tienda de aplicaciones. A continuación se muestra la misma pantalla que en la figura anterior con la versión final de la aplicación:



Fig. 6.2 Versión final de la pantalla actividades planificadas

Con estas tres fases se ha completado el desarrollo de esta parte de la aplicación de forma ágil y sin bloquear el desarrollo de las funcionalidades principales.

6.2 Estructuración de la base de datos

Una vez generadas las vistas y antes de empezar a desarrollar las funciones se ha hecho una estructuración de la base de datos teórica en base a las recomendaciones de Firebase.

Para ello hay que tener en cuenta que la aplicación consume directamente de la base de datos con el SDK de firebase, además esta base de datos es no relacional. Esto hace que sea necesario pensar en la estructura de la base de datos en función de los datos necesarios en cada visualización para evitar consultas innecesarias y optimizando así el uso. A continuación, se presenta una imagen donde se puede ver la estructura:

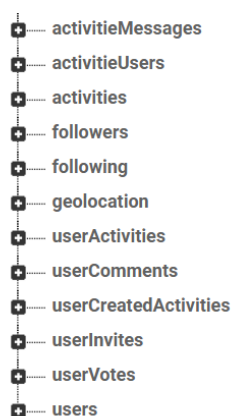


Fig. 6.3 Estructura de la base de datos

Como se puede apreciar en la imagen, terminamos obteniendo un objeto JSON con distintos nodos en función de la información requerida en pantalla. El SDK nos permite acceder a las distintas anidaciones de este objeto como si de una ruta se tratara y obtener o modificar información en ella en función de un conjunto de normas de seguridad establecidas, en este caso los usuarios que no se hayan registrado no pueden acceder a la información.

Además, cabe destacar que la base de datos actúa en tiempo real, por lo que nos permite hacer una aplicación muy reactiva de cara al usuario.

En resumen, lo más importante a la hora de crear la estructura en la base de datos es pensar en la visualización de la información en la aplicación.

6.3 Autenticación

Después de generar la primera fase de creación de vistas, la siguiente función implementada fue la de la autenticación con la aplicación. Para ello se decidió implementar la autenticación mediante correo electrónico y contraseña y la autenticación usando una cuenta de Facebook.

Para gestionar la autenticación, Firebase dota de un servicio específico que nos permite ambas. Además de esto también permite realizar la verificación del correo electrónico registrado, la modificación de este y la recuperación de la contraseña.

Para desarrollar esta funcionalidad se ha utilizado la librería AngularFire[28] para gestionar los Observables⁸ de autenticación, y el SDK JavaScript de Firebase para las funcionalidades de recuperación de la contraseña, modificación y verificación del correo electrónico, ya que la librería AngularFire no permite estas funcionalidades.

⁸ **Observables** en JavaScript se utilizan para la programación reactiva estos juntamente con las librerías adecuadas permiten que la vista de información proporcionada por la aplicaciones se actualicen de forma dinámica, los podríamos definir cómo un vector de datos asíncronos al cual estamos escuchando de forma continua.

Para empezar con el desarrollo se configuraron en la consola de administración de Firebase los dos métodos de inicio de sesión comentados. Una vez realizado se empezó creando un proveedor de autenticación en la aplicación, generando un fichero TypeScript importable por vistas de la aplicación que lo requieran.

La primera función generada fue la del registro del usuario mediante correo electrónico, está le manda a Firebase el email y la contraseña definidos por el usuario y recoge la respuesta de Firebase que contiene el identificador de usuario generado. Cuando lo recoge se guarda en la base de datos debajo de la ruta de usuarios, y manda un correo de verificación al usuario.

```
signupUser(newEmail: string, newPassword: string): firebase.Promise<FirebaseAuthState> {
  return this.af.auth.createUser({ email: newEmail, password: newPassword }).then((newUser) => {
    firebase.database().ref('/users').child(newUser.uid).set({ email: newEmail }).then(() => {
      firebase.auth().currentUser.sendEmailVerification();
    });
  });
};
```

Fig. 6.4 Función de registro

Cuando se genera este registro de forma correcta, la aplicación muestra un formulario de bienvenida donde el usuario puede terminar de registrar los datos necesarios para completar el registro.

Para la función de inicio de sesión mandamos el email y la contraseña a Firebase y este devuelve un token que se utiliza para poder utilizar la aplicación de forma segura. Una vez se obtiene el token un Observable es el que se encarga de gestionar el estado del usuario dentro de la aplicación.

```
loginUser(newEmail: string, newPassword: string): firebase.Promise<FirebaseAuthState> {
  return this.af.auth.login({email: newEmail,password: newPassword},
    {provider:AuthProviders.Password, method: AuthMethods.Password});
}
```

Fig. 6.5 Función de inicio de sesión

```
af.auth.subscribe( user => {
  if (user) {
    this.fireAuth = user.auth;
  }
}
```

Fig. 6.6 Suscripción al observador de autenticación

Una vez generado el inicio de sesión por email, el siguiente paso fue generar el inicio de sesión con Facebook. Para ello es necesario crear una aplicación de desarrollo en la plataforma y configurar el inicio de sesión con los permisos necesarios y instalar en la aplicación la dependencia nativa[29] de Facebook.

Una vez instalada la dependencia y configurada la aplicación en Facebook la función desarrollada para el inicio de sesión es la siguiente:

```

facebookLogin(): Promise<any> {
  return new Promise ((resolve, reject) =>{
    let userID: any;
    this.facebook.login(['email']).then( (response) => {
      console.log('reponse from face', response);
      const facebookCredential = firebase.auth.FacebookAuthProvider
        .credential(response.authResponse.accessToken);
      firebase.auth().signInWithCredential(facebookCredential)
        .then((success) => {
          console.log("Firebase success: " + JSON.stringify(success));
          userID = success.uid;
          console.log(userID);
          resolve (userID);
        }).catch((error) => {
          console.log("Firebase failure: " + JSON.stringify(error));
        });
    }).catch((error) => { console.log(error) });
  });
}

```

Fig. 6.7 Función de inicio de sesión con Facebook

El funcionamiento de esta es el siguiente, la aplicación se conecta con Facebook, este le devuelve a la aplicación las credenciales necesarias para el inicio de sesión, a continuación la aplicación manda estas credenciales a Firebase y las almacena para poder acceder a la aplicación mediante estas.

Finalmente una vez desarrollados estos métodos se generaron las funciones de recuperación de contraseña y modificación del correo electrónico mediante el SDK de JavaScript ya mencionado. Estas se presentan a continuación:

```

resetPassword(email: string): firebase.Promise<FirebaseAuthState> {
  return firebase.auth().sendPasswordResetEmail(email);
}

```

Fig. 6.8 Función de recuperación de la contraseña

```

updateEmail(newEmail: string, password: string): firebase.Promise<any> {
  const credential = firebase.auth.EmailAuthProvider
    .credential(firebase.auth().currentUser.email, password);

  return firebase.auth().currentUser.reauthenticate(credential).then( user => {
    firebase.auth().currentUser.updateEmail(newEmail).then( user => {
      firebase.database().ref('/users').child(firebase.auth().currentUser.uid)
        .update({ email: newEmail });
    });
  });
}

```

Fig. 6.9 Función de recuperación de correo electrónico

La función de recuperación de contraseña se conecta con Firebase indicando el correo del usuario y este le responde con un mensaje con un enlace donde el usuario puede modificar la contraseña y generar una nueva.

