



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

**Diseño e implementación de una aplicación para la
búsqueda de espacios de ocio con requisitos en
dispositivos móviles**



Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Campos Domínguez, Eric

Tutores: Castro Bleda, María José

Ruiz García, Juan Carlos

Curso 2018-2019

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Diseño de identidad visual de la aplicación **KidZone Valencia** realizado por Rosario Sanchis Font.



Resumen

Este proyecto consiste en la creación de una aplicación móvil desarrollada para sistemas operativos Android, cuya principal funcionalidad, es que cada usuario pueda crear, obtener información y compartir planes de ocio para familias con niños en Valencia.

Para el diseño y desarrollo de esta aplicación se ha estudiado el alcance que podríamos tener con respecto al tiempo disponible y el necesario para su realización. Se ha utilizado el entorno de desarrollo integrado oficial de Android llamado Android Studio como entorno de trabajo y una base de datos NoSQL flexible, escalable y en la nube denominada Cloud Firestore con el fin de almacenar y sincronizar los datos.

Palabras clave: Android, Android Studio, Firebase, aplicación, App, búsqueda, ocio, familia.

Abstract

This project consists of the creation of a mobile application developed for Android operating systems, whose main functionality is that each user can create, obtain information and share leisure plans for families with children in Valencia.

For the design and development of this application we have studied the scope that we could have with respect to the time available and the necessary for its realization. The official Android integrated development environment called Android Studio has been used as a working environment and a flexible, scalable NoSQL database in the cloud called Cloud Firestore in order to store and synchronize the data.

Keywords: Android, Android Studio, Firebase, application, App, search, leisure, family.

Resum

Aquest projecte consistix en la creació d'una aplicació mòbil desenvolupada per a sistemes operatius Android, la principal funcionalitat de la qual, és que cada usuari puga crear, obtindre informació i compartir plans d'oci per a famílies amb xiquets a València.

Per al disseny i desenvolupament d'aquesta aplicació s'ha estudiat l'abast que podríem tindre respecte al temps disponible i el necessari per a la seua realització. S'ha utilitzat l'entorn de desenvolupament integrat oficial d'Android anomenat Android Studio com a entorn de treball i una base de dades NoSQL flexible, escalable i en el núvol denominada Cloud Firestore amb la finalitat d'emmagatzemar i sincronitzar les dades.

Paraules clau: Android, Android Studio, Firebase, aplicació, App, cerca, oci, família.

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Tabla de contenidos

1. Introducción	13
1.1 Motivación	13
1.2 Objetivos	14
1.3 Organización del documento	14
1.4 Siglas y acrónimos	15
2. Estado del arte	17
2.1 Aplicaciones similares	17
2.2 Comparativa entre las aplicaciones	22
3. Diseño centrado en el usuario	25
3.1 Formulario	25
3.2 Respuestas de los usuarios	27
3.3 Conclusión	30
4. Especificación de requisitos	31
4.1 Análisis	31
4.1.1 Requisitos funcionales	31
4.1.2 Requisitos no funcionales	32
4.2 Casos de uso	33
4.2.1 Diagrama de casos de uso	33
4.2.2 Descripciones de los casos de uso	35
4.3 Bocetos de interfaz	57
4.4 Modelo de datos	62
5. Desarrollo e implementación	63
5.1 Java	63
5.2 Android	63
5.3 Android Studio	64
5.4 Aplicación Android	65
5.4.1 Processes and Threads	65
5.4.2 Manifest	65
5.4.3 Permissions	66
5.4.4 Activities	66
5.4.5 Layouts	67
5.4.6 Fragments	68



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

5.4.7 Menus	69
5.4.8 ViewPager	70
5.4.9 BottomNavigationView	71
5.4.10 Material Design	72
5.4.11 Input Events: Event Listeners.....	75
5.4.12 Toasts	75
5.4.13 Dialogs.....	75
5.4.14 Styles and Themes	76
5.4.15 App resources.....	77
5.4.16 Google Maps Android API.....	78
5.5 Firebase Google.....	79
5.5.1 Firebase Authentication	79
5.5.2 Cloud Firestore	80
5.5.3 Cloud Storage.....	81
5.6 Picasso	82
5.7 Identidad visual	83
6. Conclusiones	85
7. Ampliaciones y mejoras futuras	87
7.1 Chat	87
7.2 Menú de planes.....	87
7.3 Notificaciones.....	87
7.4 Cuenta	87
7.5 Seguidos y seguidores	87
7.6 Ayuda.....	88
7.7 Ajustes	88
7.8 Analíticas.....	88
7.9 Mantenimiento.....	88
7.10 Migración.....	88
7.11 Monetización	88
Referencias	89

Tabla de figuras

Figura 1. Capturas de pantalla de la aplicación Planandgo.....	17
Figura 2. Capturas de pantalla de la aplicación ForTwo.....	18
Figura 3. Capturas de pantalla de la aplicación Murciano.....	19
Figura 4. Capturas de pantalla de la aplicación La diversiva.....	20
Figura 5. Capturas de pantalla de la aplicación Paracrios.....	20
Figura 6. Capturas de pantalla de la aplicación Viaparents Kids Friendly	21
Figura 7. Capturas de pantalla de la aplicación Winnie.....	22
Figura 8. Portada formulario	25
Figura 9. Pregunta del formulario número 1	26
Figura 10. Pregunta del formulario número 2.....	26
Figura 11. Pregunta del formulario número 3.....	26
Figura 12. Pregunta del formulario número 4.....	27
Figura 13. Encuesta enviada por WhatsApp	27
Figura 14. Respuestas del formulario a la pregunta número 1	28
Figura 15. Respuestas del formulario a la pregunta número 2	28
Figura 16. Respuestas del formulario a la pregunta número 3: Parte 1.....	29
Figura 17. Respuestas del formulario a la pregunta número 3: Parte 2.....	29
Figura 18. Respuestas del formulario a la pregunta número 4.....	29
Figura 19. Casos de uso general	33
Figura 20. Casos de uso apartado “Visualizar menú de planes”	33
Figura 21. Casos de uso apartado “Acceder a chat”	34
Figura 22. Casos de uso apartado “Acceder a mis planes”	34
Figura 23. Casos de uso apartado “Acceder a mi cuenta”	34
Figura 24. Bienvenida: Ventana 1	57
Figura 25. Bienvenida: Ventana 2.....	57
Figura 26. Registro	57
Figura 27. Inicio de sesión.....	57
Figura 28. Restablecer contraseña	57
Figura 29. Menú de planes..	57
Figura 30. Lista de planes de un apartado del menú de planes	58
Figura 31. Plan seleccionado vista usuario autenticado	58
Figura 32. Plan seleccionado vista usuario no autenticado	58
Figura 33. Información del plan seleccionado.....	58
Figura 34. Visualización mapa.....	58
Figura 35. Comentarios vista usuario autenticado.....	58
Figura 36. Comentarios vista usuario no autenticado.....	59
Figura 37. Listado de chats	59
Figura 38. Chat abierto	59
Figura 39. Lista de contactos: Mis contactos.....	59
Figura 40. Lista de contactos: Bloqueados.....	59
Figura 41. Añadir contacto.....	59

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Figura 42. Mis planes: Favoritos	60
Figura 43. Mis planes: Creados por mí	60
Figura 44. Crear un plan: Foto y título.....	60
Figura 45. Crear un plan: Descripción y tipo.....	60
Figura 46. Crear un plan: Dirección y plazas.....	60
Figura 47. Crear un plan: Fechas y horarios.....	60
Figura 48. Crear un plan: Edad, precio, audio y vídeo	61
Figura 49. Crear un plan: Vista previa/publicar	61
Figura 50. Crear un plan: Visualización previa	61
Figura 51. Notificaciones... ..	61
Figura 52. Perfil: Vista usuario propietario	61
Figura 53. Perfil: Vista otro usuario	61
Figura 54. Planes creados (vista otro usuario)	62
Figura 55. Modelo de datos	62
Figura 56. Emulador Android Studio	64
Figura 57. Manifest (Android Studio).....	66
Figura 58. Activity (Android Studio).....	67
Figura 59. Layout (Android Studio)	68
Figura 60. Fragment (Android Studio).....	69
Figura 61. Uso de fragments (Android Studio)	69
Figura 62. Menu (Android Studio)	70
Figura 63. Menu (Aplicación)	70
Figura 64. ViewPager Adapter (Android Studio).....	71
Figura 65. ViewPager (Aplicación)	71
Figura 66. BottomNavigationView (Android Studio)	72
Figura 67. BottomNavigationView (Aplicación).....	72
Figura 68. FloatingActionButton (Aplicación)	73
Figura 69. CardView (Aplicación).....	73
Figura 70. RecyclerView (Android Studio).....	74
Figura 71. RecyclerView (Aplicación).....	74
Figura 72. Event Listener Android Studio.....	75
Figura 73. Toast (Android Studio)	75
Figura 74. Toast (Aplicación)	75
Figura 75. Dialog (Android Studio)	76
Figura 76. Dialog (Aplicación)	76
Figura 77. Styles and Themes (Android Studio).....	76
Figura 78. Styles and Themes (Aplicación).....	77
Figura 79. Resources (Android Studio)	77
Figura 80. Strings (Android Studio)	77
Figura 81. Google Maps (Aplicación)	78
Figura 82. Firebase Authentication	79
Figura 83. Cloud Firestore	81
Figura 84. Cloud Storage.....	82
Figura 85. Compresión de imágenes	82
Figura 86. Picasso (Android Studio).....	83
Figura 87. Identidad visual corporativa App KidZone Valencia.....	84

Tablas

Tabla 1. Comparativa de aplicaciones	23
Tabla 2. CU Registrar usuario.....	35
Tabla 3. CU Identificar usuario.....	35
Tabla 4. CU Restablecer contraseña	36
Tabla 5. CU Visualizar menú de planes	36
Tabla 6. CU Visualizar lista de planes de cada apartado	37
Tabla 7. CU Visualizar plan seleccionado	37
Tabla 8. CU Visualizar información	38
Tabla 9. CU Visualizar mapa	38
Tabla 10. CU Utilizar funciones de localización.....	39
Tabla 11. CU Acceder a otro plan	39
Tabla 12. CU Visualizar comentarios	40
Tabla 13. CU Añadir comentario	40
Tabla 14. CU Borrar comentario propio	41
Tabla 15. CU Dar like.....	41
Tabla 16. CU Quitar like.....	42
Tabla 17. CU Compartir por redes sociales	42
Tabla 18. CU Añadir a favoritos	43
Tabla 19. CU Acceder a “Chat”	43
Tabla 20. CU Visualizar lista de chats.....	44
Tabla 21. CU Entrar a chat	44
Tabla 22. CU Visualizar mensajes	45
Tabla 23. CU Enviar mensajes	45
Tabla 24. CU Visualizar lista de usuarios.....	46
Tabla 25. CU Crear chat individual	46
Tabla 26. CU Salir del chat	47
Tabla 27. CU Acceder a “Mis planes”	47
Tabla 28. CU Ver lista de favoritos.....	48
Tabla 29. CU Eliminar plan de favoritos.....	48
Tabla 30. CU Eliminar todos los planes de favoritos	49
Tabla 31. CU Acceder a plan de favoritos.....	49
Tabla 32. CU Ver lista de “creados por mí”	50
Tabla 33. CU Eliminar plan de “creados por mí”	50
Tabla 34. CU Eliminar todos los planes de “creados por mí”	51
Tabla 35. CU Acceder a plan de “creados por mí”	51
Tabla 36. CU Editar plan.....	52
Tabla 37. CU Crear nuevo plan	52
Tabla 38. CU Acceder a “Mi cuenta”	53
Tabla 39. CU Ver información del perfil.....	53
Tabla 40. CU Editar perfil.....	54
Tabla 41. CU Cambiar nombre	54
Tabla 42. CU Cambiar foto	55
Tabla 43. CU Cambiar biografía	55
Tabla 44. CU Cerrar sesión	56



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

1. Introducción

Estamos en pleno siglo XXI y es una realidad que el uso de la tecnología está cada vez más presente en nuestro día a día. Las aplicaciones móviles ofrecen una experiencia de usuario personalizada y una gran variedad de funcionalidades, lo que ha provocado que los usuarios usen las Apps cada vez con mayor frecuencia a lo largo del día y durante más tiempo.

Las aplicaciones de redes sociales, mensajería instantánea y ocio son las favoritas de los usuarios. Más del 80% de los minutos invertidos en el uso de Apps móviles, en todos los mercados, se consume en este tipo de aplicaciones. Esto ha propiciado que el dispositivo móvil se haya consolidado como el primer dispositivo de uso en todo el mundo. El desarrollo de las aplicaciones móviles, o Apps, está en auge y su importancia es tal, que actualmente constituye una parte relevante dentro la economía mundial (1).

En España se ha incrementado en un 5% el uso de los dispositivos móviles en el año 2018, teniendo en cuenta que un 96% de ciudadanos tiene un teléfono móvil y de ellos, un 87% son smartphones (2).

El crecimiento en el uso de dispositivos móviles es evidente. La frecuencia con la que los usuarios emplean el dispositivo móvil hace descender la frecuencia de utilización de otros dispositivos tales como los ordenadores (2).

1.1 Motivación

El principal motivo por el que se ha llevado a cabo este proyecto es ayudar, en la medida de lo posible, a las familias que estén en Valencia a encontrar buenos lugares que visitar y buenas actividades a realizar con los niños. Entre éstos podemos destacar algunos como: parques de bolas, teatros infantiles, piscinas infantiles, restaurantes con zonas para niños, etc.

Otro de los motivos por los que se ha llevado a cabo este proyecto ha sido el auge que está teniendo actualmente la tecnología utilizada para el desarrollo de aplicaciones Android, así como la utilización de smartphones y todas las posibilidades y funcionalidades que ofrecen.

Por último, cabe destacar el hecho de que a lo largo de la carrera solo se haya enseñado programación en dispositivos móviles en una asignatura de modalidad optativa, por lo que diseñar y desarrollar esta aplicación desde cero es una buena oportunidad para poner en práctica los conocimientos ya adquiridos en esta asignatura, relacionarlos con los asimilados en otras materias de la carrera y desarrollar nuevas habilidades y competencias ligadas al planteamiento y realización de proyectos.

1.2 Objetivos

El objetivo principal de este proyecto es el diseño y desarrollo de una aplicación móvil **-KidZone Valencia-** para el sistema operativo Android que se distribuya de manera gratuita a través de la tienda de aplicaciones Google Play.

Los objetivos específicos de este proyecto son los siguientes:

- Diseñar y desarrollar una aplicación móvil internacionalizada, es decir, que se adapte al idioma del dispositivo (castellano, valenciano e inglés en nuestro caso) y a los usos y costumbres asociados a cada zona (*locale* en inglés), tales como el uso de puntos, comas, escritura y lectura a izquierda o derecha.
- Facilitar la búsqueda de lugares de ocio para niños.
- Permitir a los propios usuarios crear y guardar planes, así como valorar y almacenar los que no sean suyos.
- Permitir la comunicación entre usuarios a través de chats y/o comentarios.
- Notificar a los usuarios informaciones acerca de planes y chats.
- Evaluar la acogida de la aplicación mediante pruebas piloto.
- Poner en práctica los conocimientos adquiridos durante la carrera en lo que respecta a software, bases de datos y diseño centrado en el usuario.

1.3 Organización del documento

En este apartado presentamos la organización del documento, estructurado en los siguientes capítulos:

1. Introducción: En este capítulo se realiza la introducción, la motivación, los objetivos y la estructura del proyecto.
2. Estado del arte: En esta sección se presenta una recopilación de aplicaciones que tienen relación con la nuestra, así como lo un análisis de sus características.
3. Diseño centrado en el usuario: En este capítulo se muestra una encuesta realizada a potenciales usuario de la App y los resultados y las conclusiones que se han obtenido de ésta.
4. Especificación de requisitos: En este punto se hace una descripción completa del comportamiento de la aplicación.

5. Desarrollo e implementación: Durante este apartado se describe el proceso de desarrollo e implementación junto con los aspectos técnicos.
6. Conclusiones: En este punto se detallan las conclusiones que se han obtenido tras la finalización del proyecto.
7. Ampliaciones y mejoras futuras: Durante este apartado se explican las posibles ampliaciones y mejoras a realizar en el futuro con respecto a este proyecto.
8. Referencias: En esta sección se incluyen las referencias de la documentación utilizada en este proyecto.

1.4 Siglas y acrónimos

A continuación, se exponen las siglas y los acrónimos utilizados durante este documento.

App (Application): Aplicación, es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas (3).

SDK (Software Development Kit): Kit de desarrollo de software, es un conjunto de herramientas de desarrollo de software que permite a un desarrollador de software crear una aplicación informática para un sistema concreto (4).

API (Application Programming Interface): Interfaz de programación de aplicaciones, es un conjunto de subrutinas, funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción (5).

UI (User Interface): Interfaz de usuario, es el medio con que el usuario puede comunicarse con una máquina, equipo, computadora o dispositivo, y comprende todos los puntos de contacto entre el usuario y el equipo (6).

GUI (Graphical User Interface): Interfaz gráfica de usuario, es un programa informático que actúa de interfaz de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interfaz (7).

MVP (Minimum Viable Product): Producto viable mínimo, es un producto con suficientes características para satisfacer a los clientes iniciales, y proporcionar retroalimentación para el desarrollo futuro (8).

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

2. Estado del arte

En este apartado se va a introducir un análisis de las aplicaciones existentes que tienen una temática similar a la aplicación que se propone desarrollar en este proyecto. Este análisis se realizó el 25 de febrero de 2019. Una vez finalizado este análisis se va a realizar una comparativa entre las distintas alternativas presentadas, con el objetivo de identificar un mínimo común denominador que aglutine las principales características de este tipo de Apps.

2.1 Aplicaciones similares

Planandgo

Es una aplicación gratuita destinada a que los usuarios puedan crear y unirse a planes (actividades, quedadas, viajes, etc) para así disfrutar de buenas experiencias y poder compartirlas. Tiene más de 50.000 descargas y una valoración media de 4'2/5. Su página web se puede consultar en <https://www.planandgoapp.com/>.

Requiere registrarse e iniciar sesión con alguna cuenta (Google, Facebook o Email) para poder acceder a todos los servicios y aceptar permisos de localización en caso de querer optar a todas las funcionalidades de la localización en el mapa. Si no se inicia sesión, únicamente se dispone de una vista parcial de los planes. Ofrece diversos servicios, tales como conectar con los amigos a través de las redes sociales, buscar/crear planes, ver fotos, descripciones, comentarios, valoraciones, chatear, localización en el mapa y más (véase figura 1).

