

Desarrollo de una aplicación móvil iOS para la búsqueda de lugares y actividades al aire libre en la ciudad de Barranquilla: Quillair

Valery Patrizia Madiedo Gómez

Máster universitario de Desarrollo de Aplicaciones para Dispositivos Móviles
Desarrollo de Aplicaciones para dispositivos iOS

Pau Dominkovics Coll

Carles Garrigues Olivella

27/12/2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivad a [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Desarrollo de una aplicación móvil iOS para la búsqueda de lugares y actividades al aire libre en la ciudad de Barranquilla: Quillair.</i>
Nombre del autor:	<i>Valery Patrizia Madiedo Gómez</i>
Nombre del consultor/a:	<i>Pau Dominkovics Coll</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega (mm/aaaa):	12/2021
Titulación:	<i>Máster universitario de Desarrollo de Aplicaciones para Dispositivos Móviles</i>
Área del Trabajo Final:	<i>Desarrollo de Aplicaciones para dispositivos iOS</i>
Idioma del trabajo:	Castellano
Palabras clave	iOS, aire libre, actividades
Resumen del trabajo (máximo 250 palabras): <i>Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.</i>	
<p>Para la realización de este trabajo final se decidió realizar el desarrollo de una aplicación móvil para dispositivos iOS llamada Quillair, la cual servirá para descubrir lugares y actividades al aire libre en que la comunidad quiera compartir.</p> <p>Esta aplicación va dirigida principalmente a un público juvenil o adulto, que les guste hacer ejercicio afuera, sean amantes de la naturaleza o simplemente disfrutan de un picnic en el parque. La aplicación está pensada para las personas residen en la ciudad de Barranquilla, aunque también puede utilizarse desde otras ciudades.</p> <p>En la presente memoria se describe todo el proceso para el desarrollo de la aplicación, mostrando los diseños y prototipos, hasta la implementación final. Esta aplicación se realizará para dispositivos iOS, utilizando XCode 12 y Swift 5. También incorpora librerías como Firebase, core location y mapkit con arquitectura VIP y principios SOLID.</p> <p>El resultado es una aplicación funcional, escalable y sostenible en la cual las personas pueden registrarse, ingresar, ver y agregar lugares y eventos, y compartirlos.</p>	

Abstract (in English, 250 words or less):

To carry out this final work, it was decided to develop a mobile application for iOS devices called Quillair, which will serve to discover places and outdoor activities in which the community wants to share.

This application is aimed mainly at a youth or adult audience, who like to exercise outside, are nature lovers or simply enjoy a picnic in the park. The application is designed for people residing in the city of Barranquilla, although it can also be used from other cities.

In this report the entire process for the development of the application is described, showing the designs and prototypes, until the final implementation. This application will be made for iOS devices, using XCode 12 and Swift 5. It also incorporates libraries such as Firebase, core location and mapkit with VIP architecture and SOLID principles.

The result is a functional, scalable and sustainable application in which people can register, enter, view and add places and events, and share them.

ÍNDICE

1. INTRODUCCIÓN	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	3
1.2.2 Requerimientos Funcionales	3
1.2.3 Requerimientos No Funcionales	5
1.3 Enfoque y método seguido	5
1.3.1 Arquitecturas y prácticas de desarrollo	5
1.3.2 Ciclo de vida del proyecto	6
1.3.3 Marco de trabajo	7
1.4 Planificación del Trabajo	7
1.4.1 Recursos de hardware y software	7
1.4.2 Desglose de Tareas	8
1.4.3 Estimación de tareas	8
1.4.4 Estimación de tiempos	9
1.5 Breve resumen de productos obtenidos	10
1.6 Breve descripción de los otros capítulos de la memoria	10
2. DISEÑO Y ANÁLISIS	11
2.1 Investigación y análisis	11
2.1.1 Encuesta	11
2.1.2 Entrevista	17
2.2 Diseño conceptual	18
2.3 Prototipo de la aplicación	20
2.3.1 Prototipo de baja fidelidad	21
2.3.2 Prototipo de alta fidelidad	22
2.4 Árbol de navegación	33
2.5 Evaluación	33
2.6 Diseño técnico	34
2.6.1 Casos de Uso	34
2.6.2 Arquitectura	41
2.6.3 Diagrama de base de datos	44
3. IMPLEMENTACIÓN	46
3.1 Organización de tareas	46
3.2 Cambios o modificaciones	46
3.2.1 Sobre el mínimo producto viable	46
3.2.2 Sobre el backend	47

3.2.3 Sobre elementos de las vistas	48
3.2.4 Sobre las consultas de búsqueda a la base de datos	48
3.2.5 Sobre la planeación, tiempos de desarrollo y otros problemas	49
3.3 Herramientas	49
3.3.1 IDE y lenguaje	49
3.3.2 Repositorios	49
3.3.3 Librerías	49
3.4 Desarrollo	50
3.4.1 Estructura de ficheros	50
3.5 Pruebas	51
3.6 Despliegue y distribución	51
3.7 Ejecución del código fuente	52
4. CONCLUSIONES	53
5. GLOSARIO	54
6. BIBLIOGRAFÍA	55
7. ANEXOS	57
7.1 Transcripciones des las entrevistas	57

LISTA DE FIGURAS

<i>Figura 1. SOLID principios.....</i>	<i>5</i>
<i>Figura 2. VIP Architecture.....</i>	<i>6</i>
<i>Figura 3. Metodología en espiral.....</i>	<i>7</i>
<i>Figura 4. Diagrama de Gantt. Fuente propia.....</i>	<i>10</i>
<i>Figura 5. pregunta 1 encuesta. Fuente propia.....</i>	<i>11</i>
<i>Figura 6. pregunta 2 encuesta. Fuente propia.....</i>	<i>12</i>
<i>Figura 7. pregunta 3 encuesta. Fuente propia.....</i>	<i>12</i>
<i>Figura 8. pregunta 4 encuesta. Fuente propia.....</i>	<i>13</i>
<i>Figura 9. pregunta 5 encuesta. Fuente propia.....</i>	<i>13</i>
<i>Figura 10. pregunta 6 encuesta. Fuente propia.....</i>	<i>14</i>
<i>Figura 11. pregunta 7 encuesta. Fuente propia.....</i>	<i>14</i>
<i>Figura 12. pregunta 8 encuesta. Fuente propia.....</i>	<i>15</i>
<i>Figura 13 pregunta 9 encuesta. Fuente propia.....</i>	<i>15</i>
<i>Figura 14. pregunta 10 encuesta. Fuente propia.....</i>	<i>16</i>
<i>Figura 15. pregunta 11 encuesta. Fuente propia.....</i>	<i>16</i>
<i>Figura 16. pregunta 12 encuesta. Fuente propia.....</i>	<i>17</i>
<i>Figura 17. Boceto app. Fuente propia.....</i>	<i>21</i>
<i>Figura 18. Login. Fuente propia.....</i>	<i>22</i>
<i>Figura 19. Registro. Fuente propia.....</i>	<i>23</i>
<i>Figura 20. Permisos. Fuente propia.....</i>	<i>24</i>
<i>Figura 21. Intereses. Fuente propia.....</i>	<i>25</i>
<i>Figura 22. Explorar actividades. Fuente propia.....</i>	<i>26</i>
<i>Figura 23. Detalle de actividad y compartir actividad. Fuente propia.....</i>	<i>27</i>
<i>Figura 24. Explorar lugares. Fuente propia.....</i>	<i>28</i>
<i>Figura 25. Detalle del lugar. Fuente propia.....</i>	<i>29</i>
<i>Figura 26. Crear evento. Fuente propia.....</i>	<i>30</i>
<i>Figura 27. Mis eventos. Fuente propia.....</i>	<i>31</i>
<i>Figura 28. Mi perfil. Fuente propia.....</i>	<i>32</i>
<i>Figura 29. Árbol de navegación. Fuente propia.....</i>	<i>33</i>
<i>Figura 30. Gestión de usuario. Fuente propia.....</i>	<i>34</i>
<i>Figura 31. Creación de eventos. Fuente propia.....</i>	<i>37</i>
<i>Figura 32. Creación de eventos. Fuente propia.....</i>	<i>40</i>
<i>Figura 33. Arquitectura de servicios. Fuente propia.....</i>	<i>42</i>
<i>Figura 34. Arquitectura de la escena (VIP). Fuente propia.....</i>	<i>44</i>
<i>Figura 35. Base de datos SQL. Fuente propia.....</i>	<i>45</i>
<i>Figura 36. Vista de añadir este lugar. Fuente propia.....</i>	<i>47</i>
<i>Figura 37. Base de datos noSQL. Fuente propia.....</i>	<i>48</i>
<i>Figura 38. Antes y después del diseño de ítems. Fuente propia.....</i>	<i>48</i>
<i>Figura 39. Estructura de ficheros. Fuente propia.....</i>	<i>51</i>

1. INTRODUCCIÓN

1.1 Contexto y justificación del Trabajo

El presente proyecto describe un Trabajo Final de Máster del área de desarrollo de aplicaciones para dispositivos móviles.

Este proyecto surge teniendo en cuenta el contexto actual de la pandemia. Después de afrontar en el año 2020 y a principios de 2021 largos periodos de encierro decretados por el gobierno[1], se hace evidente la falta de actividad física y la necesidad de recreación por la clausura de todos los espacios donde anteriormente se reunían grandes cantidades de personas y que se utilizaban para cubrir estas necesidades, tales como parques, cines, centros comerciales, academias, gimnasios y coliseos.

Al reabrir espacios públicos[2] y brindar nuevamente la posibilidad de salir a las personas, gracias a los avances en la vacunación en el país [3], se crea una necesidad personal de buscar distintos lugares al aire libre, donde se pudiera respirar libremente y realizar actividades deportivas, o de esparcimiento también con el fin de reconectarse con otras personas.

En la actualidad se pueden encontrar algunos lugares en aplicaciones de mapas como google maps pero no existe una guía unificada donde las personas puedan encontrar lugares y además crear un punto de encuentro. Por lo tanto, como resultado de este proyecto se obtendrá una aplicación para dispositivos iOS donde se pueda consultar actividades y lugares al aire libre en la ciudad de Barranquilla.

1.1.1 Benchmarking

En el ecosistema iOS podemos encontrar aplicaciones de mapas y planeación de viajes, o actividades que podrían funcionar en conjunto como producto sustituto al que se propone. Por lo tanto se describen en esta sección las debilidades y fortalezas de algunas de estas aplicaciones para tener en cuenta qué funcionalidades serían más importantes para la aplicación.

a. *Google Maps* [4]

Aplicación móvil de mapas y direcciones de Google.

Fortalezas

- Brinda direcciones, rutas y formas de llegar a los lugares.
- Tiene sistema de calificación y comentarios.
- Contiene vista de la calle.
- Tiene búsqueda de lugar.

- Tiene filtros por zona o categoría del lugar.
- Tiene ubicación en vivo.

Debilidades

- Muestra horarios desactualizados.
- En ocasiones la ruta establecida lleva a sitios desconocidos.
- No muestra fácilmente actividades al aire libre cercanas.
- No es específica para la ubicación de actividades y lugares abiertos.
- No tiene recomendaciones de lugares y actividades.
- No se puede agregar puntos de encuentro con amigos.

b. Mapas [5]

Aplicación móvil de mapas y direcciones de Apple.

Fortalezas

- Brinda direcciones, rutas y formas de llegar a los lugares.
- Tiene búsqueda de lugar.
- Tiene ubicación en vivo.

Debilidades

- No es útil en Barranquilla porque el mapa no está actualizado.

c. TripAdvisor [6]

Es una aplicación para planificar viajes y actividades turísticas.

Fortalezas

- Muestra restaurantes, y lugares cercanos.
- Recomienda viajes y actividades turísticas.
- Se pueden ver precios de ofertas de hoteles y planes de viaje.
- Se pueden reservar hoteles.
- Se pueden calificar los lugares y ver opiniones del público general.
- Se pueden ver precios de restaurantes, horarios e información detallada.

Debilidades

- Se pueden buscar actividades como “patinar” pero el resultado no está enfocado en la ciudad de Barranquilla.
- No tiene forma de crear puntos de encuentro.
- Está enfocada en actividades turísticas y en recomendaciones de viajes.

d. Meetup [7]

Es una aplicación para crear eventos y puntos de encuentro

Fortalezas

- Es bastante conocida.
- Tiene chat.
- Se pueden crear eventos.
- Tiene posts de la comunidad
- tiene calendario

Debilidades

- Aunque se pueden crear eventos de todo tipo se puede percibir un enfoque más laboral y serio

Podríamos también considerar aplicaciones como Facebook, instagram o whatsapp como sustitutos, ya que a través de estas se pueden crear grupos, compartir cualquier tipo de información y hablar por medio del chat con otras personas para encontrar actividades o lugares al aire libre.

Se podría decir que aunque existan aplicaciones, que combinando su uso, sirvan para encontrar lugares o actividades al aire libre cercanas, no hay una aplicación especializada en brindar este servicio específicamente y tal vez cuando el usuario piense en lo mencionado, no se tenga una opción específica, sino que puede variar entre chats y otras redes sociales.

1.2 Objetivos del Trabajo

El objetivo principal de este trabajo es desarrollar un mínimo producto viable de la aplicación Quillair, la cual muestre lugares y actividades al aire libre, y sirva de punto de encuentro para los ciudadanos de la ciudad de Barranquilla, Colombia.

1.2.1 Objetivos específicos

Se definen los siguientes objetivos teniendo en cuenta las acciones que el usuario podrá realizar dentro de la aplicación:

- O1: Registrarse en la aplicación.
- O2: Iniciar sesión en la aplicación.
- O3: Ver recomendaciones de actividades y lugares al aire libre cercanos.
- O4: Buscar o filtrar actividades y/o lugares.
- O5: Ver el detalle de una actividad o lugar.
- O6: Compartir información en redes sociales de una actividad o lugar.
- O7: Crear un evento con punto de encuentro, hora y fecha.
- O8: Compartir invitaciones de punto de encuentro en redes sociales.
- O9: Ver lista de eventos.
- O10: Crear lugar.
- O11: Ver perfil de usuario.
- O12: Puede decir que asistirá a eventos al aire libre que estén disponibles.

1.2.2 Requerimientos Funcionales

Requisito Funcional	Objetivo Relacionado
RF1. Gestión de perfiles de usuario y autenticación	
RF1.1 Registrar cuenta de usuario	O1
RF1.2 Iniciar sesión con cuenta existente	O2
RF1.3 Cerrar sesión	O2
RF1.4 Ver información de usuario ingresado	O11

RF2. Geolocalización	
RF2.1 Gestionar permisos de localización de la app	O3
RF2.2 Obtener ubicación actual del usuario	O3
RF3. Vista del Home	
RF3.1 Mostrar tarjetas con recomendaciones de actividades cercanas	O3
RF3.2 Mostrar tarjetas con recomendaciones de lugares cercanos	O3
RF3.3 Agregar barra de búsqueda con filtros por actividad	O4
RF3.4 Al hacer click en la tarjeta de lugar debe ir a la vista del detalle	O5
RF4. Detalle del lugar o actividad	
RF4.1 Mostrar información detallada del lugar (Foto, horarios, direcciones)	O5
RF4.2 Añadir botón de compartir para enviar información por redes sociales (whatsapp)	O6
RF4.3 Añadir botón para crear evento	O7
RF4.4 Al hacer click en crear evento se abre un modal de creación de evento	O7
RF4.5 Boton de like	O10
RF5. Vista modal de creación de evento	
RF5.1 Mostrar formulario de creación de evento (Título, descripción, el lugar seleccionado anteriormente, fecha y hora)	O7
RF5.2 Agregar botón de crear evento	O7
RF5.3 Al presionar el botón de crear evento se debe mostrar una alerta avisando si se ha creado o no exitosamente.	O7
RF5.3.1 Si la alerta es exitosa se agregan botones para volver y para ir a mis eventos	O7
RF5.3.2 Si la alerta es fallida se muestra botón de volver	O7
RF5.4 Esta vista tendrá un botón para cancelar la acción que devuelve al detalle del evento.	O7
RF6. Vista de mis eventos	
RF6.1 Mostrar lista de eventos ordenados desde los más recientes	O9
RF6.2 Cada ítem tiene un botón para cancelar evento	O9

RF6.3 Cada ítem tiene un botón para definir asistencia al evento	O12
RF6.4 Compartir evento en redes sociales	O8

1.2.3 Requerimientos No Funcionales

- Utilizar una arquitectura VIP (View Interactor Presenter) con Protocol Oriented Programming.
- Utilizar inyección de dependencias.
- Implementar los conceptos SOLID.
- Implementar tests unitarios.
- Implementar librería una de mapas.
- Crear una capa de Networking.

1.3 Enfoque y método seguido

1.3.1 Arquitecturas y prácticas de desarrollo

Este proyecto se desarrollará desde cero para la plataforma iOS utilizando el lenguaje de programación Swift, haciendo uso de librerías de mapas como Google Maps y Firebase para el backend. Se tendrá en cuenta en todo momento los principios SOLID[8], los cuales representan buenas prácticas, para el desarrollo y mantenimiento del código a largo plazo.