Por otra parte la función de email funciona de manera similar, ya que Firebase recibe el correo antiguo y el nuevo, y este le manda al usuario un mensaje al correo antiguo avisándole de la modificación. Una vez mandado este mensaje, la aplicación recoge la respuesta y almacena la nueva dirección de correo en la base de datos, cosa que no se hace con la contraseña.

6.4 Actividades

En este segundo apartado se presentan las funcionalidades más relevantes relacionadas con las actividades que los usuarios pueden gestionar en la aplicación y la implementación de estas, al igual que con la autenticación se ha generado otro proveedor en la aplicación para las funciones relacionadas con las actividades.

La primera funcionalidad que se presenta es la de creación de actividades. Para poder llevarla a cabo es necesario instalar la librería de GeoFire[30] que permite almacenar coordenadas en Firebase y usarlas para hacer búsquedas basadas en localización. También es necesario instalar la dependencia nativa de localización para poder acceder al gps del dispositivo, la librería de mapas de google[31] y generar un proveedor del estado de la conectividad que usa la dependencia nativa de Red para poder acceder a la información de conectividad del dispositivo.

La función recoge la información de la vista y una vez obtiene los datos comprueba el estado de la conexión mediante el proveedor generado para asegurar que no hay ningún problema para realizar la operación. Si esta da un resultado positivo empieza la creación. El primer paso es generar una ruta definida en la base de datos debajo de actividades encabezada por el identificador único de la actividad. Una vez se ha generado correctamente se genera otra ruta donde se almacenan la coordenadas de esta con la librería GeoFire. Finalmente se añade la actividad al usuario que la ha generado cómo programada. A continuación, se muestran tres imágenes que hacen referencia a las rutas de la base de datos mencionadas.



Fig. 6.10 Ruta de la actividad

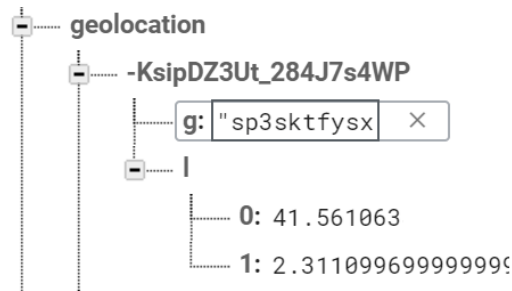


Fig. 6.11 Ruta de las coordenadas

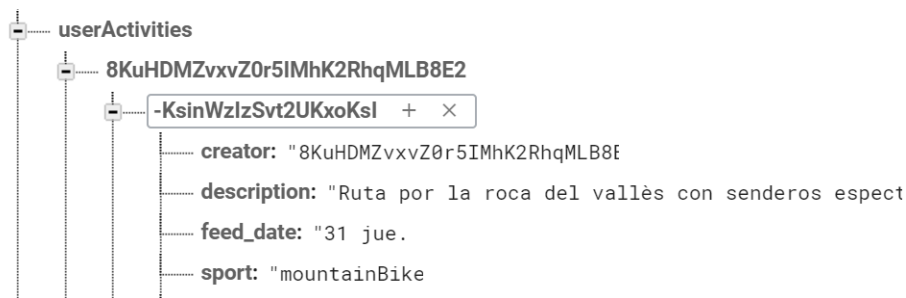


Fig. 6.12 Ruta de las actividades del usuario

Una vez generadas las actividades, la siguiente función que se ha realizado es la que se ocupa de la búsqueda de estas. Esta función utiliza las mismas dependencias que la función anterior pero esta vez para encontrar las actividades.

El funcionamiento es el siguiente, la aplicación obtiene la información del usuario en la pantalla y al igual que antes comprueba la conectividad. A continuación busca las actividades que están dentro del radio establecido por el usuario y las filtra en función del deporte, seleccionado la duración de la actividad la distancia y la fecha.

Esta función se ha generado como un Observable por lo tanto la aplicación está de forma continua observando el vector de datos generado por el filtro y esto permite que si aparecen nuevas actividades el usuario las pueda verlas en la pantalla de su dispositivo sin tener que realizar ninguna acción. En la siguiente imagen se puede ver parte de esta función donde se obtiene la localización del dispositivo y mediante la librería GeoFire se empiezan a buscar las actividades cercanas.

```

Geolocation.getCurrentPosition().then(position =>{
let geoQuery = this.geoFire.query({
  center: [position.coords.latitude, position.coords.longitude],
  radius: parseInt(searchParams.range) //Km Radius
});

geoQuery.on("key_entered", (key, location, distance) => {

```

Fig. 6.13 Parte de la función de búsqueda

Una vez realizada la función de búsqueda las siguientes funciones que se han desarrollado son las que permiten que los usuarios se inscriban a las actividades y salgan de estas.

Para que un usuario pueda inscribirse a una actividad solo es necesario que pulse el botón de inscribirse en el detalle de la aplicación. Entonces se activa la función de inscripción a la actividad, esta incluye al usuario como pendiente dentro de la ruta de participantes de una actividad y la añade como pendiente en las actividades del usuario.



Fig. 6.14 Usuario pendiente en una actividad

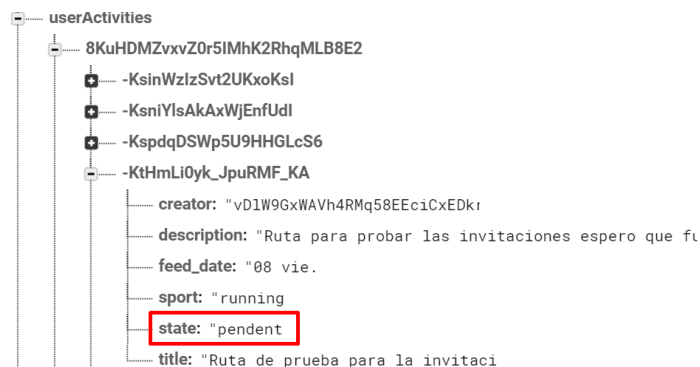


Fig. 6.15 Actividad del usuario con estado pendiente

El usuario que ha creado la actividad puede gestionar los usuarios que se han inscrito a esta y aceptar o no su participación, si el usuario acepta o rechaza la invitación el estado pendiente pasa a aceptado o rechazado.

Una vez un usuario pertenece a la actividad, puede acceder al chat interno que permite comunicarse de forma ágil con los distintos participantes de la misma.

Para desarrollar el chat se ha sacado provecho de la propiedad de base de datos en tiempo real. Ya que nos permite generar una sala de chat sin tener que configurar sockets cómo se haría si se usara un servidor propio.

