

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN



"APLICACIÓN WEB PARA LA GESTIÓN Y
RESERVAS DE ESCAPERROOMS"

TRABAJO FIN DE GRADO

Septiembre - 2022

AUTOR: Adam Gómez Mouti

TUTOR: Juan Pedro Moreno Martín-Viveros

Agradecimientos

Primero me gustaría agradecer a mi familia, por apoyarme durante todos estos años de trabajo y esfuerzo, por la ayuda y el apoyo que me han brindado para continuar y conseguir mis objetivos aunque fueran complicados de conseguir.

También agradecer a mis compañeros de clase Carmelo, Fran, Johnny y Jose Antonio, con los que he entablado una amistad que espero mantener el resto de mi vida y con los que he colaborado para seguir adelante con la carrera. Otra persona a la que tengo que agradecer es a Adrián Paredes Fuentes del Grado en Comunicación Audiovisual de la UMH, que me ha ayudado con los iconos del inicio de la aplicación.

Y por último, agradecer a todos los profesores que se han dedicado a instruirme durante todos estos años y en particular a Juan Pedro Moreno Martín-Viveros, que aceptó tutorizar este proyecto y me ha ayudado a salir de dudas cuando las he tenido.



Índice de contenido

CAPÍTULO 1. INTRODUCCIÓN	10
1.1 Introducción e importancia	10
1.2 Razones de la elección	12
1.3 Qué se pretende conseguir	12
1.4 Qué queda fuera de este proyecto	13
CAPÍTULO 2. ANTECEDENTES, ESTADO DE LA CUESTIÓN	14
CAPÍTULO 3. OBJETIVOS DEL PROYECTO	35
3.1 Objetivos principales	35
3.2 Objetivos secundarios	35
3.3 Conocimientos previos	35
3.4 Estudio y puesta en marcha.	36
CAPÍTULO 4. HIPÓTESIS DE TRABAJO	37
4.1 Elección de las herramientas	37
4.1.1 Editor de código	37
4.1.2 Lenguaje de programación	38
4.1.2.1 Flutter	38
4.1.3 Servidor y Base de datos	40
4.1.3.1 Firebase	40
4.1.3.2 Cloud Firestore	42
4.2 Equipo Físico	43
CAPÍTULO 5. METODOLOGÍA Y RESULTADOS	44
5.1 Introducción al proceso de ingeniería del software	44
5.2 Planificación	44
5.2.1 Ciclo de vida	44
5.3 Modelado UML	46
5.3.1 Descripción de actores	46

5.3.2 Diagrama de casos de uso	47
5.3.3 Diagrama de actividades	58
5.3.4 Diagrama de clases	61
5.4 Descripción de la base de datos	62
5.5 Publicación de la aplicación	64
CAPÍTULO 6. CONCLUSIONES Y PROPUESTAS	69
6.1 Conclusiones	69
6.2 Trabajo futuro.	69
ANEXO I	74
ANEXO II	91

Índice de imágenes

Imagen 1. Página principal de EscapeRank	14
Imagen 2. Submenús de EscapeRank	15
Imágenes 3 y 4. Detalles de escaperoom de EscapeRank	16
Imágenes 5 y 6. Ranking de jugadores de EscapeRank	17
Imágenes 7, 8, 9 y 10. Proceso de creación de equipo en EscapeRank	18
Imagen 11. Formulario de inicio de sesión en EscapeRank	19
Imagen 12. Pantalla de perfil en EscapeRank	19
Imagen 13. Lista de escaperooms favoritos en EscapeRank	19
Imagen 14. Lista de escaperooms realizadas en EscapeRank	20
Imagen 15. Lista de escaperooms reservadas y completadas en EscapeRank	20
Imagen 16. Página principal de Escape Radar.	21
Imágenes 17 y 18. Detalles de escaperoom de Escape Radar.	22

Imagen 19. Submenú de Escape Radar.	22
Imagen 20. Integración de GMaps en Escape Radar.	23
Imagen 21. Inicio de sesión en Escape Radar.	24
Imagen 22. Lista de escaperooms jugadas en Escape Radar.	25
Imagen 23. Lista de escaperooms favoritas en Escape Radar.	25
Imagen 24. Lista de escaperooms ocultas en Escape Radar.	25
Imagen 25. Pantalla de configuración en Escape Radar.	26
Imagen 26. Pantalla de información sobre Escape Radar.	26
Imagen 27. Pantalla de contacto en Escape Radar.	26
Imagen 28. Pantalla de inicio de Padlock.	27
Imagen 29. Opciones en el menú hamburguesa de Padlock.	27
Imagen 30. Lista de escaperooms de Padlock.	28
Imagen 31. Pantalla de detalles de escaperoom de Padlock.	29
Imagen 32. Lista de escaperooms online de Padlock.	30
Imagen 33. Pantalla de detalles de escaperoom online de Padlock.	30
Imagen 34. Lista de usuarios de Padlock.	31
Imagen 35. Perfil de un usuario de Padlock.	31
Imagen 36. Lista de eventos de Padlock.	32
Imagen 37. Pantalla de detalles de un evento de Padlock.	32
Imagen 38. Lista de escaperooms más votadas de Padlock.	32
Imagen 39. Formulario de voto de escaperoom de Padlock.	32
Imagen 40. Lista de productos en la tienda de Padlock.	33
Imagen 41. Lista de juegos de mesa ofrecidos por Padlock.	33

Imagen 42. Pantalla del perfil de usuario de Padlock.	34
Imagen 43. Logo de Visual Studio Code.	37
Imagen 44. Logo de Flutter.	38
Imagen 45. Logo de Dart.	39
Imagen 46. Logo de Firebase.	40
Imagen 47. Método en cascada.	45
Imagen 48. Diagrama de usuario no registrado	47
Imagen 49. Diagrama de usuario con cuenta	49
Imagen 50. Diagrama de administrador de escaperoom	53
Imagen 51. Diagrama de administrador	56
Imagen 52. Diagrama de actividad del inicio de sesión	59
Imagen 53. Diagrama de actividad de reserva en escaperoom	60
Imagen 54. Diagrama de clases relacionado con las reservas	61
Imagen 55. Diagrama de clases relacionado con las escaperooms	62
Imagen 56. Repositorio del proyecto en GitHub.	65
Imagen 57. Carpeta del proyecto en local	66
Imagen 58. Proyecto abierto en Visual Studio Code	67
Imagen 59. Resultado del comando flutter doctor	67
Imagen 60. Resultado del comando flutter run	68
Imagen 61. Splash screen	74
Imagen 62, 63, 64 y 65. Carrusel de preferencias	75
Imagen 66 y 67 . Formulario de registro	76
Imagen 68 y 69. Formulario de inicio de sesión	77

Imagen 70 y 71. Pantalla principal	79
Imagen 72, 73 y 74. Resultados de búsqueda	79
Imagen 75 y 76. Detalles de escaperoom	80
Imagen 77. Descripción de escaperoom	81
Imagen 79. Lista de partidas disponibles para unirse	82
Imagen 80. Formulario para unirse a la partida	82
Imagen 81. Notificación de unión con éxito	82
Imagen 82. Perfil público del usuario	83
Imagen 83. Perfil privado del usuario	84
Imagen 84. Lista de salas favoritas	84
Imagen 85. Lista de salas completadas	85
Imagen 86. Formulario para cambiar nombre de usuario y contraseña.	85
Imagen 87. Opciones del menú hamburguesa	86
Imagen 88. Ranking de usuarios	86
Imagen 89. Perfil de otro usuario	86
Imagen 90. Lista de reservas	87
Imagen 91. Detalles de reserva	87
Imagen 92. Pantalla con información sobre la aplicación	87
Imagen 93. Pantalla principal para administradores	88
Imagen 94 y 95. Formulario de creación de una escaperoom	89
Imagen 96. Detalles de escaperoom para administradores	90
Imagen 97. Menú hamburguesa para administradores	91
Imagen 98. Lista de reservas para administradores	91

Imagen 99. Datos de reserva	91
Imagen 100. Declaración de referencias a las colecciones	91
Imagen 101. Interfaz de la clase usuario	92
Imagen 102. Función get usuarios	92
Imagen 103. Declaración de multiproviders en main	93
Imagen 104. Instanciación de variables con los datos de los providers	94
Imagen 105. Ejemplo de uso de variable tipo usuario	94

Índice de tablas

Tabla 1. Descripción del rol de “Usuario no registrado”	46
Tabla 2. Descripción del rol de “Usuario con cuenta”	46
Tabla 3. Descripción del rol de “Administrador de escaperoom”	47
Tabla 4. Descripción del rol de “Administrador”	47
Tabla 5. Caso de uso 1.1	48
Tabla 6. Caso de uso 1.2	49
Tabla 7. Caso de uso 2.1	50
Tabla 8. Caso de uso 2.2	51
Tabla 9. Caso de uso 2.3	51
Tabla 10. Caso de uso 2.4	52
Tabla 11. Caso de uso 2.5	53
Tabla 12. Caso de uso 3.1	54
Tabla 13. Caso de uso 3.2	54
Tabla 14. Caso de uso 3.3	55

Tabla 15. Caso de uso 3.4	56
Tabla 16. Caso de uso 4.1	57
Tabla 17. Caso de uso 4.2	58
Tabla 18. Caso de uso 4.3	58



CAPÍTULO 1. INTRODUCCIÓN

1.1 Introducción e importancia

Una escaperoom es una actividad recreativa que consiste en escapar de una habitación dentro de un tiempo determinado descubriendo pistas, y resolviendo puzzles. Normalmente ambientado en historias de terror, de ciencia ficción y fantasía.

En la actualidad se está explorando el uso de estas actividades en el entorno escolar como forma de aprendizaje para desarrollar el trabajo en equipo y las aptitudes mentales de los niños. Similar al uso escolar, también en el mundo laboral algunas empresas han utilizado escaperooms para fomentar el trabajo en equipo y acercar a los trabajadores entre ellos, ya que deben colaborar para completar el objetivo.

Tiene unas bases fuertemente arraigadas en el cine y la literatura, aunque una de las primeras manifestaciones se remonta al boom en los años setenta en los juegos de rol siendo el más conocido Dragones y Mazmorras [1].

Volviendo al tema del cine y literatura, son formas de ocio cuya finalidad es el entretenimiento a través de una historia, poco a poco está evolucionando hasta el punto en el que el consumidor forme parte de la historia, un ejemplo son los videojuegos, donde el jugador controla al protagonista, eso hace que empatice mucho más con los personajes y tenga un impacto mayor que viendo una película o leyendo un libro.