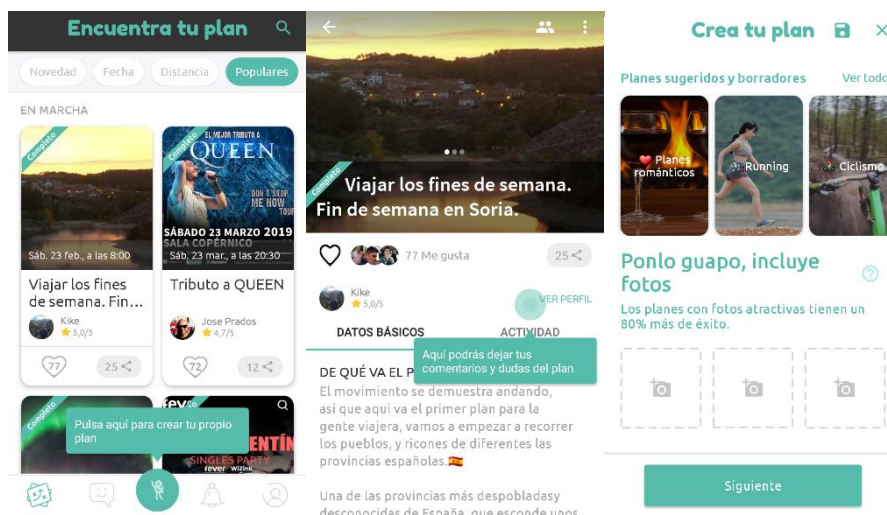


Figura 1. Capturas de pantalla de la aplicación Planandgo

ForTwo

Es una aplicación gratuita destinada a la realización de regalos, escapadas y planes en pareja. Aunque no es una App específicamente centrada en ofrecer planes para niños, su temática es muy próxima y por ella se ha incluido en este estudio. Tiene más de 10.000 descargas y una valoración media de 4'5/5. Su página web se puede consultar en <https://fortwo.es/>.

Igual que en la App anterior, requiere permisos de localización, registrarse e iniciar sesión, aunque ésta ofrece muchos servicios sin necesidad de registrarse. Muestra un menú inicial en donde se clasifican los distintos tipos de actividades ofertadas junto con imágenes que facilitan el reconocimiento de las actividades y hacen más amigable y rápida la interacción con el listado ofrecido. A su vez, cada tipo actividad general se refina en actividades más concretas a través de las distintas opciones recogidas en el menú principal.

Para cada actividad concreta se ofrece una descripción en la cual se informa del precio (si lo hubiera), lugar, fecha (o fechas) y la localización en el mapa. Dicha actividad se puede compartir a través de las redes sociales. No ofrece valoraciones de los usuarios. Este tipo de actividades son comerciales, destinadas a negocios/empresas. Esto se justifica por el hecho de que la App está orientada hacia actividades de tipo comercial y, por tanto, los anunciantes no son particulares sino negocios y/o empresas. Cabe finalmente destacar que la App incluye publicidad tal y como se muestra en la figura 2.

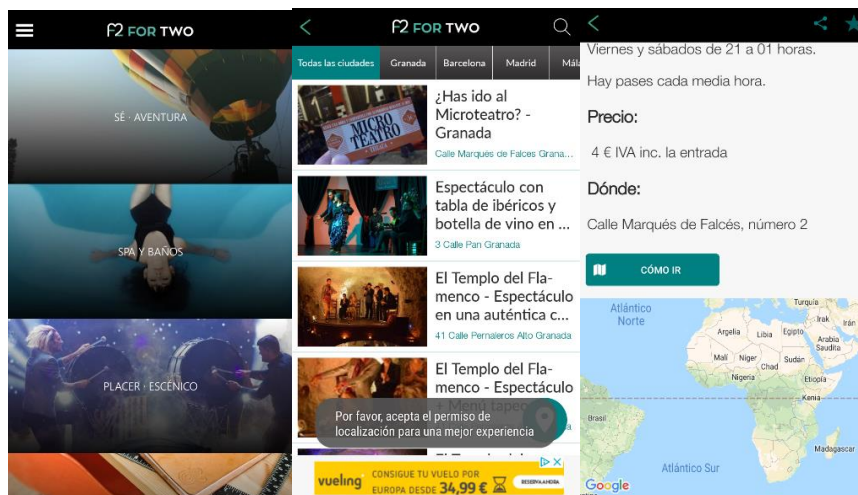


Figura 2. Capturas de pantalla de la aplicación ForTwo

Murciano

Es una aplicación gratuita, con una agenda cultural y de ocio, en la que se pueden encontrar eventos y lugares interesantes, curiosos y desconocidos de la Región de Murcia. Tiene más de 5.000 descargas y una valoración media de 4'9/5. Podemos consultar su página web en <http://www.murciano.es/>.

La App no requiere registrarse e iniciar sesión, aunque no hacerlo implica no optar a todas las funcionalidades ofrecidas. Las actividades están clasificadas según su tipo y

es posible acceder a ellas también a través de su localización en el mapa. Tampoco ofrece valoraciones de los usuarios. Este tipo de actividades son comerciales en la mayoría de los casos, publicadas por negocios y/o empresas. En cada actividad se dispone de una descripción, formas de contactar y formas de hacer una reserva y es posible compartirla a través de las redes sociales. Al ser gratuita, también incluye publicidad. Podemos observar partes de esta aplicación en la figura 3.

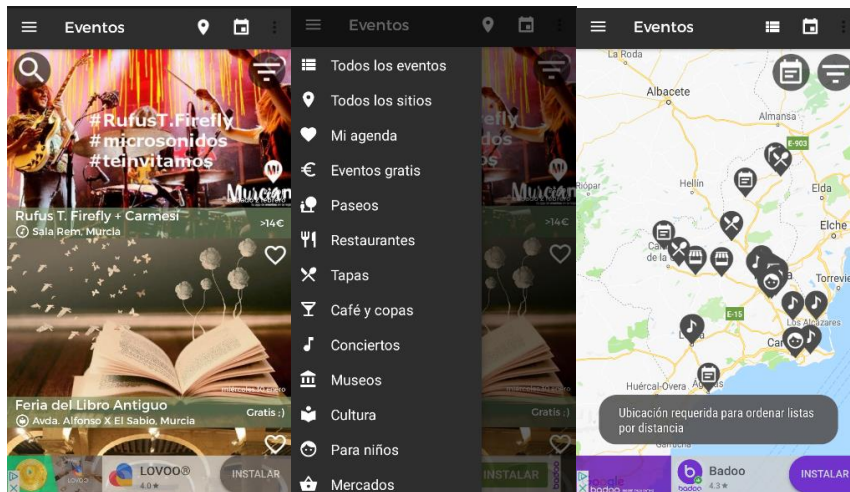


Figura 3. Capturas de pantalla de la aplicación Murciano

La diversiva

Es una aplicación gratuita que ofrece actividades para niños y planes para disfrutar en familia dentro de la provincia de Málaga. Tiene más de 100 descargas y una valoración media de 5/5. Se puede consultar su página web en <https://ladiversiva.com/>.

Tampoco requiere registrarse e iniciar sesión ni aceptar permisos de localización, pero hacerlo implicaría optar a más funcionalidades. Muestra tanto actividades ofrecidas de forma gratuita tales como talleres y actividades que son realizadas por empresas. Las actividades incluyen una descripción junto con la localización en el mapa. Es posible compartirlas a través de las redes sociales. Tampoco ofrece comentarios/valoraciones de los usuarios. Se puede ver partes de esta aplicación en la figura 4.

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

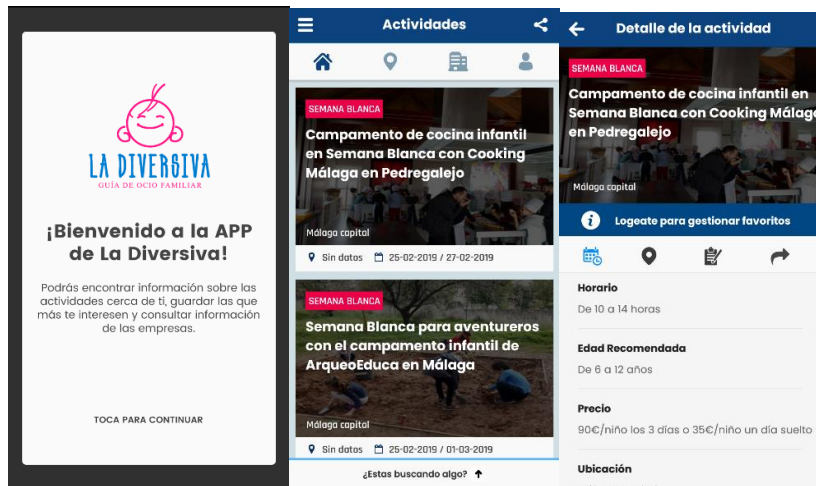


Figura 4. Capturas de pantalla de la aplicación La diversiva

Paracrios

También es una aplicación gratuita que ofrece actividades para niños y planes para disfrutar en familia. Tiene más de 100 descargas y una valoración media de 5/5. Podemos consular su página web en <https://paracrios.com/>.

No requiere registrarse ni iniciar sesión ni permisos de localización. Muestra un menú clasificando las actividades ofrecidas. Al acceder a un tipo de actividad, la App te redirige a una página web en donde se puede ver comentarios y valoraciones de la persona que ha añadido dicha información y bastante publicidad (véase figura 5).

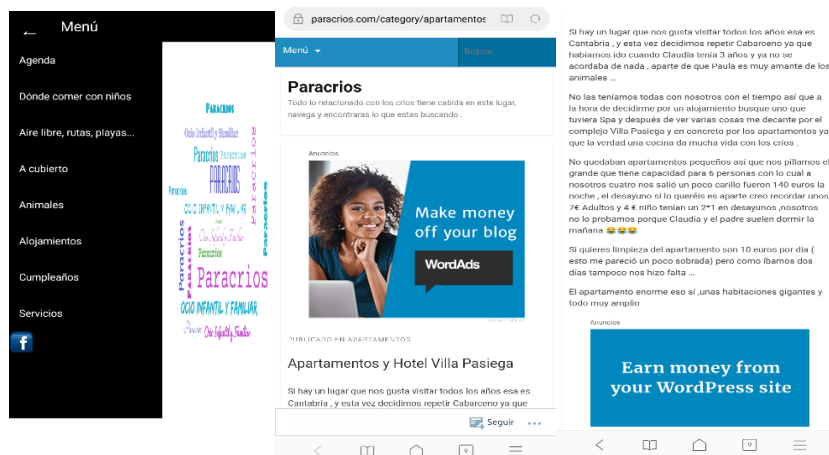


Figura 5. Capturas de pantalla de la aplicación Paracrios

Viaparents Kids Friendly

Es una aplicación gratuita destinada a que los padres y madres puedan compartir sus experiencias y ayudar así a crear un mundo más amigable para los niños en Francia. La aplicación está diseñada únicamente con el idioma francés. Tiene más de 1.000 descargas y una valoración media de 5/5. Se puede consultar su página web en <https://www.viaparents.com/>.

Igual que otras aplicaciones, ésta requiere el registro de sus usuarios y aceptar permisos de localización para acceder a todas las funcionalidades. En caso de no hacerlo, se permite optar a varios servicios, pero no a su totalidad. Se da la opción de acceder a las redes sociales en las que está presente la organización de la aplicación. Inicialmente, se muestra una pantalla con las distintas acciones que se pueden realizar, tales como buscar por proximidad, buscar en toda la App, ver favoritos, añadir actividades, ver las últimas visitas y acceder al perfil de los usuarios.

Al entrar en una actividad, se puede ver la foto asociada a dicha actividad, su descripción, valoración media, valoración por partes, experiencias de usuarios, datos de contacto, localización, la página web de la empresa que la ofrece (en caso de que haya). Podemos ver partes de esta aplicación en la figura 6.

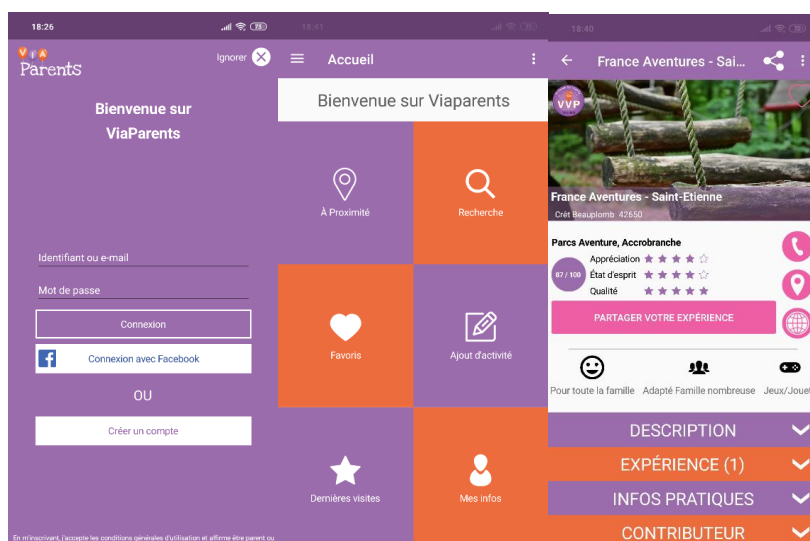


Figura 6. Capturas de pantalla de la aplicación Viaparents Kids Friendly

Winnie

Es una aplicación gratuita destinada a que los padres y madres se comuniquen entre ellos, compartan sus historias o simplemente encuentren nuevas cosas para hacer con sus hijos. La descarga de la App para dispositivos Android solo está disponible en la zona geográfica de Estados Unidos, pero puede accederse a la aplicación web desde España. La descarga para dispositivos iOS está disponible en España. Tiene más de 50.000 descargas y una valoración media de 4/5. Podemos consultar su página web en <https://winnie.com/>.

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Requiere registrarse para poder utilizar los servicios que ofrece. Para ciertos servicios también necesita permisos de localización. Inicialmente, muestra posts que han sido subidos y sus respectivas respuestas. Hay muchos foros destinados a muchos tipos de temas relacionados con los niños y la paternidad. Se pueden realizar búsquedas por temática, como son bebés, niños y hogar. Al seleccionar un foro, se pueden ver los comentarios, preguntas y respuestas, pertenecientes a éste. Se puede valorar la actividad no solo dándole un “me gusta” sino dejando un comentario en caso de desearlo.

Es posible realizar búsquedas no solo por tipo de actividad, sino también por tipo de lugar o localización como pueden ser parques, lugares para comer, jugar, comprar, viajar, etc. Dentro de esta selección, aparece un listado con las opciones disponibles, en las cuales se ve una foto, descripción general, categoría y dirección. Al seleccionar una opción, es posible acceder al sitio web, sugerir cambios en la opción, añadir fotos, dejar comentarios, guardar la opción, ver datos como el número de teléfono o la dirección o ver la localización en el mapa. Se puede ver partes de esta aplicación en la figura 7.

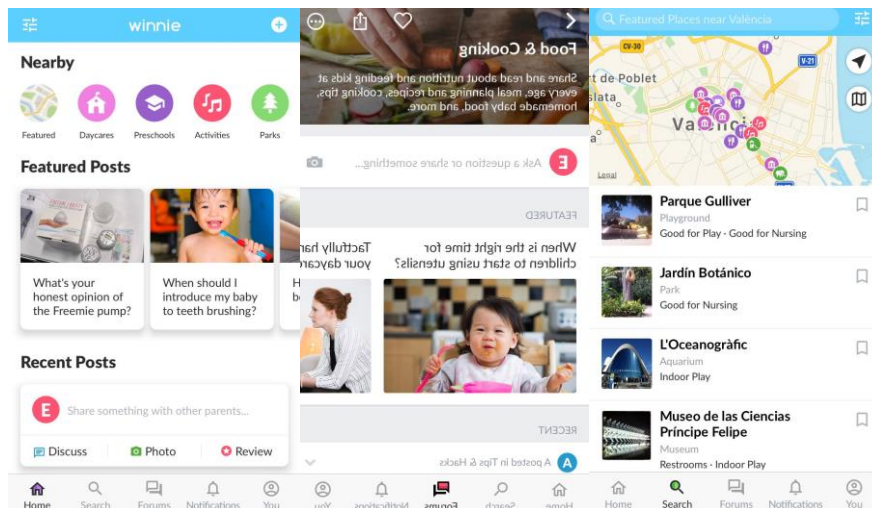


Figura 7. Capturas de pantalla de la aplicación Winnie

2.2 Comparativa entre las aplicaciones

En este apartado, vamos a realizar la comparativa entre las aplicaciones anteriormente descritas, destacando las características que consideramos más relevantes. Como hemos comentado antes, este análisis se realizó el 25 de febrero de 2019, con las características y el número de descargas que tenían entonces las aplicaciones. Se puede ver la comparativa en la tabla 1.

	aPG	aFT	aM	aLD	aPC	aVKF	aW
Gratis	Sí	Sí	Sí	Sí	Sí	Sí	Sí
Uso sin iniciar sesión	Sí	Sí	Sí	Sí	No	Sí	Sí
Registrarse e iniciar sesión obligatoriamente	No	No	No	No	Sí	No	No
Menús sencillos de búsqueda	Sí	Sí	Sí	No	Sí	No	Sí
Utilización de mapas	Sí	Sí	Sí	Sí	No	Sí	Sí
Valoración/comentarios de usuarios	Sí	No	No	No	Sí	Sí	Sí
Comunicación directa entre usuarios	Sí	No	No	No	No	No	Sí
Interfaz amigable	Sí	Sí	Sí	Sí	No	Sí	Sí
Facilidad de uso	Sí	Sí	Sí	Sí	No	Sí	Sí
Desarrollo en varios idiomas	Sí	Sí	Sí	No	No	No	No
Publicidad de otras empresas	No	Sí	Sí	No	Sí	No	No
Valoración media de la App sobre 5	4,2	4.5	4.9	5	5	5	4
Descargas de la App	+50K	+10K	+5K	+100	+100	+1K	+50K

Tabla 1. Comparativa de aplicaciones

aPG: Aplicación Planandgo

aFT: Aplicación ForTwo

aM: Aplicación Murciano

aLD: Aplicación La diversiva

aPC: Aplicación Paracrios

aVKF: Aplicación Viaparents Kids Friendly

aW: Aplicación Winnie

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Observando la tabla, llegamos a la conclusión de que todas las aplicaciones tienen una buena valoración, aunque las que presentan una mayor cantidad de descargas son aquellas que además de tener una interfaz amigable y buena facilidad de uso, utilizan mapas y permiten que los usuarios se comuniquen entre ellos.