Cuando un usuario perteneciente al chat genera un mensaje, este se añade a la sala con un identificador único, el texto del mensaje y el nombre del usuario. La aplicación para gestionar la recepción de estos mensajes se conecta con un Observable a la base de datos y cuando aparece un mensaje nuevo se actualiza en pantalla.

```
getActivitieMessages(activitieID): FirebaseListObservable<any>{
  return this.af.database.list(`/activitieMessages/${activitieID}`);
}
```

Fig. 6.16 Función de lectura de mensajes



Fig. 6.17 Sala de chat

Finalmente, cuando un usuario quiere abandonar la actividad existen tres casuísticas, la primera si un usuario es el creador de la actividad y la cierra, las coordenadas de la actividad se eliminan de la base de datos juntamente con la sala de chat y la actividad se elimina de la lista de programadas por el usuario.

La segunda, si el creador elimina la actividad los participantes pueden abandonarla con la opción de votar con un voto positivo al usuario creador. Una vez seleccionan si votan o no el usuario se elimina de la actividad y la actividad de su lista de actividades programadas.

Finalmente si un usuario desea abandonar una actividad se repite la misma casuística que en el caso anterior pero esta vez sin capacidad de voto.

6.5 Usuarios

Una vez vistas las funcionalidades disponibles para las actividades, en este apartado se mostraran el conjunto de funciones para la gestión de usuarios. Igual que en los dos casos anteriores también se ha generado un proveedor en la aplicación para las funciones relativas a los usuarios.

La primera función a tratar es la que permite la modificación de los datos del usuario, para ello es necesario la dependencia nativa de acceso al almacenamiento interno del dispositivo y a la cámara del mismo para poder modificar la imagen del perfil, y el SDK de JavaScript de Firebase para poder alojarla en el CDN del Baas.

Cuando un usuario desea modificar su perfil, puede modificar el nombre y apellido, la descripción, su deporte favorito (que se utiliza en la función de búsqueda como deporte predefinido), y la imagen de perfil.

Para ello una vez el usuario ha rellenado todos los campos con las verificaciones correspondientes, la función comprueba los campos a modificar y los sobrescribe en la base de datos. Si es la imagen la que se ha modificado, esta se procesa y se almacena en el CDN de Firebase en una subcarpeta identificada por el identificador del usuario y a continuación se recibe como respuesta la ruta pertinente que se agrega a la base de datos.

<input type="checkbox"/>	Nombre	Tamaño	Tipo	Última modificación
<input type="checkbox"/>	8KuHDMZvxvZ0r5IMhK2RhgMLB8E2/	-	Carpeta	-
<input type="checkbox"/>	br6g98fZ6WXTXiqJa5jkgxp92B2/	-	Carpeta	-
<input type="checkbox"/>	hrwQwL71FVTZxkuzDHeQgTHxWhV2/	-	Carpeta	-
<input type="checkbox"/>	vDIW9GxWAVh4RMq58EEciCxEDkr1/	-	Carpeta	-
<input type="checkbox"/>	WJZ0UPWqBzLJxJPSM5KUX2yXlu1/	-	Carpeta	-

Fig. 6.18 CDN de Firebase

Otra de las funciones principales referentes a los usuarios es la de seguir usuarios. Para su realización se genera en la base de datos una ruta con los usuarios seguidos por el usuario y otra con los que siguen al usuario donde se almacenan los identificadores de usuario, su nombre de usuario y la imagen de perfil.

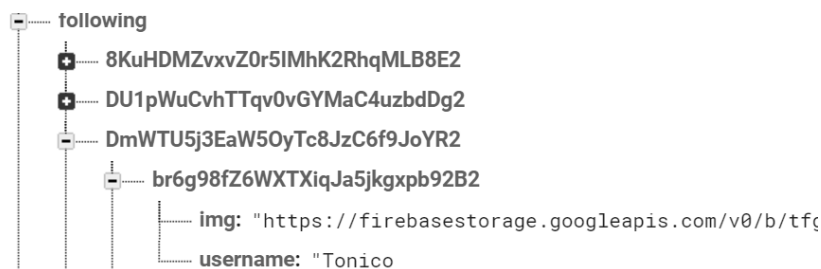


Fig. 6.19 Usuarios seguidos

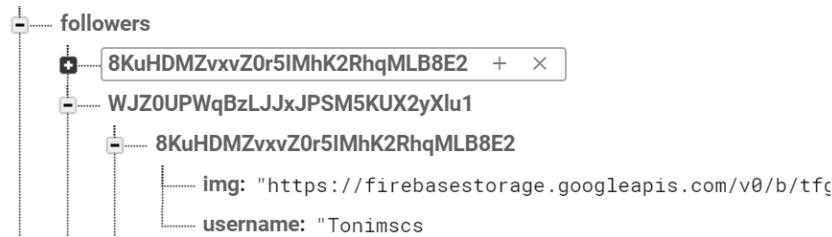


Fig. 6.20 Seguidores del usuario

La finalidad principal de seguir a un usuario es que después este puede invitarte a las actividades creadas por él. La metodología es la misma vista en la inscripción pero con el estado invitado, y esta vez el que tiene que aceptar si participar o no ya no es el creador de la actividad sino el invitado.

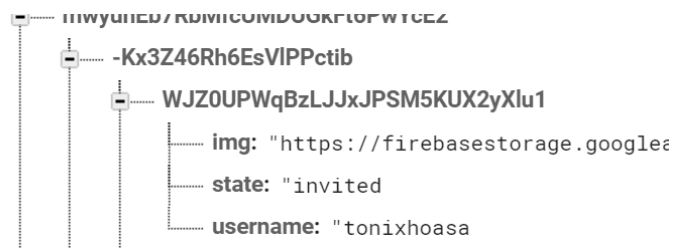


Fig. 6.21

Para poder encontrar a estos usuarios se ha realizado un buscador en función del nombre de usuario. Para ello se hace una función de mapeo a los usuarios y se retornan los que coinciden con las letras que el usuario esté escribiendo en ese momento en el buscador.

```

filterUsersList(searchTerm): any {
  return this.af.database.list(`/users`).map((users) =>{
    return users.filter((user) =>{
      return user.username.toLowerCase().indexOf(searchTerm.toLowerCase()) > -1;
    });
  });
}

```

Fig. 6.22 Función de búsqueda de usuarios

Finalmente la última función en relación con los usuarios es la de comentar en el perfil de otro usuario. Para ello se ha generado una ruta en la base de datos donde se almacenan el identificador del usuario que ha dejado el comentario, el comentario en sí mismo y el nombre de usuario.

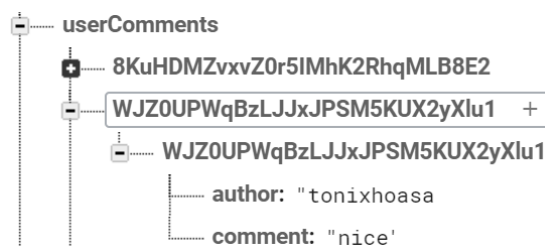


Fig. 6.23 Comentario de un usuario

6.6 Notificaciones

En este apartado se mostrará el proceso de implementación de las notificaciones push con Firebase. Para ello es necesario instalar la dependencia de FCM que nos permitirá gestionar la conexión con los servicios de Google y ver las notificaciones en el dispositivo móvil.