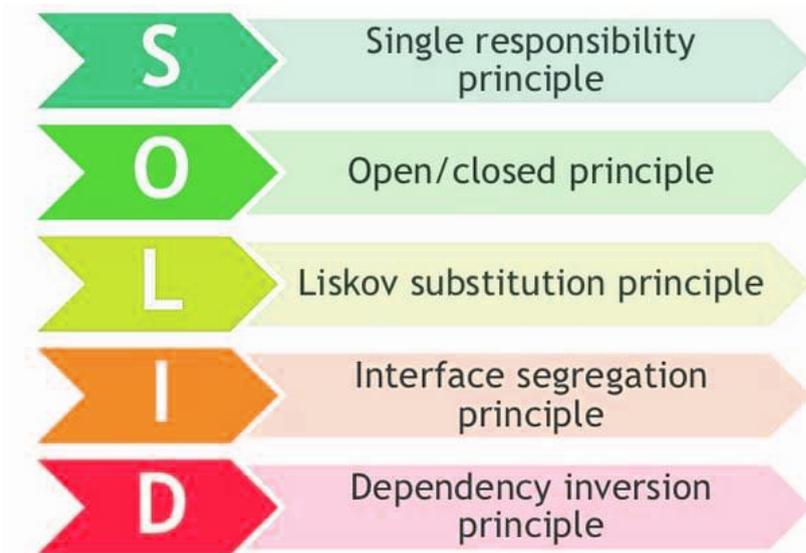


Figura 1. SOLID principles.

Para dar cumplimiento a los principios mencionados y a los requerimientos no funcionales, se ha escogido la arquitectura VIP[9] View-Interactor-Presenter, también conocida como clean swift, la cual separa el código en distintas capas, lo cual permite que se cumpla el principio de única responsabilidad (S: Single

responsability), y además para permitir una mayor facilidad al momento de hacer testing.

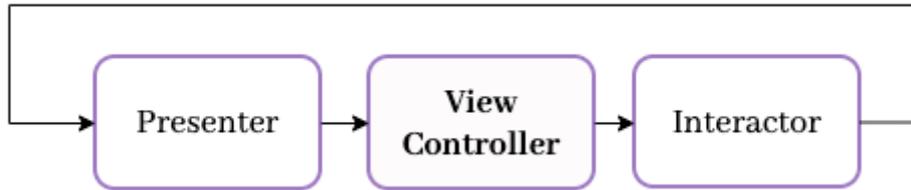


Figura 2. VIP Architecture.

También se implementará el paradigma de programación orientada a protocolos[10] el cual permite crear una abstracción de una clase sin tener que utilizar directamente su instancia, ni modificar directamente sus propiedades. Esto permite dar cumplimiento a los principios de abierto/cerrado (O: open closed principle) porque permite que las clases se abran abiertas a su extensión pero sean cerradas a su modificación, y permite dar cumplimiento al principio de segregación de interfaces (I: Interface segregation), el cual habla de hacer varias abstracciones de un objeto. Para dar cumplimiento al principio de inversión de dependencias, se utilizará el patrón de inyección de dependencias[11], el cual suministra un objeto a una clase, en vez de permitir que se cree el objeto directamente.

1.3.2 Ciclo de vida del proyecto

Para realizar este proyecto se utilizará la metodología en espiral para el desarrollo de software[12], la cual está basada en las fases de la metodología en cascada, pero en este caso las fases se repiten, aumentando en cada ciclo la complejidad de las tareas. Se elige esta metodología porque brinda gran flexibilidad al momento de desarrollar un software y además se alinea con el manifiesto *Agile*[13] el cual tiene una forma cíclica también y que pretende tener siempre un producto que se pueda mostrar al final de cada ciclo, y mejorar en los próximos ciclos. Cada ciclo se dividirá en cuatro fases:

- *Fijación de objetivos*: Se seleccionan cuáles requerimientos se van a realizar en la fase.
- *Análisis de riesgos*: Se determina el tiempo de desarrollo y los riesgos de hacer la tarea.
- *Desarrollo y pruebas*
- *Planificación*: Se finaliza el ciclo, se presenta al cliente lo realizado y se planifica que tiene prioridad en el siguiente ciclo.



Figura 3. Metodología en espiral.

1.3.3 Marco de trabajo

Para este proyecto se va a utilizar el Scrum[14] como marco de trabajo, el cual también se alinea con los principios ágiles y con el ciclo de vida en espiral. Este define cada ciclo como Sprint. Para trabajar sobre el marco de Scrum se deben tener en cuenta los siguientes protocolos:

- *Organización del backlog*: Se refiere a mantener o crear las tareas necesarias para el sprint.
- *Planificación del sprint*: Se refiere a la asignación de tareas al equipo y a la estimación de cada tarea.
- *Sprint*: Periodo de tiempo en el que se realizan las tareas.
- *Daily*: Reunión que se realiza cada día para revisar el avance. Se debe responder a las preguntas: ¿Que hice ayer?, ¿Qué haré hoy? y ¿Hay algún obstáculo?
- *Revisión del sprint*: Al finalizar el Sprint el equipo se reunirá para realizar una demostración y definir el incremento del siguiente sprint.
- *Retrospectiva*: En este punto se revisa que está funcionando, que se debe mejorar y que salió mal.

1.4 Planificación del Trabajo

1.4.1 Recursos de hardware y software

Para la realización del presente trabajo se hará uso de los siguientes recursos:

- Hardware:
 - Macbook pro 2012. Core i5. 16gb de ram.
 - iPad gen 7.
- Software:
 - XCode 12.4: Para desarrollar la aplicación.
 - XCode Simulator: Para probar la aplicación.
 - Git: Para controlar versiones de la aplicación.
 - Github: Para mantener un repositorio en la nube y generar un kanban con las tareas del proyecto.

- Google docs: Para escribir la memoria.
- Google Sheets: Para la generación de tablas y gráficos.
- Draw.io: Para el diseño de diagramas.
- Firebase: Backend.
- OBS: Para realizar captura de video en pantalla.
- Figma: Programa de prototipado.
- Procreate: Programa de dibujo.

1.4.2 Desglose de Tareas

En este apartado se muestra una estructura general de cómo se van a realizar las tareas a lo largo del proyecto teniendo en cuenta los requerimientos descritos anteriormente. A medida que pasan los sprints se definen específicamente las tareas. Se tendrá en cuenta que cada Sprint es de dos semanas, empezando lunes y acabando viernes. Se utilizará la herramienta de proyectos de Github para tener seguimiento de tareas.

- **Plan de trabajo**
 - **Preparación**
 - Definir objetivo general del proyecto.
 - Hacer un mapeo general de requerimientos para sacar un mínimo producto viable.
 - **Modelo del Sprint (Cada sprint va a repetir las siguientes fases):**
 - *Determinar objetivos generales: (Tareas)*
 - *Analizar objetivos definidos. (Estimación del sprint)*
 - *Desarrollar tareas.*
 - *Probar.*
 - *Mostrar y recibir feedback.*
 - **Primer Sprint (Descripción general)**
 - Se realizan tareas para cumplir RF1.
 - **Segundo Sprint (Descripción general)**
 - Se realizan tareas para cumplir RF2 y parte de RF3.
 - **Tercer Sprint (Descripción general)**
 - Se realizan tareas para cumplir RF3 y la mitad de RF4.
 - **Cuarto Sprint (Descripción general)**
 - Se realizan las tareas para terminar RF5 y parte de RF6.
 - **Quinto Sprint (Descripción general)**
 - Se realizan las tareas para terminar RF6 y hacer setup de despliegue.
 - **Sexto Sprint (Descripción general)**
 - Se realizan las tareas de entrega del proyecto.

1.4.3 Estimación de tareas

Para este proyecto se utilizará una estimación basada en tallas de camiseta desde la xs hasta la xl. Cada talla se relaciona con un número de la sucesión de fibonacci empezando desde uno (1).

- XS: (1 punto) siendo la tarea más pequeña, podría tener un estimado de tiempo aproximado menor a una hora.
- S: (2 puntos) un tiempo aproximado de dos horas.
- M: (3 puntos) entre 4 y 8 horas.
- L: (5 puntos) entre 9 y 24 horas.
- XL: (8 puntos) más de 24 horas.

Lo ideal es dividir las tareas de forma que cada sprint no pase de 8 puntos.

1.4.4 Estimación de tiempos

Se calcula que, como se han definido seis (6) sprints, y que cada sprint es de dos semanas, el tiempo estimado para culminar el MVP de este proyecto sería de 12 semanas (3 meses), empezando el primer sprint el día 22 de septiembre de 2021. Es decir el 14 de diciembre aproximadamente se finalizarán los sprints y el 27 se hará entrega del proyecto. Para cada semana de un sprint se trabajarán 32 horas (384 horas). Sumando a lo anterior 18 horas de preparación para un total de 402 horas.

TAREAS	HORAS	PREPARACIÓN		PRIMER SPRINT		SEGUNDO SPRINT		TERCER SPRINT		CUARTO SPRINT		QUINTO SPRINT		SEXTO SPRINT	
		15/09/2021	21/09/2021	22/09/2021	05/10/2020	06/10/2021	19/10/2021	20/10/2021	02/11/2021	03/11/2021	16/11/2021	17/11/2021	30/11/2021	01/12/2021	14/12/2021
Preparación	18	█	█												
Determinar objetivos	1			█											
Estimación	1			█											
Desarrollo	50			█											
Pruebas	10				█										
Mostrar y recibir feedback	2				█										
Determinar objetivos	1					█									
Estimación	1					█									
Desarrollo	50					█									
Pruebas	10						█								
Mostrar y recibir feedback	2						█								
Determinar objetivos	1							█							
Estimación	1							█							
Desarrollo	50							█							
Pruebas	10								█						
Mostrar y recibir feedback	2								█						
Determinar objetivos	1									█					
Estimación	1									█					
Desarrollo	50									█					
Pruebas	10										█				
Mostrar y recibir feedback	2										█				
Determinar objetivos	1											█			
Estimación	1											█			
Desarrollo	50											█			
Pruebas	10												█		
Mostrar y recibir feedback	2												█		
Revisión y correcciones	64													█	█
TOTAL	402														

Figura 4. Diagrama de Gantt. Fuente propia.

1.5 Breve resumen de productos obtenidos

- Código fuente
- Memoria del proyecto
- Demostración de uso en video
- Manual de usuario

1.6 Breve descripción de los otros capítulos de la memoria

En este apartado se describen brevemente los siguientes capítulos de la presente memoria, los cuales llevaron a conseguir los objetivos del proyecto, a través de la metodología propuesta.

- **Capítulo 2: Diseño**

Este capítulo se dedica al análisis y diseño de la aplicación centrada en el usuario, mostrando los diagramas pertinentes, los casos de uso y los prototipos realizados.

2. DISEÑO Y ANÁLISIS

2.1 Investigación y análisis

En esta fase del proyecto se realiza una investigación utilizando métodos cuantitativos y cualitativos los cuales brindarán al proyecto información valiosa para descubrir el perfil del usuario. En este caso se utiliza la encuesta y la entrevista como métodos primarios.

2.1.1 Encuesta

Para esta fase se diseñó la siguiente [encuesta](#) como método cuantitativo, la cual nos permitió capturar datos de 44 personas que residen en la ciudad de Barranquilla. A continuación analizamos los resultados de la encuesta, pregunta por pregunta:

(TODO) Poner resultados de encuestas acá con el análisis

1. ¿Te gusta realizar actividades al aire libre?

Se puede observar que a casi todos los encuestados les gusta realizar actividades al aire libre.

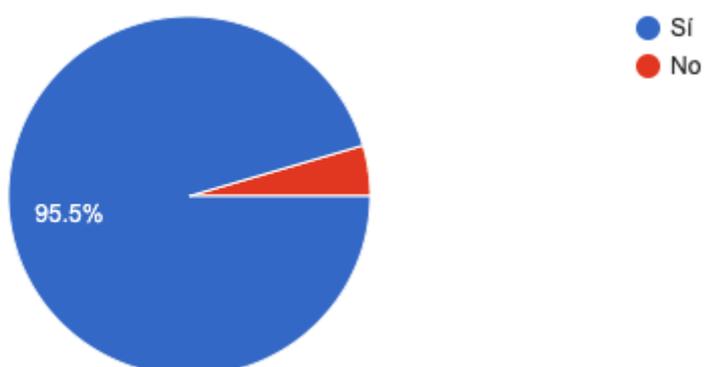


Figura 5. pregunta 1 encuesta. Fuente propia.

2. ¿Qué actividades disfrutas realizar al aire libre?

En esta pregunta los encuestados en su mayoría respondieron que les gusta hacer ejercicio en exteriores. La segunda categoría más escogida fue hacer picnic, luego correr y ver cine o teatro al aire libre. Entre las otras respuestas también los encuestados agregaron acampar, ir al mar o a piscina.

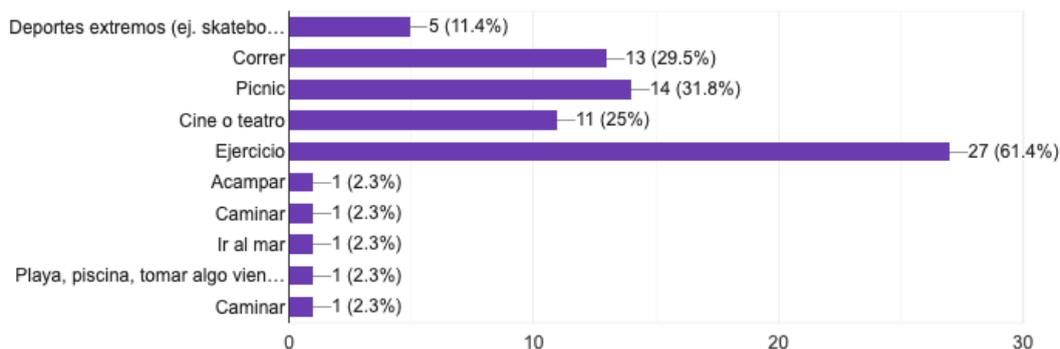


Figura 6. pregunta 2 encuesta. Fuente propia.

3. Con qué frecuencia buscas actividades o lugares al aire libre para realizar en tu ciudad?

En esta pregunta la mitad de los encuestados respondieron que los fines de semana generalmente estarían dispuestos a buscar actividades o lugares al aire libre. Lo cual puede indicar que en su mayoría salen a sitios abiertos en los fines de semana.

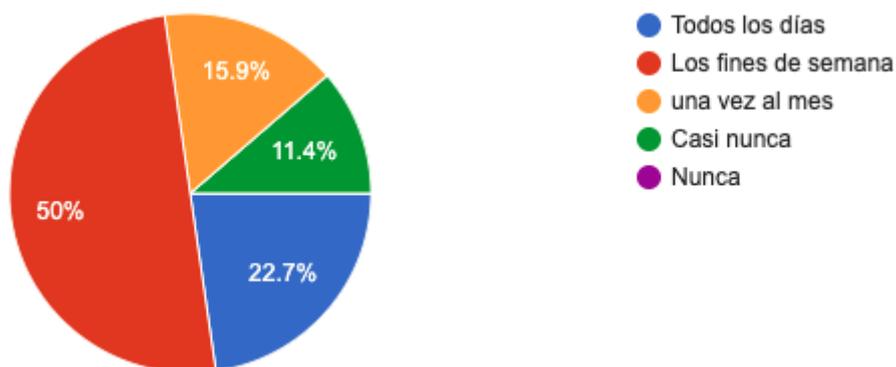


Figura 7. pregunta 3 encuesta. Fuente propia.

4. ¿Utilizas tu celular para encontrar actividades o lugares al aire libre?

En esta pregunta los encuestados respondieron en su mayoría que no usaban su celular para encontrar actividades o lugares al aire libre, lo cual podría ocurrir porque no existe una aplicación relacionada al tema o porque los encuestados prefieren descubrir lugares o actividades por otros medios.

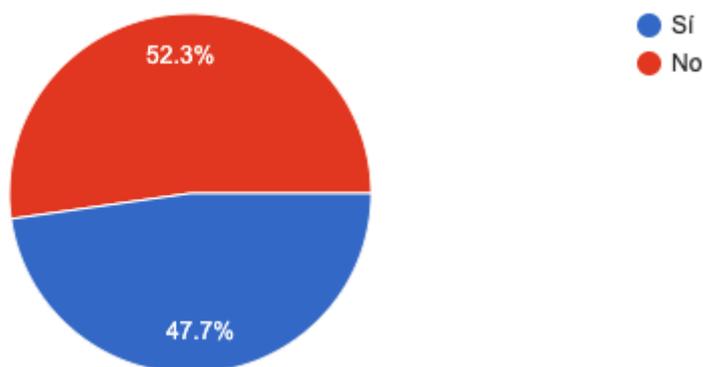


Figura 8. pregunta 4 encuesta. Fuente propia.