3. Diseño centrado en el usuario

En este capítulo presentamos la encuesta que se pasó a potenciales usuarios de la App para tener en cuenta sus puntos de vista en el diseño de la aplicación. Presentaremos el formulario creado, un resumen de las respuestas obtenidas y las conclusiones extraídas.

3.1 Formulario

La finalidad de esta encuesta es conocer el punto de vista y las opiniones de algunos de los usuarios a los que va destinada nuestra aplicación para así terminar de perfilar la App y ofrecer a los usuarios lo que éstos esperan de la misma.

Por tanto, las respuestas influirán en la implementación y más concretamente en los tipos de planes ofrecidos, el tipo de autenticación ofrecido, etc. El formulario (véanse figuras 8-12) fue creado con Google Forms (9), una herramienta ofrecida por Google para el diseño y gestión de formularios, y enviado por el autor de este trabajo y sus tutores mediante un mensaje de WhatsApp a distintos usuarios y grupos potenciales de uso de la App.



Formulario de ayuda al diseño de ValenKidZone, una App para crear, informar y compartir planes de ocio para familias con niños en valencia

Bienvenid@, los resultados de este formulario se utilizarán para saber el punto de vista de los usuarios a los que va destinada la aplicación, para así realizar las mejoras que se consideren convenientes. Muchas gracias por tu participación.

*Obligatorio



ValenKidZone

Figura 8. Portada formulario

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

1. Iniciando sesión en la aplicación, el usuario podrá no solo consultar los planes ya existentes, sino también identificarse para añadir otros nuevos, como parques, rutas o cumpleaños. ¿Cómo preferirías iniciar sesión en la aplicación? *

- Con mi cuenta de Google
- Con mi cuenta de Facebook
- Creando una cuenta en la propia aplicación

Figura 9. Pregunta del formulario número 1

2. ¿Sobre qué planes de ocio te gustaría obtener información? *

- Novedades (los últimos planes añadidos)
- Populares (los que más valoraciones positivas tienen)
- Cerca de ti
- Restaurantes
- Alojamiento
- Cumpleaños
- Rutas y playas
- Parques
- Acuático
- Entretenimiento
- Eventos
- Ver mapa (que aparezcan en el mapa todos los planes con sus respectivas temáticas)

Figura 10. Pregunta del formulario número 2

3. ¿Añadirías algún tipo de plan de ocio que no aparezca en la lista anterior? Si es así, indícanos cuál

Tu respuesta

Figura 11. Pregunta del formulario número 3

4.Si quieres pues indicarnos cómo te gustaría que fuera la aplicación (cualquier sugerencia o comentario será bienvenido)

Tu respuesta

Figura 12. Pregunta del formulario número 4

3.2 Respuestas de los usuarios

La encuesta se envió durante los días 08/06/2019 – 15/06/2019, cuando ya se tenía pensada una primera versión de la App. Como se ha comentado anteriormente, el formulario se envió a usuarios y grupos de WhatsApp potencialmente interesados por la App. En la figura 13 se muestra el mensaje enviado. Se obtuvo un total de 37 respuestas.

En esta sección presentamos un análisis estadístico las respuestas obtenidas (véanse figuras 14-18), a través de las facilidades que nos aporta la herramienta Google Forms.

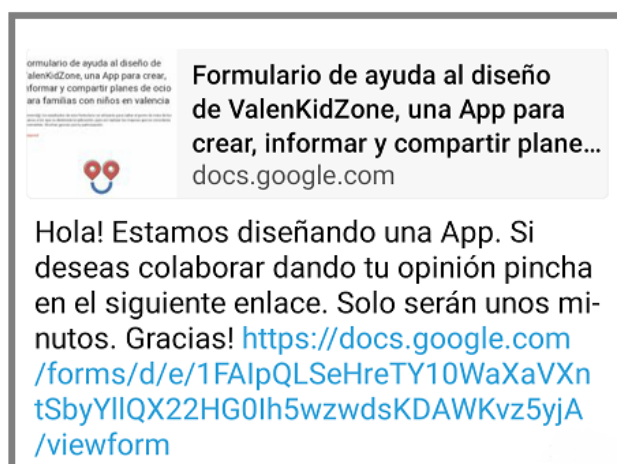


Figura 13. Encuesta enviada por WhatsApp

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

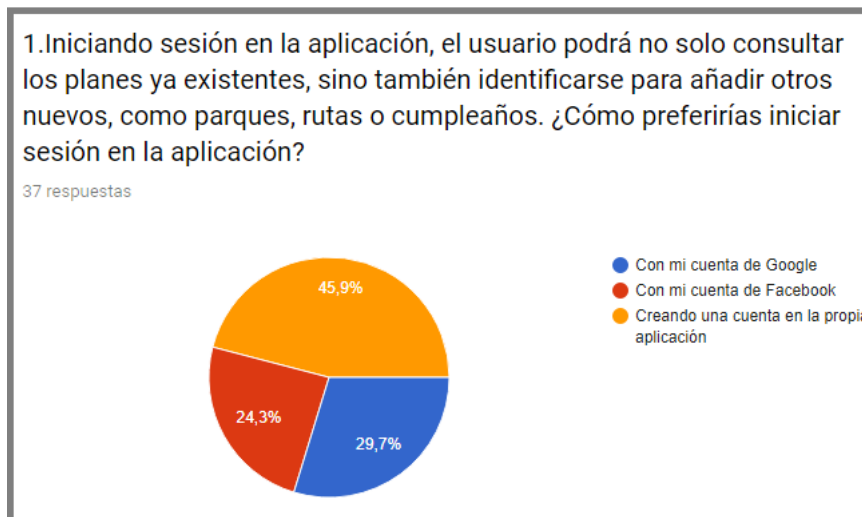


Figura 14. Respuestas del formulario a la pregunta número 1

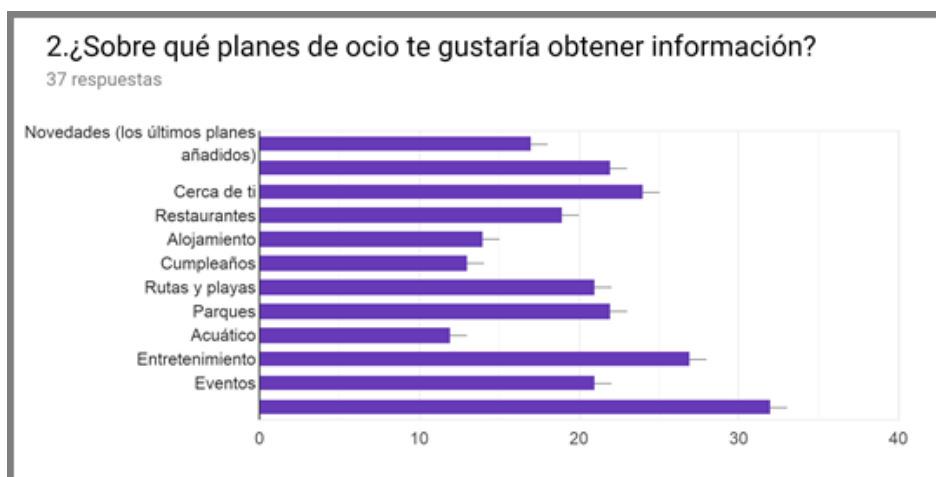


Figura 15. Respuestas del formulario a la pregunta número 2

3. ¿Añadirías algún tipo de plan de ocio que no aparezca en la lista anterior?
Si es así, indícanos cuál

13 respuestas

Eventos específicos: cuenta cuentos, manualidades, espectáculos de calle, clown,... Servicios infantiles de apoyo al ocio: canguros, tiendas, etc...

Teatro infantil

Conciertos, cuenta cuentos, obras de teatro, espacios lúdicos permanentes, exposiciones, museos, ludotecas, aunque entiendo q están comprendidos entre eventos y entretenimiento

Personalmente lo que mas me interesa son rutas en la naturaleza pata hacer con niños caminando o en bici , donde haya en algun momento una zona de picnic o mesas.

Teatro

Deportes

[REDACTED]

Deportes

Ludotecas

Parques de bolas. También por ente organiAdor. Pej. Para cosas que organizas el ayuntamienyo

Ferías

Figura 16. Respuestas del formulario a la pregunta número 3: Parte 1

Ferías

Rutas y playas

Deportes y aventura

Figura 17. Respuestas del formulario a la pregunta número 3: Parte 2

4. Si quieres pues indicarnos cómo te gustaría que fuera la aplicación (cualquier sugerencia o comentario será bienvenido)

10 respuestas

Usable "for dummies" y placentera.

En el tema del login, apple también lo va a incluir en su nueva versión de iOS 13, ya está disponible para desarrolladores.

Filtros por rango de edad

De fácil manejo

muy sencilla de manejar porque yo no tengo cuenta en redes sociales y no manejo mucho la tecnología

De color verde

Algo sobre seguridad, avisos o utilizar la propia red para que una simple pulsera bluetooth utilice la red para localizar a los peques

Que se enviará un boletín o avisos

Interocio

Que sea fácil e intuitiva

Figura 18. Respuestas del formulario a la pregunta número 4

3.3 Conclusión

Una vez obtenidas las respuestas, llegamos a la conclusión de que:

- La mayoría de los usuarios prefieren crearse una cuenta en la propia aplicación, pero también sería buena idea permitir iniciar sesión con la cuenta de Google o Facebook.
- La mayoría de tipo de planes de ocio tienen una buena valoración, destacando entre estos “Ver mapa”, “Cerca de ti” y “Entretenimiento”.
- Muchos de los tipos de planes a añadir sugeridos por los usuarios ya forman parte del menú de la versión inicial y se han sugerido otros, como por ejemplo “Deportes”, que se podrían incorporar.¹
- Se sugiere que la aplicación sea fácil e intuitiva, entre otras cosas. Esto indica que hay que hacer un esfuerzo importante para que el manejo de la App sea sencillo, pues los potenciales usuarios de la App ya saben que, en muchas ocasiones, Apps que les podrían interesar al final no se usan por estos motivos.

¹ Cabe destacar que los tutores de este trabajo son potenciales usuarios de la App, por lo que ya habían aportado ideas y sugerencias en el menú inicial.

4. Especificación de requisitos

Este capítulo se centra en la descripción completa del comportamiento del sistema, incluyendo un análisis de la aplicación, casos de uso, bocetos de interfaz y modelo de datos.

Las herramientas utilizadas para la realización de la especificación han sido:

- Draw.io para los casos de uso (10).
- NinjaMock para los bocetos de interfaz (11).
- VisualParadigmOnline para el modelo de datos (12).

4.1 Análisis

El análisis de requerimientos es una parte fundamental del proceso de desarrollo de una aplicación, cuya finalidad es obtener y detallar los requerimientos de funcionalidad y de calidad de servicio del producto, es nuestro caso, de una aplicación móvil Android.

4.1.1 Requisitos funcionales

Los requisitos funcionales describen cómo interacciona la aplicación con su entorno y cuáles van a ser su estado y funcionamiento.

Estos requisitos son:

- Autenticación de usuarios: Acceso a la aplicación de forma autenticada para usar todas las funcionalidades ofrecidas.
- Gestión de contenidos:
 - Perfiles: Información personal de los usuarios.
 - Menú de planes: Menú con los distintos planes ofrecidos en la aplicación, clasificados según su temática.
 - Planes favoritos: Planes almacenados en la lista de favoritos.
 - Planes creados: Planes creados por el usuario y almacenados en su lista de planes.
- Lenguaje: Tres idiomas que se aplican según la configuración del dispositivo: Castellano, valenciano e inglés.

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

- Mapas:
 - Mapa de Valencia: Mapa en el que se ubica el plan actual. Este mapa dispone de marcadores personalizados que destacan la ubicación de otros planes, así como información de cada plan. Para ello se utilizan las distintas funcionalidades ofrecidas por Google Maps.
 - Geolocalización: Mapa ofrecido para proporcionar la ubicación actual del dispositivo del usuario. Presenta las mismas características que el mapa descrito anteriormente.
- Comunicación entre usuarios:
 - Chats: Chats privados entre usuarios de la aplicación.
 - Comentarios: Comentarios introducidos por los usuarios para los diversos planes y que se ofrecen de manera pública a cualquier usuario que consulte cada plan.

4.1.2 Requisitos no funcionales

Los requisitos no funcionales describen los servicios que ha de ofrecer la aplicación y las restricciones asociadas a su funcionamiento.

- Disponibilidad: La aplicación debe estar disponible desde Google Play y, una vez instalada, ofrecer un funcionamiento robusto.
- Diseño y accesibilidad: La aplicación ha de seguir los patrones de diseño y accesibilidad definidos por Google.
- Usabilidad: Cualquier usuario autenticado ha de poder utilizar todas las funcionalidades disponibles.
- Escalabilidad: El uso de la aplicación ha de poder ser ampliado a más zonas geográficas de España, no solo a Valencia.
- Mantenimiento: La aplicación debe ser fácilmente mantenible y las actualizaciones deben ofrecerse a través de Google Play.
- Diseño: La aplicación debe tener interfaces gráficas claras y amigables, respetando el patrón de diseño de interfaces de Google.
- Integración: Se debe poder integrar la aplicación en cualquier dispositivo de la plataforma Android.
- Rendimiento y optimización: La aplicación debe funcionar de una forma óptima, disminuyendo los tiempos de espera al mínimo posible. El consumo de batería y de datos debe ser adecuado, así como las peticiones al servidor. Se debe utilizar una caché para las imágenes y se tiene que comprimir las imágenes antes de ser subidas al servidor.

4.2 Casos de uso

Comenzaremos a describir el comportamiento dinámico de la aplicación mediante las distintas interacciones entre el usuario y la App. Incluiremos un diagrama de casos de uso, dividido en partes debido a su tamaño (véanse figuras 19-23), y las correspondientes descripciones (véanse tablas 2-44).

4.2.1 Diagrama de casos de uso

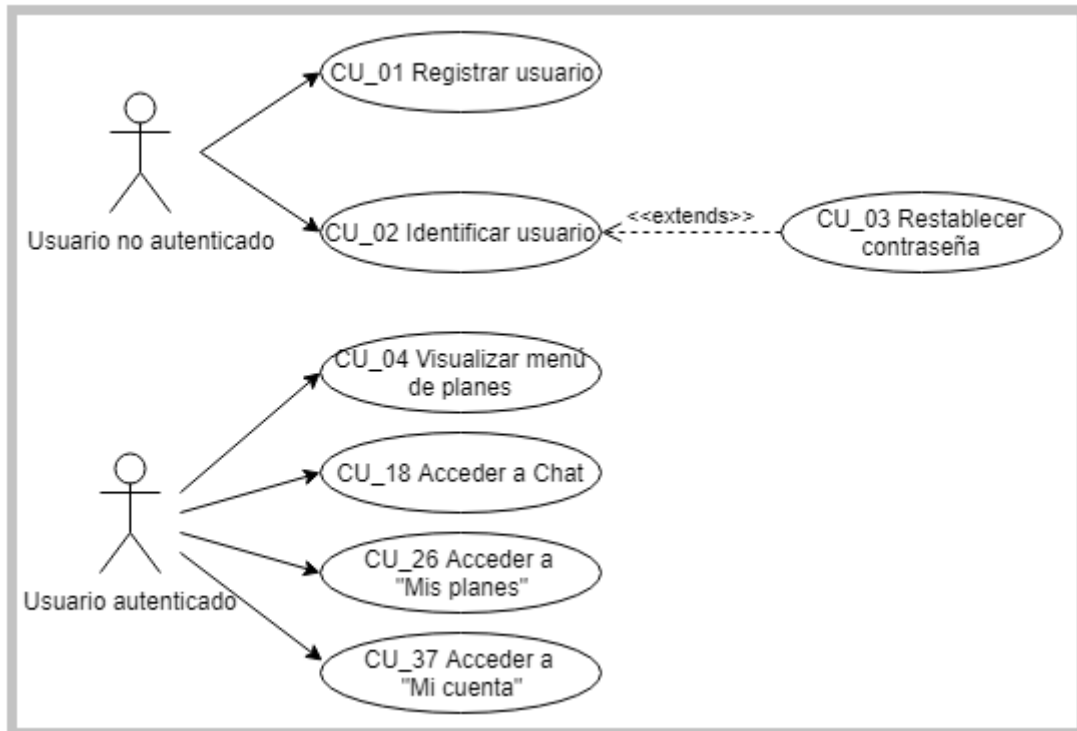


Figura 19. Casos de uso general

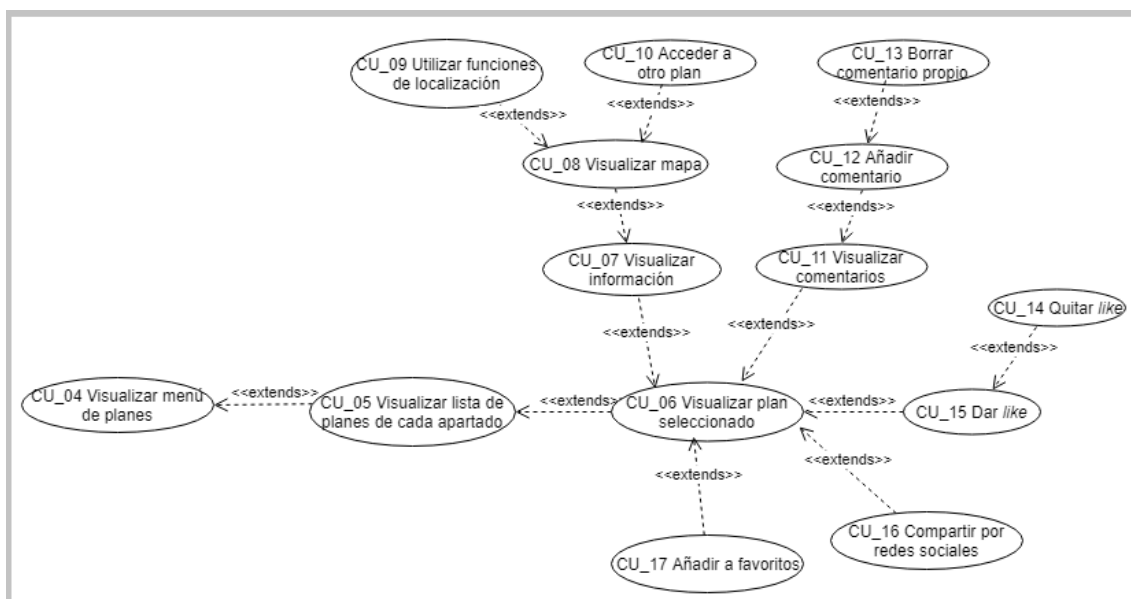


Figura 20. Casos de uso apartado "Visualizar menú de planes"