En cuanto a las escaperooms, en un principio fueron simples puzzles y minijuegos, pero últimamente se están basando en historias en las que el protagonista directo es el jugador y hace que la inmersión sea muchísimo mayor.

Las escaperooms son un negocio joven que llegó a España en 2013, pero hasta el 2016 no empezó a expandirse llegando a 150 compañías dedicadas a

estas actividades, ya en 2018 se registraron 677, un aumento del 451% estimando que el sector mueve alrededor de 20 millones de euros en España [2]. En 2019 siguió creciendo, alcanzando hasta 974 empresas y autónomos, con 1.721 salas de escape, contando con aproximadamente 3.000 empleos directos y 10.000 indirectos en servicios auxiliares [3]. *“La Asociación Española de Juegos en Vivo Escape Room (Aejever) publicaba en abril de 2020 un informe en el que afirmaba que, en 2019, existían cerca de 1.000 empresas con este tipo de juegos en España, sumando un volumen de negocio total de 173,5 millones de euros”* [6].

En 2020 el sector sufrió gravemente las consecuencias de la pandemia del COVID-19, ya que durante el confinamiento era imposible participar en este tipo de actividades de ocio, se estima que un 25% de las salas cerraron y no volvieron a abrir, ya que se pasó de un volumen de reservas cercano al 80% en 2019 a un 15% de reservas. Está claro que no fue el único negocio que se vio afectado por la pandemia pero la naturaleza de ocio que tiene dió lugar a una gran repercusión económica en la que muchos trabajadores perdieron su empleo [5]. *“La Asociación Española de Juegos en Vivo Escape Room (Aejever) calcula que el cierre forzado habría supuesto unas pérdidas aproximadas de 25 millones de euros para el sector”* [6].

[4] Por otra parte, se llevó a cabo un estudio de mercado el verano de 2021, en comparación a 2019. Se realizó un conteo de las empresas, que pasaron de 974 a 700, y de 1.721 salas a 1.500. Los siguientes datos son de un 10% de las 700 actuales.

- Sobre los Escaperooms: Comentan que la gran mayoría escoge la sala de escape tradicional ya que, últimamente, con los avances que se hacen en el ocio están surgiendo nuevos tipos de experiencias como por ejemplo las de realidad virtual, lo que tiene sentido, ya que el equipamiento VR supone una gran inversión que muchas empresas no están dispuestas a afrontar. La temática de los juegos más comunes suelen ser las clásicas de aventuras y misterio con la dificultad media de 3 y 4 candados sobre 5. La duración de las partidas rondan los 60-90 minutos. Sobre los idiomas predominan el español y

el inglés, incluyendo los idiomas regionales de las zonas en las que se sitúe cada sala.

- Sobre los Clientes: Indican que la mayoría son locales y aficionados de la misma ciudad que ya han participado en algún evento del estilo, suelen ser grupos de 4-5 personas de entre 24-35 años. Con un precio entre 60-80 euros, las reservas que se realizan muestran que tres cuartas partes de estas se hacen en la misma semana en la que van a realizar la actividad ya sea mediante la web de la escaperoom, o llamando al establecimiento. La mayoría de jugadores suele repetir si el establecimiento tiene más de un juego. Las épocas clave de reservas son en diciembre y enero siendo la mayoría en fines de semana entre las 16:00 y 19:59.

1.2 Razones de la elección

En cuanto a las razones de la elección, se ha escogido esta temática debido a que se trata de un mercado nuevo que está en auge, que todavía no está muy explorado. Hay bastante donde mejorar y aportar ideas. Teniendo en cuenta que apenas existen gestores de reservas especializados en este sector y sería de gran ayuda tanto para organizadores como para usuarios. Además de hacer más accesible la entrada a este entorno a gente que pudiera sentirse intimidada.

1.3 Qué se pretende conseguir

Utilizando conocimientos y tecnologías aprendidas dentro y fuera del grado se busca realizar una aplicación que permita a una persona reservar sesiones para participar en una escaperoom y en el caso en el que no cumpla el mínimo de jugadores la posibilidad de unirse a una partida reservada por otro grupo que tenga huecos libres.

Y por la parte de los representantes de las escaperooms la posibilidad de ofrecer sus actividades al público.

1.4 Qué queda fuera de este proyecto

De las ideas iniciales que tenía planeadas desarrollar quedan fuera un apartado de noticias relacionadas con escaperooms que podrían interesar al usuario, p.e. apertura de nuevas salas de escape, premios, eventos, etc.

Otra funcionalidad que no se ha implementado, sería la interacción entre usuarios, chats privados, chats grupales, organizaciones de jugadores y más.

También queda fuera la opción de registrarse e iniciar sesión con Google y Facebook.

Por último, se queda en el tintero la posibilidad de por la parte de los administradores de las escaperooms, realizar cambios en las reservas, como cancelarlas y modificar los datos de las escaperooms publicadas, aunque esto se puede conseguir borrándose y volviendo a crearlas.

CAPÍTULO 2. ANTECEDENTES, ESTADO DE LA CUESTIÓN

En la actualidad, existen pocas aplicaciones móviles a nivel nacional que sirvan exclusivamente el mismo propósito que la aplicación desarrollada para este trabajo. Pero sí cabe mencionar que al realizar una rápida búsqueda en internet hay páginas como (www.escaperoomlover.com o www.escapistas.club) que suplen esta necesidad. También existen páginas informativas de actividades de ocio entre las que se encuentran las escaperooms, aunque claramente con una información mucho menos elaborada sobre éstas.

A continuación, se enumeran una serie de aplicaciones con funcionalidades similares a las de la aplicación desarrollada: EscapeRank, Escape Radar y Padlock.

EscapeRank

Para empezar se ha probado EscapeRank, un buscador de escaperooms en España. Al abrir la aplicación muestra un buscador en la parte superior y una serie de salas categorizadas entre novedades, cercanas al usuario y temáticas.

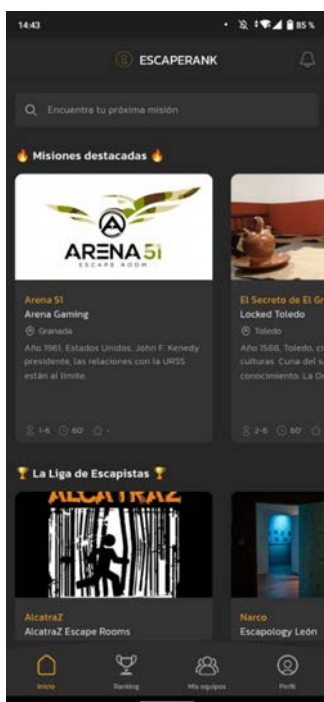


Imagen 1. Página principal de EscapeRank

Tiene 4 submenús que dan al inicio que se han comentado, un ranking, un creador y consultor de equipos propios, y el perfil.

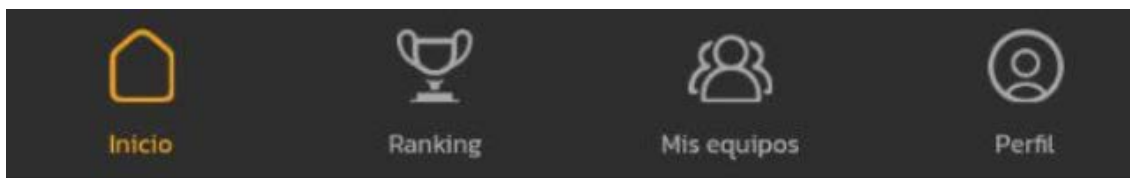
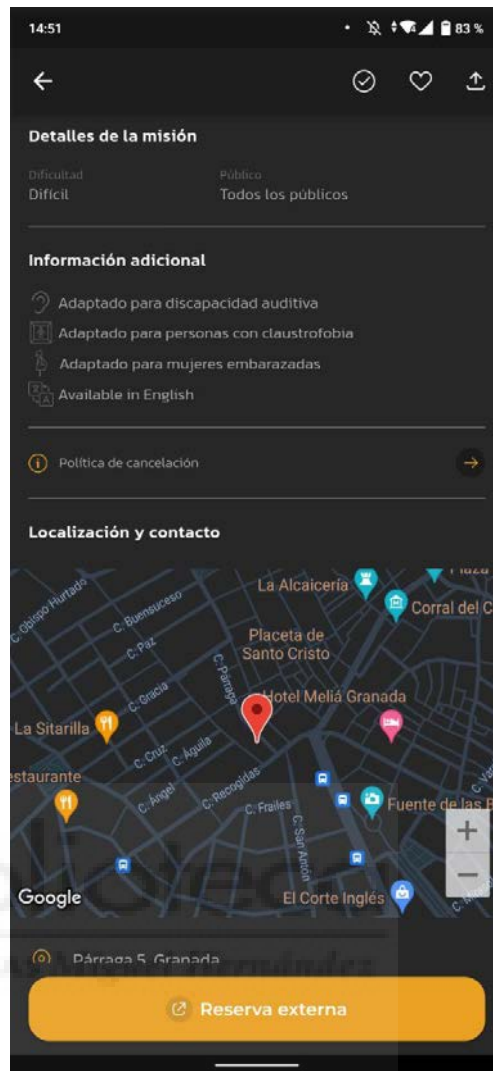