5. ¿Qué apps utilizas o has utilizado para encontrar actividades al aire libre?

Esta pregunta pretendía obtener información sobre las aplicaciones que más usaban los encuestados para encontrar actividades al aire libre. Se puede observar que la más usada es instagram y la segunda opción más escogida es ninguna app. Entre las otras opciones más escogidas se encuentran google maps, facebook y whatsapp, pero esto podría ocurrir porque el encuestado no conoce o no existe una aplicación específica para el tema.

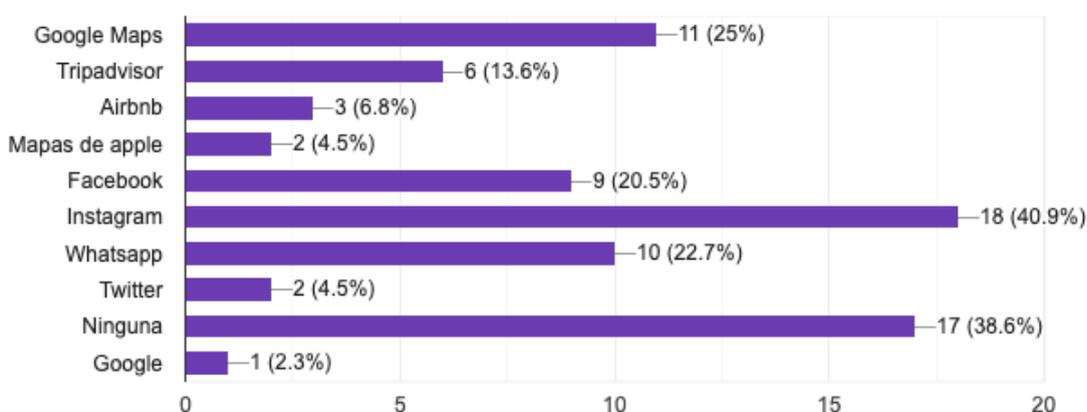


Figura 9. pregunta 5 encuesta. Fuente propia.

6. ¿Qué apps utilizas para encontrar lugares al aire libre?

En esta pregunta se pretendía obtener información sobre qué apps usan los encuestados para encontrar lugares al aire libre, y se observa que los usuarios en su mayoría no usan ninguna aplicación, o usan en su mayoría instagram o google maps. Es posible que utilicen dichas aplicaciones porque son las que más utilizan en su diario vivir.

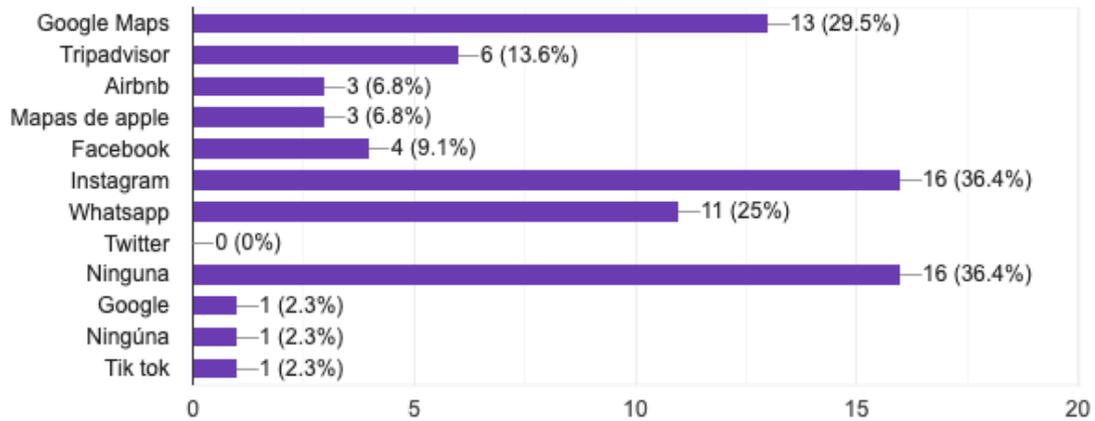


Figura 10. pregunta 6 encuesta. Fuente propia.

7. ¿Alguna vez has planeado encontrarte con personas en lugares al aire libre?

Esta pregunta estaba enfocada en las necesidades de los usuarios de planear puntos de encuentro en lugares al aire libre. Se puede observar que la gran mayoría de usuarios planea encuentros en exteriores, lo cual se podría utilizar para validar funcionalidades de creación de eventos en la aplicación.

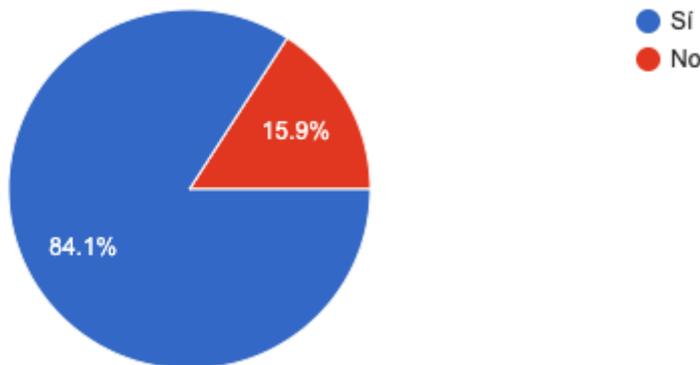


Figura 11. pregunta 7 encuesta. Fuente propia.

8. ¿Con qué frecuencia utilizas lugares al aire libre como punto de encuentro con otras personas?

La mitad de los encuestados respondieron que ocasionalmente utilizan lugares al aire libre como punto de encuentro con otras personas, y un 29% afirmó que casi siempre. Lo cual podría brindar información sobre la frecuencia de uso de la característica de creación de eventos en la aplicación.

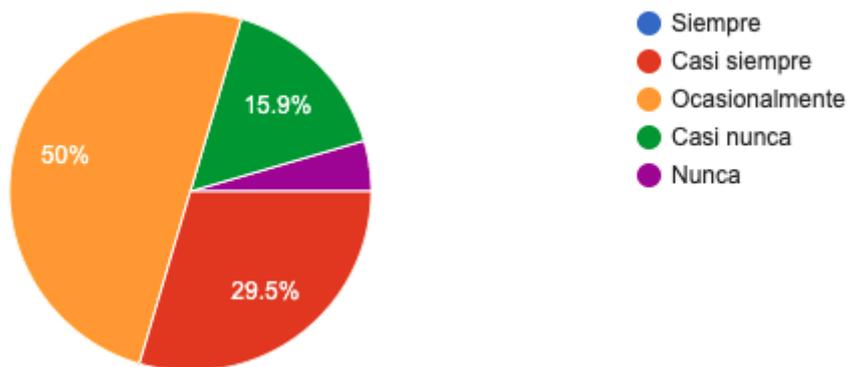


Figura 12. pregunta 8 encuesta. Fuente propia.

9. ¿Qué apps utilizas para crear puntos de encuentro?

Esta pregunta estaba enfocada en investigar las aplicaciones que utilizan los encuestados para crear puntos de encuentro, o eventos. En este caso la gran mayoría respondió que utilizaban Whatsapp o ninguna aplicación. Esto podría apuntar a que los encuestados prefieren hablar directamente con las personas o por chat.

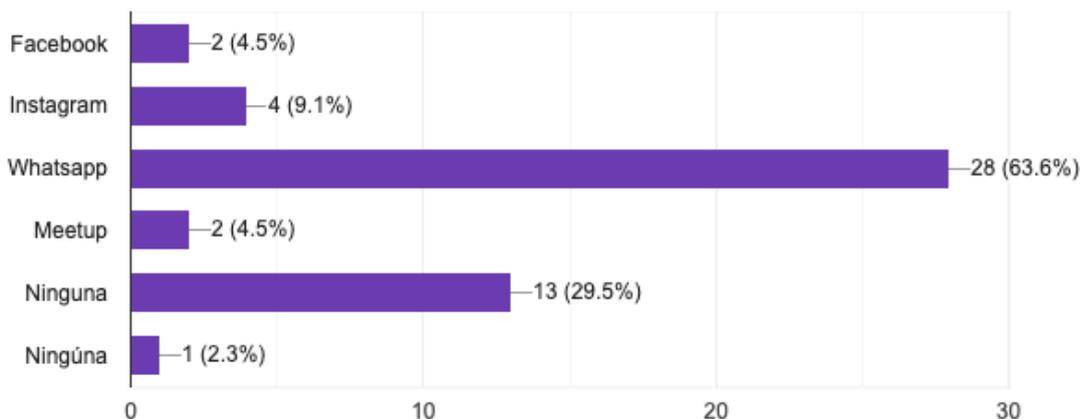


Figura 13. pregunta 9 encuesta. Fuente propia.

10. ¿Crees que sería útil una app donde pudieras encontrar actividades al aire libre en tu ciudad?

Esta pregunta se realiza para validar el objetivo principal de la aplicación el cual es permitir el descubrimiento de lugares y actividades al aire libre a las personas. Podemos observar que a la mayoría de personas les gusta la idea.

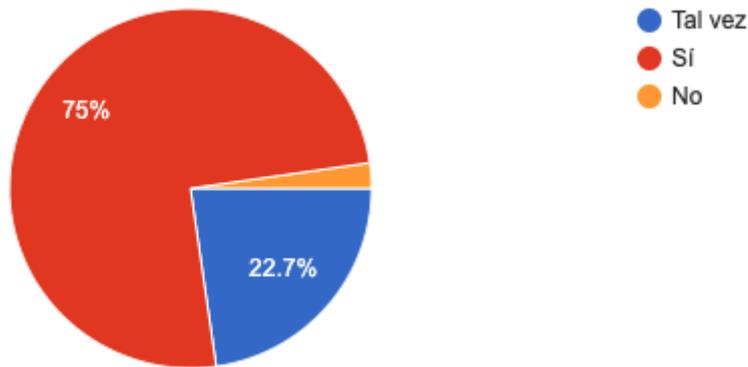


Figura 14. pregunta 10 encuesta. Fuente propia.

11. *¿Crees que sería útil una app donde pudieras crear y compartir tus eventos al aire libre?*

Esta pregunta está orientada a validar la funcionalidad de creación de eventos de la aplicación. Se puede observar que la mayoría de personas cree que sería útil crear y compartir eventos.

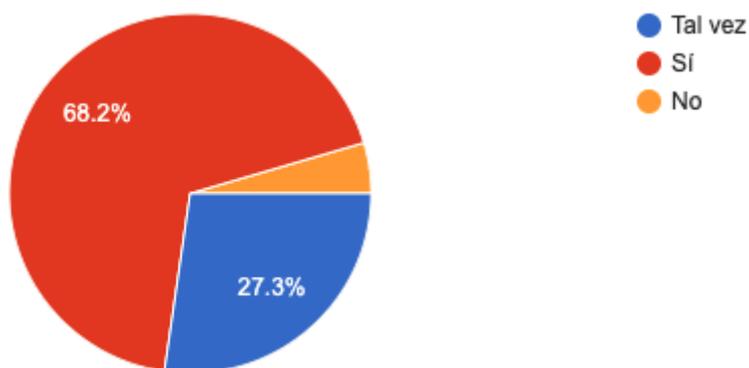


Figura 15. pregunta 11 encuesta. Fuente propia.

12. *¿Qué edad tienes?*

Esta pregunta está orientada a perfilar nuestro usuario teniendo en cuenta su edad. Podemos observar que la mayoría de los encuestados tiene de 25 a 40 años.

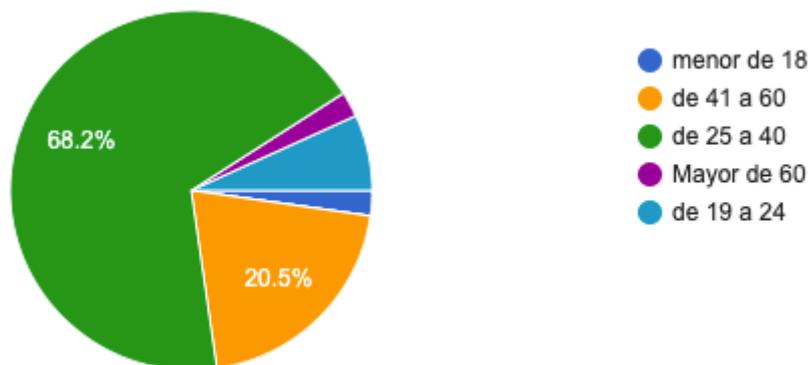


Figura 16. pregunta 12 encuesta. Fuente propia.

2.1.2 Entrevista

Como método cualitativo, para profundizar sobre el tema se realizaron entrevistas a un grupo de personas de la ciudad de Barranquilla que consideran que realizan a menudo actividades al aire libre, con un formato de preguntas abiertas donde se le brinda a la persona un espacio para expresarse libremente y brindar su opinión. Las entrevistas se realizaron por teléfono y las transcripciones se pueden ver en el anexo 6.1.

Analizando las respuestas se observa que los entrevistados prefieren salir a lugares abiertos, principalmente por temor a contagiarse de covid. Uno dijo que prefería espacios abiertos porque trabaja en espacios cerrados, otra persona dijo que dependía del clima.

Las actividades que los entrevistados realizan al aire libre son variadas, entre jugar Pokemon Go, salir a caminar, patinar o a trotar, sacar al perro y tomar cerveza con los amigos, conciertos, manifestaciones. En este caso las personas explicaron que les gustaba ver el paisaje o porque las actividades que realizan al aire libre no se pueden hacer en sitios cerrados.

Sobre las apps que utilizan para descubrir lugares al aire libre los entrevistados respondieron que utilizan google maps, facebook, instagram y whatsapp, porque se pueden observar rutas o porque tienen o siguen grupos de personas con los mismos intereses. Una persona mencionó que utiliza las páginas del gobierno para encontrar eventos culturales.

Las personas entrevistadas comentaron que salen a menudo a lugares abiertos, ya sea todos los días o por lo menos una vez a la semana, para hacer ejercicio y para entretenerse. En general todos dijeron que sí elegirían lugares abiertos para planear salidas con amigos, algunos opinaron que tienen más flexibilidad de esa forma y por las restricciones de covid en espacios cerrados. También comentaron que Whatsapp es el medio de comunicación preferido para compartir estos eventos porque ya tienen grupos creados y se puede encontrar a todas las personas en esa misma app.

Al preguntarles si creían que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil, se puede observar distintas respuestas, entre no y preferir seguir utilizando whatsapp porque es más directo, a creer que sí, para encontrar nuevos lugares para patinar y planear eventos de sus comunidades, como eventos de la comunidad de pokemon go o scouts, y también para encontrar cosas nuevas que hacer.

Para finalizar la entrevista, se les preguntó a los entrevistados a qué otro tipo de personas les sería útil una aplicación de este estilo y comentaron que a personas que planeaban eventos deportivos en los parques, o encuentros de baile, o personas que quieran realizar eventos de mayor tamaño. También comentaron que podría ser útil para personas que no conocen la ciudad y entusiastas de la naturaleza.

2.2 Diseño conceptual

Habiendo finalizado la fase de investigación, se pasa a utilizar los datos obtenidos para crear los perfiles de usuario, escenarios de uso, y un diagrama de flujo de la aplicación.

2.2.1 Perfiles de usuario

A partir de la información recolectada en las encuestas y las entrevistas, podemos definir a nuestro usuario como:

Una persona que vive en Barranquilla entre los 25 y los 40 años, que le gustan las actividades al aire libre, que va a lugares abiertos para hacer ejercicio. Que le gusta salir los fines de semana y que utiliza lugares al aire libre como punto de encuentro con otras personas.

2.2.2 Ficha de persona y escenarios de usuario

Para entender mejor y traducir los datos recolectados, se crean las siguientes fichas de personas, las cuales muestran ejemplos de potenciales usuarios de la aplicación tomando las características encontradas anteriormente en la investigación y ubicándolos en un contexto donde interactúan con el sistema.

P1 - Deportista

Nombre: Andrea Jimenez.

Edad: 43 años.

Trabajo: Entrenadora deportiva.

Familia: Separada con un hijo de 7 años.

Localidad: Barrio El tabor, Barranquilla.



Metas

- Mantener un estilo de vida saludable.
- Conectarse con la naturaleza.
- Pasar tiempo con su hijo.

Frustraciones

- No tener donde encontrar actividades relacionadas con la naturaleza en la ciudad.
- Hacer siempre lo mismo los fines de semana.
- No poder encontrar una comunidad de deportistas.

Biografía

Andrea es una persona apasionada por el deporte y la vida saludable. Corre todos los días dos horas en la madrugada en el malecón del río para ver el amanecer, luego se va al parque bosques del norte a entrenar a sus clientes. Después de su jornada laboral regresa a casa a recibir a su hijo que llega de la escuela. Ella trabaja todos los días pero le encanta pasar tiempo con su familia por lo tanto hace lo posible para terminar su jornada temprano y en la tarde salir con su hijo a disfrutar de los parques.