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

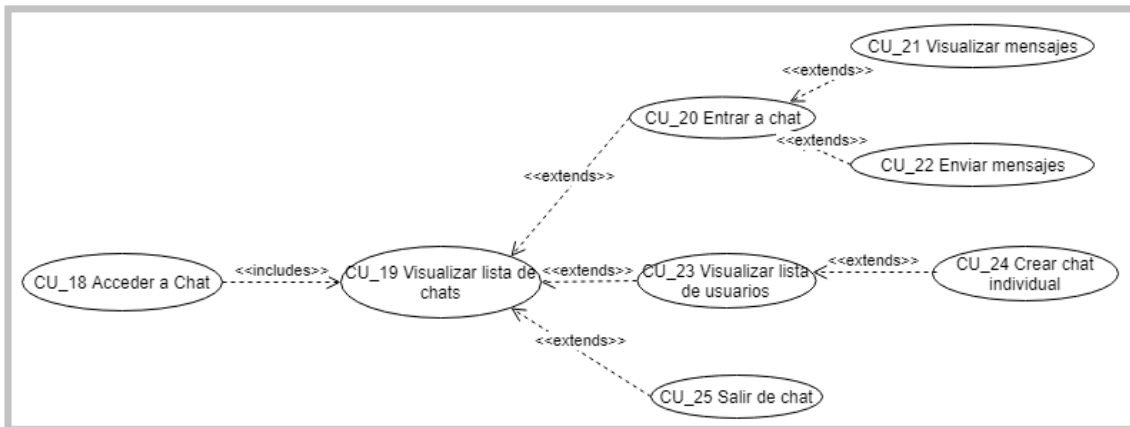


Figura 21. Casos de uso apartado "Acceder a chat"

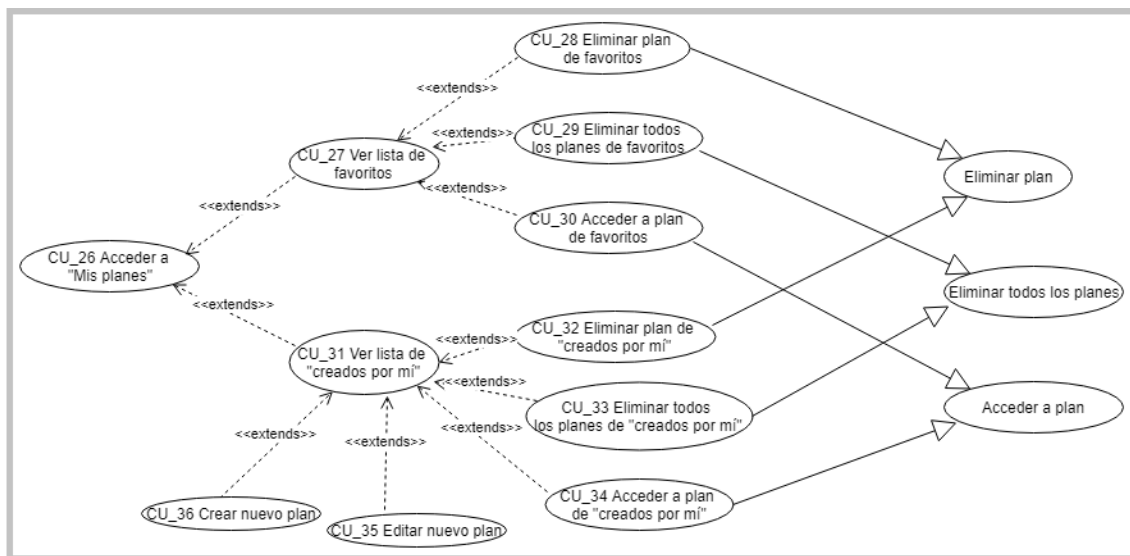


Figura 22. Casos de uso apartado "Acceder a mis planes"

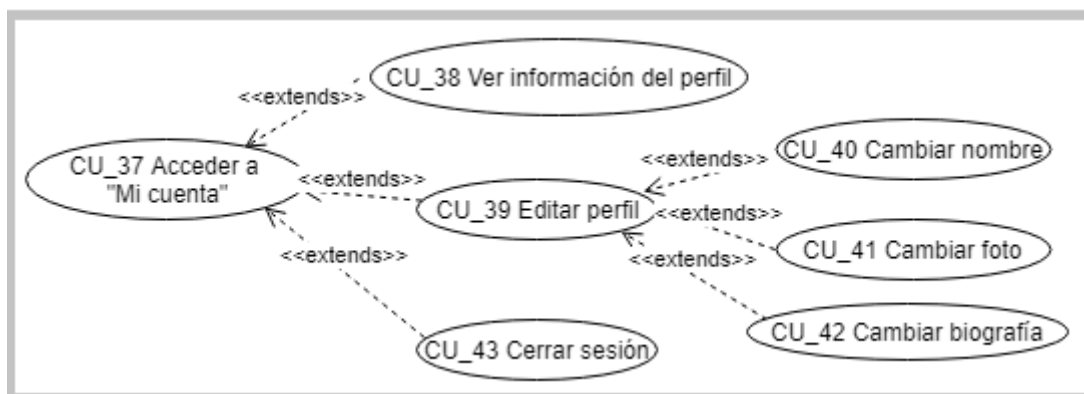


Figura 23. Casos de uso apartado "Acceder a mi cuenta"

4.2.2 Descripciones de los casos de uso

<u>Caso de uso 1: Registrar usuario</u>	
Actor	Usuario no autenticado
Resumen	Registrarse la primera vez que se va a utilizar la aplicación para así acceder con una cuenta de usuario
Precondición	Iniciar la aplicación
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Pulsar en el botón "Registrarse" 6. Pulsar en el enlace de validación recibido en la dirección de correo electrónico
Postcondición	El usuario ha sido registrado en la aplicación
Excepciones	La dirección de correo electrónico no debe estar siendo usada por otro usuario
Boceto de interfaz	Figura 26

Tabla 2. CU Registrar usuario

<u>Caso de uso 2: Identificar usuario</u>	
Actor	Usuario no autenticado
Resumen	Iniciar sesión para acceder a la aplicación con la cuenta de usuario que se ha generado al registrarse
Precondición	Iniciar la aplicación
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Pulsar en el botón "Iniciar sesión"
Postcondición	El usuario ha sido identificado en la aplicación
Excepciones	Correo electrónico y/o contraseña no válidos
Boceto de interfaz	Figura 27

Tabla 3. CU Identificar usuario

<u>Caso de uso 3: Restablecer contraseña</u>	
Actor	Usuario no autenticado
Resumen	Acceder al apartado de restablecer contraseña para recibir un email con dicho fin
Precondición	La cuenta de usuario debe existir
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Pulsar en "He olvidado mi contraseña" 3. Introducir dirección de correo electrónico 4. Pulsar en el botón "Restablecer contraseña" 5. Pulsar en el enlace de restablecimiento recibido en la dirección de correo electrónico 6. Cambiar contraseña
Postcondición	La contraseña ha sido modificada
Excepciones	Correo electrónico no registrado
Boceto de interfaz	Figura 28

Tabla 4. CU Restablecer contraseña

<u>Caso de uso 4: Visualizar menú de planes</u>	
Actor	Usuario autenticado
Resumen	Visualizar el menú de planes e interactuar con él para hacer una mejor búsqueda del plan deseado
Precondición	<p>El usuario debe estar autenticado</p> <p>El menú de planes debe existir y no estar vacío</p>
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección de correo electrónico 4. Introducir contraseña 5. Acceder al menú de planes
Postcondición	Menú de planes visualizado por el usuario
Excepciones	
Boceto de interfaz	Figura 29

Tabla 5. CU Visualizar menú de planes

<u>Caso de uso 5: Visualizar lista de planes de cada apartado</u>	
Actor	Usuario autenticado
Resumen	Visualizar los planes pertenecientes a un apartado del menú de planes que haya sido seleccionado
Precondición	La lista de planes del apartado debe existir y no estar vacía
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú
Postcondición	Lista de planes visualizada por el usuario
Excepciones	
Boceto de interfaz	Figura 30

Tabla 6. CU Visualizar lista de planes de cada apartado

<u>Caso de uso 6: Visualizar plan seleccionado</u>	
Actor	Usuario autenticado
Resumen	Visualizar un plan seleccionado perteneciente a un apartado del menú, así como todas las acciones posibles a realizar sobre éste
Precondición	El plan debe existir en la lista de planes
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista
Postcondición	Plan visualizado por el usuario
Excepciones	
Boceto de interfaz	Figura 31

Tabla 7. CU Visualizar plan seleccionado

<u>Caso de uso 7: Visualizar información</u>	
Actor	Usuario autenticado
Resumen	Ver la información referente al plan seleccionado (descripción, lugar, recomendaciones, etc)
Precondición	El plan debe tener una sección de “información”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección de “información”
Postcondición	Información del plan visualizada por el usuario
Excepciones	
Boceto de interfaz	Figura 33

Tabla 8. CU Visualizar información

<u>Caso de uso 8: Visualizar mapa</u>	
Actor	Usuario autenticado
Resumen	Ver la posición en el mapa del lugar de realización del plan y dar la opción de generar una ruta hasta él
Precondición	El plan debe tener un apartado de “ver mapa”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección “información” 9. Pulsar en la opción de “ver mapa”
Postcondición	Mapa del plan visualizado por el usuario
Excepciones	No se han aceptado los permisos de localización
Boceto de interfaz	Figura 34

Tabla 9. CU Visualizar mapa

<u>Caso de uso 9: Utilizar funciones de localización</u>	
Actor	Usuario autenticado
Resumen	Utilizar las funciones de localización ofrecidas por Google Maps
Precondición	Debe haberse accedido a la opción de “ver mapa”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección de “información” 9. Pulsar en la opción de “ver mapa” 10. Pulsar en el marcador 11. Seleccionar opción ofrecida por Google Maps
Postcondición	Usadas las funcionalidades de Google Maps
Excepciones	
Boceto de interfaz	

Tabla 10. CU Utilizar funciones de localización

<u>Caso de uso 10: Acceder a otro plan</u>	
Actor	Usuario autenticado
Resumen	Navegar por el mapa y acceder a la información de otro plan al que pertenece otro marcador
Precondición	Debe haberse accedido a la opción de “ver mapa”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección de “información” 9. Pulsar en la opción de “ver mapa” 10. Pulsar en el marcador de otro plan 11. Pulsar en “Ver plan”
Postcondición	Visualización de otro plan
Excepciones	
Boceto de interfaz	

Tabla 11. CU Acceder a otro plan

<u>Caso de uso 11: Visualizar comentarios</u>	
Actor	Usuario autenticado
Resumen	Ver los comentarios de los demás usuarios acerca del plan
Precondición	El plan debe tener una sección de “comentarios”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección “comentarios”
Postcondición	Comentarios del plan visualizados por el usuario
Excepciones	
Boceto de interfaz	Figura 35

Tabla 12. CU Visualizar comentarios

<u>Caso de uso 12: Añadir comentario</u>	
Actor	Usuario autenticado
Resumen	Añadir una opinión o vivencia propia de acuerdo con el plan seleccionado
Precondición	En la sección de “comentarios” debe estar la opción de añadir comentarios
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección “comentarios” 9. Añadir comentario
Postcondición	Comentario añadido por el usuario en la sección “comentarios” del plan
Excepciones	
Boceto de interfaz	Figura 35

Tabla 13. CU Añadir comentario

Caso de uso 13: Borrar comentario propio	
Actor	Usuario autenticado
Resumen	Borrar un comentario añadido por el mismo usuario en el plan seleccionado
Precondición	En la sección de “comentarios” debe estar la opción de borrar comentario, en caso de que lo haya escrito el usuario que lo desea borrar
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Acceder a la sección “comentarios” 9. Buscar el comentario a borrar 10. Borrar el comentario
Postcondición	Comentario borrado por el usuario de la sección “comentarios” del plan
Excepciones	
Boceto de interfaz	Figura 35

Tabla 14. CU Borrar comentario propio

Caso de uso 14: Dar like	
Actor	Usuario autenticado
Resumen	Valorar positivamente un plan en su conjunto. Los usuarios pueden ver los likes totales del plan.
Precondición	El plan debe tener la opción de dar like
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Pulsar en el icono de dar like
Postcondición	Plan valorado positivamente por el usuario
Excepciones	
Boceto de interfaz	Figura 31

Tabla 15. CU Dar like

<u>Caso de uso 15: Quitar like</u>	
Actor	Usuario autenticado
Resumen	Quitar una valoración positiva que se había dado a un plan en su conjunto. Los usuarios pueden ver los likes totales del plan.
Precondición	El plan debe haber sido valorado positivamente con anterioridad y debe tener la opción de quitar like
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Pulsar en el icono de quitar like
Postcondición	Plan con una valoración positiva menos
Excepciones	
Boceto de interfaz	Figura 31

Tabla 16. CU Quitar like

<u>Caso de uso 16: Compartir por redes sociales</u>	
Actor	Usuario autenticado
Resumen	Compartir el plan seleccionado por las redes sociales que el usuario elija
Precondición	El plan debe tener la opción de compartir por redes sociales
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Pulsar en el icono de compartir
Postcondición	Plan compartido por el usuario mediante redes sociales
Excepciones	
Boceto de interfaz	Figura 31

Tabla 17. CU Compartir por redes sociales

<u>Caso de uso 17: Añadir a favoritos</u>	
Actor	Usuario autenticado
Resumen	Añadir el plan seleccionado a la lista de favoritos del usuario
Precondición	El plan debe tener la opción de añadir a favoritos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al menú de planes 6. Seleccionar un apartado del menú 7. Seleccionar un plan de la lista 8. Pulsar en “añadir a favoritos”
Postcondición	Plan añadido a la lista de favoritos por el usuario
Excepciones	
Boceto de interfaz	Figura 31

Tabla 18. CU Añadir a favoritos

<u>Caso de uso 18: Acceder a “Chat”</u>	
Actor	Usuario autenticado
Resumen	Acceder a la lista de chats de las que dispone el usuario, así como todas las opciones posibles a realizar
Precondición	El usuario debe estar autenticado
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “chat” del menú inferior de navegación
Postcondición	Apartado “chat” del menú inferior de navegación accedido por el usuario
Excepciones	
Boceto de interfaz	Figura 37

Tabla 19. CU Acceder a “Chat”

<u>Caso de uso 19: Visualizar lista de chats</u>	
Actor	Usuario autenticado
Resumen	Ver los chats abiertos con los demás usuarios
Precondición	El usuario debe disponer de chats abiertos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “chat” del menú inferior de navegación 6. Visualizar la lista de chats
Postcondición	Lista de chats visualizada por el usuario
Excepciones	
Boceto de interfaz	Figura 37

Tabla 20. CU Visualizar lista de chats

<u>Caso de uso 20: Entrar a chat</u>	
Actor	Usuario autenticado
Resumen	Entrar a un chat abierto con otro usuario
Precondición	El usuario debe disponer de chats abiertos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “chat” del menú inferior de navegación 6. Visualizar la lista de chats 7. Pulsar en uno de los chats
Postcondición	Chat accedido por el usuario
Excepciones	
Boceto de interfaz	Figura 38

Tabla 21. CU Entrar a chat

<u>Caso de uso 21: Visualizar mensajes</u>	
Actor	Usuario autenticado
Resumen	Entrar a un chat abierto con otro usuario y leer los mensajes de dicho chat
Precondición	El chat debe contener mensajes enviados y/o recibidos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "chat" del menú inferior de navegación 6. Visualizar la lista de chats 7. Pulsar en uno de los chats 8. Visualizar mensajes
Postcondición	Mensajes vistos por el usuario
Excepciones	
Boceto de interfaz	Figura 38

Tabla 22. CU Visualizar mensajes

<u>Caso de uso 22: Enviar mensajes</u>	
Actor	Usuario autenticado
Resumen	Entrar a un chat abierto con otro usuario, escribir mensajes y enviarlos
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "chat" del menú inferior de navegación 6. Visualizar la lista de chats 7. Pulsar en uno de los chats 8. Escribir un mensaje 9. Pulsar en el icono de enviar
Postcondición	Mensaje enviado por el usuario
Excepciones	
Boceto de interfaz	

Tabla 23. CU Enviar mensajes

<u>Caso de uso 23: Visualizar lista de usuarios</u>	
Actor	Usuario autenticado
Resumen	Ver la lista de usuarios de los que dispone el usuario en la aplicación
Precondición	El usuario debe tener una lista de usuarios
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “chat” del menú inferior de navegación 6. Pulsar en la sección de “usuarios” 7. Visualizar lista de usuarios
Postcondición	Lista de usuarios visualizada por el usuario
Excepciones	
Boceto de interfaz	Figuras 39 y 40

Tabla 24. CU Visualizar lista de usuarios

<u>Caso de uso 24: Crear chat individual</u>	
Actor	Usuario autenticado
Resumen	Crear un nuevo chat individual con uno de los usuarios almacenados en la lista de usuarios
Precondición	<p>El usuario debe tener al usuario, con el que desea comunicarse, en la lista de usuarios.</p> <p>La lista de usuarios debe tener la opción de crear chat individual</p>
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “chat” del menú inferior de navegación 6. Pulsar en la sección de “usuarios” 7. Visualizar lista de usuarios 8. Pulsar en el nombre del usuario con el que se desea establecer el chat
Postcondición	Chat individual creado por el usuario
Excepciones	
Boceto de interfaz	Figura 41

Tabla 25. CU Crear chat individual

Caso de uso 25: Salir del chat	
Actor	Usuario autenticado
Resumen	Salir de un chat
Precondición	El usuario debe estar dentro de ese chat. El chat debe tener la opción de salir
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "chat" del menú inferior de navegación 6. Acceder al chat 7. Volver atrás
Postcondición	Chat abandonado por el usuario
Excepciones	
Boceto de interfaz	

Tabla 26. CU Salir del chat

Caso de uso 26: Acceder a "Mis planes"	
Actor	Usuario autenticado
Resumen	Acceder a la lista de planes del usuario clasificados en "favoritos" y "creados por mí"
Precondición	El usuario debe estar autenticado
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "mis planes" del menú inferior de navegación
Postcondición	Apartado "mis planes" del menú inferior de navegación accedido por el usuario
Excepciones	
Boceto de interfaz	Figuras 42 y 43