Una vez instalada la dependencia es necesario añadir la aplicación al proyecto de Firebase con el identificador del paquete de esta. Esto nos proporcionará un fichero con la información necesaria para la conexión con los servicios de Google proporcionados por Firebase, como pueden ser las notificaciones push o las analíticas. Este fichero lo tendremos que almacenar en la carpeta raíz de la aplicación para que la dependencia de FCM lo detecte correctamente.

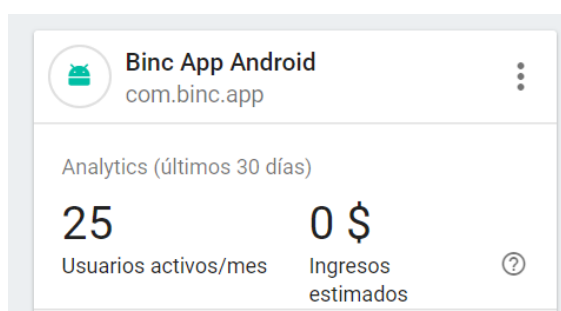


Fig. 6.24 Aplicación en la consola de Firebase

Una vez se ha hecho la configuración podemos proceder a enviar las notificaciones push desde la consola de Firebase. Esta nos permitirá programar la hora y la duración, enviar mensajes personalizados y filtrar los usuarios a los que llegan las notificaciones. Parte de esta consola se muestra en la siguiente imagen:

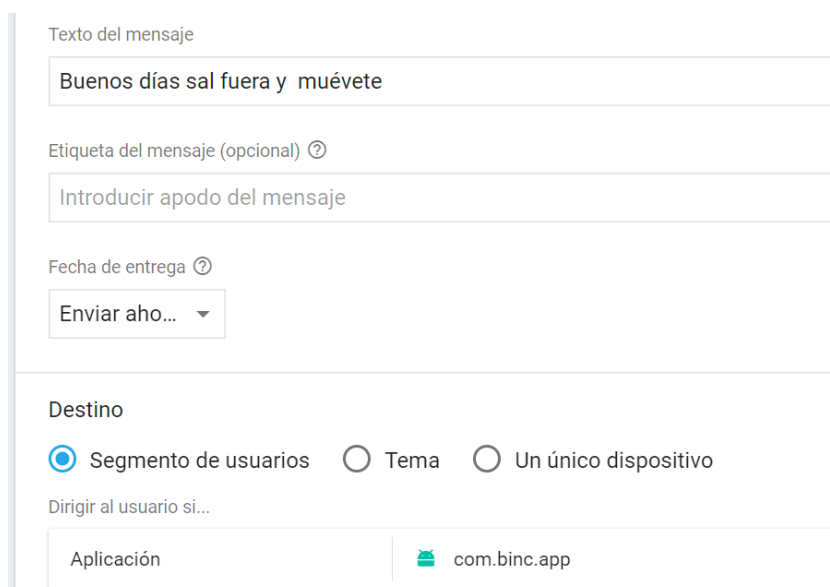
The image shows a screenshot of the notification configuration form in the Firebase console. It includes several input fields and options: 'Texto del mensaje' with the value 'Buenos días sal fuera y muévete'; 'Etiqueta del mensaje (opcional)' with the placeholder 'Introducir apodo del mensaje'; 'Fecha de entrega' with a dropdown menu set to 'Enviar aho...'; 'Destino' with radio buttons for 'Segmento de usuarios' (selected), 'Tema', and 'Un único dispositivo'; and 'Dirigir al usuario si...' with a dropdown menu set to 'Aplicación' and the package ID 'com.binc.app'.

Fig. 6.25 Consola para enviar notificaciones

Una vez mandamos la notificación aparecerá en los dispositivos que cumplan el criterio establecido. A continuación se muestra una captura de pantalla con la notificación recibida en el dispositivo móvil:

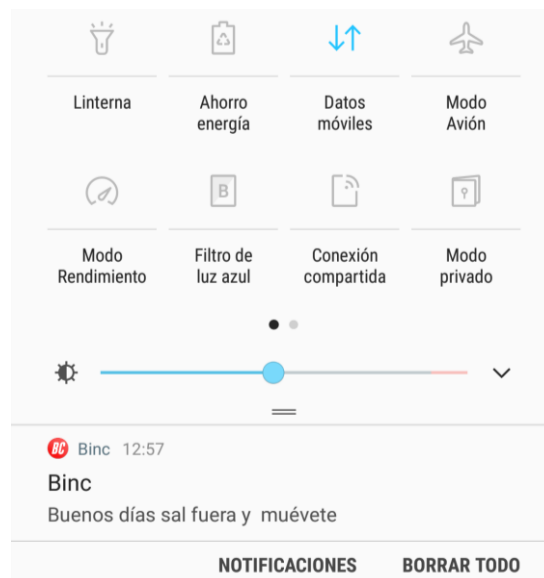


Fig. 6.26 Notificación recibida por el móvil

6.6 Analíticas

Una vez implementadas las notificaciones, se han implementado las analíticas. Para ello también será necesario el uso de la aplicación generada en Firebase, pero esta vez usaremos el fichero descargado en el apartado anterior para configurar el servicio de analíticas.

Con este servicio se puede ver el comportamiento y el uso que le están dando los usuarios a la aplicación. para poder implementarlo es necesario utilizar la dependencia nativa de Firebase Analytics. Esta inicialmente genera conflictos con la dependencia anterior ya que las dos intentan generar un plugin de acceso con el fichero de servicios de Google. Para evitar este conflicto es necesario eliminar la dependencia de Google Services de la dependencia de analíticas en el fichero plugin.xml de este.

```
<!-- <dependency id="cordova-support-google-services" version="~1.0.0"/> -->
```

Fig. 6.27 Dependencia que genera los conflictos

Si se ha realizado correctamente las analíticas empezaran a recolectarse en Firebase. De forma automática genera un conjunto de datos que nos permiten ver la cantidad de usuarios activos, el tiempo medio, las desinstalaciones, las primeras aberturas y más, estos se pueden consultar en la documentación[32].

Nombre del evento ↑	Recuento	↔	Valor	↔	Usuarios	↔	Marcar como conversión
app_remove	24	-	-	-	24	-	<input type="checkbox"/>
app_update	1	-	-	-	1	-	<input type="checkbox"/>
▶ CreateActivitie 1	6	-	-	-	2	-	<input type="checkbox"/>
first_open	25	-	-	-	25	-	<input checked="" type="checkbox"/>
notification_dismiss	2	-	-	-	1	-	<input type="checkbox"/>
notification_foreground	12	-	-	-	9	-	<input type="checkbox"/>
notification_open	4	-	-	-	2	-	<input type="checkbox"/>
notification_receive	18	-	-	-	10	-	<input type="checkbox"/>

Fig. 6.28 Eventos recogidos en Firebase

Además de los eventos automáticos se han creado dos eventos personalizados, el primero se manda al crear una actividad y contiene el deporte de esta, y el segundo cuando se hace una búsqueda de actividades también con el deporte seleccionado. Con estos dos eventos personalizados se podrá saber la cantidad de búsquedas y creaciones de actividad y el deporte preferido de los usuarios.