Escenario

tarde del Sábado, Andrea llega del trabajo, toma un baño y se alista para salir al parque con su hijo. Pero ella quiere hacer algo diferente hoy, así que abre la aplicación Quillair, y revisa opciones de actividades cercanas. En el buscador escribe la palabra infantil, y encuentra que se está realizando un evento de teatro infantil en un parque cercano a su casa, el cual era un evento al que nunca antes había podido asistir. Ella presiona en el botón de asistir, y comparte el evento a su hermana para que ella también pueda enterarse. Ambas van al evento con sus hijos y pasan una tarde amena y diferente.

P2 - Scout

Nombre: José Cervantes.

Edad: 21 años.

Trabajo: Colabora en los Scouts y estudia ingeniería.

Familia: Vive con su abuela y su padre.

Localidad: Barrio paraíso, Barranquilla.



Metas

- Sacar buenas notas en el semestre.
- Salir con sus amigos.
- Apoyar en los eventos Scout.

Frustraciones

- Tener que programar eventos de los Scout por chat.
- No poder compartir con sus amigos, fácilmente los eventos en los que está participando.

Biografía

José es un joven tranquilo que vive con su padre y su abuela. Él se encuentra en cuarto semestre de ingeniería industrial. Se levanta temprano para ir a la universidad de lunes a viernes. Por las tardes estudia y planea actividades Scout para el grupo. Los viernes por las noches sale con sus amigos a la plaza a tomar aire y hablar de su vida con ellos. Los fines de semana está siempre ocupado cumpliendo su labor como Scout.

Escenario

Es jueves por la tarde, José llega de la universidad y su líder dice que lo ayude a planear un entrenamiento Scout, porque no encuentra un espacio abierto dentro de la ciudad que tenga una cancha con piso de arena. José abre Quillair, selecciona lugares, escribe en el buscador “cancha de arena” y encuentra que en malecón del río hay una cancha con las condiciones que se necesitan. José crea un evento y lo comparte con su líder, y su líder se da cuenta que puede encontrar lugares así en esta aplicación, la instala y la comparte con todos los Scouts que van a participar el siguiente Sábado para que se enteren del evento.

2.3 Prototipo de la aplicación

En este punto se toma la información recolectada para generar un prototipo de un mínimo producto viable. En primer lugar se trabaja un boceto de las pantallas, y basados en este se genera un prototipo de alta fidelidad, teniendo en cuenta el estilo nativo de iOS y la navegación. Para realizar el prototipo de baja fidelidad se utilizó el programa *Procreate* y para el de alta fidelidad se hizo uso del programa *Figma*.

2.3.1 Prototipo de baja fidelidad

En esta fase se le da importancia a los conceptos generales de la aplicación y generar una guía que será la base para continuar con el prototipo de alta fidelidad. Se logra el siguiente bosquejo, donde se dibujan a grandes rasgos una primera impresión de la aplicación con una navegación básica. Se presentan algunas pantallas que serían las principales de la aplicación: Login, registro, intereses, permisos, home, detalle, creación de evento, lista de eventos y perfil.

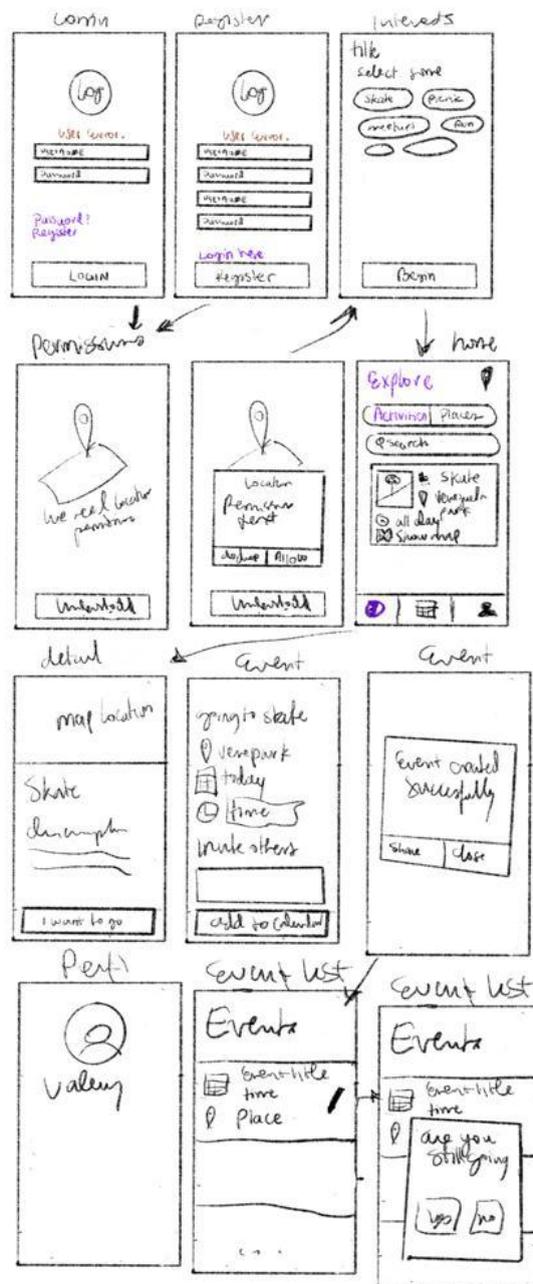


Figura 17. Boceto app. Fuente propia

2.3.2 Prototipo de alta fidelidad

En esta fase se ha tomado como base el bosquejo anterior y se han tenido en cuenta el diagrama de flujo y los diagramas de caso de uso para generar un prototipo con navegación, más completo y realista. En el siguiente [enlace](#) se puede observar una representación del prototipo con un acercamiento bastante parecido a lo que sería considerado como una aplicación nativa de iOS. El flujo de este prototipo omite la pantalla de *splash* e inicia directamente en la vista del Login. Los logotipos e iconos usados no son los oficiales.

Login

Es la primera pantalla que ve el usuario (después del *splashscreen*). En esta pantalla el usuario podrá escribir su usuario y contraseña e iniciar sesión. Al iniciar sesión el usuario podrá ver directamente el “*home*” de la aplicación. Desde esta pantalla se puede navegar hacia el registro.

9:41

Quillair

Usuario

Contraseña

Por favor introduce los datos

[¿Problemas para iniciar sesión?](#)

[¿No tienes cuenta? Regístrate](#)

Iniciar sesión

Figura 18. Login. Fuente propia.

Registro

En esta pantalla el usuario puede brindar datos como su nombre, correo y contraseña para obtener sus credenciales de inicio en la aplicación. El botón de registro hará que el usuario navegue hacia la pantalla de permisos. Desde esta pantalla se puede volver hacia el inicio de sesión si ya se tiene credenciales.

9:41

Quillair

Tu nombre

Correo electrónico

Contraseña

Confirma tu contraseña

Por favor introduce los datos

[¿Ya tienes cuenta? Ingresar aquí](#)

Registrarme

Figura 19. Registro. Fuente propia.

Permisos

En esta pantalla, el usuario podrá brindar permisos de localización a la aplicación para que la misma pueda brindar sugerencias personalizadas cercanas al usuario. Al continuar en este paso, el usuario verá la pantalla de intereses.

9:41 

Bienvenido a Quillair

Permisos de localización
Quillair utiliza tu localización para brindarte sugerencias de lugares y actividades cerca de tí

[Permitir siempre](#)

[Permitir mientras está en uso](#)

[No permitir](#)

Para una mejor experiencia por favor brinda los permisos de localización

[Continuar](#)

Figura 20. Permisos. Fuente propia.

Intereses

En esta pantalla el usuario puede brindar información de sus intereses. La app recibirá palabras separadas por comas las cuales convertirá en “tags”, las cuales se pueden eliminar. Al continuar en este punto se pasa a la vista principal o “home”.

9:41

Bienvenido a Quillair

Antes de continuar, escoge cuales son tus intereses

Escribe tus intereses

Bailar × Perros ×

Futbol ×

Continuar

Figura 21. Intereses. Fuente propia.

Home

Esta es la pantalla principal, donde el usuario puede ver sugerencias de lugares cercanos de acuerdo a sus intereses o actividades en lugares cercanos. Haciendo click en el botón de localizar, la aplicación vuelve a pedir la ubicación del usuario para actualizar las actividades cercanas.

Esta vista también contiene un “tabbed-view” el cual tiene opciones para ver el perfil, los eventos y explorar el home.

En esta pantalla se ha dividido la información en:

Explorar Actividades

Al hacer clic en el segmento de “lugares / actividades” el usuario podrá cambiar las sugerencias o buscar de acuerdo a lo seleccionado. En el caso de actividades se mostrarán eventos que estén ocurriendo cerca. Al hacer clic en una actividad se muestra el detalle de la actividad.

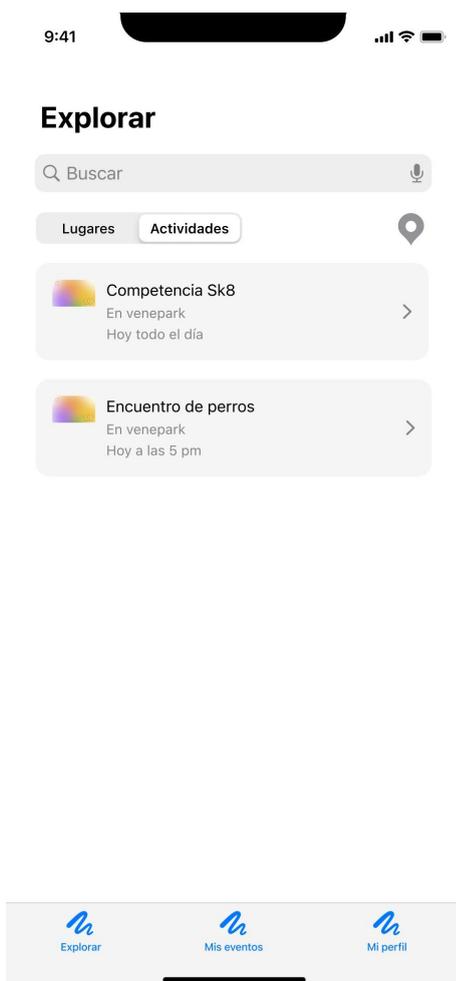


Figura 22. Explorar actividades. Fuente propia.

Detalle de la Actividad

En esta vista el usuario puede ver el mapa de donde se está realizando el evento, la hora, la cantidad de asistentes y la descripción detallada del evento. El usuario puede utilizar el botón de compartir para enviar la información del evento a otras personas y puede decir que asistirá al evento, lo cual agrega un ítem a su lista de eventos. El usuario puede volver al “home” desde esta pantalla.

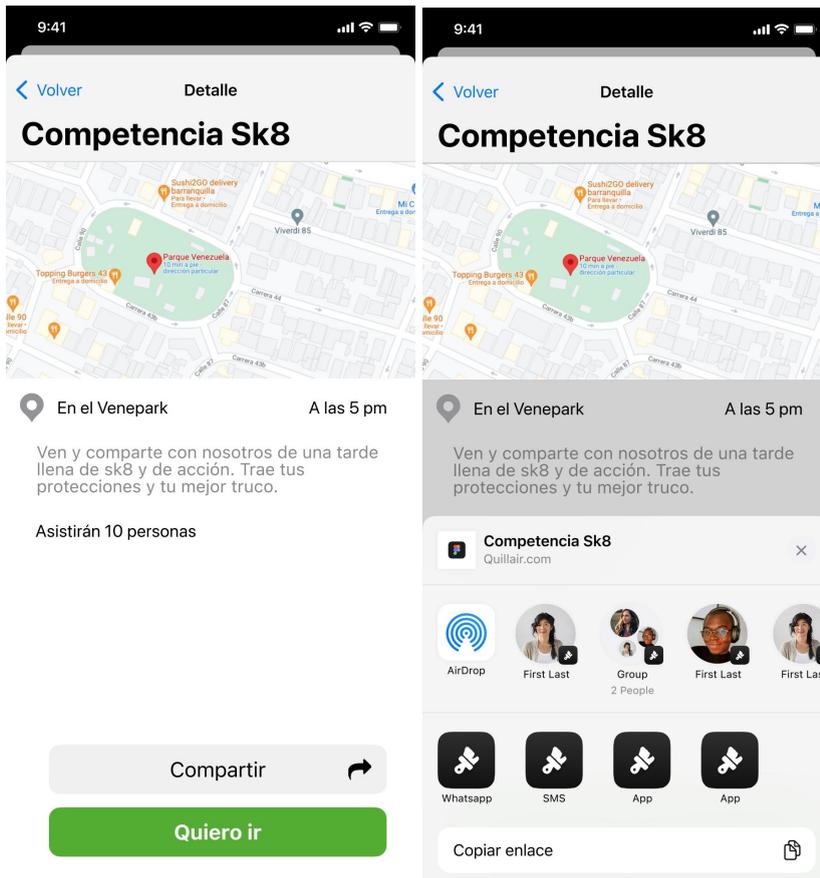


Figura 23. Detalle de actividad y compartir actividad. Fuente propia.

Explorar Lugares

Al seleccionar lugares en el segmento de “lugares / actividades”, la aplicación mostrará lugares cercanos al usuario, permitirá al usuario buscar cualquier lugar de acuerdo los intereses o temas. Al hacer clic en un lugar, la aplicación navegará hacia el detalle del lugar.

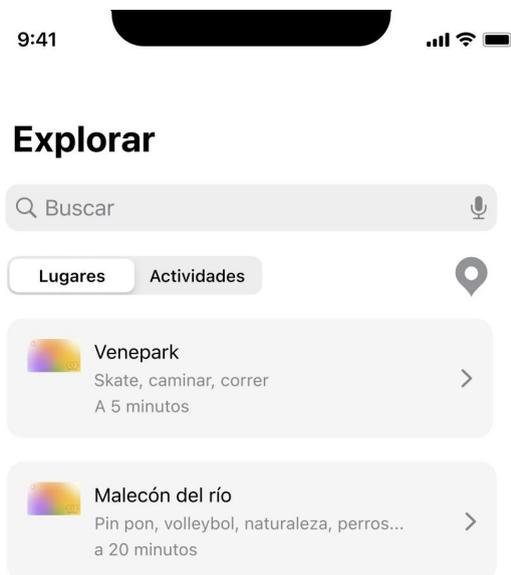


Figura 24. Explorar lugares. Fuente propia.

Detalle del lugar

En esta vista el usuario puede ver el mapa del lugar, una descripción y un aproximado de tiempo de desplazamiento. El usuario puede compartir el lugar (así como se muestra en los eventos), y navegará hacia la pantalla de crear evento haciendo clic en el botón de “Quiero ir”. El usuario puede volver al “home” desde esta pantalla.

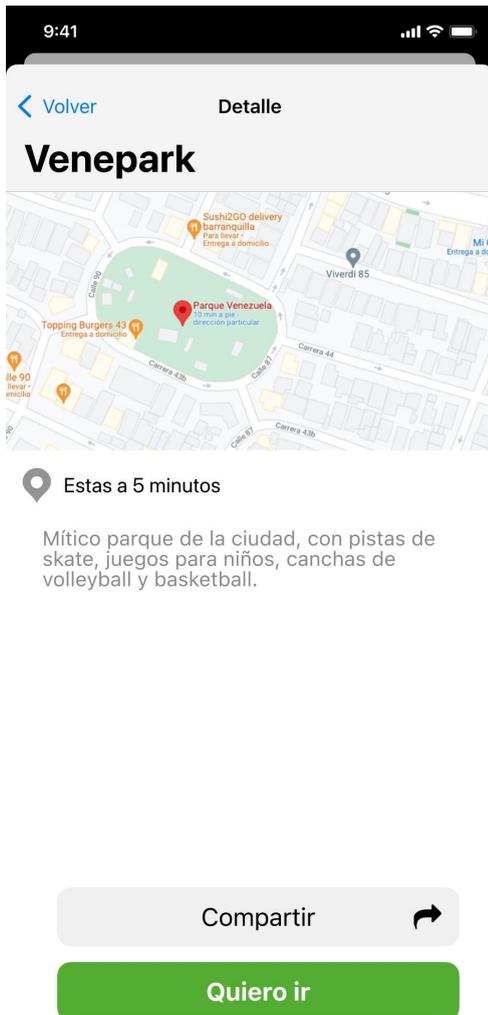


Figura 25. Detalle del lugar. Fuente propia.

Crear evento

En esta vista el usuario ya ha seleccionado un lugar y ha decidido que quiere crear un evento. El usuario podrá ingresar datos como título, descripción, fecha y hora del evento. También puede elegir si un evento es público, lo cual permitiría que todos los usuarios puedan ver el evento. En caso contrario, solo podrán ver el evento los usuarios que tengan el link. Al crear un evento exitosamente el usuario podrá volver al “home” o ir a “mis eventos”.