Tabla 27. CU Acceder a "Mis planes"

<u>Caso de uso 27: Ver lista de favoritos</u>	
Actor	Usuario autenticado
Resumen	Visualizar los planes que han sido almacenados en la lista de planes favoritos del usuario
Precondición	El usuario debe tener planes almacenados en la lista de planes favoritos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “favoritos” 7. Visualizar la lista de favoritos
Postcondición	Lista de favoritos visualizada por el usuario
Excepciones	
Boceto de interfaz	Figura 42

Tabla 28. CU Ver lista de favoritos

<u>Caso de uso 28: Eliminar plan de favoritos</u>	
Actor	Usuario autenticado
Resumen	Eliminar un plan de la lista de planes favoritos
Precondición	El usuario debe tener el plan almacenado en la lista de planes favoritos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “favoritos” 7. Visualizar la lista de favoritos 8. Mantener pulsado en el plan a eliminar 9. Seleccionar eliminar
Postcondición	Plan de la lista de favoritos eliminado por el usuario
Excepciones	
Boceto de interfaz	

Tabla 29. CU Eliminar plan de favoritos

<u>Caso de uso 29: Eliminar todos los planes de favoritos</u>	
Actor	Usuario autenticado
Resumen	Eliminar todos los planes de la lista de planes favoritos
Precondición	El usuario debe tener planes almacenados en la lista de planes favoritos
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “favoritos” 7. Pulsar en el icono de borrar todos los planes
Postcondición	Todos los planes de la lista de favoritos eliminados por el usuario
Excepciones	
Boceto de interfaz	Figura 42

Tabla 30. CU Eliminar todos los planes de favoritos

<u>Caso de uso 30: Acceder a plan de favoritos</u>	
Actor	Usuario autenticado
Resumen	Acceder a la visualización de un plan a través de la lista de “favoritos”
Precondición	El usuario debe tener planes almacenados en la lista de planes “favoritos”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “favoritos” 7. Pulsar en un plan
Postcondición	Plan accedido por el usuario
Excepciones	
Boceto de interfaz	

Tabla 31. CU Acceder a plan de favoritos

<u>Caso de uso 31: Ver lista de “creados por mí”</u>	
Actor	Usuario autenticado
Resumen	Visualizar los planes que han sido almacenados en la lista de planes “creados por mí” del usuario
Precondición	El usuario debe tener planes almacenados en la lista de planes “creados por mí”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Visualizar la lista de “creados por mí”
Postcondición	Lista de “creados por mí” visualizada por el usuario
Excepciones	
Boceto de interfaz	Figura 43

Tabla 32. CU Ver lista de “creados por mí”

<u>Caso de uso 32: Eliminar plan de “creados por mí”</u>	
Actor	Usuario autenticado
Resumen	Eliminar un plan de la lista de “creados por mí”
Precondición	El usuario debe tener el plan almacenado en la lista de planes “creados por mí”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Visualizar la lista de “creados por mí” 8. Mantener pulsado en el plan 9. Seleccionar eliminar
Postcondición	Plan de la lista de “creados por mí” eliminado por el usuario
Excepciones	
Boceto de interfaz	

Tabla 33. CU Eliminar plan de “creados por mí”

Caso de uso 33: Eliminar todos los planes de “creados por mí”	
Actor	Usuario autenticado
Resumen	Eliminar todos los planes de la lista de planes “creados por mí”
Precondición	El usuario debe tener planes almacenados en la lista de planes “creados por mí”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Pulsar en el icono de “borrar todos los planes”
Postcondición	Todos los planes de la lista de “creados por mí” eliminados por el usuario
Excepciones	
Boceto de interfaz	Figura 43

Tabla 34. CU Eliminar todos los planes de “creados por mí”

Caso de uso 34: Acceder a plan de “creados por mí”	
Actor	Usuario autenticado
Resumen	Acceder a la visualización de un plan a través de la lista de “creados por mí”
Precondición	El usuario debe tener planes almacenados en la lista de planes “creados por mí”
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Pulsar en un plan
Postcondición	Plan accedido por el usuario
Excepciones	
Boceto de interfaz	

Tabla 35. CU Acceder a plan de “creados por mí”

<u>Caso de uso 35: Editar plan</u>	
Actor	Usuario autenticado
Resumen	Editar un plan creado por el usuario, el cual pertenece a la lista de “creados por mí”
Precondición	El plan debe haber sido creado con anterioridad
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Mantener pulsado en un plan 8. Seleccionar editar
Postcondición	Plan editado por el usuario y almacenado en la lista de “creados por mí”
Excepciones	
Boceto de interfaz	

Tabla 36. CU Editar plan

<u>Caso de uso 36: Crear nuevo plan</u>	
Actor	Usuario autenticado
Resumen	Crear un nuevo plan que será visualizado por los demás usuarios y añadido a la lista de “creados por mí”
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “mis planes” del menú inferior de navegación 6. Pulsar en la sección de “creados por mí” 7. Pulsar en el icono de “crear nuevo plan” 8. Crear nuevo plan
Postcondición	Plan creado por el usuario y almacenado en la lista de “creados por mí”
Excepciones	Es obligado introducir una imagen, un título, una descripción y una dirección. La fecha de inicio debe ser menor que la fecha de fin.
Boceto de interfaz	Figuras 44-50

Tabla 37. CU Crear nuevo plan

<u>Caso de uso 37: Acceder a “Mi cuenta”</u>	
Actor	Usuario autenticado
Resumen	Acceder al perfil de la cuenta del usuario en la aplicación
Precondición	El usuario debe estar autenticado
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “Mi cuenta” del menú inferior de navegación
Postcondición	Apartado “Mi cuenta” del menú inferior de navegación accedido por el usuario
Excepciones	
Boceto de interfaz	Figura 52

Tabla 38. CU Acceder a “Mi cuenta”

<u>Caso de uso 38: Ver información del perfil</u>	
Actor	Usuario autenticado
Resumen	Visualizar la información del perfil de la cuenta del usuario en la aplicación.
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “Mi cuenta” del menú inferior de navegación 6. Visualizar la información del perfil
Postcondición	Información del perfil de “Mi cuenta” visto por el usuario
Excepciones	
Boceto de interfaz	Figura 52

Tabla 39. CU Ver información del perfil

Caso de uso 39: Editar perfil	
Actor	Usuario autenticado
Resumen	Acceder al apartado de edición del perfil del usuario
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “Mi cuenta” del menú inferior de navegación 6. Pulsar en “Editar perfil”
Postcondición	Apartado de editar perfil accedido por el usuario
Excepciones	
Boceto de interfaz	

Tabla 40. CU Editar perfil

Caso de uso 40: Cambiar nombre	
Actor	Usuario autenticado
Resumen	Editar el nombre del usuario
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado “Mi cuenta” del menú inferior de navegación 6. Pulsar en “Editar perfil” 7. Cambiar nombre del usuario 8. Pulsar en el botón “Guardar cambios”
Postcondición	Nombre de usuario editado por el usuario
Excepciones	Para pulsar en el botón “Guardar cambios” se debe haber modificado el nombre. El nombre no puede estar siendo usado por otro usuario
Boceto de interfaz	

Tabla 41. CU Cambiar nombre

<u>Caso de uso 41: Cambiar foto</u>	
Actor	Usuario autenticado
Resumen	Cambiar la foto del usuario
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "Mi cuenta" del menú inferior de navegación 6. Pulsar en "Editar perfil" 7. Cambiar la foto del usuario 8. Pulsar en el botón "Guardar cambios"
Postcondición	Foto de usuario cambiada por el usuario
Excepciones	No se ha aceptado el permiso de almacenamiento. Para pulsar en el botón "Guardar cambios" se debe haber seleccionado una foto
Boceto de interfaz	

Tabla 42. CU Cambiar foto

<u>Caso de uso 42: Cambiar biografía</u>	
Actor	Usuario autenticado
Resumen	Editar la biografía del usuario
Precondición	
Pasos	<ol style="list-style-type: none"> 1. Abrir la aplicación 2. Seleccionar modo de acceso 3. Introducir dirección electrónica 4. Introducir contraseña 5. Acceder al apartado "Mi cuenta" del menú inferior de navegación 6. Pulsar en "Editar perfil" 7. Cambiar biografía del usuario 8. Pulsar en el botón "Guardar cambios"
Postcondición	Biografía del usuario editada por el usuario
Excepciones	Para pulsar en el botón "Guardar cambios" se debe haber modificado la biografía
Boceto de interfaz	

Tabla 43. CU Cambiar biografía

Caso de uso 43: Cerrar sesión	
Actor	Usuario autenticado
Resumen	Cerrar sesión con la cuenta con la que se ha accedido a la aplicación. El usuario pasa a ser “no autenticado”.
Precondición	
Pasos	<ol style="list-style-type: none">1. Abrir la aplicación2. Seleccionar modo de acceso3. Introducir dirección electrónica4. Introducir contraseña5. Acceder al apartado “Mi cuenta” del menú inferior de navegación6. Pulsar en el icono de “opciones”7. Pulsar en “cerrar sesión”
Postcondición	Sesión cerrada por el usuario
Excepciones	
Boceto de interfaz	

Tabla 44. CU Cerrar sesión

4.3 Bocetos de interfaz

Una vez definidos los casos de uso, pasamos a mostrar la interfaz de la App (véanse figuras 24-54). Esta interfaz se ha diseñado para que sea clara, concisa, amigable y respete los preceptos que impone el Material Design de Android.

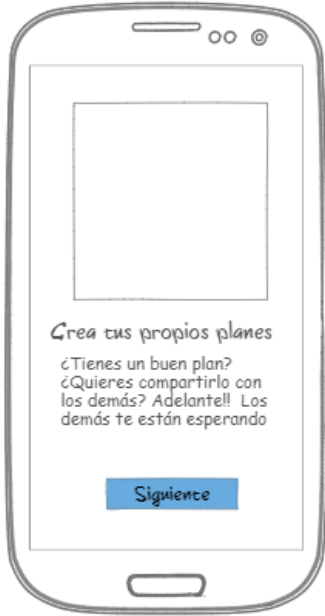


Figura 24. Bienvenida: Ventana 1

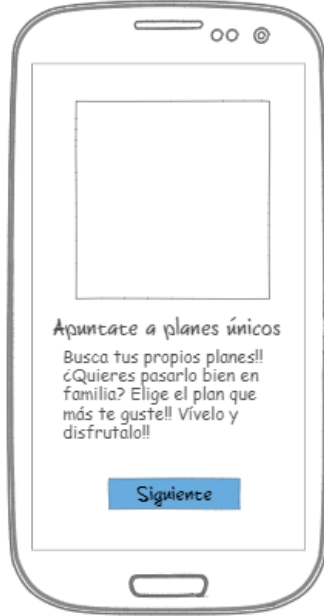


Figura 25. Bienvenida: Ventana 2



Figura 26. Registro



Figura 27. Inicio de sesión



Figura 28. Restablecer contraseña

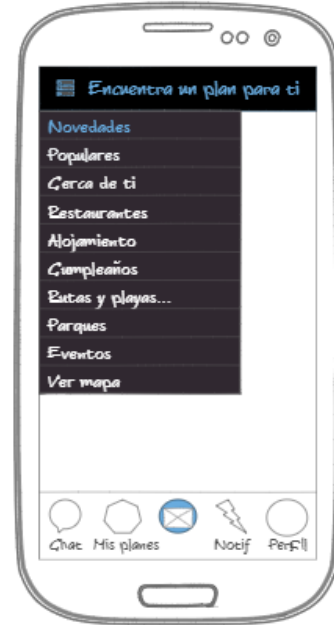


Figura 29. Menú de planes

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles



Figura 30. Lista de planes de un apartado del menú de planes



Figura 31. Plan seleccionado vista usuario autenticado



Figura 32. Plan seleccionado vista usuario no autenticado

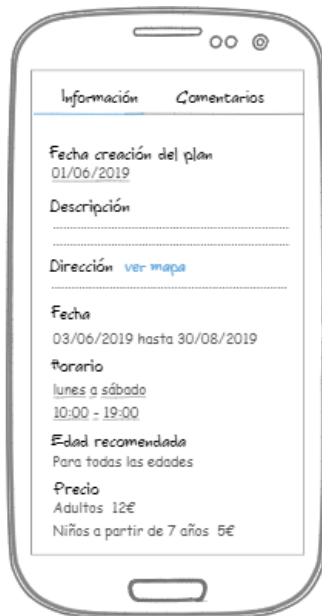


Figura 33. Información del plan seleccionado

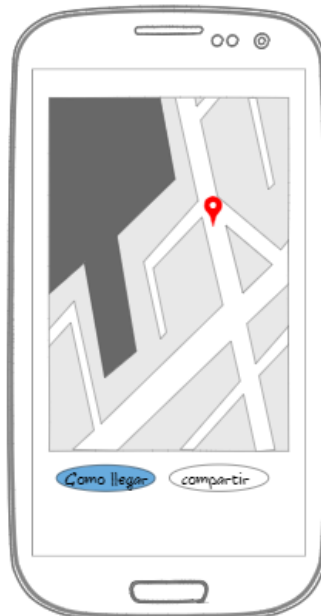


Figura 34. Visualización mapa



Figura 35. Comentarios vista usuario autenticado



Figura 36. Comentarios vista usuario no autenticado

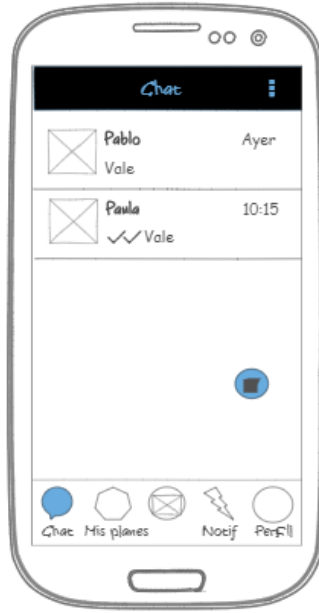


Figura 37. Listado de chats

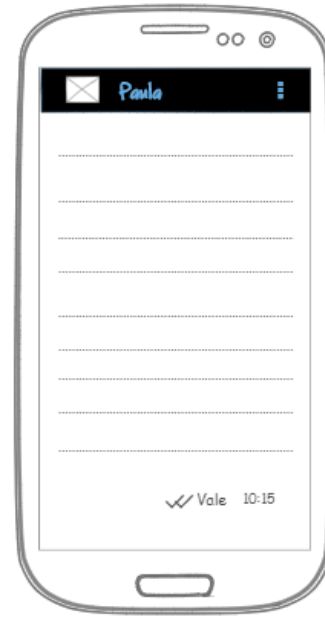


Figura 38. Chat abierto

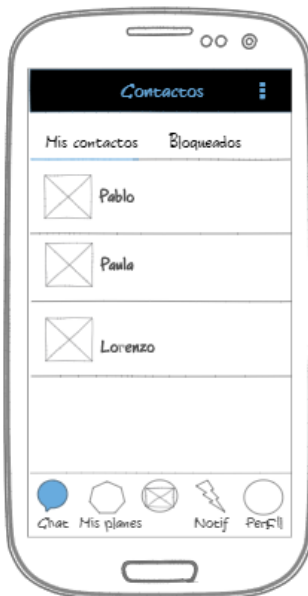


Figura 39. Lista de contactos: Mis contactos

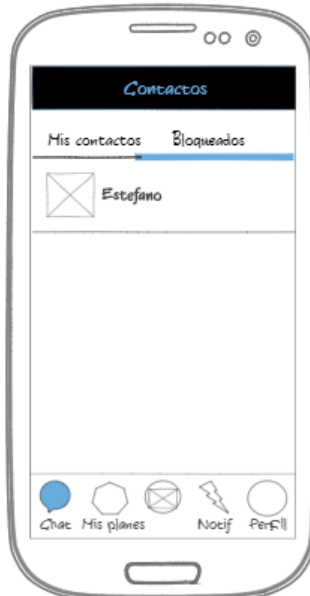


Figura 40. Lista de contactos: Bloqueados



Figura 41. Añadir contacto

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles



Figura 42. Mis planes: Favoritos



Figura 43. Mis planes: Creados por mí

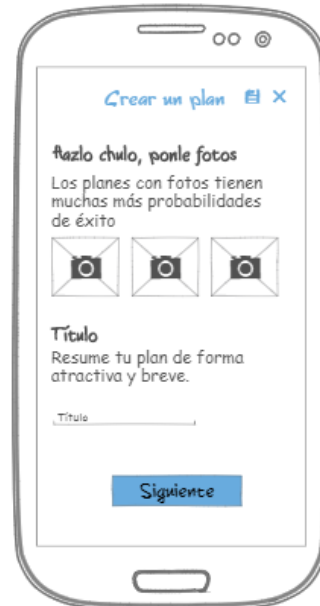


Figura 44. Crear un plan: Foto y título

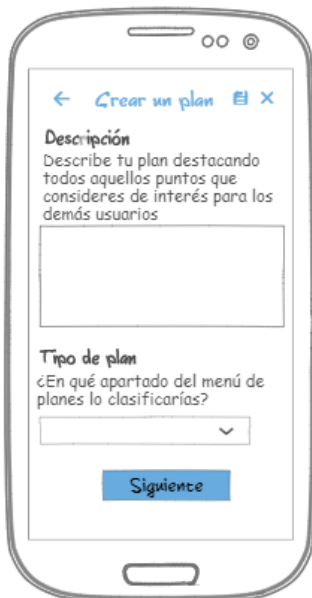


Figura 45. Crear un plan: Descripción y tipo



Figura 46. Crear un plan: Dirección y plazas



Figura 47. Crear un plan: Fechas y horarios



Figura 48. Crear un plan: Edad, precio, audio y vídeo



Figura 49. Crear un plan: Vista previa/publicar



Figura 50. Crear un plan: Visualización previa



Figura 51. Notificaciones

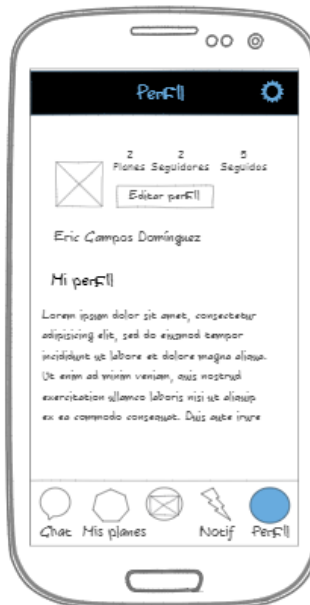


Figura 52. Perfil: Vista usuario propietario



Figura 53. Perfil: Vista otro usuario

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles



Figura 54. Planes creados (vista otro usuario)

4.4 Modelo de datos

Una aplicación sin conexión es actualmente una App de utilidad muy limitada. En nuestro caso todos los planes están almacenados en un backend que gestiona una base de datos y que es accesible para todos los dispositivos que instalen nuestra App.