CAPÍTULO 7. PUESTA A PRODUCCIÓN

En este capítulo se mostrará el proceso necesario para publicar la aplicación a la tienda de aplicaciones de Android una vez desarrollada. Aunque la aplicación esté preparada para empaquetarse también para IOS, se ha decidido lanzarla solo en la tienda de Android debido a que para publicarla en la tienda de Apple se necesita un ordenador con el sistema operativo de la compañía y una licencia de 100 euros.

En cambio para Android nos sirve cualquier ordenador para generar la aplicación y el precio de la licencia es de 25 euros, así que dadas las características y el propósito del proyecto se ha tomado esta decisión.

7.1 Preparación del Backend

Antes de generar la aplicación, se ha creado un nuevo proyecto en Firebase que será el que usará la aplicación de producción. De esta forma evitamos que los datos generados por la aplicación de desarrollo se mezclen con los datos generados por la aplicación subida a la store, a continuación se muestran los dos proyectos en la consola de Firebase.

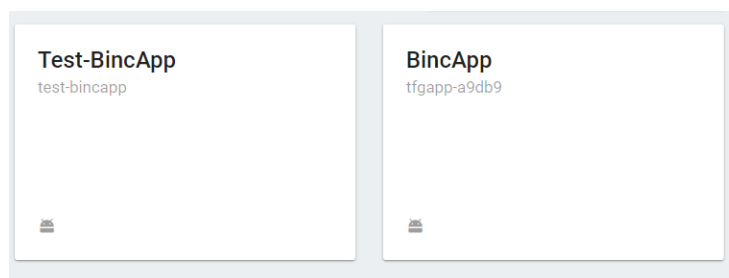


Fig. 7.1 Proyectos de Firebase

Además de los datos también es necesario hacer la migración de la cuenta de analíticas y de notificaciones push con el nuevo fichero de servicios de Google generado.

Además de esto también será necesario actualizar los permisos de la aplicación generada en Facebook para que esté disponible por todos los usuarios y no solo los administradores.

7.2 Preparación de la aplicación para producción

Una vez desarrollada la aplicación y actualizada con los datos del nuevo proyecto de Firebase, hay que prepararla para poder subirla a la tienda de aplicaciones. A continuación se muestran los pasos que se han seguido para ello.

Primero se ha eliminado el plugin de Cordova para la consola, una vez hecho esto se ha generado el apk con la siguiente comanda:

```
$ ionic build android --release --prod
```

Fig. 7.2 Comanda para generar el Apk

Con esto se consigue generar una aplicación para producción sin firmar, que es el paso que se debe aplicar a continuación con la herramienta keytool del JDK de Java.

```
$ jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore my-release-key.keystore android-release-unsigned.apk alias_name
```

Fig. 7.3 Comanda para firmar el apk

Una vez firmada la aplicación es necesario comprimirla mediante la herramienta Zipalign del SDK de Android.

```
$ zipalign -v 4 android-release-unsigned.apk androidApp.apk
```

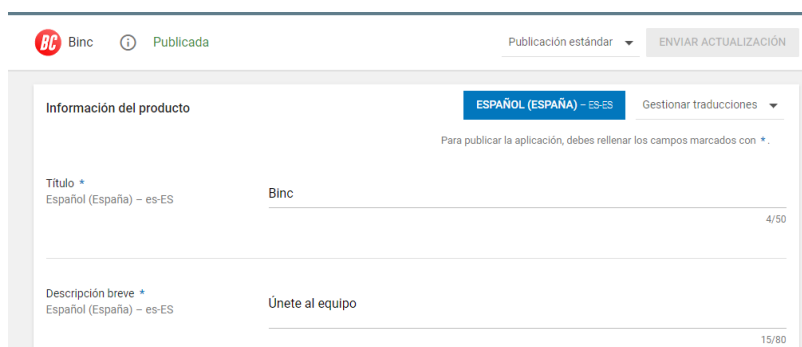
Fig. 7.4 Comanda para comprimir la aplicación

Mediante este conjunto de comandas se consigue generar un apk listo para producción. El último factor a tener en cuenta es actualizar los hashes de clave de la aplicación de Facebook, ya que al firmar la aplicación habrá que añadir uno nuevo en base a la nueva firma para que funcione correctamente.

7.3 Subida a la tienda de aplicaciones

Con la aplicación generada y el backend actualizado se ha procedido a subir la aplicación a la tienda de aplicaciones de Google, para ello es necesario disponer de un cuenta de Google y pagar la licencia de desarrollador de 25 euros, con esto ya se podrá empezar a subir la aplicación.

El primer paso que se ha realizado ha sido el de rellenar el formulario con toda la información descriptiva necesaria para que la aplicación esté disponible en la tienda. S continuación se muestra una imagen con parte de dicho formulario:



The screenshot shows the 'Información del producto' (Product Information) section of the Google Play Store developer console. At the top, it displays the app name 'Binc' and its status as 'Publicada' (Published). There are buttons for 'Publicación estándar' (Standard Publication) and 'ENVIAR ACTUALIZACIÓN' (SEND UPDATE). The language is set to 'ESPAÑOL (ESPAÑA) - ES-ES'. A note states: 'Para publicar la aplicación, debes rellenar los campos marcados con *.' (To publish the application, you must fill in the fields marked with *). The form contains two required fields: 'Título' (Title) with the value 'Binc' and a character count of 4/50; and 'Descripción breve' (Short description) with the value 'Únete al equipo' and a character count of 15/80.

Fig. 7.5 Formulario de información de la aplicación

Aparte de la información que se puede apreciar en la imagen también hay que añadir las imágenes necesarias para visualizarse en la tienda y contestar otro formulario para determinar la clasificación de contenido. En este caso al ser una red social la puntuación obtenida es de mayores de doce años.

Con el formulario creado se puede proceder a la distribución de la aplicación, para el proyecto se ha decidido publicar la aplicación en modo beta pública con límite de mil descargas y solamente para la tienda de España. Esto hace que los usuarios tengan que formar parte del programa de betas de Google o disponer de una invitación para poder obtener la aplicación.

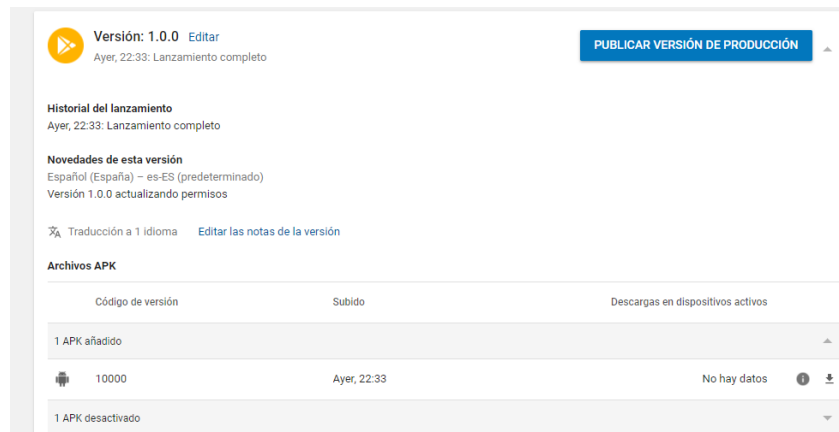


Fig. 7.6 Versión de la aplicación publicada

De esta manera se consigue un control mayor sobre el uso de la aplicación ya que al estar en una fase tan inicial no es recomendable publicarla de forma global.