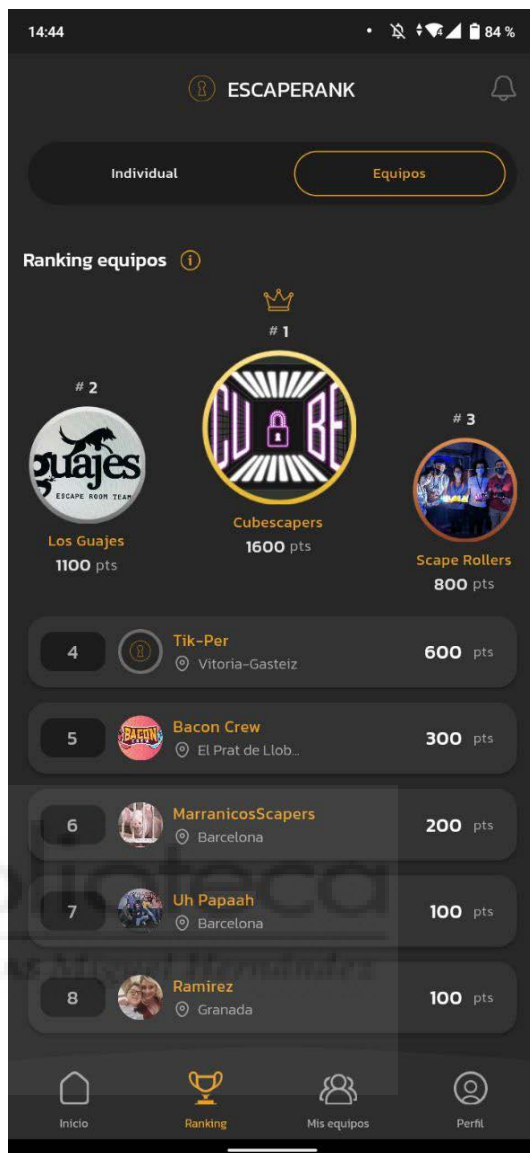
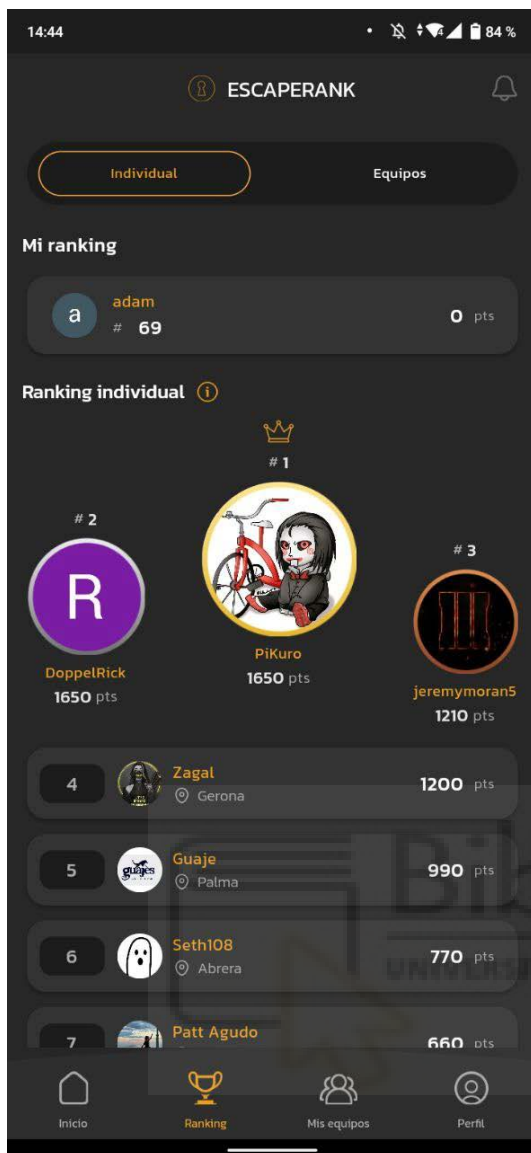
Imagen 2. Submenús de EscapeRank

Cuando se pulsa en cualquier sala te muestra toda la información sobre esta como la ubicación de la sala, la temática, el número de jugadores y tiempo de juego, el rango de precios, una descripción, la empresa a la que pertenece (permite pulsar y muestra una lista de las salas que tiene), algunos detalles (dificultad y público), las políticas de cancelación, un widget de Google Maps con la localización exacta, la información de contacto (nº tlf, email, página web) y un botón de reserva que te lleva a la página web de la sala. También puedes compartir la sala y marcarla como favorita en la parte superior.



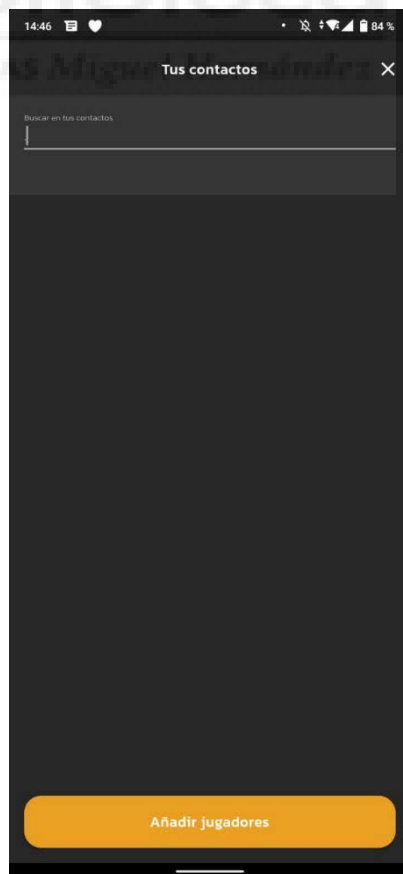
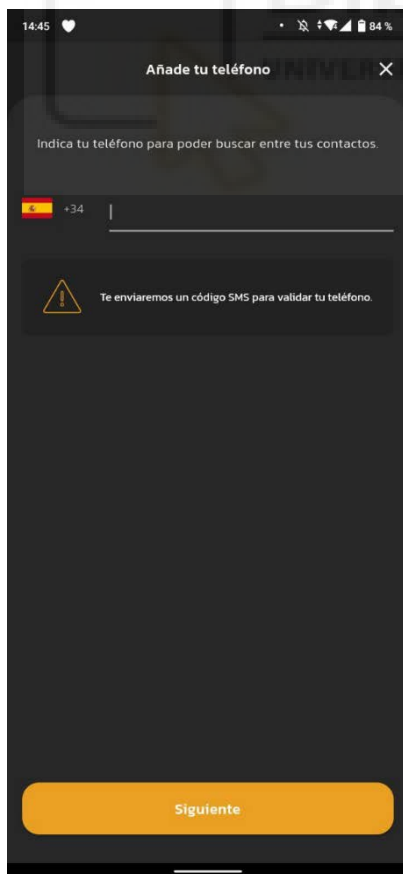
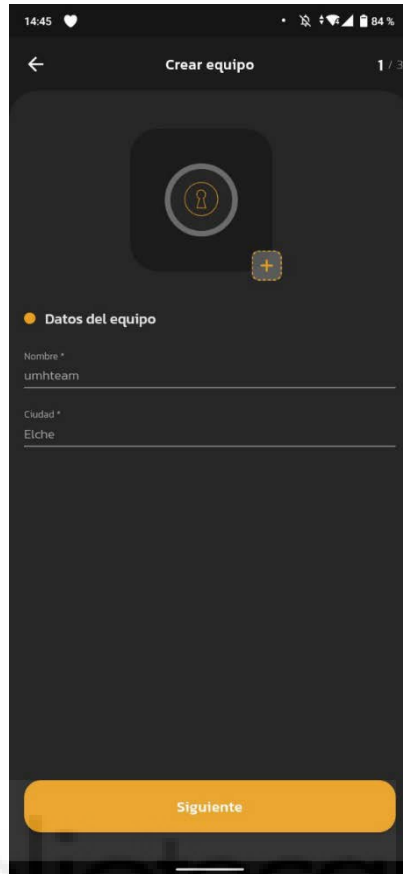
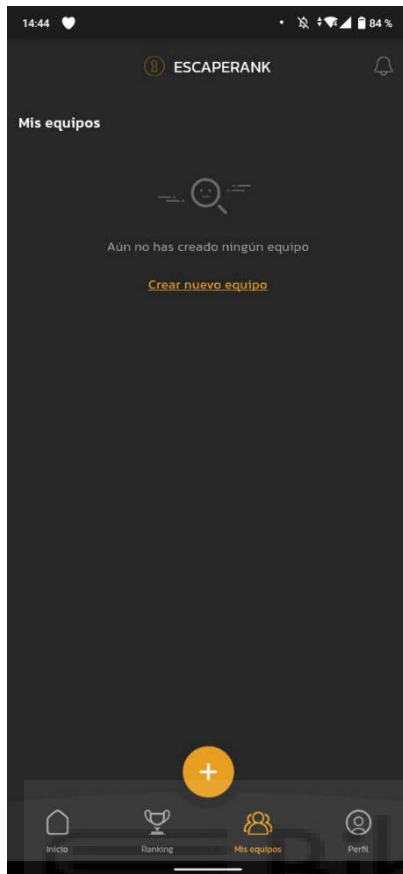
Imágenes 3 y 4. Detalles de escaperoom de EscapeRank

El submenú de ranking muestra una lista tanto de los jugadores como de los equipos que más puntos han conseguido a base de participar en escaperooms.



Imágenes 5 y 6. Ranking de jugadores de EscapeRank

El submenú de “Mis equipos” mostraría una lista de equipos a los que se pertenece y al intentar crear un nuevo equipo hay un proceso que va desde los datos del equipo, tras ello pide el número de teléfono para poder añadir en el siguiente paso nuestros contactos al equipo.



Imágenes 7, 8, 9 y 10. Proceso de creación de equipo en EscapeRank

Del mismo modo que el submenú de favoritos, el de perfil te pide iniciar sesión o crear una cuenta, pudiendo acceder con Google y Facebook. Con la sesión iniciada muestra tu nombre, localización, un botón de configuración y otro de cerrar sesión. El botón de configuración te permite cambiar datos personales como el nombre y apellidos, la fecha de nacimiento, el teléfono y la ciudad, tiene otro submenú donde puedes cambiar tu nombre de usuario y la contraseña.