Para esta base de datos se ha usado un modelo no relacional para soportar todo el almacenamiento de la información. Se puede observar cómo se estructura el almacenamiento de la información de los usuarios, planes, comentarios y chats viendo la figura 55.

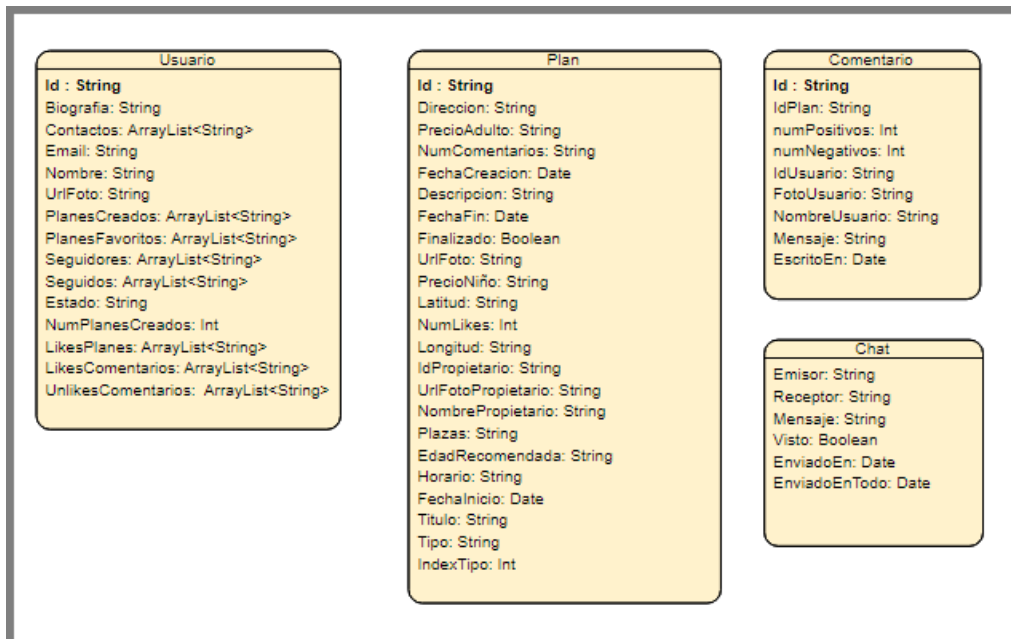


Figura 55. Modelo de datos

5. Desarrollo e implementación

Se ha utilizado Java como lenguaje de programación y Android Studio como entorno de desarrollo debido a que es el actual entorno de desarrollo integrado respaldado por Google y dispone de múltiples herramientas para el desarrollo de aplicaciones destinadas a dispositivos con sistema operativo Android.

También se ha utilizado Firebase Google, una plataforma ubicada en la nube que ofrece multitud de servicios, como son los de mensajería, almacenamiento y notificaciones entre otros. En este proyecto, se utilizan los servicios de almacenamiento con bases de datos no relaciones que ofrece la plataforma.

Además de lo anterior, se ha usado la librería Picasso para la gestión de imágenes y caché.

5.1 Java

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos, que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible (13).

El código generado por el compilador Java es independiente de la arquitectura y podría ejecutarse en un entorno UNIX, Mac o Windows. Sus principales características son (14):

- Lenguaje totalmente orientado a Objetos y simple.
- Interpretación y compilación a la vez.
- Distribución, robustez, portabilidad, dinamicidad, multihebra y rendimiento.
- Disponibilidad de un amplio conjunto de bibliotecas.
- Indiferencia a la arquitectura.

5.2 Android

Android es un sistema operativo móvil desarrollado por Google, basado en el Kernel de Linux y otros software de código abierto. Fue diseñado inicialmente para dispositivos con pantalla táctil como los teléfonos inteligentes o las tabletas, aunque actualmente su uso se extiende a los relojes inteligentes, los televisores y también a los automóviles (15).

El sistema se estructura en capas, siendo la capa de aplicación, aquella en la que se localizan tanto las Apps que constituyen el mínimo común denominador de todo sistema Android, tales como los contactos, ajustes, y alarmas, como las Apps desarrolladas por terceros y que los usuarios instalarán para adaptar el uso de sus dispositivos a sus necesidades. Cabe señalar que las Apps pueden desarrollarse tanto en Java como en Kotlin (15).



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Por debajo de esta capa está la capa de servicios que incluye todas las bibliotecas de servicio para las aplicaciones. Parte de esta capa está desarrollada en Java y parte en C/C++ por una cuestión de rendimiento y eficiencia (15).

El runtime de Android aglutina todos los controladores de dispositivos y bibliotecas de bajo nivel del sistema. En su mayor parte, esta capa del sistema está desarrollada en C e interactúa estrechamente con el núcleo del sistema, que es, como ya hemos mencionado anteriormente, un núcleo Linux. El sistema es bastante complejo y actualmente admite el desarrollo de aplicaciones (15).

5.3 Android Studio

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en IntelliJ IDEA. Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan tu productividad durante la compilación de Apps para Android, como son (16):

- Un sistema de compilación avanzada basado en Gradle.
- Un entorno unificado en el que se puede realizar desarrollos para todos los dispositivos Android.
- Integración de plantillas de código y GitHub para ayudar a compilar funciones comunes de las Apps e importar ejemplos de código.
- Soporte incorporado para Google Cloud Platform, lo que facilita la integración de Google Cloud Messaging y App Engine.
- Un conjunto de emuladores que permiten la prueba y depuración de las aplicaciones, evitando así la necesidad de tener que utilizar dispositivos físicos (véase figura 56).

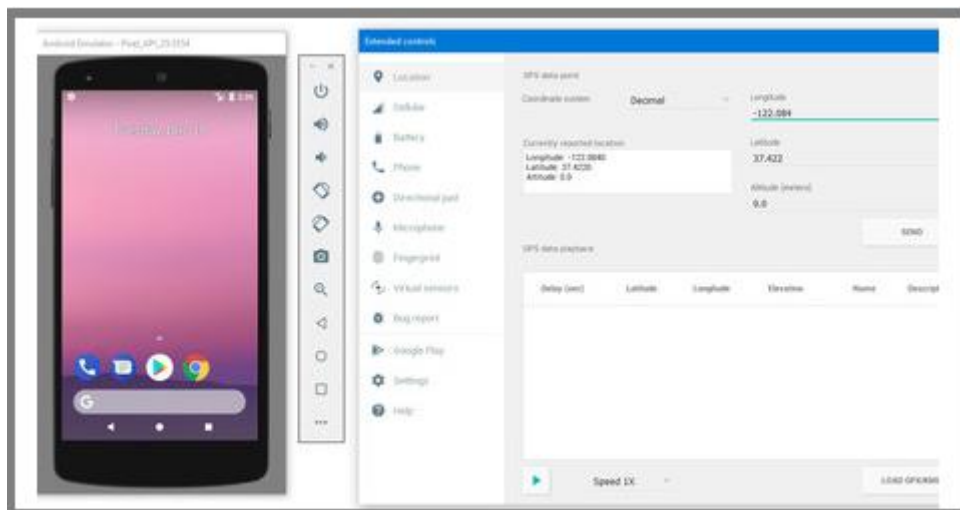


Figura 56. Emulador Android Studio

5.4 Aplicación Android

Para comprender cómo se estructuran y comportan las Apps Android, es necesario conocer y entender una serie de conceptos particulares de este tipo de aplicaciones. Las siguientes subsecciones pretenden introducir estos conceptos y explicarlo de manera sucinta.

El lector interesado podrá ahondar en las explicaciones proporcionadas bien utilizando la referencia² en línea que para tal fin se suministra o a través de cualquiera de los múltiples libros disponibles para el desarrollo de aplicaciones móviles Android. Sin embargo, la rápida evolución de la plataforma y sus APIs de soporte, hace que los libros queden desfasados rápidamente, por ello nuestro consejo es siempre utilizar la web oficial Android Developer para resolver cualquier duda que pueda aparecer.

5.4.1 Processes and Threads

Cuando un componente de la aplicación se inicia y la aplicación no tiene ningún otro componente ejecutándose, el sistema de Android inicia un nuevo proceso para la aplicación con un único subproceso de ejecución. De manera predeterminada, todos los componentes de la misma aplicación se ejecutan en el mismo proceso y subproceso (que se denomina "subproceso principal") (17).

Si el componente de una aplicación se inicia y ya existe un proceso para la aplicación, el componente se inicia dentro de ese proceso y usa el mismo subproceso de ejecución.

Sin embargo, se puede configurar que diferentes componentes de la aplicación se ejecuten en procesos separados y se pueden crear subprocesos adicionales para cualquier proceso (17).

5.4.2 Manifest

Todas las aplicaciones deben tener un archivo "AndroidManifest.xml" (con ese nombre exacto) en el directorio raíz. El archivo de manifiesto proporciona información esencial sobre la aplicación al sistema Android, información que el sistema debe tener para poder ejecutar el código de la App (18). Se puede ver un ejemplo del contenido de este archivo en la figura 57.

Entre otras cosas, el archivo de manifiesto hace lo siguiente (18):

- Nombra el paquete de Java para la aplicación. El nombre del paquete sirve como un identificador único para la aplicación.
- Describe los componentes de la aplicación, como las actividades, los servicios, los receptores de mensajes y los proveedores de contenido que la integran.
- Determina los procesos que alojan a los componentes de la aplicación.

² <https://developer.android.com/>



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

- Declara los permisos que debe tener la aplicación para acceder a las partes protegidas de una API e interactuar con otras aplicaciones.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.example.myapplication" >

    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

    <application
        android:name=".Utils.Global"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher_mine"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_mine_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">

        <activity
            android:name=".Activities.LauncherActivity"
            android:theme="@style/Theme.AppCompat.NoActionBar">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

Figura 57. Manifest (Android Studio)

5.4.3 Permissions

Un punto central del diseño de la arquitectura de seguridad de Android consiste en que ninguna aplicación, de manera predeterminada, tiene permiso para realizar operaciones que pudieran tener consecuencias negativas para otras aplicaciones, el sistema operativo o el usuario. Esto incluye leer datos privados de los usuarios o escribir en ellos, leer archivos de otra aplicación o escribir en ellos, acceder a una red, mantener el dispositivo activo y más (19).

Debido a que cada aplicación de Android funciona en una zona de pruebas de proceso, las aplicaciones deben compartir recursos y datos de manera explícita. Hacen esto declarando los permisos que necesitan para capacidades adicionales no previstas por la zona de pruebas básica. Las aplicaciones declaran estáticamente los permisos que necesitan y el sistema Android solicita al usuario su consentimiento (19). Podemos ver un ejemplo del uso de los permisos en la figura 57.

5.4.4 Activities

Una Activity es un componente de la aplicación que contiene una pantalla con la que los usuarios pueden interactuar para realizar una acción, como marcar un número telefónico, tomar una foto, enviar un correo electrónico o ver un mapa. A cada

actividad se le asigna una ventana en la que se puede dibujar su interfaz de usuario (véase figura 58). Toda actividad está representada en el archivo "Manifest" (20).

Una aplicación generalmente consiste en múltiples actividades vinculadas de forma flexible entre sí. Normalmente, una actividad en una aplicación se especifica como la actividad "principal" que se presenta al usuario cuando este inicia la aplicación. Cada actividad puede a su vez iniciar otra actividad para poder realizar diferentes acciones. Cada vez que se inicia una actividad nueva, se detiene la actividad anterior, pero el sistema conserva la actividad en una pila (la "pila de actividades"). Cuando se inicia una actividad nueva, se la incluye en la pila de actividades y capta el foco del usuario (20).

La pila de actividades cumple con el mecanismo de pila "el último en entrar es el primero en salir", por lo que, cuando el usuario termina de interactuar con la actividad actual y presiona el botón Atrás, se quita de la pila (y se destruye) y se reanuda la actividad anterior (20).

```
public class ExampleActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_example);  
  
        //...  
    }  
}
```

Figura 58. Activity (Android Studio)

5.4.5 Layouts

Definen la estructura visual para una interfaz de usuario, como la IU para una actividad o widget de una App (ver figura 59). Se puede declarar un diseño de dos maneras (21):

1. Declarar elementos de la IU en XML. Android proporciona un vocabulario XML simple que coincide con las clases y subclases de vistas, como las que se usan para widgets y diseños.
2. Crear una instancia de elementos del diseño en tiempo de ejecución. En la aplicación se pueden crear objetos View y ViewGroup (y manipular sus propiedades) programáticamente.

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

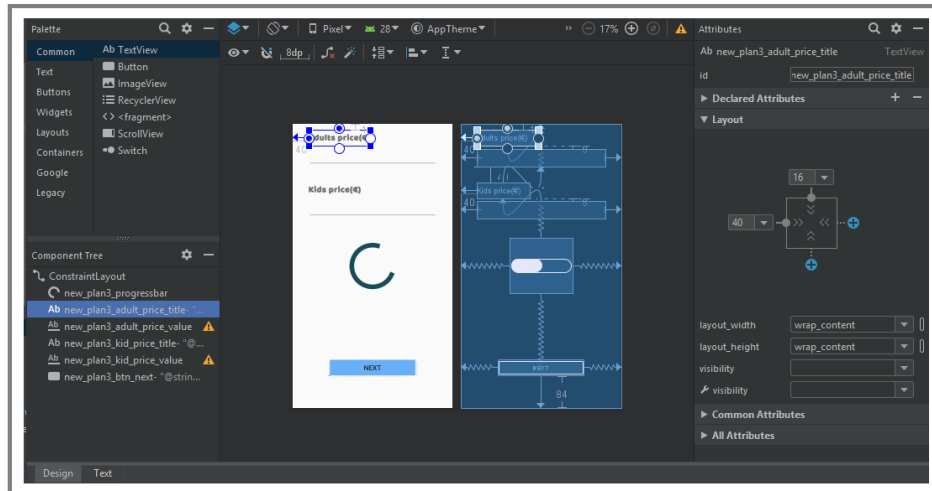


Figura 59. Layout (Android Studio)

5.4.6 Fragments

Un Fragment representa un comportamiento o una parte del layout en una Activity (véanse figuras 60 y 61). Se pueden combinar múltiples fragmentos en una sola actividad para crear una interfaz multipanel y volver a usar un fragmento en múltiples actividades. Se puede pensar en un fragmento como una sección modular de una actividad que tiene su ciclo de vida propio, recibe sus propios eventos de entrada y que se puede agregar o quitar mientras la actividad se esté ejecutando (algo así como una "subactividad" que se puede volver a usar en diferentes actividades) (22).

Un fragmento siempre debe estar integrado a una actividad y el ciclo de vida del fragmento se ve directamente afectado por el ciclo de vida de la actividad anfitriona. Por ejemplo, cuando la actividad está pausada, también lo están todos sus fragmentos, y cuando se destruye, lo mismo ocurre con todos los fragmentos (22).

Sin embargo, mientras una actividad se está ejecutando (está en el estado del ciclo de vida reanudada), se puede manipular cada fragmento de forma independiente; por ejemplo, para agregarlos o quitarlos (22).

Cuando se realiza una transacción de fragmentos, también se pueden agregar a una pila de actividades administrada por la actividad; cada entrada de la pila de actividades en la actividad es un registro de la transacción de fragmentos realizada. La pila de actividades permite al usuario invertir una transacción de fragmentos (navegar hacia atrás) al presionar el "botón atrás" (22).

```

public class ChatFragment extends Fragment {

    public ChatFragment() { }

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        //...
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_chat, root: null);
        //...
        return view;
    }
}

```

Figura 60. Fragment (Android Studio)

```

@Override
public boolean onNavigationItemSelected (MenuItem menuItem){
    switch (menuItem.getItemId()) {
        case R.id.main_bottom_menu_chat:
            tag = "ChatFragment";
            fragment = getSupportFragmentManager().findFragmentByTag(tag);
            if (fragment == null) {
                fragment = new ChatFragment();
            }
            getSupportActionBar().setTitle(getString(R.string.main_title_chat_string));
            break;

        case R.id.main_bottom_menu_my_plans:
            tag = "MyPlansFragment";
            fragment = getSupportFragmentManager().findFragmentByTag(tag);
            if (fragment == null) {
                fragment = new MyPlansFragment();
            }
            getSupportActionBar().setTitle(getString(R.string.main_title_my_plans_string));
            break;

        //...
    }
    getSupportFragmentManager() .FragmentManager
        .beginTransaction() .FragmentManagerTransaction
        .replace(R.id.main_fragment, fragment, tag) .FragmentManagerTransaction
        .commit();
    return true;
}

```

Figura 61. Uso de fragments (Android Studio)

5.4.7 Menus

Los menús son un componente común de la interfaz de usuario en muchos tipos de aplicaciones (ver figura 62 y 63). Para proporcionar una experiencia de usuario conocida y uniforme, se debe usar las Menu API para presentar al usuario acciones y otras opciones en las actividades (23).



```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/account_menu_help"
    android:icon="@drawable/ic_help"
    android:title="@string/account_menu_help_string"
    android:visible="true"
    app:showAsAction="always"/>

  <item
    android:id="@+id/account_menu_settings"
    android:title="@string/account_menu_settings_string"
    android:visible="true"
    app:showAsAction="never"/>

  <item
    android:id="@+id/account_menu_log_out"
    android:title="@string/account_menu_log_out_string"
    android:visible="true"
    app:showAsAction="never"/>
</menu>
```

Figura 62. Menu (Android Studio)

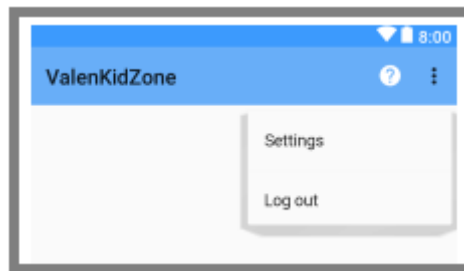


Figura 63. Menu (Aplicación)

5.4.8 ViewPager

Administrador de diseño que permite al usuario voltear hacia la izquierda y hacia la derecha a través de las páginas de datos. Proporciona una implementación de un PagerAdapter para generar las páginas que muestra la vista (24) (véanse figuras 64 y 65).