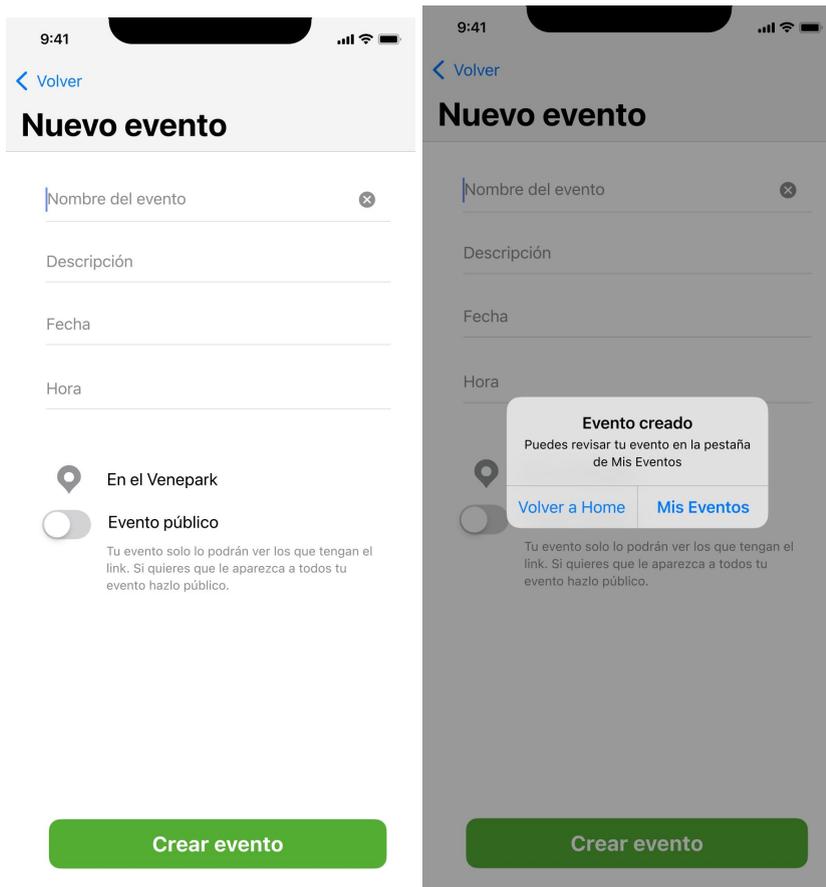


Figura 26. Crear evento. Fuente propia.

Mis eventos

En esta vista el usuario puede encontrar una lista de eventos a los cuales dijo que asistirá o que ha organizado. Por el momento desde esta vista el usuario solo puede decidir, compartir el link del evento o decidir si ya no asistirá al evento o si va a cancelar el evento. Al cancelar o decir que ya no va a asistir, la aplicación refrescará la lista, removiendo dichos eventos.

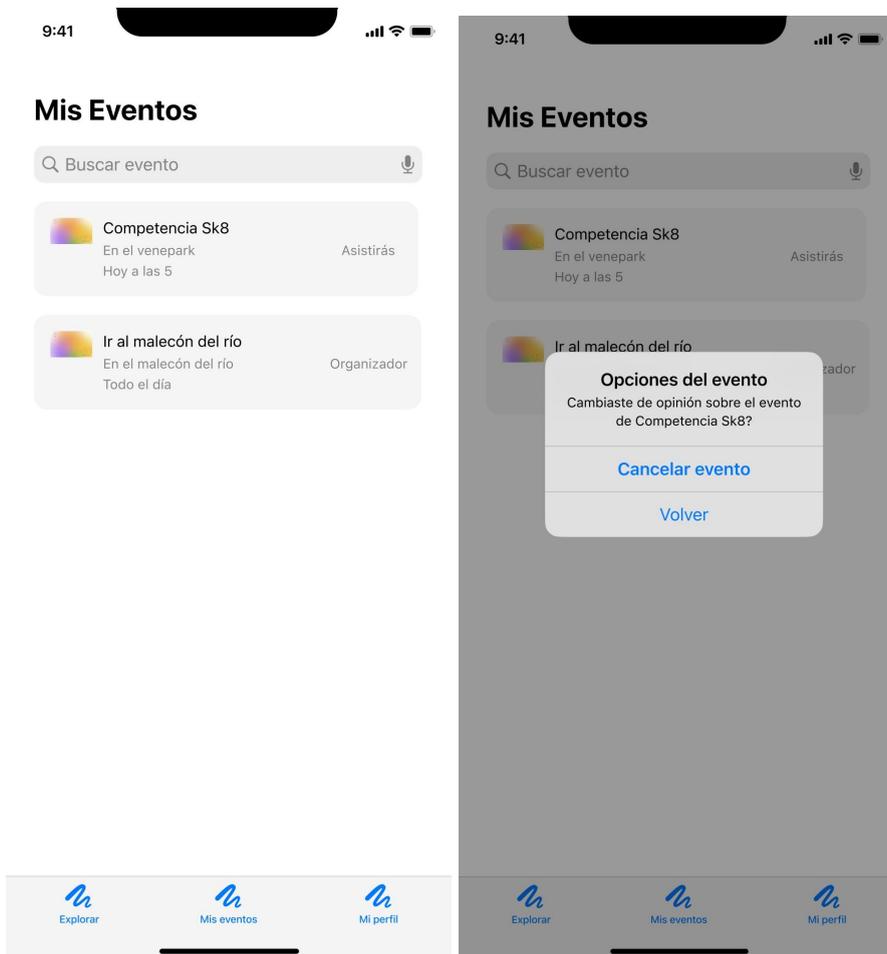


Figura 27. Mis eventos. Fuente propia.

Perfil

En esta vista el usuario puede agregar una foto de perfil, y editar sus preferencias (intereses). El usuario también podrá cerrar sesión.

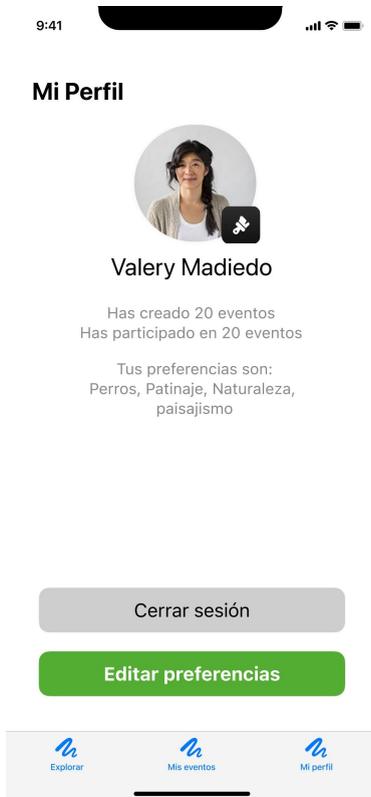


Figura 28. Mi perfil. Fuente propia.

2.4 Árbol de navegación

En esta sección se muestra un diagrama de flujo que muestra la navegación de la aplicación.

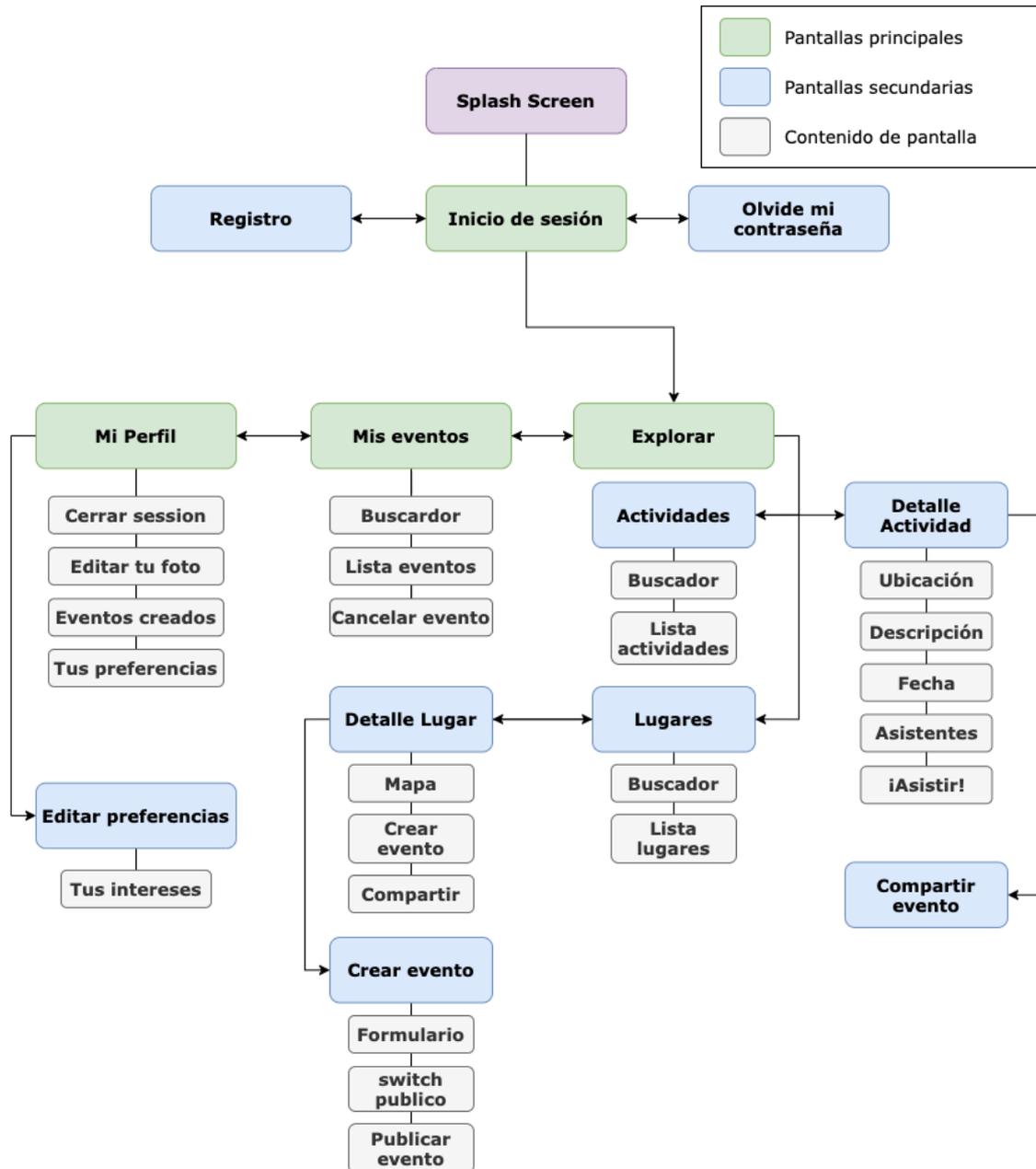


Figura 29. Árbol de navegación. Fuente propia.

2.5 Evaluación

- Al realizar el prototipo de alto nivel se puede observar que no tiene íconos oficiales ni logotipos.
- Al revisar el diseño se evalúa la falta de la funcionalidad para añadir lugares y se añade al diagrama de caso de uso.
- Se recomienda basar los modelos en arquitectura de base de datos noSql de bases de datos, debido a que se piensa implementar Firebase.

2.6 Diseño técnico

En la presente fase se irán describiendo los diagramas de caso de uso, arquitectura y diagrama de clases, teniendo en cuenta el diseño anteriormente presentado.

2.6.1 Casos de Uso

Este apartado definirá las funcionalidades del sistema y cómo las usarán los distintos actores. En esta sección se utiliza la palabra evento en distintas ocasiones para referirse a las actividades al aire libre que muestra el sistema.

Actores

En esta aplicación tiene un solo tipo de usuario pero este usuario se podría dividir en tres actores: Usuario, creador de eventos y consumidor de eventos.

Gestión de usuarios

En el siguiente diagrama se pueden observar las acciones que puede realizar el actor (usuario) relacionadas a la gestión de usuarios. Se describen casos como registro, inicio de sesión, dar permisos, preferencias, ver perfil, cerrar sesión y recuperar contraseña.

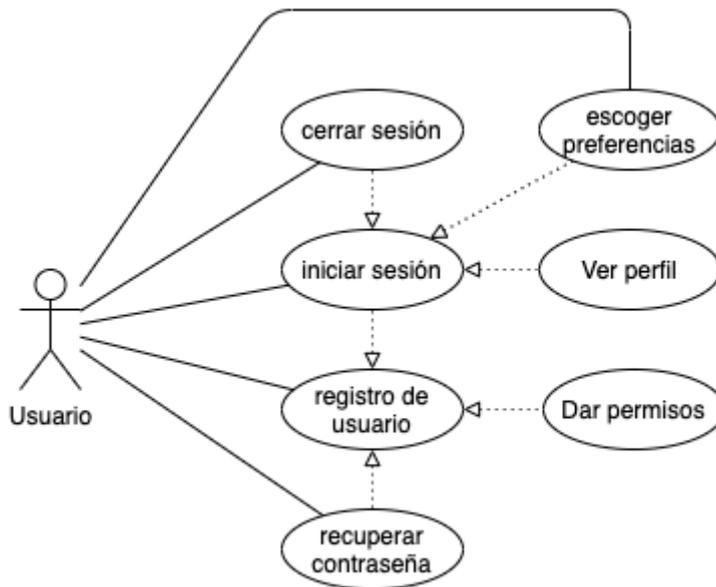


Figura 30. Gestión de usuario. Fuente propia.

C1: Registro de usuario	
Descripción	Registrarse en el sistema
Actores	Usuario

Precondiciones	Tener acceso a internet. Tener la aplicación descargada. Ingresar un correo válido. Cumplir con las condiciones de la contraseña.
Flujo	Desde la pantalla de inicio de sesión, el usuario hace click en el enlace de registro.
Postcondiciones	Si el correo no está registrado y no hubo problemas con la conexión, el registro será exitoso. Si el correo ya existe se mostrará un mensaje.

C2: Inicio de sesión	
Descripción	Iniciar sesión en el sistema
Actores	Usuario
Precondiciones	Tener acceso a internet. Tener la aplicación descargada. Ingresar un correo válido. Cumplir con las condiciones de la contraseña.
Flujo	Al iniciar la aplicación aparece la pantalla de inicio de sesión. También se puede acceder desde el registro al hacer click en el link de inicio de sesión o al cerrar la sesión que estaba abierta.
Postcondiciones	Si el usuario ya está registrado y no hubo problemas con la conexión, el inicio de sesión será exitoso y se pasará a la vista del home. Si ocurre algún problema de conexión o si las credenciales son incorrectas se mostrará un mensaje.

C3: Permisos del sistema	
Descripción	La aplicación mostrará un diálogo para pedir permisos de geolocalización con las opciones correspondientes del sistema operativo.
Actores	Usuario
Precondiciones	Haberse registrado exitosamente en la aplicación.
Flujo	Después que el registro fué exitoso, la app muestra una pantalla donde solicita al usuario brindar permisos.
Postcondiciones	Al continuar mostrará la pantalla de intereses.

C4: Intereses	
Descripción	El usuario puede escribir sus intereses para mostrar sugerencias personalizadas

Actores	Usuario
Precondiciones	Haber realizado el registro exitosamente.
Flujo	Esto ocurre después de haber visto la pantalla de permisos. Se mostrará la pantalla de Intereses donde el usuario puede escribir sus preferencias relacionadas a actividades o lugares.
Postcondiciones	Si no hay problemas de conexión, al continuar se habrán guardado correctamente los intereses y se pasará al home.

C5: Ver perfil	
Descripción	El usuario puede ver su información de perfil y editarla
Actores	Usuario
Precondiciones	Haber realizado el inicio de sesión exitosamente.
Flujo	El usuario navegó hacia el perfil utilizando el enlace de perfil en las pestañas inferiores de la app.
Postcondiciones	Se muestran los datos del usuario, de los cuales podrá editar su foto y sus preferencias. También podrá cerrar sesión desde este punto.

C6: Cerrar sesión	
Descripción	El usuario cierra su sesión en la app
Actores	Usuario
Precondiciones	Haber realizado el inicio de sesión exitosamente y haber navegado hacia la vista de perfil.
Flujo	El usuario hace click en el botón de cerrar sesión que se encuentra en la vista del perfil.
Postcondiciones	Se regresa al inicio de sesión, el usuario ya no tiene acceso a las otras vistas. Se ha eliminado su token del sistema.

C7: Recuperar contraseña	
Descripción	El usuario puede cambiar su contraseña
Actores	Usuario
Precondiciones	Haber realizado el registro exitosamente.
Flujo	El usuario navegó desde inicio de sesión utilizando el enlace correspondiente.

Postcondiciones	Se le pide al usuario que agregue un correo electrónico donde se enviarán los datos para realizar la recuperación de la cuenta.
------------------------	---

Gestión de eventos

En el siguiente diagrama se muestran los casos relacionados a la gestión de eventos. Esta sección también contiene casos en los que representan la búsqueda de lugares y detalle del lugar. En este caso el actor es el creador de eventos, el cual representa al usuario cuando realiza las acciones que lo llevan a crear un evento y gestionarlo.