Una vez publicada, Google hace un test de la aplicación durante unas cuatro horas antes de que esta sea pública. A continuación se muestran los resultados principales de este test:

Device model	Avg. CPU (Percent) ⓘ	Avg. network sent (Bytes/Sec) ⓘ	Avg. network received (Bytes/Sec) ⓘ	Avg. memory (Bytes) ⓘ	Startup time (ms) ⓘ
Galaxy J1 Ace ⓘ	18,71 %	519	1,2 K	29 M	0
Mate 9 ⓘ	3,76 %	1,3 K	4,1 K	168 M	381
Galaxy S7 Edge ⓘ	2,16 %	816	1,8 K	171 M	741
Galaxy J7(2016) ⓘ	2,27 %	450	1,5 K	83 M	882
LG G6 ⓘ	5,92 %	784	2,3 K	183 M	865
Pixel ⓘ	5,96 %	1 K	2,4 K	156 M	608
Pixel ⓘ	3,28 %	855	2,6 K	137 M	843
Xperia XZ Premium ⓘ	1,50 %	87	181	144 M	642
P8 Lite ⓘ	2,90 %	332	1,3 K	55 M	1,5 K
Moto G4 Play ⓘ	9,09 %	1,2 K	1,8 K	91 M	1,1 K

Fig. 7.7 Dispositivos probados en el test de google

Con esto realizado, la aplicación se empieza a mostrar en la tienda de aplicaciones.

CAPÍTULO 8. PLANIFICACIÓN DEL TRABAJO

En este capítulo se muestra como se ha realizado la planificación para llevar a cabo el proyecto.

El conjunto de tareas realizadas se pueden dividir en cinco grandes bloques, que son los siguientes:

Estudio previo: Durante esta fase se hizo la investigación y prueba del conjunto de plataformas de desarrollo móvil existentes, la investigación de los backend as a service más afines a las particularidades del proyecto y la investigación del estado actual del mercado de aplicaciones móviles deportivas.

Diseño: esta fase sirvió para definir el diseño conceptual de la aplicación y para investigar los conceptos utilizados de experiencia de usuario.

Desarrollo: La fase de desarrollo incluye toda la programación de la aplicación de inicio al fin.

Publicación a la tienda: Aquí se contemplan el conjunto de horas necesarias para preparar la aplicación para producción y para aprender a subir la aplicación a la tienda hasta realizarlo.

Documentación: La parte de documentación hace referencia a las horas invertidas para realizar la memoria del trabajo.

A continuación se presenta una tabla con la planificación del trabajo:

Tabla 8.1. Planificación del trabajo

Bloque de tareas	Estimación del tiempo en días
Estudio previo	21
Diseño	4
Desarrollo	82
Publicación a la tienda	3
Documentación	15
Total	125

Las fases de estudio previo, diseño y publicación a la tienda se hicieron de forma independiente, las fases de desarrollo y documentación se han ido intercalando en el tiempo de creación del proyecto.

CAPÍTULO 9. CONCLUSIONES

En este último capítulo se comentarán las conclusiones finales del proyecto: hablando de los objetivos conseguidos, las conclusiones personales y las posibles implementaciones realizables en un futuro.

9.1 Objetivos conseguidos

En el inicio de esta memoria se han definido cinco objetivos a cumplir con el desarrollo de la aplicación y se han definido las distintas funcionalidades. La consecución de estos objetivos ha sido desarrollar una aplicación móvil de principio a fin utilizando un servicio como servidor.

El propósito principal que se ha querido conseguir es la generación de una aplicación que permita a los usuarios conocer gente nueva con la que practicar deporte y así incentivar los principios de una vida activa. Esto se ha conseguido con la creación de una red social que permite a la gente generar actividades deportivas al aire libre y que otros las encuentren para poder participar en ellas.

Aunque al principio esto pudiera parecer muy costoso de desarrollar, usando las tecnologías de una aplicación híbrida y un backend as a service se ha obtenido una velocidad de desarrollo mucho mayor que con un servidor tradicional. Desarrollando todas las partes del proyecto de forma independiente hubiera sido realmente difícil de cumplir en el plazo establecido.

Además se han conseguido desarrollar todas las funciones definidas para que los usuarios puedan utilizar la aplicación cumpliendo con el propósito principal descrito.

Finalmente se ha publicado la aplicación en modo beta a la tienda de aplicaciones de Google. Aunque hubiera estado bien publicarla también en la tienda de Apple, las limitaciones del ecosistema de la compañía han dificultado que esto se realizara.

9.2 Conclusiones personales

La realización de este trabajo me ha aportado experiencia en el desarrollo de aplicaciones multiplataforma desde el inicio del desarrollo hasta la publicación de estas. Aunque Ionic era un framework que ya conocía en su versión uno he obtenido un conocimiento mucho más profundo del framework en la segunda versión y del lenguaje Typescript que era la primera vez que utilizaba.

Utilizar Firebase como Baas me ha permitido descubrir una forma de desarrollar completamente distinta a la que había visto hasta ahora y ver que actualmente existen muchas más posibilidades para la arquitectura del desarrollo de una aplicación.

El resultado final de la combinación de ambas tecnologías ha resultado ser sorprendentemente ágil. Eso sí con limitaciones a las cuales se ha adaptado el proyecto.

Finalmente, el hecho de realizar una aplicación de principio a fin me ha aportado una visión mucho más amplia de todos los factores a tener en cuenta cuando se quiere desarrollar para el mundo real y de su complejidad.

9.2 Futuras implementaciones

Los objetivos principales de este proyecto se han cumplido, aun así si la aplicación se decidiera mantener en el tiempo y escalar a un público general fuera de la beta desarrollada, sería recomendable realizar las siguientes mejoras:

Realizar y tramitar una política de uso, cumpliendo con la normativa del reglamento general de protección de datos aplicable a partir del 25 de mayo de 2018.

Optimizar la función de búsqueda de la aplicación con la nueva estructura de base de datos de Firebase, que facilita las consultas. Esta actualmente se encuentra en versión alfa.

Modificar el plan de Firebase gratuito a uno de pago para escalar la aplicación de forma correcta a medida que aumentan los usuarios.

Con estas mejoras la aplicación estaría preparada para poder funcionar de forma abierta en España.