ViewPager se usa más a menudo en conjunto con Fragment, lo cual es una forma conveniente de suministrar y administrar el ciclo de vida de cada página (24).

Se han implementado adaptadores estándar para usar fragmentos con ViewPager, que cubren los casos de uso más comunes. Estos son FragmentPagerAdapter y FragmentStatePagerAdapter; Cada una de estas clases tiene un código simple que muestra cómo construir una interfaz de usuario completa con ellas (24).

```

public class MyPlansPagerAdapter extends FragmentPagerAdapter {
    private Context context;

    public MyPlansPagerAdapter(FragmentManager fm, Context context) {
        super(fm);
        this.context = context;
    }

    @Override
    public Fragment getItem(int position) {
        Fragment result = null;
        switch (position) {
            case 0:
                result = new MyPlansFavouritesFragment();
                break;
            case 1:
                result = new MyPlansCreatedFragments();
                break;
        }
        return result;
    }

    @Override
    public int getCount() { return 2; }

    @Override
    public CharSequence getPageTitle(int position) {
        switch (position) {
            case 0:
                return ("Favourites");
            case 1:
                return ("Created by me");
        }
        return super.getPageTitle(position);
    }
}

```

Figura 64. ViewPager Adapter (Android Studio)

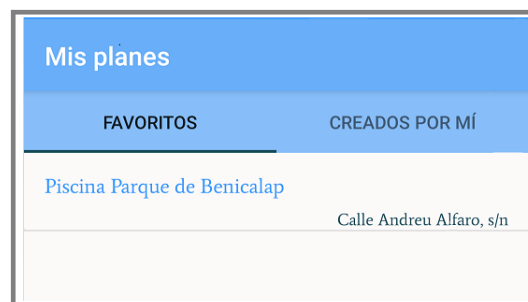


Figura 65. ViewPager (Aplicación)

5.4.9 BottomNavigationView

Representa una barra de navegación inferior estándar para la aplicación. Las barras de navegación inferiores facilitan la exploración y el cambio entre las vistas de nivel superior con un solo toque. Se debe utilizar cuando la aplicación tiene entre tres y cinco destinos de nivel superior (25) (ver figuras 66 y 67).

El contenido de la barra se puede rellenar especificando un archivo de recursos de menú. El título de cada elemento del menú, el icono y el estado activado se utilizan para mostrar los elementos de la barra de navegación inferior. Los elementos del menú también se pueden usar para seleccionar mediante programación qué destino está activo actualmente (25).

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

```
20 <android.support.design.widget.BottomNavigationView
21     android:id="@+id/main_bottomNavigationView"
22     android:layout_width="0dp"
23     android:layout_height="wrap_content"
24     android:background="@color/colorPrimaryDark"
25     app:itemIconTint="@color/main_bottom_menu_color"
26     app:itemTextColor="@color/main_bottom_menu_color"
27     app:layout_constraintBottom_toBottomOf="parent"
28     app:layout_constraintEnd_toEndOf="parent"
29     app:layout_constraintHorizontal_bias="1.0"
30     app:layout_constraintStart_toStartOf="parent"
31     app:menu="@menu/main_bottom_menu" />
32
33 </android.support.constraint.ConstraintLayout>
```

Figura 66. BottomNavigationView (Android Studio)

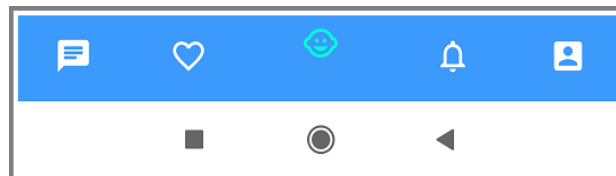


Figura 67. BottomNavigationView (Aplicación)

5.4.10 Material Design

El diseño de materiales es una guía completa para el diseño visual, de movimiento e interacción en plataformas y dispositivos (26).

Para usarlo en las aplicaciones de Android, se sigue las pautas definidas en la especificación de diseño de material y se usan los nuevos componentes y estilos disponibles en la biblioteca de soporte de diseño de materiales (26).

5.4.10.1 FloatingActionButton

Los botones de acción flotantes se utilizan para un tipo especial de acción promovida. Se distinguen por un ícono en forma de círculo que flota sobre la interfaz de usuario y tienen comportamientos de movimiento especiales relacionados con la transformación, el lanzamiento y el punto de anclaje de transferencia (27) (ver figura 68).

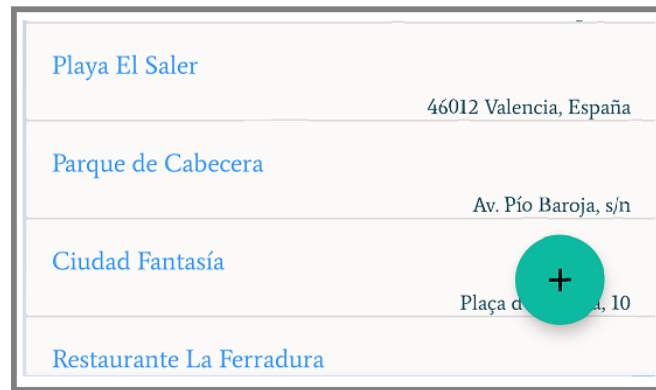


Figura 68. FloatingActionButton (Aplicación)

5.4.10.2 CardView

Es un FrameLayout con una esquina redondeada de fondo y sombra (véase figura 69). Utiliza la propiedad “elevation” a partir de la versión Lollipop (Android versión 5.0) para las sombras y recurre a una implementación de sombra emulada personalizada en plataformas más antiguas (28).

Debido a la naturaleza costosa del recorte de esquinas redondeadas, en plataformas anteriores a Lollipop, CardView no recorta sus elementos secundarios que se intersecan con esquinas redondeadas. En su lugar, agrega relleno para evitar dicha intersección (28).

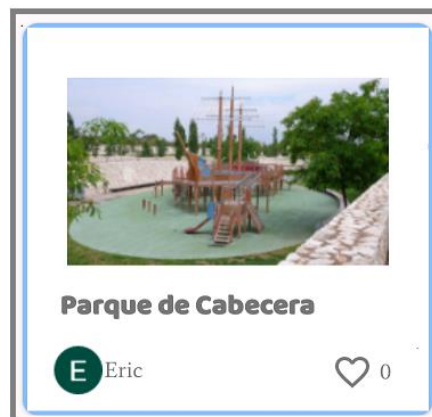


Figura 69. CardView (Aplicación)

5.4.10.3 RecyclerView

Es una vista flexible para proporcionar una ventana limitada en un gran conjunto de datos. Introduce un nivel adicional de abstracción entre Adapter y LayoutManager para poder detectar cambios de conjuntos de datos en lotes durante un cálculo de diseño (29) (ver figuras 70 y 71).

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Esto evita que layoutManager siga los cambios del adaptador para calcular las animaciones. También ayuda con el rendimiento porque todos los enlaces de vista ocurren al mismo tiempo y se evitan los enlaces innecesarios (29).

Por esta razón, hay dos tipos de métodos de posición relacionados (29):

- Posición de diseño: Posición de un elemento en el último cálculo de diseño. Esta es la posición desde la perspectiva del layoutManager.
- Posición del adaptador: Posición de un elemento en el adaptador. Esta es la posición desde la perspectiva del Adaptador.

```
public class MyPlansAdapter extends RecyclerView.Adapter<MyPlansAdapter.ViewHolder> {  
  
    private Context context;  
    private int layout;  
    private ArrayList<MyPlans> data;  
    private OnMyPlansClickListener clickListener;  
    private OnMyPlansLongClickListener clickLongListener;  
    private List<View> itemViewList = new ArrayList<>();  
  
    public MyPlansAdapter(Context context, int resource, ArrayList<MyPlans> data,  
        MyPlansAdapter.OnMyPlansClickListener listenerShort,  
        MyPlansAdapter.OnMyPlansLongClickListener listenerLong ) {...}  
  
    @Override  
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {...}  
  
    @Override  
    public void onBindViewHolder(final ViewHolder holder, final int position) {...}  
  
    @Override  
    public int getItemCount() { return data.size(); }  
  
    static class ViewHolder extends RecyclerView.ViewHolder {...}  
  
    public interface OnMyPlansClickListener {...}  
  
    public interface OnMyPlansLongClickListener {...}  
  
}
```

Figura 70. RecyclerView (Android Studio)

Restaurante italiano Sorsi e Morsi	Carrer del Dr. Serrano, 11
Inarya fiestas cumpleaños	Carrer del Mestre Valls, 45
Hotel Magic Aqua Rock Gardens	Calle de Bonn, 6
Gran hotel las fuentes	Avinguda de les Fonts, 26

Figura 71. RecyclerView (Aplicación)

5.4.11 Input Events: Event Listeners

Un receptor de eventos es una interfaz en la clase View que contiene un solo método de callback. Estos métodos serán llamados por el framework de Android cuando la vista con la cual se ha registrado el gestor sea iniciada por la interacción del usuario con el elemento en la UI (30).

Estos métodos son los únicos habitantes de su respectiva interfaz. Para definir uno de estos métodos y manejar sus eventos, se implementa la interfaz anidada en su actividad o se define como una clase anónima (30). Se puede observar un ejemplo en la figura 72.

```
Button btnSaveChanges = findViewById(R.id.edit_profile_save_changes);
btnSaveChanges.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        //...
    }
});
```

Figura 72. Event Listener Android Studio

5.4.12 Toasts

Un aviso proporciona información simple sobre una acción en una pequeña ventana emergente. Solo ocupa la cantidad de espacio necesario para el mensaje, y la actividad en curso permanece visible y admite la interacción. Los avisos desaparecen automáticamente después de un tiempo (31) (ver figuras 73 y 74).

```
Toast.makeText(EditProfileActivity.this, "Hola!", Toast.LENGTH_LONG).show();
```

Figura 73. Toast (Android Studio)



Figura 74. Toast (Aplicación)

5.4.13 Dialogs

Un diálogo es una ventana pequeña que le indica al usuario que debe tomar una decisión o ingresar información adicional. Un diálogo no ocupa toda la pantalla y generalmente se usa para eventos modales que requieren que los usuarios realicen alguna acción para poder continuar (32) (véanse figuras 75 y 76).



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

```
AlertDialog.Builder myBuilder = new AlertDialog.Builder( context: EditProfileActivity.this);
myBuilder.setTitle(getString(R.string.edit_profile_image_title_string))
    .setMessage(getString(R.string.edit_profile_image_message_string))

    .setPositiveButton(android.R.string.yes, new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            //...
        }
    })

    .setNegativeButton(android.R.string.no, listener: null)
    .setIcon(R.drawable.ic_dialog_photo)
    .create()
    .show();
```

Figura 75. Dialog (Android Studio)

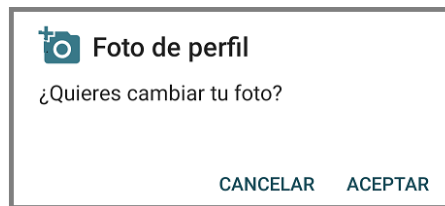


Figura 76. Dialog (Aplicación)

5.4.14 Styles and Themes

Los estilos y temas en Android permiten separar los detalles del diseño de la estructura y del comportamiento de la interfaz de usuario, similar a las hojas de estilo en el diseño web (33) (ver figuras 77 y 78).

Un estilo es una colección de atributos que especifican la apariencia de una sola View. Dicho estilo puede especificar atributos como el color de fuente, el tamaño de fuente, el color de fondo y mucho más (33).

Un tema es un tipo de estilo que se aplica a toda una aplicación, actividad o jerarquía de vistas, no solo a una vista individual. Cuando aplica su estilo como un tema, cada vista en la aplicación o actividad aplica cada atributo de estilo que admite. Los temas también pueden aplicar estilos a elementos que no se ven, como la barra de estado y el fondo de la ventana (33).

```
1 <resources>
2   <!-- Base application theme -->
3   <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
4     <item name="colorPrimary">@color/colorPrimary</item>
5     <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
6     <item name="colorAccent">@color/colorAccent</item>
7   </style>
8 </resources>
```

Figura 77. Styles and Themes (Android Studio)

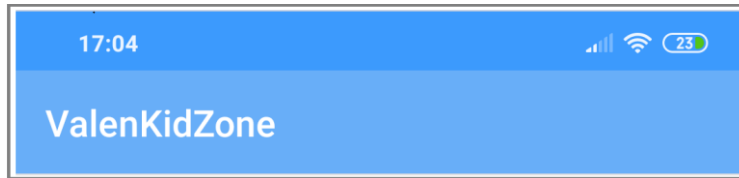


Figura 78. Styles and Themes (Aplicación)

5.4.15 App resources

Siempre se debe externalizar los recursos para aplicaciones como imágenes y strings del código, para poder mantenerlos de forma independiente. También se debe proporcionar recursos alternativos para configuraciones de dispositivos específicos, agrupándolos en directorios de recursos con un nombre especial (34) (véanse figuras 79 y 80).

En tiempo de ejecución, Android utiliza el recurso adecuado según la configuración actual. Por ejemplo, se puede proporcionar un diseño de interfaz de usuario diferente según el tamaño de la pantalla, definir un layout distinto (portaría/landscape) que el dispositivo aplicará en base a cómo lo utilice su usuario, o strings diferentes según la configuración de idioma (34).

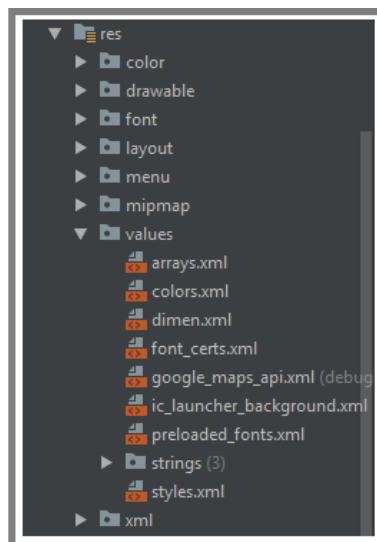


Figura 79. Resources (Android Studio)

Key	Resource Folder	Untranslatable	Default Value	Catalan (ca)	Spanish (es)
account_fragment_edit_string	app/src/main/res	<input type="checkbox"/>	Edit profile	Editar perfil	Editar perfil
account_fragment_followers_string	app/src/main/res	<input type="checkbox"/>	Followers	Seguidors	Seguidores
account_fragment_following_string	app/src/main/res	<input type="checkbox"/>	Following	Seguits	Seguidos
account_fragment_info_title_string	app/src/main/res	<input type="checkbox"/>	Biography	Biografia	Biografía
account_fragment_plan_string	app/src/main/res	<input type="checkbox"/>	Plans	Plans	Planes
account_menu_help_string	app/src/main/res	<input type="checkbox"/>	Help	Ajuda	Ayuda
account_menu_log_out_string	app/src/main/res	<input type="checkbox"/>	Log out	Tancar sessió	Cerrar sesión

Figura 80. Strings (Android Studio)

5.4.16 Google Maps Android API

Con el SDK de Maps para Android, se puede agregar mapas basados en los datos de Google Maps a la aplicación (ver figura 81). La API maneja automáticamente el acceso a los servidores de Google Maps, la descarga de datos, la visualización del mapa y la respuesta a los gestos del mapa (35).

También se pueden usar llamadas a la API para agregar marcadores, polígonos y superposiciones a un mapa básico, y para cambiar la vista del usuario de un área de mapa en particular. Estos objetos proporcionan información adicional para las ubicaciones del mapa y permiten la interacción del usuario con el mapa (35).

La API permite agregar a un mapa elementos como son (35):

- Iconos anclados a posiciones específicas en el mapa (marcadores).
- Conjuntos de segmentos de línea (polilíneas).
- Segmentos cerrados (polígonos).
- Gráficos de mapa de bits anclados a posiciones específicas en el mapa (superposiciones de terreno).
- Conjuntos de imágenes que se muestran en la parte superior de los mosaicos del mapa base (superposiciones de mosaico).



Figura 81. Google Maps (Aplicación)

5.5 Firebase Google

Es una plataforma ubicada en la nube, integrada con Google Cloud Platform, que usa un conjunto de herramientas para la creación y sincronización de proyectos que serán dotados de alta calidad, haciendo posible el crecimiento del número de usuarios y dando resultado también a la obtención de una mayor monetización (36).

Esta plataforma proporciona una serie de ventajas (36):

- Sincronización fácil de los datos de proyectos sin tener que administrar conexiones o escribir lógica de sincronización compleja.
- Uso de un conjunto de herramientas multiplataforma.
- Uso de la infraestructura de Google y escalabilidad automática para cualquier tipo de aplicación.
- Creación proyectos sin necesidad de un servidor.

La plataforma también ofrece una alta variedad de servicios, de entre las cuáles se ha usado **Firestore Authentication** para la autenticación segura de usuarios, **Cloud Firestore** para la creación y utilización de bases de datos y **Cloud Storage** para el almacenamiento de las imágenes.

5.5.1 Firebase Authentication

Proporciona servicios de backend, SDK fáciles de usar y bibliotecas de IU ya elaboradas para autenticar a los usuarios en la App (ver figura 82). Admite la autenticación mediante contraseñas, números de teléfono, proveedores de identidad federada populares, como Google, Facebook y Twitter, y mucho más (37).

Para que un usuario acceda a la App, primero se debe obtener sus credenciales de autenticación del usuario. Estas credenciales pueden ser la dirección de correo electrónico y la contraseña del usuario, o un token OAuth de un proveedor de identidad federada. A continuación, se debe pasar estas credenciales al SDK de Firebase Authentication. Luego, los servicios de backend verificarán esas credenciales y mostrarán una respuesta al cliente (37).



Figura 82. Firebase Authentication

5.5.2 Cloud Firestore

Es una base de datos flexible y escalable para la programación en servidores, dispositivos móviles y la Web desde Firebase y Google Cloud Platform (38) (véase figura 83).

Mantiene los datos sincronizados entre Apps cliente a través de agentes de escucha en tiempo real y ofrece asistencia sin conexión para dispositivos móviles y la Web, por lo que se pueden compilar Apps con capacidad sin importar la latencia de la red ni la conectividad a Internet. También ofrece una integración sin interrupciones con otros productos de Firebase y Google Cloud Platform, incluido Cloud Functions (38).

A partir del modelo de datos NoSQL de Cloud Firestore, almacena los datos en documentos que contienen campos que se asignan a valores. Estos documentos se almacenan en colecciones, que son contenedores para los documentos y que se pueden usar para organizar tus datos y compilar consultas (38).