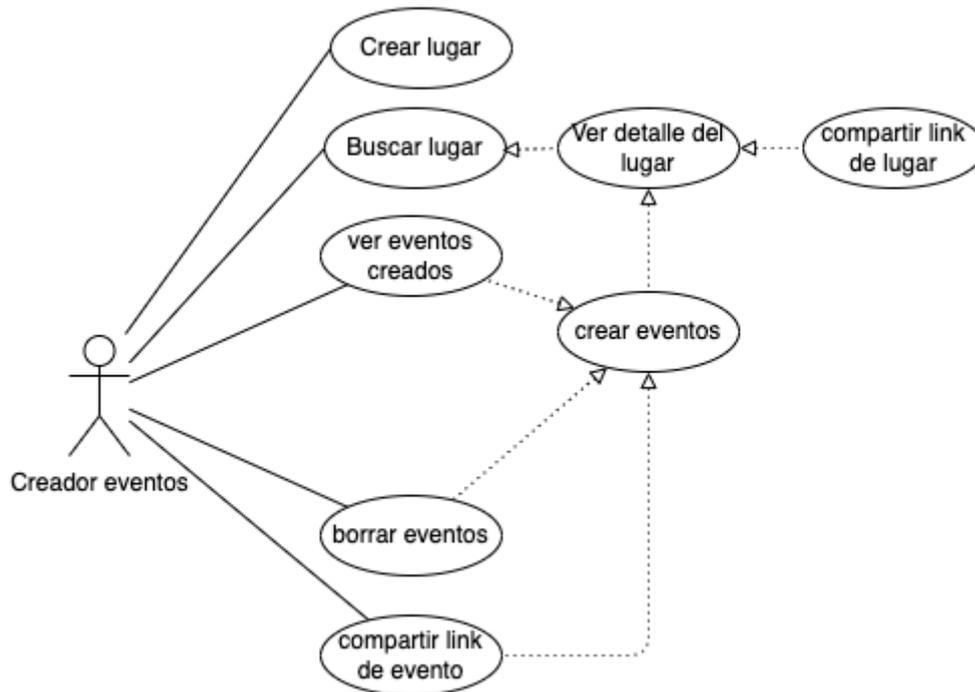


Figura 31. Creación de eventos. Fuente propia.

C8: Buscar lugar	
Descripción	El actor puede buscar lugares para crear sus eventos
Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet. Estar en la vista del home.
Flujo	El actor hace clic en el segmento para seleccionar lugares e ingresa una palabra en el buscador.
Postcondiciones	Se muestran una lista de lugares relacionados a la palabra insertada.

C9: Ver detalle del lugar	
Descripción	El actor puede ver más información sobre el lugar

Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet. Haber seleccionado lugares en el segmento superior de explorar en el home.
Flujo	El actor hace clic en un ítem de lugar
Postcondiciones	Se muestra una pantalla con el mapa de ubicación del lugar y más información.

C10: Compartir link del lugar	
Descripción	El actor puede compartir el enlace del lugar
Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet. Estar en el detalle del lugar.
Flujo	El actor hace clic en el botón de compartir
Postcondiciones	Se abre el panel de compartir del sistema con las opciones para compartir el link

C11: Ver eventos creados	
Descripción	El creador de eventos puede ver una lista de los eventos que ha creado y que ha dicho que asistirá
Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet.
Flujo	El usuario navegó desde el home hacía "mis eventos" haciendo click sobre el enlace en las pestañas inferiores.
Postcondiciones	Se muestran una lista de los eventos que el usuario ha creado y ha dicho que asistirá

C12: Crear evento	
Descripción	El creador de eventos puede publicar un evento en la app
Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet. El usuario se encuentra en el detalle de un lugar.
Flujo	El usuario hace click en el botón de crear evento en la pantalla de detalle de lugar.

Postcondiciones	Se muestra una pantalla de creación de evento, donde se le pide al usuario que ingrese el título, descripción, fecha y hora del evento y defina si el evento es público o no.
------------------------	---

C13: Borrar evento	
Descripción	El creador de eventos puede cancelar un evento y se eliminará del sistema.
Actores	Creador de eventos
Precondiciones	Haber creado un evento o haber comunicado la asistencia a un evento. El actor se encuentra en "mis eventos"
Flujo	El actor hace clic en el ítem del evento
Postcondiciones	Se muestra un diálogo preguntando si quiere cancelar Al cancelar se actualizará la lista de eventos.

C14: Compartir link del evento	
Descripción	El actor puede compartir el link del evento
Actores	Creador de eventos
Precondiciones	Haber iniciado sesión. Haber ingresado a la lista de mis eventos. Tener eventos en mis eventos.
Flujo	El usuario hace click largo en el ítem del evento
Postcondiciones	Se abre el panel de compartir del sistema con las opciones para compartir el link

C15: Añadir lugar	
Descripción	El actor puede agregar lugares
Actores	Creador de eventos
Precondiciones	Haber realizado el inicio de sesión exitosamente. Tener conexión a internet. Estar en la vista del home. Hacer click en el botón de lugar.
Flujo	El actor ingresa los datos del lugar y una imagen.
Postcondiciones	El lugar se guarda y aparecerá en la lista de lugares.

Descubrir actividades

En este diagrama se muestra al actor, consumidor de eventos, como un actor que busca actividades al aire libre, en el sistema.

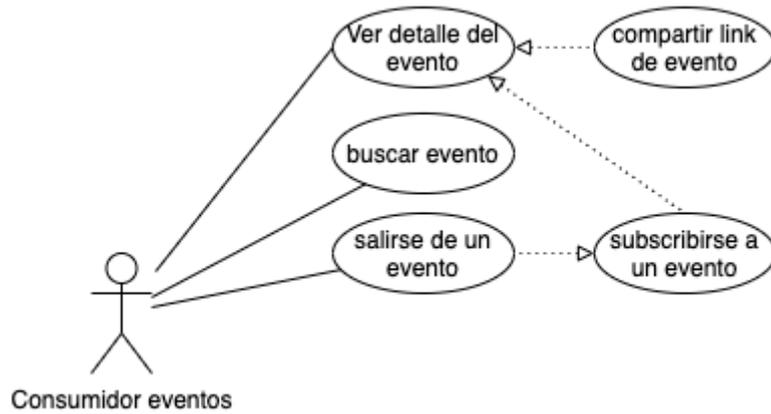


Figura 32. Creación de eventos. Fuente propia.

C12: Buscar evento	
Descripción	El actor puede ingresar palabras en el buscador y descubrir actividades al aire libre en el sistema.
Actores	Consumidor de eventos
Precondiciones	Tener conexión a internet. Haber iniciado sesión exitosamente. Estar en la vista de explorar.
Flujo	El actor selecciona la opción de actividades en el segmento superior. El actor ingresa una palabra en el buscador.
Postcondiciones	Se muestra una lista de actividades (o eventos) al aire libre relacionadas a la palabra insertada.

C13: Salirse de un evento	
Descripción	El actor puede decir que ya no quiere asistir a un evento.
Actores	Consumidor de eventos
Precondiciones	Tener conexión a internet. Haber iniciado sesión exitosamente. Haber dicho que asistía a un evento. Haber ingresado a la vista de mis eventos.
Flujo	El actor hace clic en el ítem del evento.
Postcondiciones	Se muestra un diálogo preguntando al actor si ya no quiere asistir al evento. Si el actor hace el cambio, la lista ya no mostrará el evento.

C14: Ver detalle del evento	
Descripción	El actor puede ver más información del evento
Actores	Consumidor de eventos

Precondiciones	Tener conexión a internet. Haber iniciado sesión exitosamente. Estar en la pestaña "explorar" del home.
Flujo	El actor selecciona un ítem de la lista de actividades que resultaron de la búsqueda
Postcondiciones	Se muestra una pantalla donde se puede observar el mapa y más información de la actividad

C14: Suscribirse a un evento	
Descripción	El actor puede decir que va a asistir a un evento creado (actividad al aire libre)
Actores	Consumidor de eventos
Precondiciones	Tener conexión a internet. Haber iniciado sesión exitosamente. Estar en el detalle del evento
Flujo	El actor hace clic en el botón de quiero ir.
Postcondiciones	El botón cambiará su texto a asistirás y aumentará uno al contador de asistentes. También se agregará el evento a la lista de mis eventos.

C15: Compartir link del evento	
Descripción	El actor puede compartir el enlace de la actividad
Actores	Consumidor de eventos
Precondiciones	Tener conexión a internet. Haber iniciado sesión exitosamente. Estar en el detalle del evento.
Flujo	El actor hace clic en el botón de compartir.
Postcondiciones	Se abre el panel de compartir del sistema con las opciones para compartir el link

2.6.2 Arquitectura

En el primer capítulo, en la sección 1.3.1 se definió el uso de SOLID, y de la arquitectura MVP para el presente proyecto. En esta sección se presenta una representación visual de la arquitectura, mostrando la división de las capas y sus componentes. Este diagrama también muestra una aproximación de las clases e interfaces a utilizar.

Arquitectura de los servicios

Tomando en consideración los principios SOLID, empezamos definiendo las dependencias que se van a utilizar:

- Keychain para guardar el token y la información del usuario.
- Firebase para suplir la necesidad de un backend.
- UIImagePickerController para la selección y toma de imágenes.

El paquete o carpeta *Services* contendrá clases que dividen el uso de cada una de las dependencias anteriores, creando funciones que hacen referencia únicamente las funcionalidades que se utilizarán de cada dependencia. Cada clase implementa un *Manager* el cual contiene solo los métodos que serán públicos a la aplicación. Se realiza de esta forma porque si en un futuro se quiere cambiar alguno de los servicios solo sería que el nuevo servicio implemente el *Manager* y no se debería cambiar más nada en la aplicación.

En el siguiente diagrama se puede observar de una manera gráfica las capas:

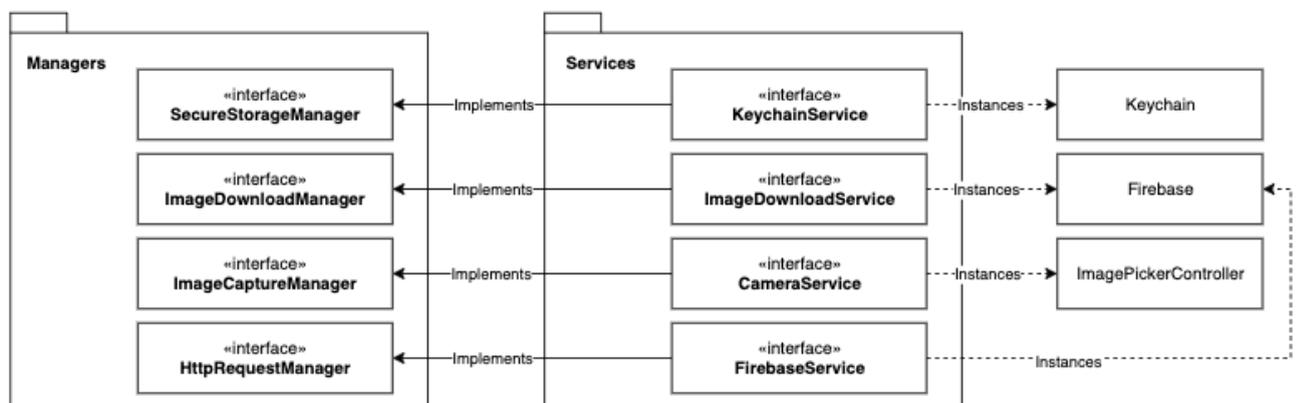


Figura 33. Arquitectura de servicios. Fuente propia.

Arquitectura de la escena

En esta sección mostramos cómo está conformada la arquitectura de una escena teniendo en cuenta VIP. En los diagramas definimos la arquitectura de la escena, la cual contiene worker, interactor, viewcontroller, presenter, businessmodel y router y cómo interactúa con los servicios.

Tendremos un folder de *factory* donde crearemos clases que instancian los managers necesarios para cada escena. El *factory* de login por ejemplo solo podrá instanciar *SecureStorageManager* y *HttpRequestManager*, porque son los que necesita.

Pasando a la escena se pueden nombrar los siguientes componentes:

- *Router*: Tiene la responsabilidad de inyectar las dependencias desde el *factory* hacia el *ViewController*. También se encarga de la navegación, teniendo funciones para presentar la siguiente vista.
- *ViewController*: Tiene la responsabilidad de manipular los elementos de la vista. Este recibe como dependencia el *interactor*.

- *Presenter*: Se encarga de presentar solo la información que necesita la vista y se encarga de validar los datos de los formularios. Tiene una instancia del *ViewController*.
- *Interactor*: Se encarga de realizar la lógica del negocio. Toda operación que se necesite realizar deberá ubicarse aquí. Este hace un llamado a las funciones de los *workers*. También contiene los datos que se van a enviar a otra vista.
- *Worker*: Se encarga de hacer llamados a los servicios externos al negocio. Por ejemplo: pedir al servicio de http que haga un get a los lugares.
- *BusinessModel*: Es la clase donde se definen los modelos específicos de la escena. Esta clase se puede utilizar para definir los componentes de una escena en forma de *structs*, por ejemplo, definir los elementos de la vista, o los elementos de una petición.

Se debe aclarar los siguientes puntos:

- Que todos los componentes anteriores se comunican a través de protocolos.
- Que el router inyecta a todos los componentes las dependencias necesarias.
- Que el *worker* puede usar directamente los modelos para realizar tareas de decodificación por ejemplo, pero las demás clases utilizarán los modelos de *BusinessModel*.

El siguiente diagrama muestra un ejemplo de lo descrito anteriormente. Para este caso se utilizó la escena de *Login* para realizar el diagrama. Para el resto de escenas se tiene la misma estructura de la escena con las mismas clases y protocolos.

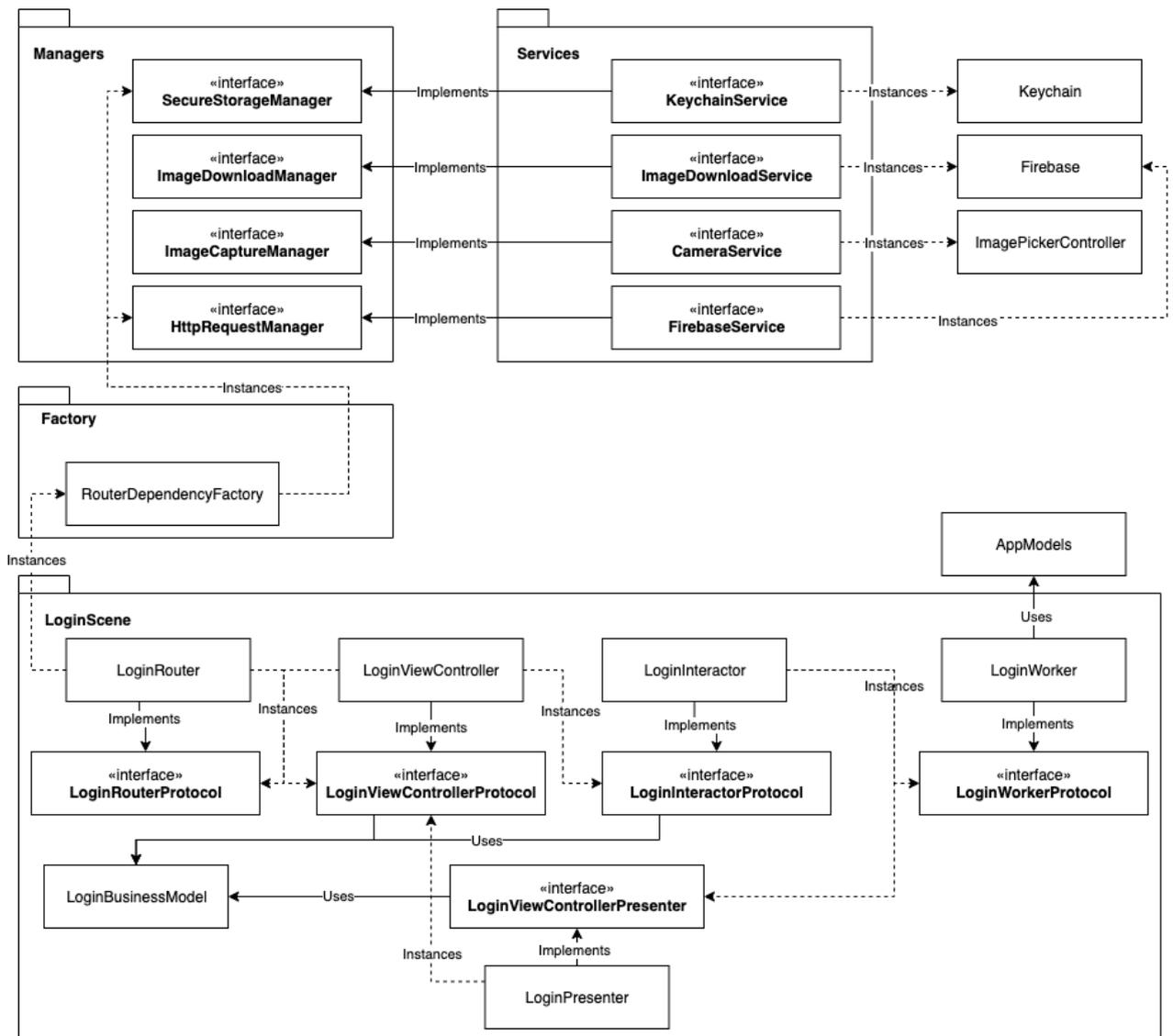


Figura 34. Arquitectura de la escena (VIP). Fuente propia.