BIBLIOGRAFÍA

- [1] **Documento de la Organización Mundial de la Salud**, Recomendaciones mundiales sobre la actividad física.(en línea).[Última consulta: 15 de septiembre del 2017]. URL:
http://apps.who.int/iris/bitstream/10665/44441/1/9789243599977_spa.pdf
- [2] **Datos sobre la insuficiencia de actividad física** en adultos por continente de la Organización mundial de la salud.(en línea).[Última consulta: 19 de septiembre del 2017]. URL:
<http://apps.who.int/gho/data/node.main.A893?lang=en>
- [3] **Plan de acción global de la organización mundial de la salud**, para las enfermedades no transmisibles. (en línea).[Última consulta: 15 de septiembre del 2017]. URL:
http://apps.who.int/iris/bitstream/10665/910%4384/1/9789241506236_en_g.pdf?ua=1
- [4] **Página oficial de Strava**. (en línea).[Última consulta: 17 de septiembre del 2017]. URL:
<https://www.strava.com/dashboard>
- [5] **Página oficial de Meetup**. (en línea).[Última consulta: 17 de septiembre del 2017]. URL:
<https://www.meetup.com/es-ES/>
- [6] **Página oficial de Runnator**. (en línea).[Última consulta: 17 de septiembre del 2017]. URL:
<https://www.runator.com/es>
- [7] **Página oficial de Timpick**. (en línea).[Última consulta: 17 de septiembre del 2017]. URL:
<http://www.timpik.com/>
- [8] **Página oficial de Decathlon sport meeting**. (en línea).[Última consulta: 17 de septiembre del 2017]. URL:
<https://sportmeeting.decathlon.com/>
- [9] **Anuario de estadísticas deportivas**. Informe del ministerio de cultura y deporte con los resultados estadísticos relacionados con la práctica de deporte (en línea). [Última consulta: 17 de septiembre del 2017]. URL:
https://www.mecd.gob.es/servicios-al-ciudadano-mecd/dms/mecd/servicios-al-ciudadano-mecd/estadisticas/deporte/anuario-deporte/AED-2017/Graficos_Anuario_de_Estadisticas_Deportivas_2017.pdf

- [10] **Informe Mobile en España y el mundo 2017**. Informe realizado por ditrendia sobre el uso del móvil tanto en España como en el mundo (en línea). [Última consulta: 17 de septiembre del 2017]. URL: http://www.amic.media/media/files/file_352_1289.pdf
- [11] **Definición de Personas**. Concepto utilizado en diseño centrado en el usuario (en línea). [Última consulta: 17 de septiembre del 2017]. URL: [https://es.wikipedia.org/wiki/Persona_\(experiencia_de_usuario\)](https://es.wikipedia.org/wiki/Persona_(experiencia_de_usuario))
- [12] **Disponibilidad del nombre Binc**. Web para comprobar la disponibilidad del nombre de la aplicación (en línea). [Última consulta: 17 de septiembre del 2017]. URL: <https://www.namecheck.com/search?searchterm=binc>
- [13] **Psicología del color**. Teoría del color Eva Heller (en línea). [Última consulta: 17 de septiembre del 2017]. URL: <https://www.namecheck.com/search?searchterm=binc>
- [14] **Tipografía utilizada**. Web de dafont fuente de la tipografía gobold: [Última consulta: 17 de septiembre del 2017]. URL: <http://www.dafont.com/es/gobold.font>
- [15] **Guía de estilos**. Guía de estilos material design elaborada por google: [Última consulta: 17 de septiembre del 2017]. URL: <https://material.io/guidelines/style/icons.html#>
- [16] **Api de inicio de sesión** Api de inicio de sesión de Facebook: [Última consulta: 17 de septiembre del 2017]. URL: <https://developers.facebook.com/docs/facebook-login>
- [17] **Xamarin** Página principal de la plataforma de desarrollo móvil xamarin: [Última consulta: 25 de septiembre del 2017]. URL: <https://www.xamarin.com/platform>
- [18] **React Native** Página principal de la plataforma de desarrollo móvil react native: [Última consulta: 25 de septiembre del 2017]. URL: <https://facebook.github.io/react-native/>
- [19] **Flutter** Página principal de la plataforma de desarrollo móvil flutter: [Última consulta: 25 de septiembre del 2017]. URL: <https://flutter.io/>
- [20] **Ionic** Página principal del framework de desarrollo móvil ionic: [Última consulta: 25 de septiembre del 2017]. URL: <http://ionicframework.com/>
- [21] **Firebase** Página principal del servicio Baas firebase: [Última consulta: 30 de septiembre del 2017]. URL: <https://firebase.google.com/products/>

- [22] **Back4App** Página principal del servicio Baas: [Última consulta: 30 de septiembre del 2017]. URL:
<https://www.back4app.com/pricing>
- [23] **Parse server** Página principal del backend open source: [Última consulta: 30 de septiembre del 2017]. URL:
<http://parseplatform.org/>
- [24] **Kinvey** Página principal de la plataforma de servicios: [Última consulta: 30 de septiembre del 2017]. URL:
<https://www.kinvey.com/platform/>
- [25] **Node** Página principal denode: [Última consulta: 10 de octubre del 2017]. URL:
<https://nodejs.org/es/>
- [26] **Cordova** Página principal de cordova: [Última consulta: 10 de octubre del 2017]. URL:
<https://cordova.apache.org/>
- [27] **Typescript** Página principal typescript: [Última consulta: 10 de octubre del 2017]. URL:
<https://www.typescriptlang.org/>
- [28] **AngularFire** Página de github de la librería AngularFire: [Última consulta: 15 de octubre del 2017]. URL:
<https://github.com/angular/angularfire2>
- [29] **Ionic Native** Página de documentación oficial: [Última consulta: 15 de octubre del 2017]. URL:
<https://ionicframework.com/docs/native/>
- [30] **GeoFire** Repositorio oficial de la librería: [Última consulta: 15 de octubre del 2017]. URL:
<https://github.com/firebase/geofire-js>
- [31] **Api de Google Maps**: Página oficial: [Última consulta: 15 de octubre del 2017]. URL:
<https://developers.google.com/maps/?hl=es-419>
- [32] **Firebase Analytics**: Lista de eventos recolectados automáticamente: [Última consulta: 15 de octubre del 2017]. URL:
<https://support.google.com/firebase/answer/6317485?hl=en>
- [33] **Binc**: página de la aplicación en la tienda de android: [Última consulta: 22 de octubre del 2017]. URL:
<https://play.google.com/store/apps/details?id=com.binc.app&hl=es>