Los documentos admiten varios tipos de datos diferentes, desde strings y números simples, hasta objetos anidados complejos. También se puede crear subcolecciones dentro de documentos y crear estructuras de datos jerárquicas que se ajustan a escala a medida que la base de datos crece (38).

Además, las consultas de Cloud Firestore son expresivas, eficientes y flexibles. Es posible crear consultas superficiales para recuperar datos en el nivel del documento, sin la necesidad de recuperar la colección completa ni las subcolecciones anidadas (38).

Para mantener actualizados los datos de las Apps sin tener que recuperar toda la base de datos cada vez que haya una actualización, se puede agregar agentes de escucha en tiempo real. Estos notifican con una instantánea de los datos cada vez que los datos de tus Apps cliente escuchan para detectar cambios y recuperan solo los cambios nuevos (38).

Se puede proteger el acceso a los datos en Cloud Firestore con Firebase Authentication y las reglas de seguridad de Cloud Firestore para Android, iOS y JavaScript, o con la administración de identidades y accesos (IAM) para los lenguajes en el lado del servidor (38).

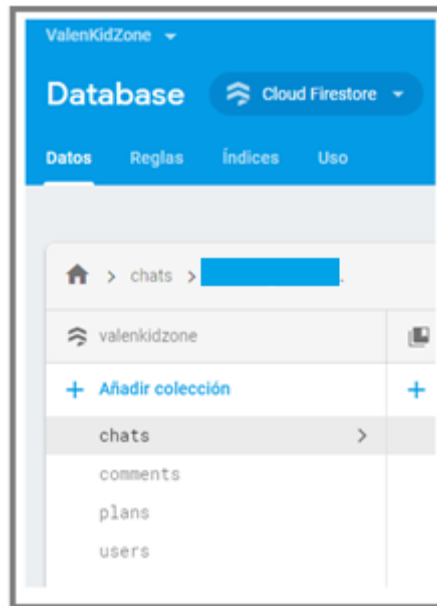


Figura 83. Cloud Firestore

5.5.3 Cloud Storage

Es un servicio de almacenamiento de objetos potente, simple y rentable construido para la escala de Google (ver figura 84). Los SDK de Firebase para Cloud Storage agregan la seguridad de Google a las operaciones de carga y descarga de archivos para las Apps de Firebase, sin importar la calidad de la red. Se pueden usar los SDK para almacenar imágenes, audio, video y otros tipos de contenido generado por el usuario (39).

Se usan los SDK de Firebase para Cloud Storage para subir y descargar archivos directamente de los clientes. Si la conexión a la red es deficiente, el cliente puede reintentar la operación donde la dejó de inmediato, lo cual les ahorra tiempo y ancho de banda a los usuarios (39).

Cloud Storage almacena los archivos en un depósito de Google Cloud Storage y los hace accesibles a través de Firebase y Google Cloud. Esto permite tener la flexibilidad para subir y descargar archivos de clientes móviles a través de los SDK de Firebase y realizar procesamiento en el servidor, como el filtrado de imágenes o la transcodificación de video mediante Google Cloud Platform (39).

Cloud Storage se escala automáticamente, lo que significa que no es necesario migrar a ningún otro proveedor (39).

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

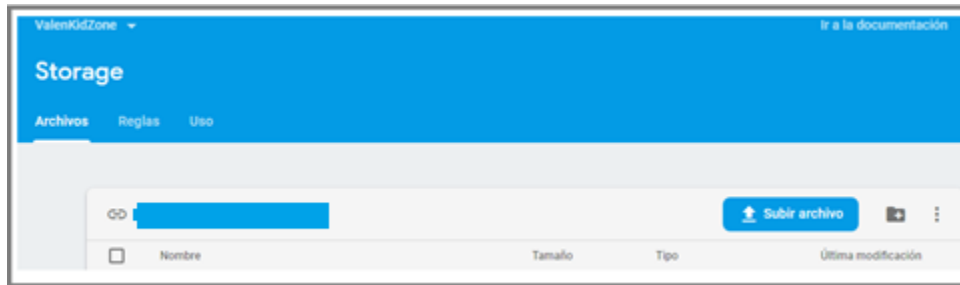


Figura 84. Cloud Storage

Para disminuir el espacio que van a ocupar las imágenes a almacenar en este servicio de almacenamiento, dichas imágenes han sido comprimidas por código (ver figura 85).

```
try {
    Bitmap bmp = MediaStore.Images.Media.getBitmap(getContentResolver(), path);
    ByteArrayOutputStream baos = new ByteArrayOutputStream();
    if (path.toString().endsWith(".jpg"))
        bmp.compress(Bitmap.CompressFormat.JPEG, quality: 25, baos);
    else
        bmp.compress(Bitmap.CompressFormat.PNG, quality: 25, baos);
    byte[] byteData = baos.toByteArray();
    riversRef = pictureDBref.child(currentUser.getUid()).child("profilePicture");
    riversRef.putBytes(byteData).addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {
        @Override
        public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {
            riversRef.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {
                @Override
                public void onSuccess(Uri uri) {
                    firebaseUri = uri;
                    updateData();
                }
            });
        }
    });
} catch (Exception e) {
    Log.e("tag: "Compressing error", e.toString());
}
```

Figura 85. Compresión de imágenes

5.6 Picasso

Picasso es una librería que permite cargar imágenes sin problemas en una aplicación y de una forma relativamente sencilla (40) (véase figura 86).

Una gran cantidad de errores habituales en la carga de imágenes son gestionados automáticamente por Picasso (40):

- Administración del reciclado de los ImageView y cancelación de la descarga en un adapter.
- Transformación compleja de imágenes con un uso de memoria mínimo.
- Caching automático de memoria y disco.

Picasso transforma las imágenes para que encajen mejor en los layouts y así reducir el tamaño en memoria. También se puede especificar transformaciones personalizadas para efectos más avanzados (40).

Soporta poder mostrar ambos tipos de imágenes, las que se muestran antes de que se descargue la imagen real, o la que se mostraría en caso de no poder descargarse. Recursos, activos, archivos, proveedores de contenido son soportados por Picasso como fuentes de imágenes (40).

Para facilitar el desarrollo, se puede habilitar una opción que muestra una pequeña cinta en las esquinas que indica el estado de la imagen (carga de la red, disco o memoria) (40).

```
Picasso.with(getContext()).load(uri.toString())
    .networkPolicy(NetworkPolicy.OFFLINE)
    .resize(targetWidth: 300, targetHeight: 300)
    .centerCrop()
    .transform(new CircleTransform())
    .into(imgAccount, new Callback() {
        @Override
        public void onSuccess() {
        }

        @Override
        public void onError() {
            Picasso.with(getContext()) Picasso
                .load(uri.toString()) RequestCreator
                .resize(targetWidth: 300, targetHeight: 300) RequestCreator
                .centerCrop() RequestCreator
                .transform(new CircleTransform()) RequestCreator
                .into(imgAccount);
        }
    });
```

Figura 86. Picasso (Android Studio)

5.7 Identidad visual

El color, en esencia, es mucho más que un fenómeno óptico. Todos los colores tienen un significado propio, todos crean un impacto determinado en nuestro cerebro, y por ello, la psicología del color es hoy en día una herramienta básica y esencial para la actual neuromercadotecnia (41).

Comprender de qué manera reacciona el consumidor ante determinados estímulos cromáticos puede hacer que se eleve el índice de uso y venta, y aunque su efecto no siempre sea infalible al 100%, se observan patrones de reacción semejantes que nos vienen a demostrar que, efectivamente, la psicología del color tiene su utilidad (41).

La psicología del color es un campo de estudio que está dirigido a analizar el efecto del color en la percepción y la conducta humana. Los colores tienen diferentes interpretaciones y significados en cada cultura, pero debemos saber que el color influye de manera asombrosa en el estado de ánimo (42).

Por ello hemos creado una guía de uso de identidad visual corporativa que se muestra en la figura 87. Este diseño ha sido creado por Rosario Sanchis Font, con un significado de felicidad (globos, payasos y familias felices) y el localizador de Google. La tipografía del logo es "Letters for Learners", gratis para uso personal o académico.



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

Si la App se hiciera comercial o divulgativa sin usos académicos a gran escala, habría que pagar derechos de uso. Esta memoria, asimismo, sigue estos aspectos de diseño: colores y tipografía.

La App se ha nombrado "**KidZone Valencia**" de forma que, si se amplía a otras zonas o ciudades, se puede cambiar el nombre sin perder la identidad de la marca.

**Identidad Visual Corporativa
App KidZone Valencia
Mayo 2019**



Tipografía principal: *Letters for Learners*

Tipografía secundaria (recomendada textos largos): Arial

Colores corporativos:

■ Rojo: (RGB) R201 G91 B80 / (Pantone) 7418 C / (CMYK) C0% M 83,29% Y66,46% K0% / Hex #c95b50

■ Azul: (RGB) R37 G107 B167 / (Pantone) 7683 C / (CMYK) C95% M 50% Y9% K0% / Hex #256ba7

Negro y Blanco

Figura 87. Identidad visual corporativa App KidZone Valencia

6. Conclusiones

En lo que respecta a la realización de este trabajo de final de grado, no sólo la mayor parte de los objetivos técnicos se han cumplido, sino que se ha realizado un estudio de mercado para guiar determinadas cuestiones de diseño, como la autenticación de los usuarios o el conjunto de categorías de plan ofrecidas a los usuarios.

Esta aplicación alberga muchísimas funcionalidades y es necesario más tiempo del que se ha empleado para la realización de todas ellas. La subida de la App al Google Play requiere de un trabajo adicional de depuración y prueba que se está llevando a cabo. En un plazo breve de tiempo se espera subir la App para comenzar a realizar pruebas reales con la misma.

De cara al diseño, y desarrollo de la aplicación, ha sido necesario poner en práctica muchos conocimientos adquiridos durante la carrera, como son el diseño centrado en el usuario y el desarrollo de software, pero también aprender otros nuevos, como el uso de Google Maps Android API y Firebase Google.

La aplicación está diseñada para ser intuitiva y amigable. Se han seguido los patrones de diseño de Google y el estudio de la psicología del color. Los colores más utilizados han sido azul (calma, confianza y seguridad), verde (salud, naturaleza, bienestar y paz) y blanco (pureza, felicidad y sinceridad). También hemos solicitado el punto de vista de los usuarios y ya con éste hemos incidido en las características de la App.

Cabe destacar el uso de fragments y adapters en muchas partes de nuestra aplicación, lo que la convierte en una App que ha sido desarrollada con buenas y recomendadas prácticas. Esto aporta ventajas como asincronía en el comportamiento de la App, lo que la hace más reactiva de cara a sus usuarios, y una gestión más eficiente del GUI.

Durante el desarrollo, nos hemos encontrado con múltiples problemas tales como los retardos de respuesta por parte de los servicios de Firebase Google y el funcionamiento no esperado de ciertos elementos como fragments y viewpagers, los cuáles hemos conseguido solucionar o mitigar de la mejor forma posible. Pero sin duda, el mayor problema al que nos hemos enfrentado ha sido el tiempo del que hemos dispuesto para la realización de esta aplicación.

En conclusión, hemos creado una App completa, lo que se denomina un MVP, y tenemos múltiples posibles mejoras futuras identificadas y que vamos a incorporar en la primera versión que se va a suministrar a un grupo controlado de usuarios potenciales con el objetivo de terminar de perfilar el producto y comenzar a validar las mejoras a introducir.



Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles

7. Ampliaciones y mejoras futuras

En este último apartado vamos a describir algunas de las posibles ampliaciones y mejoras futuras de nuestra aplicación.

7.1 Chat

En esta parte, deberíamos añadir más funcionalidades además de las que ya hay, tales como crear chat grupal, bloquear, desbloquear, crear una lista de contactos, vaciar chats, eliminar chats, etc.

7.2 Menú de planes

Con respecto al menú de planes, tendríamos que añadir la opción de reportar un plan y de valorar los comentarios de la gente, así como llevar a cabo un análisis de los planes más buscados y los que menos, para valorar cambios en el menú.

7.3 Notificaciones

Esta parte habría que hacerla entera, de momento solo dispone de una imagen de aviso. La idea es utilizar esta parte de la aplicación para avisar al usuario de cambios realizados en planes que les gustan, o de personas a las que siga, o de creación de nuevos planes, y más.

7.4 Cuenta

Habría que valorar si los usuarios consideran necesario la sección de biografía además de si deberíamos de añadir alguna otra funcionalidad en este apartado. También tendría que poder visitarse el perfil de otras personas.

7.5 Seguidos y seguidores

Habilitar la opción de seguir a otros usuarios, así como dejar de seguir. La finalidad es añadir las novedades acerca de este usuario seguido al apartado de notificaciones.

7.6 Ayuda

En esta parte tendríamos que crear algún vídeo tutorial y/o guía de uso para los usuarios. También una guía de respuesta de las preguntas frecuentes que tengan los usuarios de nuestra aplicación y un apartado describiendo quiénes somos los creadores de la App y como podrían ponerse en contacto con nosotros.

7.7 Ajustes

Con respecto a los ajustes, se podrían añadir varias cosas, entre las cuáles destacarían la opción de cambiar de dirección de correo electrónico y de borrar la cuenta.

7.8 Analíticas

Como hemos mencionado en alguna de las mejores propuestas anteriormente, sería una buena idea medir el uso que hacen los usuarios de nuestra aplicación, no solo en lo que respecta a la elección de planes, sino también a la pulsación de botones y más.

7.9 Mantenimiento

Las actualizaciones deberían suministrarse siempre a través del store oficial de Apps Android (Google Play). Esto significa que hemos de introducir en la arquitectura software de la App patrones de diseño que permitan actualizar/mantener fácilmente la App, corrigiendo errores, y proponiendo a los usuarios nuevas funcionalidades que, aunque desconocidas actualmente, la App ha sido ya pensada para facilitar la futura integración de dichas funcionalidades.

7.10 Migración

Una vez se suba la aplicación al Google Play, se podría estudiar la migración de la aplicación hacia los dispositivos con sistema operativo iOS.

7.11 Monetización

La aplicación es gratuita, pero pasado un tiempo se podrían añadir contenidos extras de pago, así como buscar sponsors.

Referencias

1. **Olivero, Emma. Uso de dispositivos móviles.** [En línea] pickaso, 13 de 12 de 2017. <https://pickaso.com/2017/estudio-tiempo-uso-dispositivos-moviles>
2. **Juste, M. Las cifras de uso de internet.** [En línea] expansion, 01 de 02 de 2018. <http://www.expansion.com/economia-digital/innovacion/2018/02/01/5a72e73a22601db2288b4658.html>
3. **App (Application).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica
4. **SDK (Software Development Kit).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software
5. **API (Application Programming Interface).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones
6. **UI (User Interface).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Interfaz_de_usuario
7. **GUI (Graphical User Interface).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario
8. **MVP (Minimum Viable Product).** [En línea] Wikipedia. https://es.wikipedia.org/wiki/Producto_viable_m%C3%ADnimo
9. **Google Forms.** [En línea] <https://www.google.es/intl/es/forms/about/>
10. **Draw.io.** [En línea] <https://about.draw.io/>
11. **NinjaMock.** [En línea] <https://ninjamock.com/>
12. **VisualParadigmOnline.** [En línea] <https://online.visual-paradigm.com/es/>
13. **Java.** [En línea] Wikipeda. [https://es.wikipedia.org/wiki/Java_\(lenguaje_de_programaci%C3%B3n\)](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n))
14. **Java.** [En línea] Wikipedia. https://es.wikibooks.org/wiki/Programaci%C3%B3n_en_Java/Caracter%C3%ADsticas_del_lenguaje
15. **Android.** [En línea] Wikipedia. <https://es.wikipedia.org/wiki/Android>
16. **Android Studio.** [En línea] Android Developers. <https://developer.android.com/studio/intro?hl=es-419>
17. **Processes and threads.** [En línea] Android Developers. <https://developer.android.com/guide/components/processes-and-threads>



18. Manifest. [En línea] Android Developers.

<https://developer.android.com/guide/topics/manifest/manifest-intro?hl=es-419>

19. Permissions. [En línea] Android Developers.

<https://developer.android.com/guide/topics/security/permissions.html?hl=es-419>

20. Activities. [En línea] Android Developers.

<https://developer.android.com/guide/components/activities.html?hl=ES>

21. Layouts. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/declaring-layout>

22. Fragments. [En línea] Android Developers.

<https://developer.android.com/guide/components/fragments>

23. Menus. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/menus>

24. ViewPager. [En línea] Android Developers.

<https://developer.android.com/reference/androidx/viewpager/widget/ViewPager.html>

25. BottomNavigationView. [En línea] Android Developers.

<https://developer.android.com/reference/android/support/design/widget/BottomNavigationView>

26. Material design. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/look-and-feel>

27. FloatingActionButton. [En línea] Android Developers.

<https://developer.android.com/reference/android/support/design/widget/FloatingActionButton>

28. CardView. [En línea] Android Developers.

<https://developer.android.com/reference/android/support/v7/widget/CardView>

29. RecyclerView. [En línea] Android Developers.

<https://developer.android.com/reference/kotlin/androidx/recyclerview/widget/RecyclerView>

30. Event listeners. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/ui-events#EventListeners>

31. Toasts. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/notifiers/toasts>

32. Dialogs. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/dialogs>

33. Styles and themes. [En línea] Android Developers.

<https://developer.android.com/guide/topics/ui/look-and-feel/themes.html>

34. App resources. [En línea] Android Developers.

<https://developer.android.com/guide/topics/resources/providing-resources>

- 35. Google Maps Android API.** [En línea] Google Developers.
<https://developers.google.com/maps/documentation/android-sdk/intro>
- 36. Firebase Google.** [En línea] Wikipedia. <https://es.wikipedia.org/wiki/Firebase>
- 37. Firebase Authentication.** [En línea] Firebase Google.
<https://firebase.google.com/docs/auth?hl=es-419>
- 38. Cloud Firestore.** [En línea] Firebase Google.
<https://firebase.google.com/docs/firestore?hl=es-419>
- 39. Cloud Storage.** [En línea] Firebase Google.
<https://firebase.google.com/docs/storage>
- 40. Picasso.** [En línea] Desarrollador Android. <https://desarrollador-android.com/librerias/square/picasso/>
- 41. Psicología del color.** [En línea] Wikipedia.
https://es.wikipedia.org/wiki/Psicolog%C3%ADa_del_color
- 42. Psicología del color.** [En línea] Lamenteesmaravillosa.
<https://lamenteesmaravillosa.com/psicologia-del-color-curiosidades/>

Diseño e implementación de una aplicación para la búsqueda de espacios de ocio con requisitos en dispositivos móviles