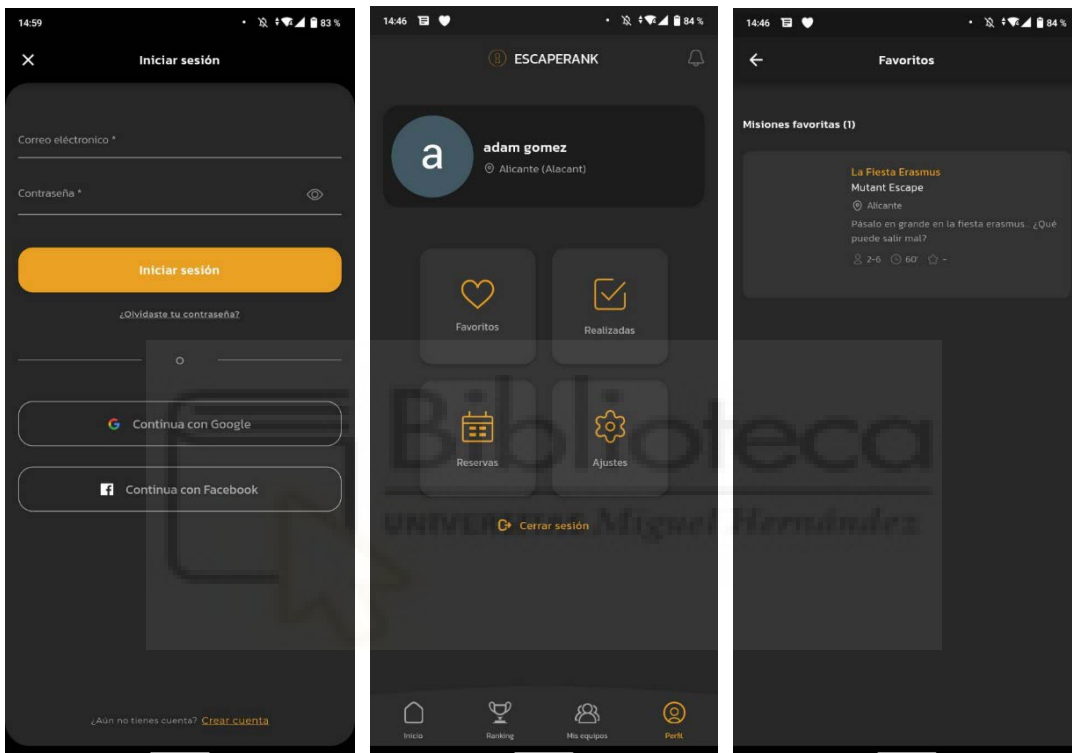


Imagen 11. Formulario de inicio de sesión en EscapeRank

Imagen 12. Pantalla de perfil en EscapeRank

Imagen 13. Lista de escaperooms favoritos en EscapeRank

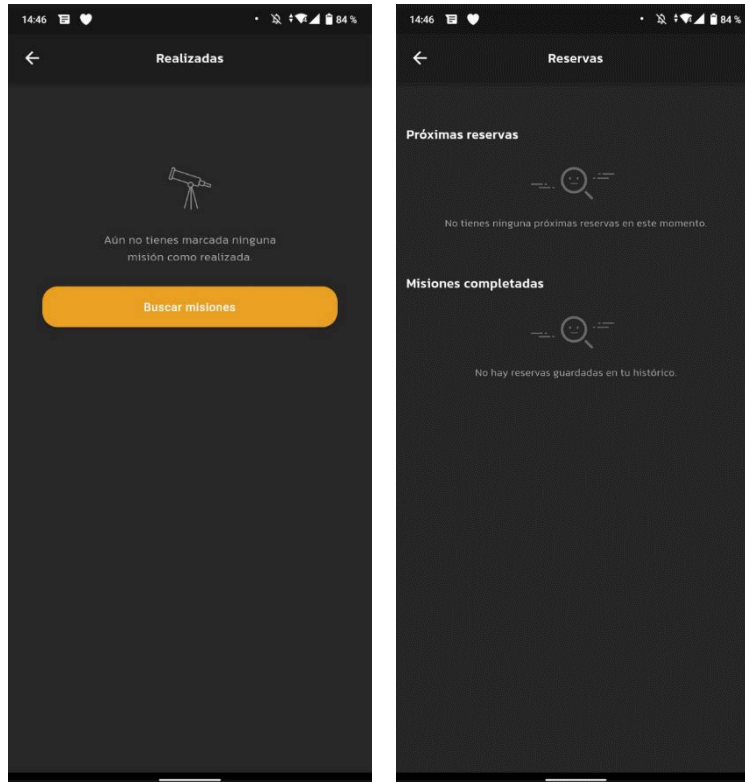


Imagen 14. Lista de escaperooms realizadas en EscapeRank

Imagen 15. Lista de escaperooms reservadas y completadas en EscapeRank

Para concluir, la aplicación cuenta con muchas de las funcionalidades que tenía pensado añadir como el buscador, la opción de marcar como favoritos o la distinción entre temáticas, pero al ser un simple buscador de salas de escape poco más se le podría añadir, en mi opinión es una buena aplicación de búsqueda de salas de escape.

Escape Radar

La segunda aplicación se ha probado es Escape Radar. Similar a la anterior, es un buscador de salas de escape en España. El inicio muestra un buscador, un grid con las principales ciudades que al pulsar muestran las salas de escape más famosas de la ciudad, salas nuevas y las más votadas.

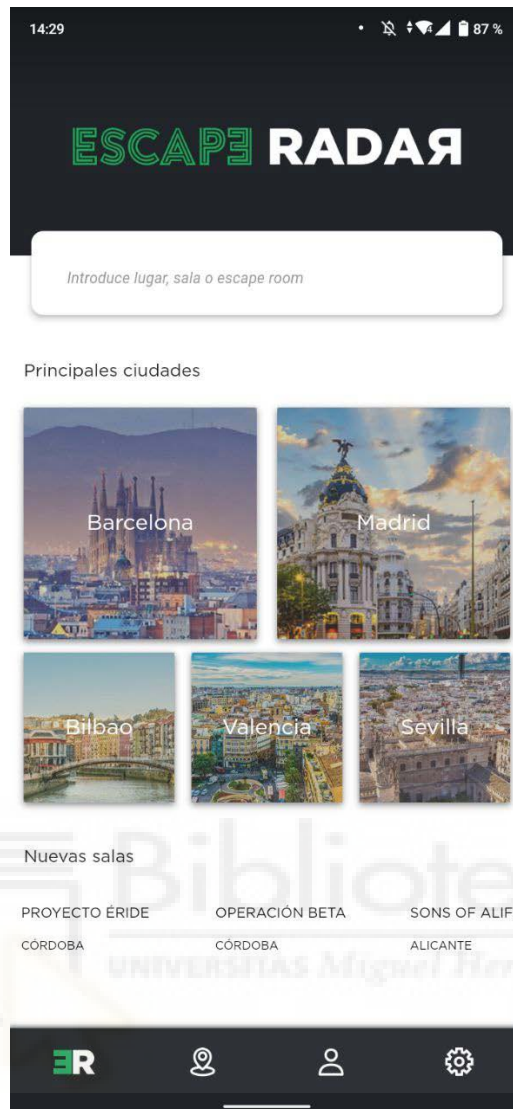
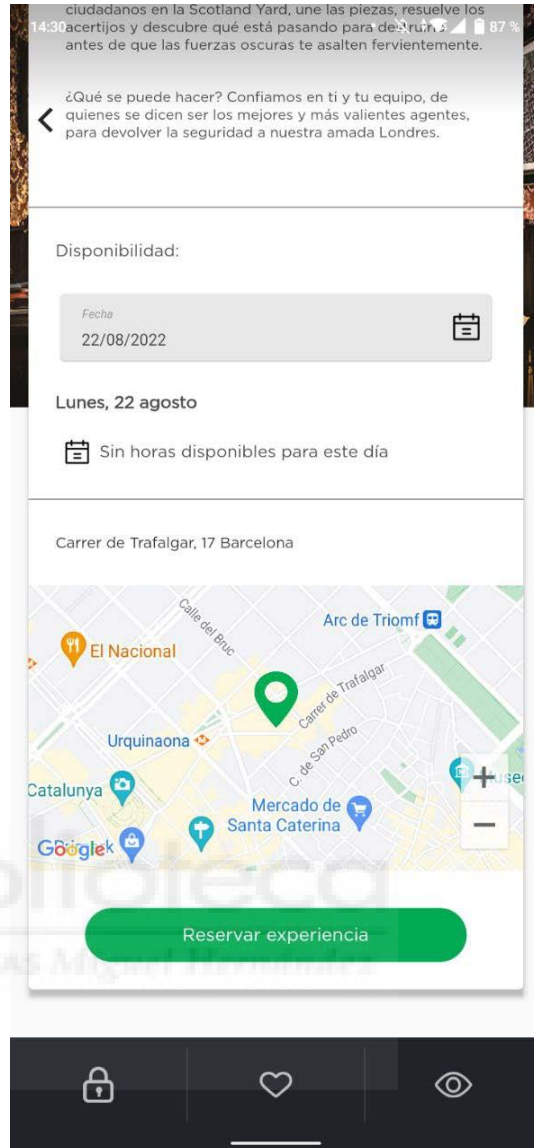


Imagen 16. Página principal de Escape Radar.

Cuando pulsas en una escaperoom te muestra el nombre, la ciudad, nº de jugadores, dificultad, tiempo y precio, una descripción, la disponibilidad utilizando un widget de calendario que eligiendo un día te muestra justo debajo las horas que hay disponibles, una integración de GMaps señalando la escaperoom y un botón para reservar que abre la página de la sala.



Imágenes 17 y 18. Detalles de escaperoom de Escape Radar.

Tiene 4 submenús siendo uno el inicio, otro una integración de Google Maps, un submenú para ver tu perfil y otro submenú de configuración.

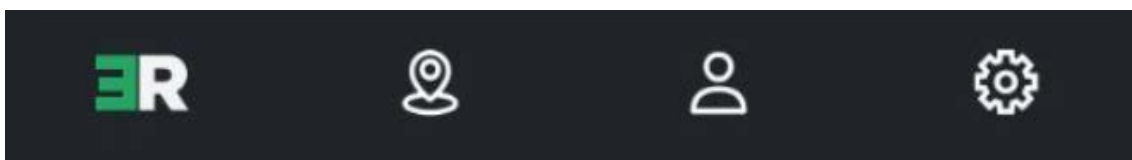


Imagen 19. Submenú de Escape Radar.

De la integración de Google Maps cabe destacar que señala en todo momento las salas de escape que hay registradas y al pulsarla te muestra la página con los datos de la sala.