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

ANNEXOS

TÍTULO DEL TFG: Aplicación de quedadas para realizar actividades deportivas

TITULACIÓN: Grado en Ingeniería Telemática

AUTOR: Antoni Mas Canet

DIRECTOR: Antoni Oller Arcas

FECHA: 23 de octubre del 2017

ANNEXO I: WIREFRAMES

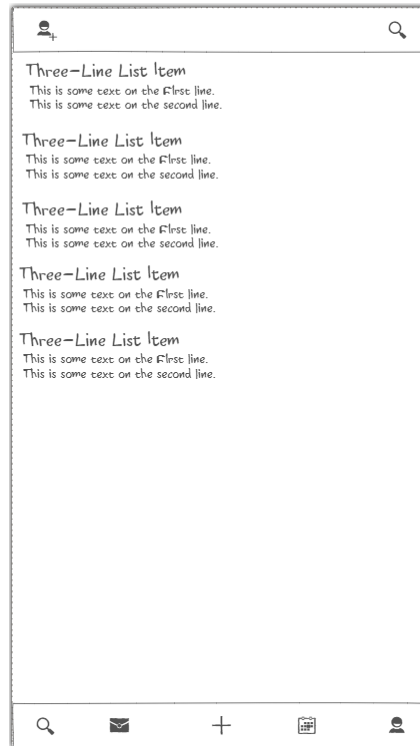


Fig. I.1 Wireframe de la página principal

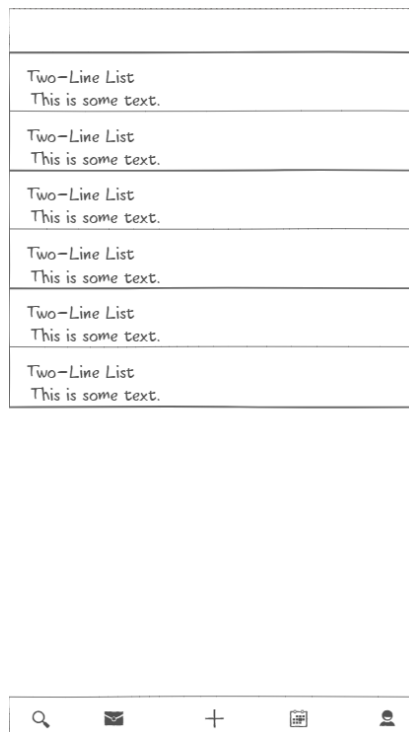


Fig. I.2 Wireframe de la página de invitaciones

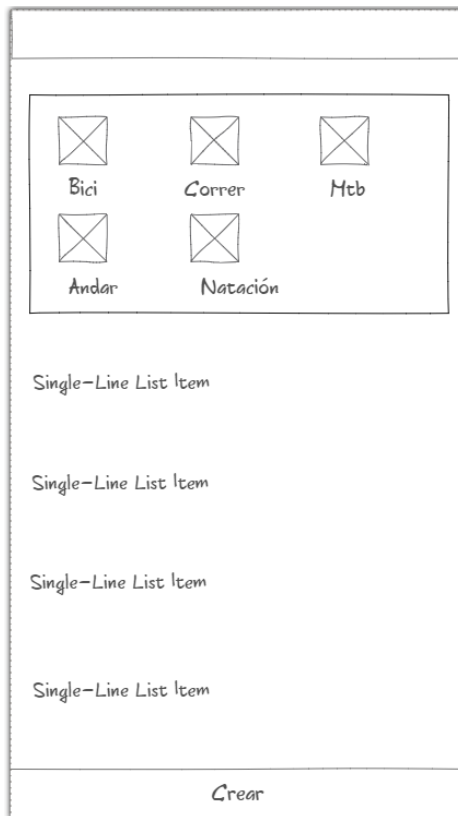


Fig. I.3 Wireframe de la página de creación de actividad



Fig. I.4 Wireframe de la página de actividades programadas

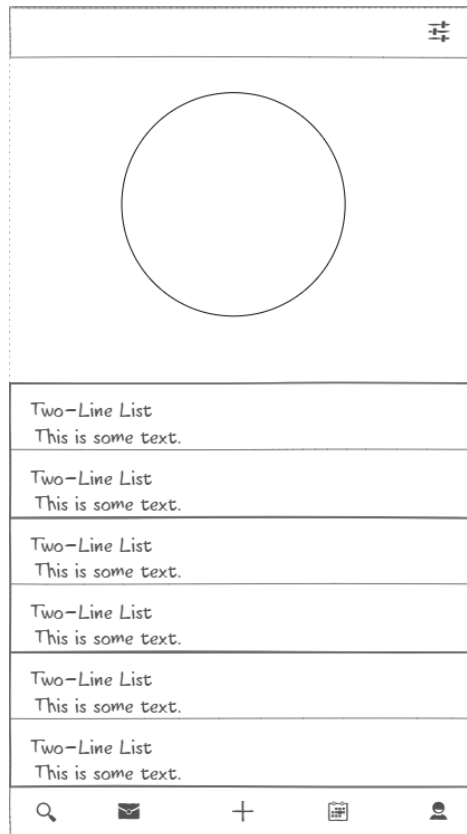


Fig. I.5 Wireframe del perfil de usuario

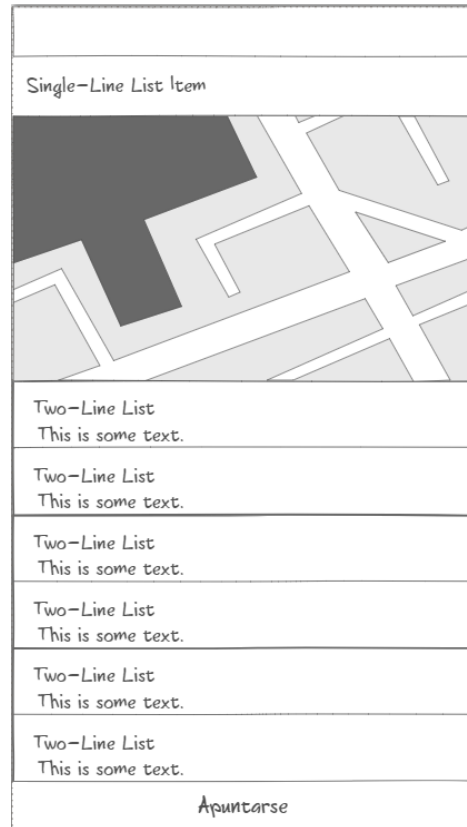


Fig. I.6 Wireframe del detalle de una actividad

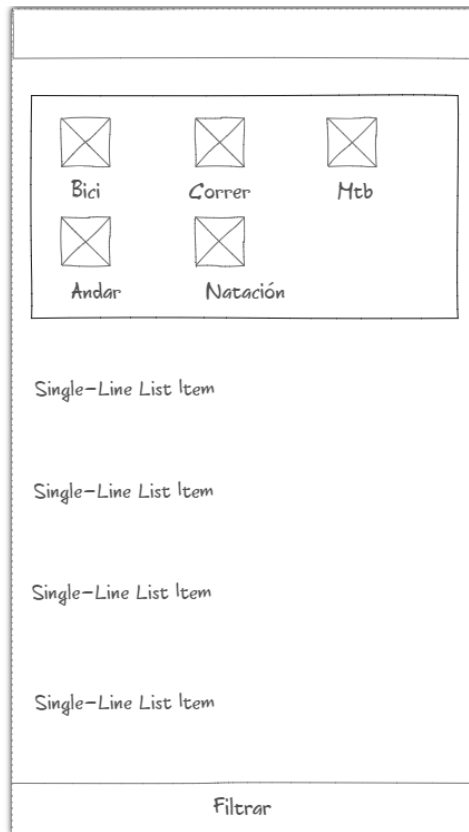


Fig. I.7 Wireframe del buscador de actividades



Fig. I.8 Wireframe del buscador de usuarios