2.6.3 Diagrama de base de datos

En esta sección se muestra una representación de los modelos sql de la base de datos de la aplicación en caso de que se quiera realizar una base de datos tradicional.

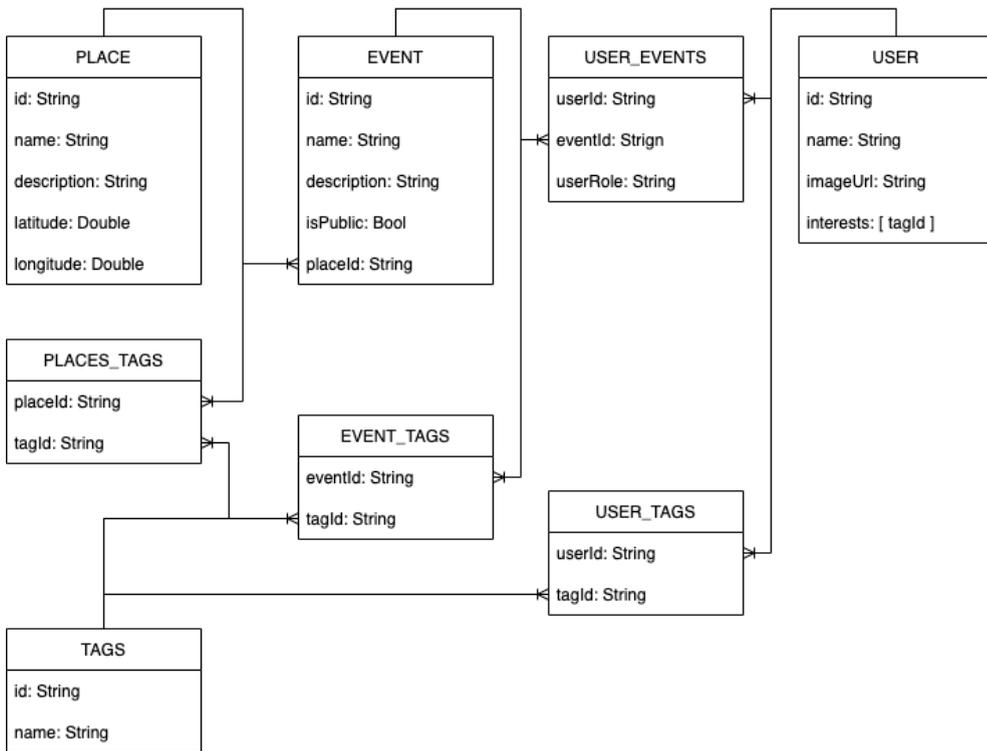


Figura 35. Base de datos SQL. Fuente propia.

3. IMPLEMENTACIÓN

En esta sección se detallarán las decisiones tomadas sobre la etapa de implementación del producto descrito en anteriormente. Se plantean y justifican diversos cambios introducidos sobre el diseño, la estructura general del código y se muestra el funcionamiento de la aplicación móvil.

3.1 Organización de tareas

Este proyecto se implementó teniendo en cuenta cada vista del prototipo. Anteriormente se definió que se utilizará Scrum como marco de trabajo, pero en la marcha se cambió a un modelo más simplificado como es el *kanban* debido a la falta de tiempo para realizar el análisis de puntaje y los recuentos diarios. Para esto se hizo uso de la herramienta de proyectos en Github, donde se generaron 4 columnas para mayor simplicidad:

- *Analisis*
- *Bloqueado*
- *En progreso*
- *Terminado*

Debido a que cada *épica* comprendía una funcionalidad se subdividió por tareas en tres secciones, arquitectura, funcionalidad y componentes visuales. La sección de arquitectura se componía de una tarea por cada capa de la vista, por ejemplo una tarea era hacer la capa del *interactor*, otra tarea la del *presenter*, etc. Para las tareas de funcionalidades, se describieron cada una de ellas en cada vista, y se crearon tareas de cada componente visual que se tenía que agregar. Esto se realizó de esta manera, debido a que con este método se podía hacer un seguimiento más cercano de cada elemento de la vista, ya que las escenas se componen de muchos archivos y capas era mucho más fácil olvidar partes de la vista. Se puede observar el proyecto en el [presente link](#).

3.2 Cambios o modificaciones

En este apartado se describen los cambios más importantes que se realizaron durante la implementación.

3.2.1 Sobre el mínimo producto viable

En el diseño del prototipo no se tenía planteado permitir que el usuario agregase lugares porque el tiempo no alcanzaría para realizar esta vista. Al reflexionar sobre este tema se concluyó que tomaba más tiempo que una sola persona agregase de forma manual a la base de datos todos los lugares, ante la posibilidad de que muchos usuarios ingresen la información por su cuenta.

Se plantea una nueva vista llamada *Añadir Lugar* la cual permite al usuario agregar el lugar donde se encuentra, con los datos de nombre y descripción y una fotografía. Se decidió que sólo se podrían agregar lugares donde el usuario se encuentre debido al corto plazo que se tenía para realizar esta vista.

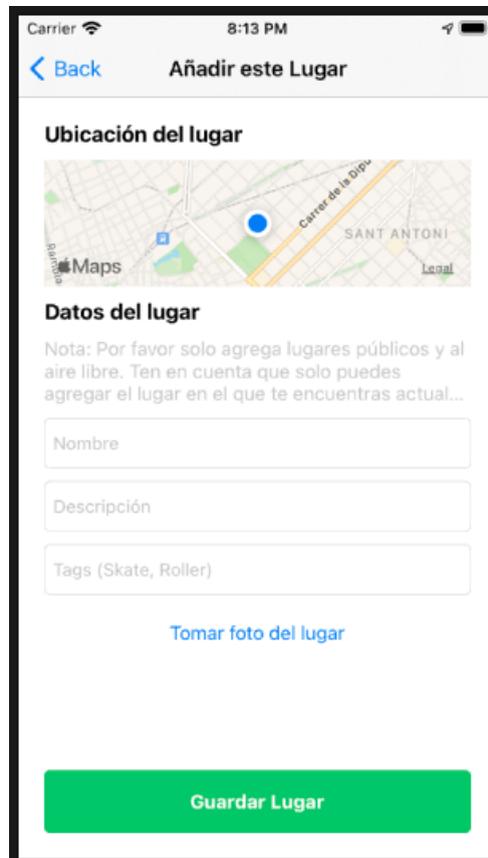


Figura 36. Vista de añadir este lugar. Fuente propia.

3.2.2 Sobre el backend

En el diseño se planteó un diagrama de base de datos SQL, pero por falta de experiencia en backend, se tomó la decisión de utilizar Firebase la cual es una solución basada en noSQL y sin servidores. En teoría esta decisión ahorró tiempo y recursos debido a que todo el tema de autenticación, base de datos y almacenamiento de imágenes quedó unificado en la misma herramienta.

Para esto se tuvo que realizar un nuevo planteamiento de los modelos y del diagrama de base de datos, el cual se describe en la siguiente figura:

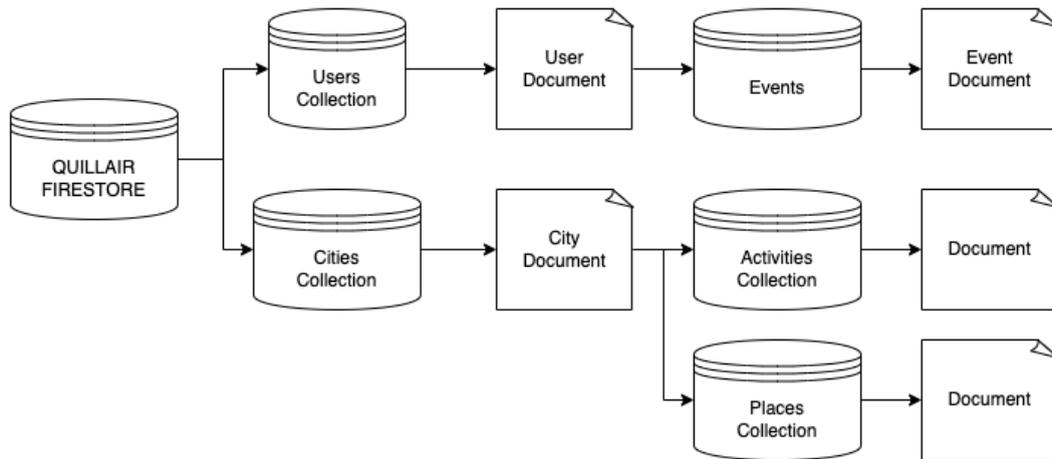


Figura 37. Base de datos noSQL. Fuente propia.

3.2.3 Sobre elementos de las vistas

Se decide, por falta de tiempo, cambiar los siguientes elementos:

- En la vista de *Explorar* y *Mis eventos*, se podía observar un estilo de carta en los ítems de las listas el cual se cambió por un estilo de celda más simplificado.
- Se optó por mostrar las etiquetas y no agregar el tiempo de llegada debido a que no se contaban con el tiempo y experiencia necesarios para realizar cálculo de rutas.



Figura 38. Antes y después del diseño de ítems. Fuente propia.

- Por falta de tiempo se decidió no agregar el estilo *chip* a las etiquetas, sino que se dejó como un texto plano.

3.2.4 Sobre las consultas a la base de datos

Se decidió usar el método de búsqueda más básico que tiene Firestore, el cual busca por orden de mayor o igual al texto, y solo por la propiedad de nombre, debido a la falta de experiencia con bases de datos noSQL. Esto causa que las búsquedas no tengan en cuenta la inclusión de una subcadena de caracteres en los títulos de los ítems, ni se pueda crear una búsqueda cruzada agregando títulos y etiquetas.

Se decidió también que para facilitar las consultas para el mvp, y tal vez abrir el mercado de la aplicación a otros países, las colecciones de lugares y actividades se harían por la ciudad donde se encuentra el usuario, es decir si un usuario está en Barcelona, se muestra lo relacionado a esta ciudad. Del

mismo modo, si una persona agrega un lugar o actividad estarán ligadas a la ciudad donde se encuentra. Para obtener la ciudad se optó por realizar un reverse geolocation, el cual da como resultado la ciudad, si se introduce la latitud y longitud.

3.2.5 Sobre la planeación, tiempos de desarrollo y otros problemas

Durante el desarrollo de este proyecto surgieron hechos imprevistos que afectaron el tiempo de desarrollo, la motivación y la moral. El mes de noviembre el cual contenía gran parte del desarrollo se vio interrumpido por movimientos de ciudad, horarios de trabajo prolongados y otros problemas personales que lograron inevitablemente retrasar en gran medida el desarrollo del presente proyecto. Para compensar se decidió ocupar una mayor capacidad horaria en días libres, fines de semana e incluso tomar prestadas horas del trabajo.

3.3 Herramientas

3.3.1 IDE y lenguaje

El desarrollo de esta aplicación se realizó en XCode 12.4, utilizando el lenguaje de programación Swift 5, debido a que esa es la versión a la que se pudo acceder en la máquina utilizada.

3.3.2 Repositorios

Se utilizó GIT y Github como herramientas de control de versiones y para alojar el proyecto en la nube, debido a que son gratis y libres para usar. El proyecto se encuentra alojado en [este repositorio](#).

3.3.3 Librerías

Este proyecto utiliza *Cocoa Pods* como manejador de librerías debido a que es el más conocido, y porque casi todas las herramientas de google se encuentran disponibles por medio de este manejador.

Se utilizaron librerías externas como:

- Firebase auth, para la autenticación de usuarios, debido a que facilita todos los procesos de inicio de sesión, registro y recuperación de contraseña.
- Firebase Firestore, como base de datos del proyecto ya que provee una API sencilla de utilizar y rápida de implementar.
- Firebase Storage, ya que permite el alojamiento de imágenes y archivos de forma sencilla.

Se utilizaron librerías internas como:

- URL Session, para descargar imágenes, debido a que no requiere instalación adicional.
- Core Location, para obtener la ubicación del usuario.

- MapKit, para mostrar la ubicación en el mapa, ya que no requiere de pagos, como google maps, ni instalaciones adicionales.

La mayor parte de estas librerías (con excepción de mapkit), se implementaron a través de una clase de servicio, la cual tenía los métodos de utilidad para la aplicación y estos métodos se abstraen por medio de protocolos llamados Managers. Estos protocolos fueron utilizados por los routers para inyectar los servicios al resto de capas de la escena.

Particularmente para el caso de Firebase y de CoreLocation, se necesitó más tiempo de lo habitual para descifrar la forma de desacoplarse de la vista y llevarlo a cumplir con VIP. Con el firestore de Firebase se observó que funciona de una forma muy particular para realizar consultas, lo que llevó a crear interfaces muy grandes con consultas muy específicas para cada caso, por ejemplo: para consultar un documento dentro de una colección, era una función y para consultar un documento que se encontrase en una colección con más niveles de profundidad tocaba crear otra, llegando al punto de tener varias funciones parecidas pero que hacían cosas en diferentes niveles de las colecciones.

3.4 Desarrollo

Para desarrollar este proyecto se tuvieron muy en cuenta las tareas que se asignaron al momento de planear. Describiendo a grandes rasgos el proceso de desarrollo, naturalmente se decidió realizar cada vista en el orden que iban apareciendo en el sistema. Por ejemplo, ya que *Login* sería la primera vista que el usuario vería, sería la primera que se debía hacer.

Por otra parte, se tuvo en cuenta que cada vista tendría la necesidad particular de implementar un servicio o librería, y en ese caso se realizaba primero el código del servicio y posteriormente la vista.

Para implementar las capas de VIP Clean Swift en cada escena, se realizaba el siguiente proceso:

- Se iniciaba creando el *NIB* con los componentes de la vista, porque era solo agregar los elementos que ya estaban definidos en el prototipo.
- Se continuaba por la capa de *Business Model*, la cual debía describir las funcionalidades de la escena y se iba a usar en las otras capas.
- Luego se hacía el *Worker* debido a que se encargaba de realizar el llamado a los servicios y pasar los datos transformados al *interactor*.
- Luego se continúa con el *interactor*, el cual dejábamos incompleto debido a que se necesitaba hacer el *presenter* y el *view controller*. En este punto se iban haciendo estas tres capas cíclicamente hasta terminarlas.
- Por último el router, ya que era la capa que creaba la vista y los demás componentes.

3.4.1 Estructura de ficheros

Este proyecto está compuesto por 4 (cuatro) grandes ficheros:

- Models: Contiene los modelos generales de la aplicación.
- Managers: Contiene los protocolos que representan abstracciones de los servicios (o dependencias).
- Services: Contiene las clases que implementan librerías o servicios, tales como url Session, Firebase o core location, las cuales se ubican aquí porque están desacopladas de las vistas.
- Scenes: Contiene todas las vistas divididas por folders.
 - Cada escena contiene los siguientes elementos:
 - Nib, ViewController, Worker, Interactor, Presenter, BusinessModels, Router.
 - En ocasiones contienen una carpeta adicional para las celdas de las tablas.

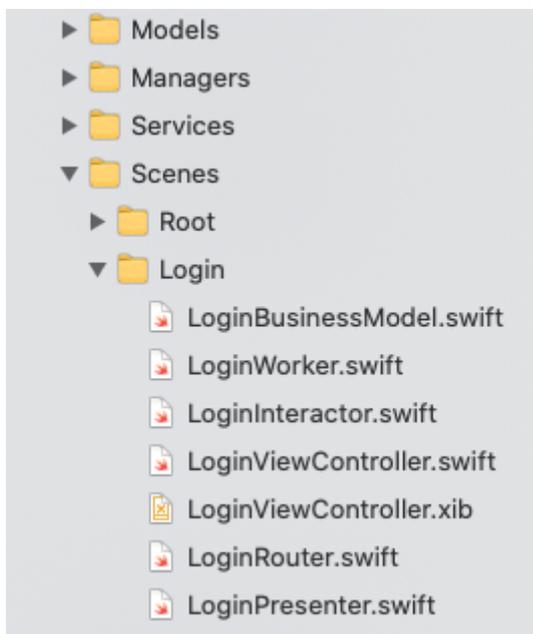


Figura 39. Estructura de ficheros. Fuente propia.

3.5 Pruebas

Para este proyecto se contaba con un tiempo limitado y recursos limitados, por lo tanto se tomó la decisión de realizar las pruebas de cada funcionalidad de forma manual por una persona ajena al desarrollo, haciendo las veces de control de calidad manual, ya que esta persona debía reportar errores encontrados y validar las funcionalidades.