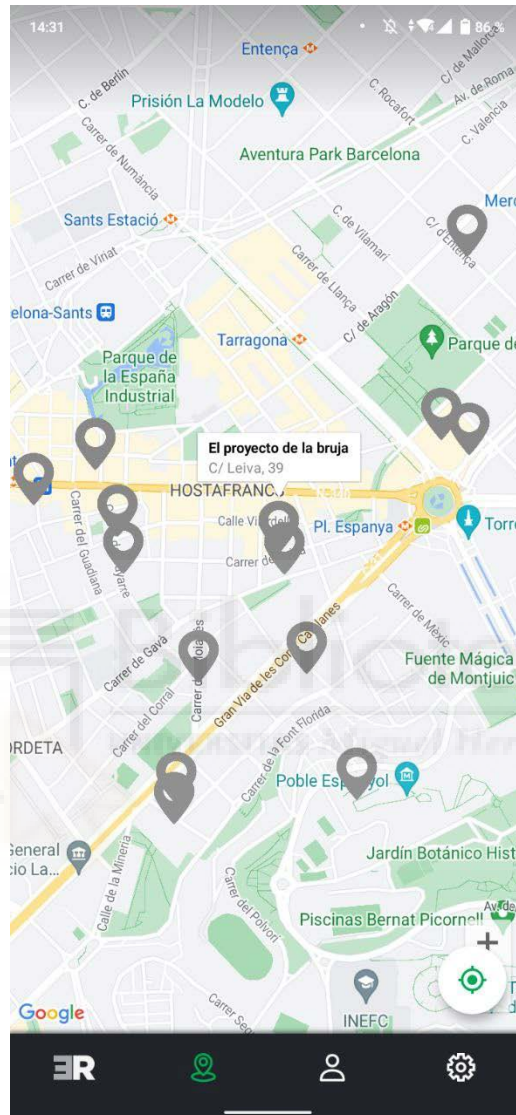


Imagen 20. Integración de GMaps en Escape Radar.

Del submenú del perfil se espera lo típico, un formulario de inicio de sesión y botones de inicio con Google y Facebook. Con la sesión iniciada muestra algo más que la aplicación anterior, ya que aquí se pueden conseguir logros y muestra el número de salas que has jugado y el porcentaje de éxito al resolverlas, justo debajo muestra tres listas una de las salas que se han jugado, otra de las favoritas y una lista en la que se encuentran las salas que se han ocultado.

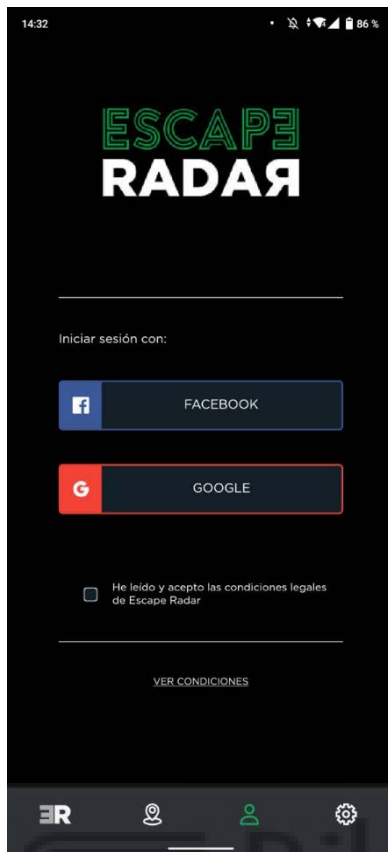


Imagen 21. Inicio de sesión en Escape Radar.

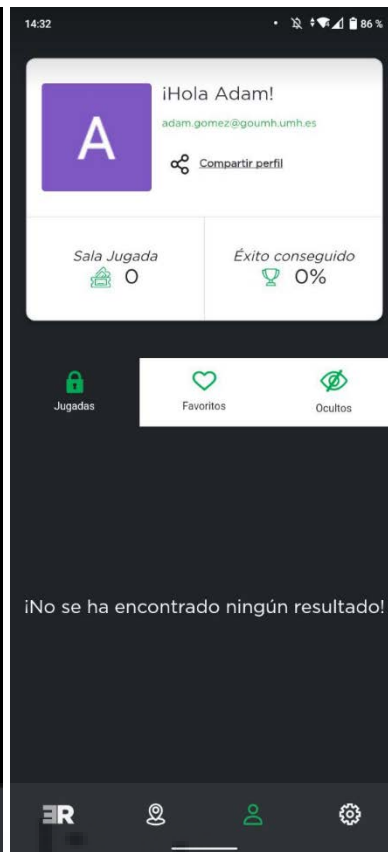


Imagen 22. Lista de escaperooms jugadas en Escape Radar.

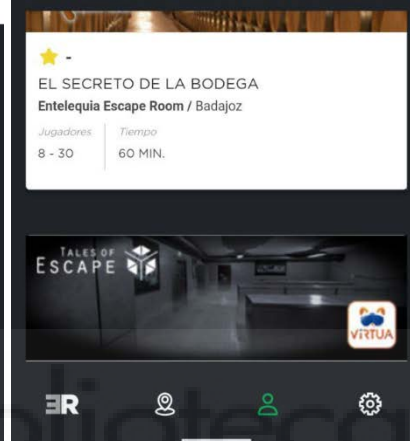
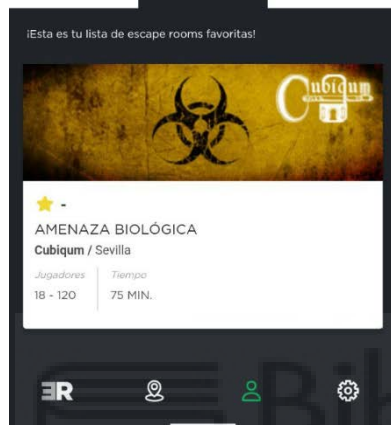
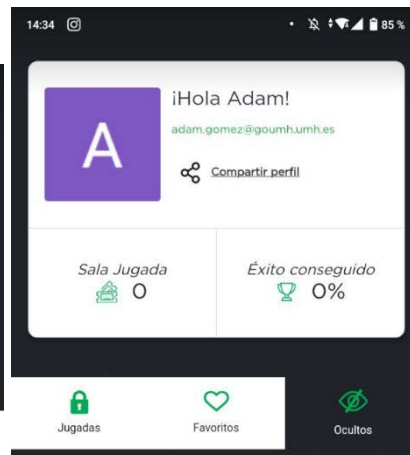
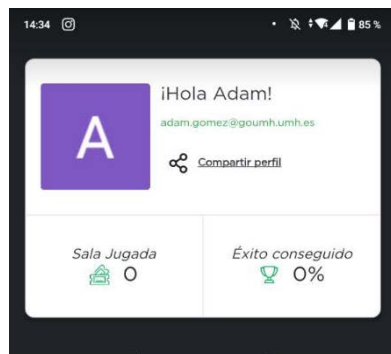


Imagen 23. Lista de escaperooms favoritas en Escape Radar.

Imagen 24. Lista de escaperooms ocultas en Escape Radar.

Por último, el menú de configuración tiene las típicas opciones con información sobre la app, las condiciones legales, información de contacto y el botón de cerrar sesión.

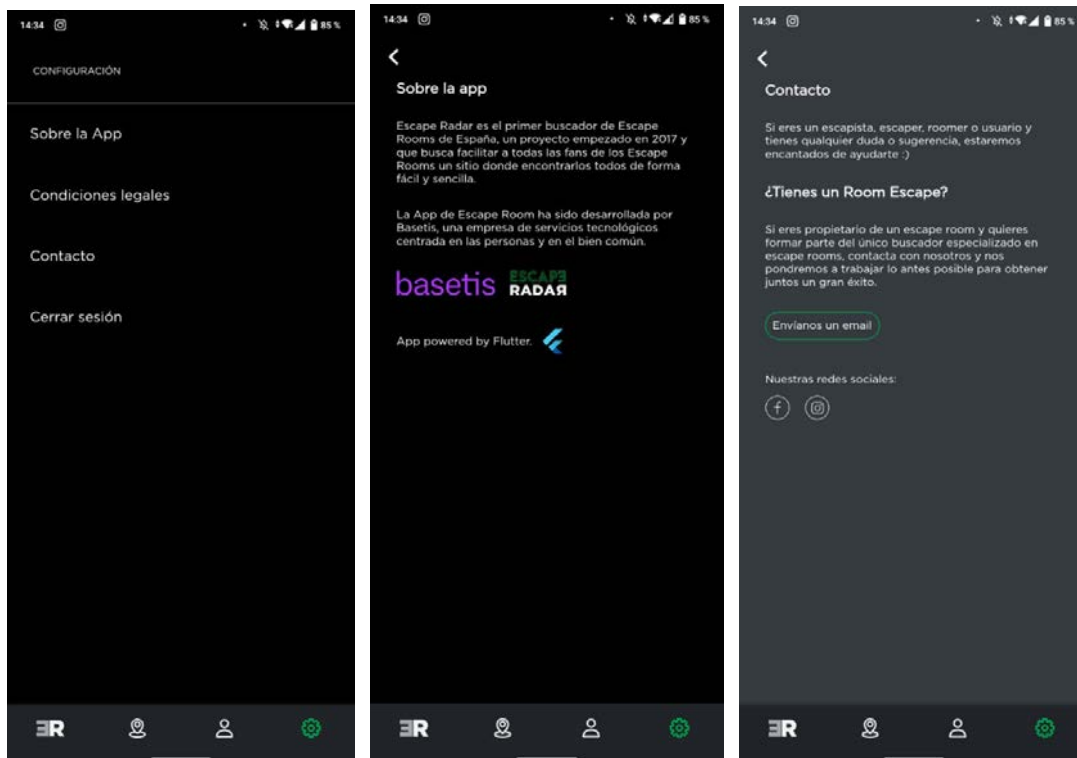


Imagen 25. Pantalla de configuración en Escape Radar.

Imagen 26. Pantalla de información sobre Escape Radar.

Imagen 27. Pantalla de contacto en Escape Radar.

Padlock

Por último, se ha testado Padlock, una app visualmente menos cuidada que las anteriores, pero con la misma funcionalidad. Al iniciarla muestra una imagen de un candado con un menú hamburguesa en la esquina superior izquierda, tres botones en la parte inferior que llevan a una página de contacto, al Instagram y a la página web, y en la esquina superior derecha un botón que lleva al perfil del usuario.



Imagen 28. Pantalla de inicio de Padlock.

Al abrir el menú hamburguesa muestra todas estas opciones:

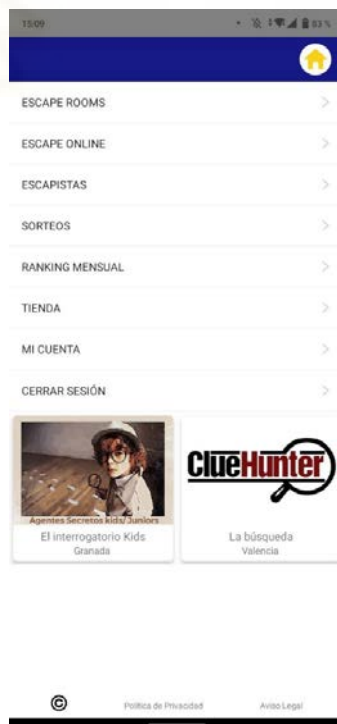


Imagen 29. Opciones en el menú hamburguesa de Padlock.

En Escaperooms muestra una lista de salas de escape recomendadas y un buscador con un filtro para especificar la temática, la ubicación, ordenar por precio e incluir salas ocultas.

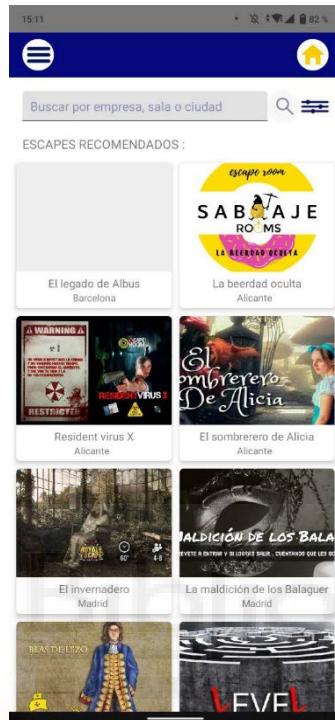


Imagen 30. Lista de escaperooms de Padlock.

Al seleccionar una escaperoom, muestra toda la información relativa, una descripción, darle una nota de hasta cinco estrellas, añadirla a una lista de completadas, ocultarla, y la opción de marcarla como favorita. También unos botones que dirigen a la página web de la sala, al Instagram, a la ubicación (abre GMaps señalando la localización exacta), y un botón de reserva que abre la página web en la app.

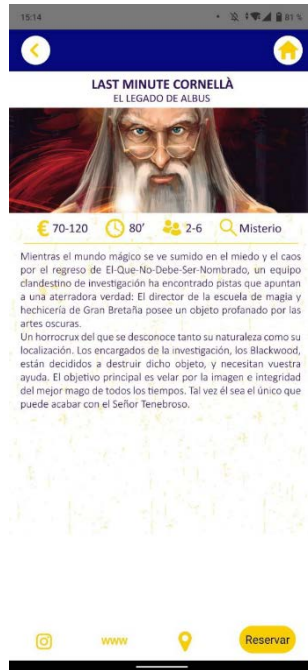


Imagen 31. Pantalla de detalles de escape room de Padlock.

La segunda opción, Escape Online, abre una lista de escape rooms que se pueden jugar desde casa haciendo puzzles y minijuegos desde la página web, algunas son de pago y otras son totalmente gratuitas y permite filtrarlas. Al escoger una muestra el precio (si tiene), el número de jugadores, la temática, una descripción, y enlaces al Instagram y a la página web donde hacer la escape room.

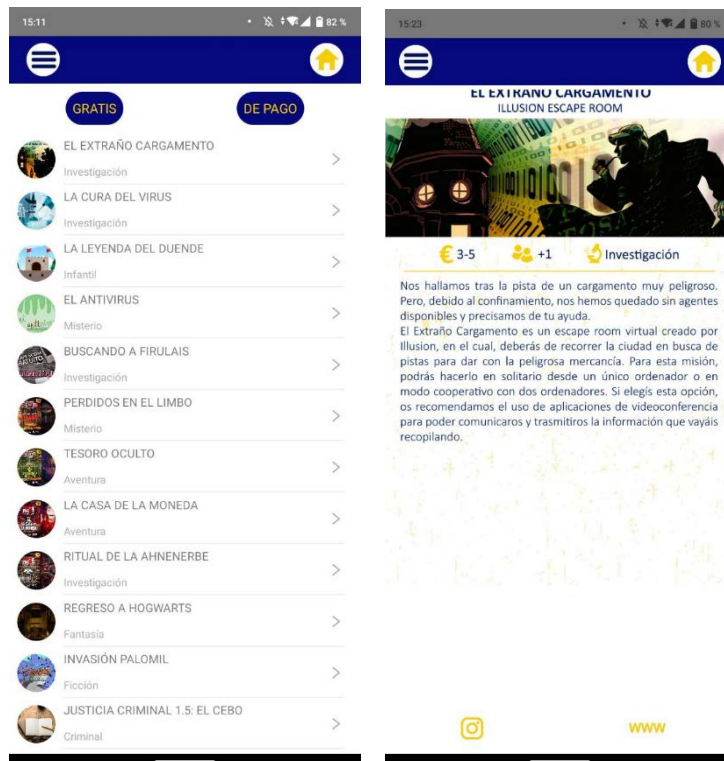


Imagen 32. Lista de escaperooms online de Padlock.

Imagen 33. Pantalla de detalles de escaperoom online de Padlock.

La siguiente opción, Escapistas, muestra una lista de jugadores registrados en Padlock, con un buscador para encontrar jugadores por su alias. Al seleccionar uno muestra la ciudad del jugador, una bio puesta por el jugador, el número de escaperooms que han solucionado, unas medallas que indican logros realizados y enlaces al Instagram del jugador y a la página web (si no hay ninguna no lleva a ningún sitio).



Imagen 34. Lista de usuarios de Padlock.



Imagen 35. Perfil de un usuario de Padlock.

La cuarta opción, Sorteos, muestra unos enlaces a sorteos realizados por la app, al pulsar en uno muestra la información sobre el sorteo y un enlace a la publicación del sorteo en Instagram.



Imagen 36. Lista de eventos de Padlock.

Imagen 37. Pantalla de detalles de un evento de Padlock.

A continuación, Ranking mensual, muestra una lista de las salas de escape más votadas en el último mes, hay un botón para votar tus salas favoritas poniendo el nombre y la ciudad.

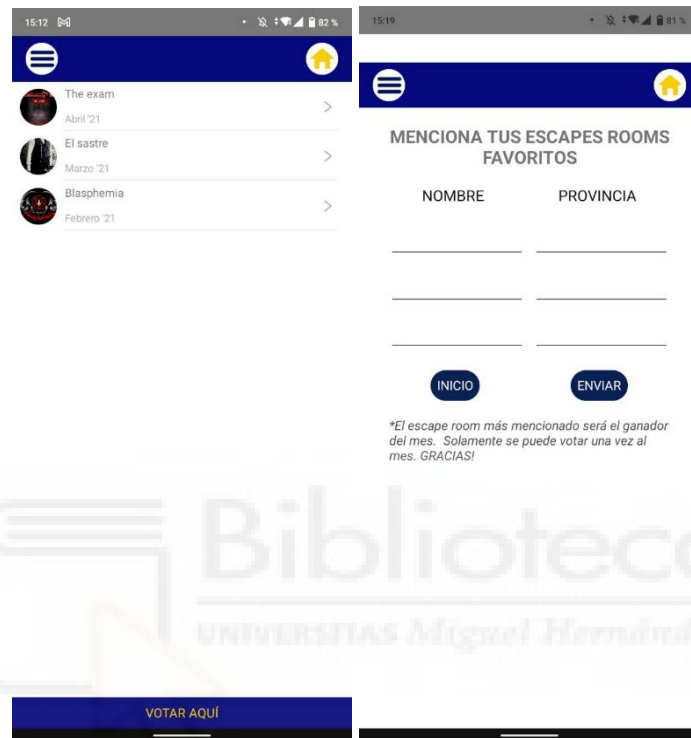


Imagen 38. Lista de escaperooms más votadas de Padlock.

Imagen 39. Formulario de voto de escaperoom de Padlock.

La sexta opción, Tienda, abre una lista de productos como juegos de mesa, ropa y libros. Al seleccionar alguno de los productos te redirige al producto en Amazon o en cualquier otra plataforma de comercio electrónico.

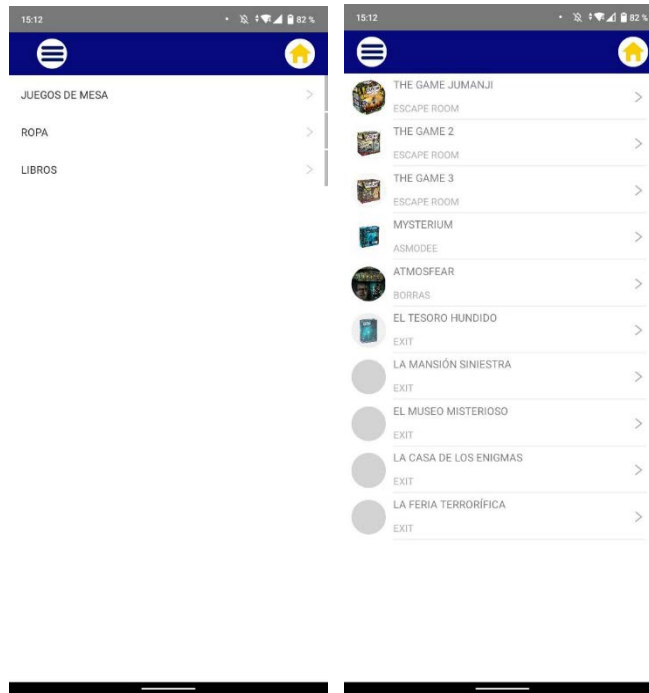


Imagen 40. Lista de productos en la tienda de Padlock.

Imagen 41. Lista de juegos de mesa ofrecidos por Padlock.

Por último se encuentran las opciones de Mi cuenta y Cerrar sesión, la primera redirige al perfil del usuario igual que el botón en la pantalla de inicio, y el segundo te abre una página con un botón para confirmar el cierre de la sesión.

Debajo de todas las opciones hay un par de escaperooms destacadas y al final enlaces a las políticas de privacidad y aviso legal que llevan a la página web de la app.

Para finalizar en la página del usuario muestra el nombre y unos números que no explican pero se supone que son el número de salas a las que se le ha dado una nota, salas completadas y, salas favoritas ya que al pulsar en cada una de las opciones muestra una lista de salas a las que se le han marcado de esas maneras.

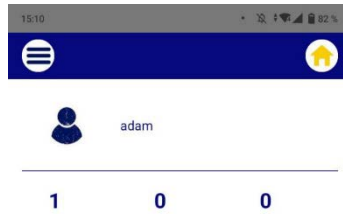


Imagen 42. Pantalla del perfil de usuario de Padlock.