Para la creación de cada prueba se toman los casos de uso descritos en la etapa de diseño como las premisas que se deben cumplir. A esta persona se le pidió que utilizara la aplicación cada vez que un *issue* se pasaba a finalizado y se cerraba el *issue* cuando la persona daba el visto bueno sobre todas las funcionalidades evaluadas.

De este modo se validaron todos los casos y los errores posibles y se llegaba a un acuerdo sobre las tareas que se debían cambiar, corregir o poner en el backlog.

3.6 Despliegue y distribución

Para el presente proyecto, se realizó la adquisición de una cuenta de appstore connect, pero no es posible presentar la aplicación por este medio para la entrega del mismo ante el jurado, debido a que la aplicación tiene que pasar obligatoriamente por un periodo de revisión para beta abierta o producción, el cual puede demorar semanas o meses si es la primera vez que se publica una aplicación, como lo es en este caso.

3.7 Ejecución del código fuente

Como alternativa al despliegue se entregan las siguientes instrucciones de ejecución:

- Descargar código fuente.
- Tener instalado CocoaPods:
`sudo gem install cocoapods`
- En el fichero raíz del proyecto ejecutar:
`pod install`
- Abrir el workspace del proyecto generado por CocoaPods
- Ejecutar el proyecto en el simulador.

4. CONCLUSIONES

Tras haber concluido todas las etapas del proyecto, se evalúa el trabajo realizado y se concluye que en términos generales se logró desarrollar el mínimo producto viable de la aplicación a pesar de el corto plazo y de la falta de recursos para realizarla. Se aprendió que por mucho que se realice una planeación estructurada de los pasos a realizar y los procesos, siempre hay contratiempos y que se debe seguir adelante con los objetivos a pesar de que no se pueda controlar las variables externas.

Otra de las conclusiones a las que se llegó con este proyecto, es que realizar código con una arquitectura VIP Clean Swift, toma mucho más tiempo de lo habitual y hasta se vuelve tedioso con el tiempo, pero se observan los beneficios después de haberla practicado, ya que los archivos quedan mejor estructurados, más pequeños y desacoplados entre sí, lo que en un futuro se puede traducir a mayor escalabilidad y capacidad de hacer pruebas unitarias.

Por otro lado se pudo observar que la implementación de Firebase, al inicio no fue tan sencilla como se esperaba, ya que se tenía que realizar a través de las capas definidas por la arquitectura, lo cual tomó varios días en ser resuelto. También queda como reflexión que Firebase es un producto muy particular y al final aunque quedó desacoplada la dependencia de la vista, los protocolos se tuvieron que hacer específicamente para las implementaciones de firebase, por lo tanto había una interfaz muy grande solo de firestore porque cada petición a la base de datos necesitaba cosas muy específicas.

Tal vez se puede hacer una reflexión sobre la arquitectura elegida, puede ser que para un proyecto pequeño no era necesario elevar tanto la complejidad, y tal vez se hubiera beneficiado de otras arquitecturas tales como MVVM, o incluso nuevas tecnologías como SwiftUI, pero lo que queda claro es que a pesar de que la complejidad de la arquitectura fue elevada, tener claros los principios de SOLID abren las puertas al mantenimiento, modificaciones y adiciones al código.

Se debe decir que los cambios realizados en la implementación fueron en su mayoría causados por los imprevistos y la falta de tiempo para completar ciertas cosas. Aunque algunos de los cambios restaron en temas de estilos e incluso tal vez en algunas funcionalidades, sin estos cambios no se hubiera podido presentar un mínimo producto viable en condiciones, el cual probablemente yo pueda seguir construyendo y puliendo e incluso en un futuro cercano, pueda publicarlo sin temor a que Appstore lo rechace y hacerlo algo más grande.

5. GLOSARIO

App: Aplicación.

API: Servicio o interfaz externos a la aplicación.

Clean code: Código limpio. Se refiere a los principios SOLID.

Factory: Clase que contiene las dependencias que se inyectarán a un módulo en específico.

Framework: Estructura definida que se presenta como base del desarrollo de software.

Interactor: Es el componente de código que recibe los resultados de las peticiones a la API y los envía al controlador.

iOS: Sistema operativo desarrollado por Apple para iPhone.

Keychain: Dependencia de iOS para guardar información encriptada y de forma segura.

Presenter: Es un componente de código que muestra los datos a la vista.

SOLID: Principios de código limpio.

Tabbed View: Vista tabulada. Componente nativo de UIKit para contener distintos enlaces a vistas.

Tags: representación visual de etiquetas.

UI: Interfaz de usuario.

UIKit: Framework de iOS que maneja los componentes visuales y el ciclo de vida de las vistas al desarrollar una aplicación.

VIP: Arquitectura que representa vista, interactor y presentador.

View: Vista. En el contexto de iOS comprende un elemento de la interfaz de usuario.

ViewController: Porción de una aplicación iOS que contiene funcionalidad e interfaz de usuario.

Worker: Es el componente de código que realiza las peticiones a los servicios y los envía al interactor.

6. BIBLIOGRAFÍA

- [1] Alcaldía de Barranquilla. (2021, 7 mayo). *Decretos durante la emergencia – Alcaldía de Barranquilla*. Recuperado 25 de octubre de 2021, de <https://www.barranquilla.gov.co/salud/coronavirus/decretos-durante-la-emergencia-coronavirus-barranquilla>
- [2] Delgans, L. H. (2020, 30 agosto). *Barranquilla está lista para la reapertura, aseguran expertos*. El Tiempo. Recuperado 10 de octubre de 2021, de <https://www.eltiempo.com/colombia/barranquilla/barranquilla-lista-para-la-reapertura-534586>
- [3] Minsalud. (s. f.). *Plan nacional de vacunación*. Ministerio de Salud Colombiano. Recuperado 10 de octubre de 2021, de <https://www.minsalud.gov.co/Paginas/Asi-avanza-el-Plan-Nacional-de-Vacunacion-contr-el-covid-1.aspx>
- [4] Google Maps. (2021, 18 octubre). *Google Maps - trafico y comida*. App Store. Recuperado 10 de octubre de 2021, de <https://apps.apple.com/co/app/google-maps-trafico-y-comida/id585027354>
- [5] Apple. (2021, 15 octubre). *Mapas*. App Store. Recuperado 10 de octubre de 2021, de <https://apps.apple.com/co/app/mapas/id915056765>
- [6] Tripadvisor. (2021, 20 octubre). *Tripadvisor: planifica viajes*. App Store. Recuperado 10 de octubre de 2021, de <https://apps.apple.com/co/app/tripadvisor-planifica-viajes/id284876795>
- [7] Meetup. (2010, 12 junio). *Meetup: Local groups & events*. App Store. Recuperado 10 de octubre de 2021, de <https://apps.apple.com/us/app/meetup-local-groups-events/id375990038>
- [8] colaboradores de Wikipedia. (2021, 14 julio). *SOLID*. Wikipedia, la enciclopedia libre. Recuperado 10 de octubre de 2021, de <https://es.wikipedia.org/wiki/SOLID>
- [9] Lukacs, M. (2019, 15 julio). *Part 1: What is clean swift(VIP), a brief overview - Martin Lukacs*. Medium. Recuperado 10 de octubre de 2021, de https://medium.com/@mlukacs_42903/part-1-what-is-clean-swift-vip-a-brief-overview-a43d4d66e664
- [10] Pluralsight. (s. f.). *Protocol oriented programming*. Recuperado 10 de octubre de 2021, de <https://www.pluralsight.com/guides/protocol-oriented-programming-in-swift>

[11] colaboradores de Wikipedia. (2021, 17 marzo). *Inyección de dependencias*. Wikipedia, la enciclopedia libre. Recuperado 10 de octubre de 2021, de https://es.wikipedia.org/wiki/Inyecci%C3%B3n_de_dependencias

[12] colaboradores de Wikipedia. (2021b, septiembre 20). *Desarrollo en espiral*. Wikipedia, la enciclopedia libre. Recuperado 10 de octubre de 2021, de https://es.wikipedia.org/wiki/Desarrollo_en_espiral

[13] Atlassian. (s. f.). *What is Agile?* Recuperado 10 de octubre de 2021, de <https://www.atlassian.com/agile>

[14] Atlassian. (s. f.-a). *Scrum: qué es, cómo funciona y por qué es excelente*. Recuperado 10 de octubre de 2021, de <https://www.atlassian.com/es/agile/scrum>

7. ANEXOS

7.1 Transcripciones des las entrevistas

E1

Juan Roca, 28 años.

- *Responde y justifica tu respuesta en las siguientes preguntas:*
 - *¿Cuándo sales prefieres lugares abiertos o cerrados?*
Abiertos. Porque mi trabajo no es al aire libre y cuando tengo tiempo quiero salir.
 - *¿Qué actividades realizas o has realizado en lugares al aire libre?*
Salir a correr, porque estoy en movimiento, puedo ver el paisaje mientras corro y me gusta tomar una ruta diferente cada vez.
 - *¿Utilizas alguna aplicación para descubrir lugares abiertos (al aire libre) en tu ciudad, si es así cual?*
Google maps nada más, porque me dice más o menos qué zonas son transitables. Pero esa app no se enfoca tanto en los corredores sino en los vehículos.
 - *¿Qué tan a menudo vas a lugares abiertos?*
Una o dos veces por semana. Porque tengo otros hobbies que me gusta hacer en casa.
 - *¿Elegirías lugares abiertos para planear salidas con amigos?*
Sí, porque es más fácil y más flexible tener un punto de encuentro al aire libre.
 - *¿Qué medios de comunicación utilizas para comunicar ese tipo de eventos?*
Whatsapp. Porque es donde está todo el mundo.
 - *¿Crees que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil para ti?*
La verdad es que no. Porque me parece más cómodo escribir por whatsapp donde quieren quedar cada uno y que me lo digan directamente. Como mi grupo de amigos es pequeño siento que es más fácil de esa forma.
 - *¿Para qué otro tipo de personas crees que sería útil una app de este estilo?*
Creo que esta app serviría mucho a los scouts, u organizaciones que normalmente hacen actividades al aire libre. Grupos de personas que hagan actividades físicas precisamente porque es mucho más difícil organizar grupos grandes de personas por whatsapp.

E2

Javier Salgado. 28 años.

- *Responde y justifica tu respuesta en las siguientes preguntas:*
 - *¿Cuándo sales prefieres lugares abiertos o cerrados?*
Depende, en temporadas normales un 50/50... pero por salud y Covid, en este momento prefiero un poco lugares abiertos.
 - *¿Qué actividades realizas o has realizado en lugares al aire libre?*
Salir a correr, a jugar Pokémon Go (incluyendo eventos), ver un partido, o a tomar cervezas y hablar con amigos.
 - *¿Utilizas alguna aplicación para descubrir lugares abiertos (al aire libre) en tu ciudad, si es así cual?*
No. Eventualmente, con eventos de Pokémon Go, por WhatsApp nos ponemos de acuerdo.
 - *¿Qué tan a menudo vas a lugares abiertos?*
De 3 a 5 veces por semana a correr. Eventualmente fines de semana.
 - *¿Elegirías lugares abiertos para planear salidas con amigos?*
Sí, ejemplo, ver un partido o solo hablar con amigos para tomar unas cervezas.
 - *¿Qué medios de comunicación utilizas para comunicar ese tipo de eventos?*
WhatsApp o llamada si la persona no responde, porque es más rápido de contactar para mí.
 - *¿Crees que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil para ti?*
Podría ser útil, más aún en el contexto del Pokémon Go, para cuadrar encuentros de eventos. Actualmente cuadrarnos por WhatsApp porque tenemos un grupo.
 - *¿Para qué otro tipo de personas crees que sería útil una app de este estilo?*
Para personas que organizan eventos deportivos, tipo salidas a correr, a montar bicicleta, hacer deporte en los parques y para los grupos que practican danza en las plazas.

E3

Lorena Gómez. 55 años.

- *Responde y justifica tu respuesta en las siguientes preguntas:*
 - *¿Cuándo sales prefieres lugares abiertos o cerrados?*
Prefiero lugares abiertos, porque prefiero evitar contagios.
 - *¿Qué actividades realizas o has realizado en lugares al aire libre?*

Caminatas porque no tengo que ir al gimnasio y reuniones al aire libre.

- *¿Utilizas alguna aplicación para descubrir lugares abiertos (al aire libre) en tu ciudad, si es así cual?*
Instagram, facebook y whatsapp, porque están todos mis amigos.
- *¿Qué tan a menudo vas a lugares abiertos?*
Los fines de semana para distraerme y todos los días para hacer ejercicio.
- *¿Elegirías lugares abiertos para planear salidas con amigos?*
Sí. Porque se siente más seguro.
- *¿Qué medios de comunicación utilizas para comunicar ese tipo de eventos?*
Instagram, whatsapp o facebook, porque son las únicas apps que utilizo para conectarme con la gente.
- *¿Crees que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil para ti?*
Sí porque podría enterarme de los eventos de carnaval por ejemplo.
- *¿Para qué otro tipo de personas crees que sería útil una app de este estilo?*
Todas las personas que desean tener actividades físicas deportivas o recreativas de todo tipo de edad.

E4

Patricia Niño. 21 años.

- *Responde y justifica tu respuesta en las siguientes preguntas:*
 - *¿Cuándo sales prefieres lugares abiertos o cerrados?*
Últimamente prefiero lugares al aire libre, sobre todo porque estoy retomando el hobby de patinar. También me gusta por ejemplo ir a un restaurante y mirar si tiene terraza por el tema del covid.
 - *¿Qué actividades realizas o has realizado en lugares al aire libre?*
Ir a patinar, llevar a mi perro a pasear. Pues es que acá no hay sitios cerrados para patinar. Y lo del perro obviamente porque necesita salir a hacer sus cosas.
 - *¿Utilizas alguna aplicación para descubrir lugares abiertos (al aire libre) en tu ciudad, si es así cual?*
Pues para descubrir creo que instagram o lo que me digan mis amigos en whatsapp, pero si quiero buscar un parque o algo por el estilo uso google maps, porque me dice como llegar.
 - *¿Qué tan a menudo vas a lugares abiertos?*

Casi todos los días. Porque voy al parque con mi perro y a veces voy a patinar.

- *¿Elegirías lugares abiertos para planear salidas con amigos?*
Sí. Porque ahora con el tema del covid muchos lugares cerrados dejaron de permitir más de 6 personas reunidas, y pues me da como asco si veo el sitio lleno.
- *¿Qué medios de comunicación utilizas para comunicar ese tipo de eventos?*
Whatsapp. Porque tengo mi grupo de amigos ahí.
- *¿Crees que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil para tí?*
Pues, yo diría que para ver qué cosas hay para hacer en la ciudad y conocer gente, por ejemplo que patina, estaría bien.
- *¿Para qué otro tipo de personas crees que sería útil una app de este estilo?*
Para la gente que cuadra partidos y se pone en los parques a hacer volleyball o fútbol. O que hacen skate.

E3

Melany Gonzalez.

- *Responde y justifica tu respuesta en las siguientes preguntas:*
 - *¿Cuándo sales prefieres lugares abiertos o cerrados?*
Prefiero lugares abiertos si hay sombra y tengo la comodidad/facilidad de sentarme. Porque con el sol no salgo ni a la esquina. Y pues si está lloviendo tampoco.
 - *¿Qué actividades realizas o has realizado en lugares al aire libre?*
Trotar, caminar, sentarme con alguien a hablar, conciertos, manifestaciones, fumar, bailar, pasear las mascotas
 - *¿Utilizas alguna aplicación para descubrir lugares abiertos (al aire libre) en tu ciudad, si es así cual?*
En Facebook, y en menor medida Instagram. En páginas de la secretaría distrital de cultura a veces publican eventos que son al aire libre, ferias, conciertos, cosas así. O en páginas de casas culturales.
 - *¿Qué tan a menudo vas a lugares abiertos?*
Dos veces al mes cuando es por esparcimiento. Pata trotar son casi cinco días a la semana
 - *¿Elegirías lugares abiertos para planear salidas con amigos?*
Sí los elegiría. Pasar un rato en la playa o en un parque, acampar, jugar paintball, algún concierto. Esos son los planes que haría con amigos en lugares abiertos.

- *¿Qué medios de comunicación utilizas para comunicar ese tipo de eventos?*

Generalmente eso se planea por whatsapp.

- *¿Crees que una app que encapsule la creación de eventos con actividades y lugares al aire libre, sería útil para ti?*

Sí creo que sería útil, para mí es útil cualquier herramienta que ofrezca sugerencias de actividades que se puedan hacer en la ciudad. Porque de otra manera termina uno haciendo siempre lo mismo.

- *¿Para qué otro tipo de personas crees que sería útil una app de este estilo?*

Podría ser útil para personas que no conocen la ciudad. También para entusiastas de la naturaleza, los que observan aves o estrellas, por ejemplo. Para casi cualquiera.