Para concluir, las aplicaciones que se han revisado hacen un buen trabajo, unas mejor que otras, muchas de las funcionalidades que se han visto van a estar implementadas en la app de este trabajo, aunque se añadirán ideas propias que podrían aportar positivamente a la experiencia de usuario.

CAPÍTULO 3. OBJETIVOS DEL PROYECTO

3.1 Objetivos principales

El objetivo principal de este trabajo de fin de grado es conseguir una aplicación útil y funcional. Y ofrecer a los usuarios finales una plataforma en la que puedan consultar, descubrir y disfrutar de una variedad de salas de escape que se adapten a sus gustos y preferencias. Para ello se va a hacer uso de tecnologías que no se estudian en la carrera como Flutter, Dart, Firebase, y Github entre otras.

También demostrar en un proyecto de este tamaño los conocimientos obtenidos en el grado. Así como mostrar nuevas tecnologías que son utilizadas en multitud de proyectos y no se ha tenido la oportunidad de aprender en el curso.

3.2 Objetivos secundarios

Otro objetivo también es crear una comunidad de usuarios que compartan el gusto por las escaperooms y que sirva de nexo para incentivar la participación de cada vez más gente en esta actividad de ocio.

3.3 Conocimientos previos

Antes de comenzar a trabajar en este proyecto conocía el concepto de “Escaperooms”, tenía entendido que eran un desafío que hacía uso del trabajo en equipo, la astucia y el ingenio para resolver un puzzle.

Para llevar a cabo el proyecto se va a hacer uso de metodologías de planificación aprendidas en la asignatura de Ingeniería del Software. Otra competencia aprendida en la carrera que ha sido útil para la realización del proyecto ha sido la programación orientada a objetos y las estructuras de

datos, que ha ayudado en gran medida para organizar y modularizar todos los componentes del proyecto para tener una mayor claridad del código.

Y por último, el conocimiento aprendido en las asignaturas de bases de datos, que si bien no se ha utilizado un modelo relacional (NoSQL) para el backend del proyecto, si que ha ayudado a entender la naturaleza de los datos y como referenciarlos para mantener la integridad.

3.4 Estudio y puesta en marcha.

Para llevar a cabo este proyecto se han tenido que realizar multitud de configuraciones e instalaciones.

Para empezar la herramienta de la que se ha hecho uso para escribir el código y la compilación es Visual Studio Code. Es una herramienta de la que se habla más en profundidad en el siguiente apartado, en la que se ha tenido que instalar algunas extensiones para poder desarrollar la aplicación.

La extensión Flutter es una extensión que añade toda la infraestructura de se necesita para poder editar, ejecutar y hacer debug. Y automáticamente instala la extensión de Dart que es la que se encarga de compilar los proyectos y permite hacer uso de shortcuts para realizar acciones como crear un proyecto o actualizar dependencias.

Y otra herramienta ha sido el Virtual Device Manager (VDM) de Android Studio, que ha permitido ver el desarrollo de la aplicación en vivo. Haciendo uso de las extensiones de Visual Studio Code ha podido instalar la apk en el VDM y pudiendo así ver los cambios realizados con la función “Hot reload” de Flutter.

CAPÍTULO 4. HIPÓTESIS DE TRABAJO

4.1 Elección de las herramientas

4.1.1 Editor de código



Imagen 43. Logo de Visual Studio Code.

En la fase de análisis hice una valoración entre los muchos editores de código que hay disponibles como Notepad++, Sublime y Visual Studio Code entre otros.

Finalmente la herramienta que se ha escogido para escribir el código ha sido Visual Studio Code [7]. Es un editor de código que ofrece multitud de opciones debido a su gran repositorio de extensiones que facilitan el desarrollo con shortcuts, autocompletado de sintaxis y visualmente amigable entre otras muchas ventajas.

Es una herramienta desarrollada por Microsoft y naturalmente está constantemente en evolución con actualizaciones casi semanales, una gran comunidad de usuarios y una documentación completa a la que suelen hacer referencia la mayoría de frameworks y lenguajes, que recomiendan su uso para el desarrollo.

4.1.2 Lenguaje de programación

4.1.2.1 Flutter



Imagen 44. Logo de Flutter.

La aplicación ha sido desarrollada en Flutter[9]. Flutter es un SDK de código fuente abierto de desarrollo de aplicaciones móviles y lanzado por Google en mayo de 2017. Permite crear aplicaciones móviles nativas haciendo uso de solamente una base de código. Esto significa que se pueden crear aplicaciones tanto para iOS como para Android usando un lenguaje de programación.

Flutter consta fundamentalmente de dos partes[10]:

- Un SDK (Software Development Kit): Es una colección de herramientas que ayuda en el desarrollo de aplicaciones, como por ejemplo, herramientas para compilar tu código en código máquina nativo (código para iOS y Android).
- Un Framework (Biblioteca de UI basada en widgets): Se trata de un conjunto de módulos de interfaz de usuario que se pueden reutilizar (botones, entradas de texto, deslizadores, etc.) que puedes personalizar para tus propias necesidades.

4.1.2.2 Dart



Imagen 45. Logo de Dart.

El lenguaje de programación que se usa en Flutter es Dart[10]. Originalmente llamado Dash, Dart es un lenguaje que fue creado por Google en octubre de 2011 y que se ha ido mejorando desde su creación. Dart es un lenguaje que se centra en el desarrollo del front-end, y se usa tanto para crear aplicaciones móviles como páginas web.

El lenguaje Dart es tipado de tipo seguro, utiliza la comprobación de tipo estático para asegurar que el valor de una variable siempre coincide con el tipo estático de la variable. También conocido como tipado seguro. Aunque los tipos son obligatorios, las anotaciones de tipo son opcionales debido a la inferencia de tipos. El sistema de tipado de Dart es también flexible, permitiendo el uso de un tipo dinámico combinado con comprobaciones en tiempo de ejecución, cosa que ha sido útil durante el desarrollo para experimentar o para el código que necesita ser especialmente dinámico.

Dart ofrece null safety sólida, lo que significa que los valores no pueden ser nulos a menos que usted diga que pueden serlo. Con la seguridad null safety, Dart puede protegerse de las excepciones de nulos en tiempo de ejecución a través del análisis de código estático. A diferencia de muchos otros lenguajes con null safety, cuando Dart determina que una variable es no anulable, esa variable es siempre no anulable. Si se inspecciona el código en ejecución en el depurador, se verá que la no anulabilidad se mantiene en tiempo de ejecución (de ahí la null safety).

4.1.3 Servidor y Base de datos

4.1.3.1 Firebase



Imagen 46. Logo de Firebase.

Para el back-end de la aplicación se ha utilizado Firebase[11]. Firebase es una plataforma creada por Envolv, una empresa establecida en 2011 que ofrecía a los desarrolladores una API para facilitar la integración de un chat en línea en sitios web.

Al poco tiempo, cuando sus fundadores, Andrew Lee y James Tamplin, descubrieron que los desarrolladores encontraron otra utilidad a esta API, haciendo uso de ella para la sincronización de datos de aplicaciones en tiempo real. Entonces, decidieron diferenciar esas dos funcionalidades siendo una de ellas Firebase, que fue lanzada un año más tarde[12].

Así nació Firebase Realtime Database, y debido a la gran financiación con la que contó la empresa y al gran número de usuarios que hacían uso de éste, lanzaron en 2014 otros dos más: Firebase Authentication y Firebase Hosting, asegurándose un liderazgo como back-end móvil como servicio (MBaaS).

Al ver que tuvo tanto éxito entre los usuarios, Google en 2014, adquirió Firebase y dio lugar a un crecimiento exponencial de la plataforma.

El funcionamiento de la plataforma se puede dividir en tres grupos según su objetivo:

- **Compilación:** permite iniciar el back-end sin administrar servidores y escalar conforme la aplicación va creciendo.
- **Lanzamiento y supervisión:** permite gestionar el lanzamiento de la aplicación y supervisar el buen funcionamiento de la misma.
- **Participación:** se puede aumentar la participación de los usuarios en la app, comprendiendo cómo estos la utilizan, personalizarla para diferentes segmentos de usuarios, ejecutar experimentos para probar ideas de nuevas funcionalidades, etc.

Algunas de las ventajas de la plataforma son:

- **Es multiplataforma**, ya que se encuentra disponible tanto para plataformas móviles como para la web. Además, tecnologías en auge comienzan a tener también soporte como Node, **Flutter**, Unity...
- **Ofrece un gran poder de crecimiento** al tener la capacidad de gestionar con facilidad a todos los usuarios de las aplicaciones.
- Se trata de una **herramienta ágil, sencilla y que permite olvidarse de la infraestructura**. Firebase proporciona APIs integradas en un sólo SDK, que permite centrarse en resolver los problemas de nuestros clientes en lugar de dedicar tiempo a construir una infraestructura compleja.
- **Inicio gratuito y escalable:** una de las características más llamativas de Firebase es que la mayoría de sus módulos son gratuitos. Algunos tienen ciertas limitaciones, que se pueden desbloquear con un plan de pagos, pero que solo serán necesarios en el caso de que la aplicación lo necesite.
- **Crecimiento continuo:** constantemente se van creando nuevos módulos y éstos cada vez son compatibles con nuevas tecnologías.

Por otra parte, existen aspectos negativos de Firebase que se han de tener en cuenta:

- **No es open source.**

- **No dispone de bases de datos relacionales**, Firestore y Realtime Database tienen una estructura NoSQL y puede generar complicaciones al realizar operaciones complejas.
- **No todos los servicios son gratuitos**, hay servicios que comienzan siendo gratuitos y tan pronto el usuario exceda el límite establecido, se efectuarán los cargos bajo el plan de pago.
- **Se ejecuta sólo en Google Cloud**, al ser parte de Google no hay ninguna opción de ejecutar Firebase en otros proveedores en la nube como AWS o Azure.
- **Requiere tiempo de aprendizaje**, aunque Firebase proporciona gran documentación sobre cómo integrar cada módulo y ejemplos de cada uno de ellos, hay que dedicar tiempo a conocer cómo integrarlos en nuestra aplicación.

4.1.3.2 Cloud Firestore

Cloud Firestore[13], es la base de datos NoSQL de Firebase, flexible, escalable y en la nube con sirve para almacenar y sincronizar los datos en tiempo real para el desarrollo en servidores, dispositivos móviles y la web. Los datos se estructuran en colecciones.

Una de las características más especiales que tiene es la “asistencia sin conexión”, ya que hace uso del almacenamiento en caché de los datos que usa la app, y permite que se pueda leer, escribir y consultar información sin que el dispositivo se encuentre conectado a internet. En el momento que se recupera la conexión, Cloud Firestore sincroniza todos los cambios locales en la nube.

Cloud Firestore está disponible en los SDK nativos de Android, iOS, Web, Node.js, Java, **Flutter**, Python, Unity, C++ y Go, además de las API de REST y RPC.

4.2 Equipo Físico

Para el desarrollo de este trabajo de fin de grado se ha usado un ordenador de sobremesa, teclado, monitor, y ratón. A continuación, se detallan las especificaciones de dichos componentes:

- Ordenador de sobremesa.
 - Marca (Placa base): MSI.
 - Modelo: MSI B550M A-PRO AM4.
 - Procesador: AMD Ryzen 5 5600X 6-Core Processor (12 CPUs), ~ 3.7GHz
 - Memoria RAM: 2 stick de 8GB de tipo DDR4 con una velocidad de 3600MHz.
 - Disco duro de 2 TB más un SSD de 512 GB.
 - Gráfica: NVIDIA® GeForce® MSI RTX 3070 Ti VENTUS 3X 8G OC
 - Sistema operativo: Windows 10 Pro
- Teclado.
 - Marca: Razer.
 - Modelo: Razer Cynosa Chroma.
- Monitor.
 - Marca: Samsung.
 - Modelo: Samsung C32JG56.
- Ratón.
 - Marca: Logitech.
 - Modelo: Logitech G402.

CAPÍTULO 5. METODOLOGÍA Y RESULTADOS

5.1 Introducción al proceso de ingeniería del software

Al ser un proyecto de desarrollo de software, se debe seleccionar un método a seguir entre todos los existentes en la ingeniería del software, como el modelo en cascada, el modelo en espiral, el modelo en V o el modelo evolutivo.

Los métodos en cascada y en V son métodos lineales, mientras que el modelo en espiral y el evolutivo son iterativos. Al no tener planes, en principio, de evolucionar el proyecto tras la presentación, se ha decidido seguir el método en cascada. Otro motivo es que es un proyecto realizado solo por una persona, si hubiera trabajado con un equipo lo más probable es que hubiera seleccionado una metodología de trabajo ágil como Scrum.

5.2 Planificación

5.2.1 Ciclo de vida

En la ingeniería de software el ciclo de vida se ha establecido como uno de los cimientos para la correcta ejecución de cualquier proyecto de diseño del software. Con el fin de garantizar que se cumplen todos los requisitos y de que se obtenga un software de calidad.

Se podría definir como una sucesión de pasos por los que el producto va evolucionando hasta convertirse en software de alta calidad.

Como se ha mencionado en el apartado de introducción, se ha escogido el ciclo de vida en cascada[14], ya que es un proyecto de pequeña escala y se ha considerado, fuera de los métodos ágiles, el ciclo natural de un proyecto.

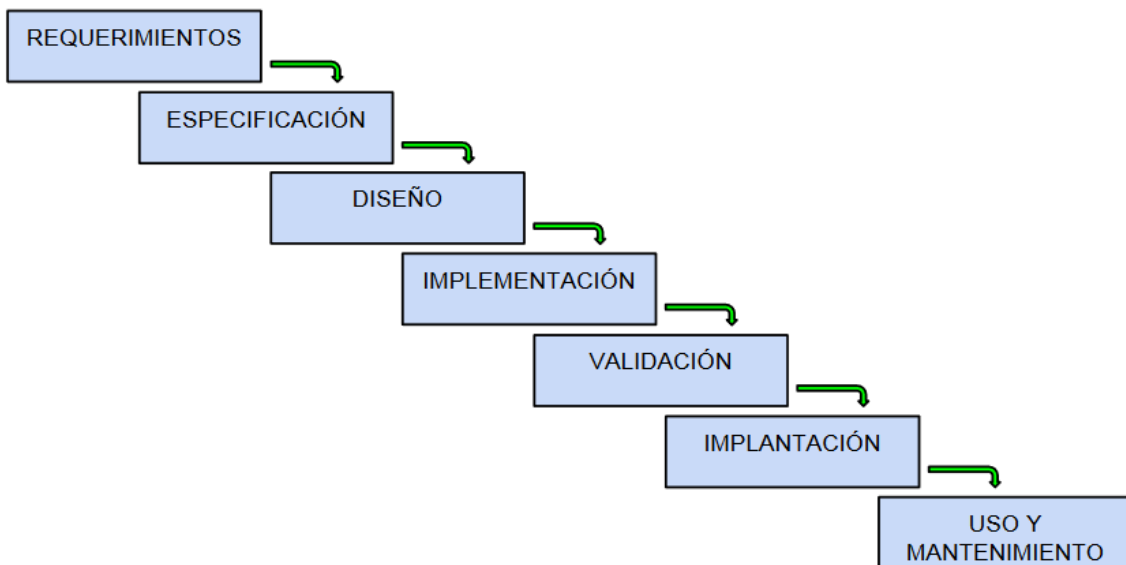


Imagen 47. Método en cascada.

Se compone de 7 pasos que se pueden agrupar en 3 grupos:

- **Análisis:** agrupa los pasos de requerimientos y especificación. En este grupo se establecen las metas del proyecto, la realización de estudios de viabilidad, y la asignación de los recursos al resto de etapas del proyecto, entre otras.
- **Desarrollo:** engloba los 3 pasos centrales del ciclo, diseño, implementación y validación. Aquí es donde en orden se realiza el diseño, donde se estudian las posibles opciones de implementación para el software que se ha de construir. Una vez establecido un diseño comienza la etapa de implementación, que consiste en el desarrollo del código que al final será el producto en sí. Terminada la fase de implementación comienza la fase de validación, que consiste en realizar que el producto funcione y cumpla todos los requisitos establecidos.
- **Implantación y mantenimiento:** en este punto se encuentra la fase de implantación, y la de uso y mantenimiento. En el paso de la implantación se pone en producción el producto creado y validado en los pasos anteriores. Por último, la fase de uso y mantenimiento consiste en la implementación de mejoras o correjimiento de errores en caso de que los hubiera.

5.3 Modelado UML

5.3.1 Descripción de actores

Para empezar, se ha realizado una descripción de uno de los elementos que componen la parte de ingeniería del software, los actores:

Actor	Usuario no registrado
Descripción	Un usuario no registrado es cualquier tipo de persona que entre a la aplicación y no esté registrada, no puede hacer ningún tipo de actividad como navegar por la pantalla principal de la aplicación o hacer reservas.
Casos de uso	C.U. 1.1, C.U. 1.2

Tabla 1. Descripción del rol de “Usuario no registrado”

Actor	Usuario con cuenta
Descripción	Un usuario con cuenta es uno que ya tiene una cuenta registrada en la aplicación y tiene la intención de realizar alguno de los casos de uso, tales como hacer una reserva o simplemente consultar las escaperooms disponibles
Casos de uso	C.U. 2.1, C.U. 2.2, C.U. 2.3, C.U. 2.4, C.U. 2.5

Tabla 2. Descripción del rol de “Usuario con cuenta”

Actor	Administrador de escaperoom
Descripción	Un usuario que se ha registrado como administrador, y tiene como objetivo añadir una actividad de escaperoom o consultar las reservas a cualquiera de las actividades que haya añadido.

Casos de uso	C.U. 3.1, C.U. 3.2, C.U. 3.3, C.U. 3.4, C.U. 3.5
--------------	--

Tabla 3. Descripción del rol de “Administrador de escaperoom”

Actor	Administrador
Descripción	Es el tipo de usuario que tiene un control total sobre los aspectos administrativos de la aplicación, que trabaja desde las herramientas de Firebase. Podrá realizar operaciones como la gestión de usuarios o comprobar la integridad de los datos en Cloud Firestore.
Casos de uso	C.U. 4.1, C.U. 4.2, C.U. 4.3

Tabla 4. Descripción del rol de “Administrador”

5.3.2 Diagrama de casos de uso

Los diagramas de este apartado exponen todos los actores y todas las acciones que pueden llevar a cabo en la aplicación:

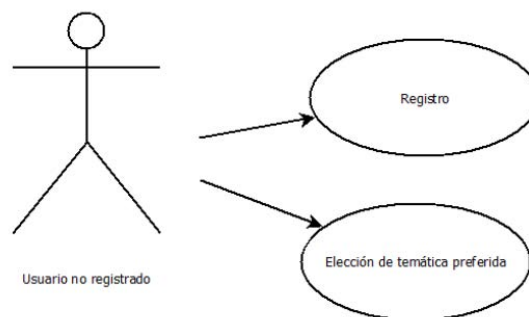


Imagen 48. Diagrama de usuario no registrado

C.U. 1.1	Registrarse en la aplicación.
----------	-------------------------------

Actor/es	Usuario no registrado.
Descripción	Un usuario se quiere registrar en la aplicación para hacer uso de los servicios que ofrece la aplicación.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa.
Secuencia normal	P1: El usuario sin cuenta ingresa en la aplicación. P2: Ejecuta o se salta el C.U. 1.2. P3: Introduce los datos requeridos por la aplicación. P4: Una vez introducidos le da al botón de "Registrarse".
Poscondición	PostC1: El usuario pasa a ser un usuario registrado a un usuario con cuenta.
Excepciones	ExC1: El usuario ha introducido un dato erróneo en cualquiera de los campos, a lo que saltará un error notificándole de la errata tras presionar el botón de "Registrarse".

Tabla 5. Caso de uso 1.1

C.U. 1.2	Elección de temáticas preferidas.
Actor/es	Usuario no registrado.
Descripción	Un usuario al abrir la aplicación por primera vez se encuentra un carrusel de preguntas sobre sus gustos en las escaperooms.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario nunca haya abierto la aplicación previamente.
Secuencia normal	P1: El usuario ingresa en la aplicación por primera vez. P2: Lee las preguntas y elige las opciones acordes con sus gustos.
Poscondición	PostC1: Las preferencias del usuario se han guardado y se tendrán en cuenta para las recomendaciones.

Excepciones	ExC1: El usuario presiona el botón de “Saltar” y avanza directamente al C.U. 1.1.
-------------	---

Tabla 6. Caso de uso 1.2

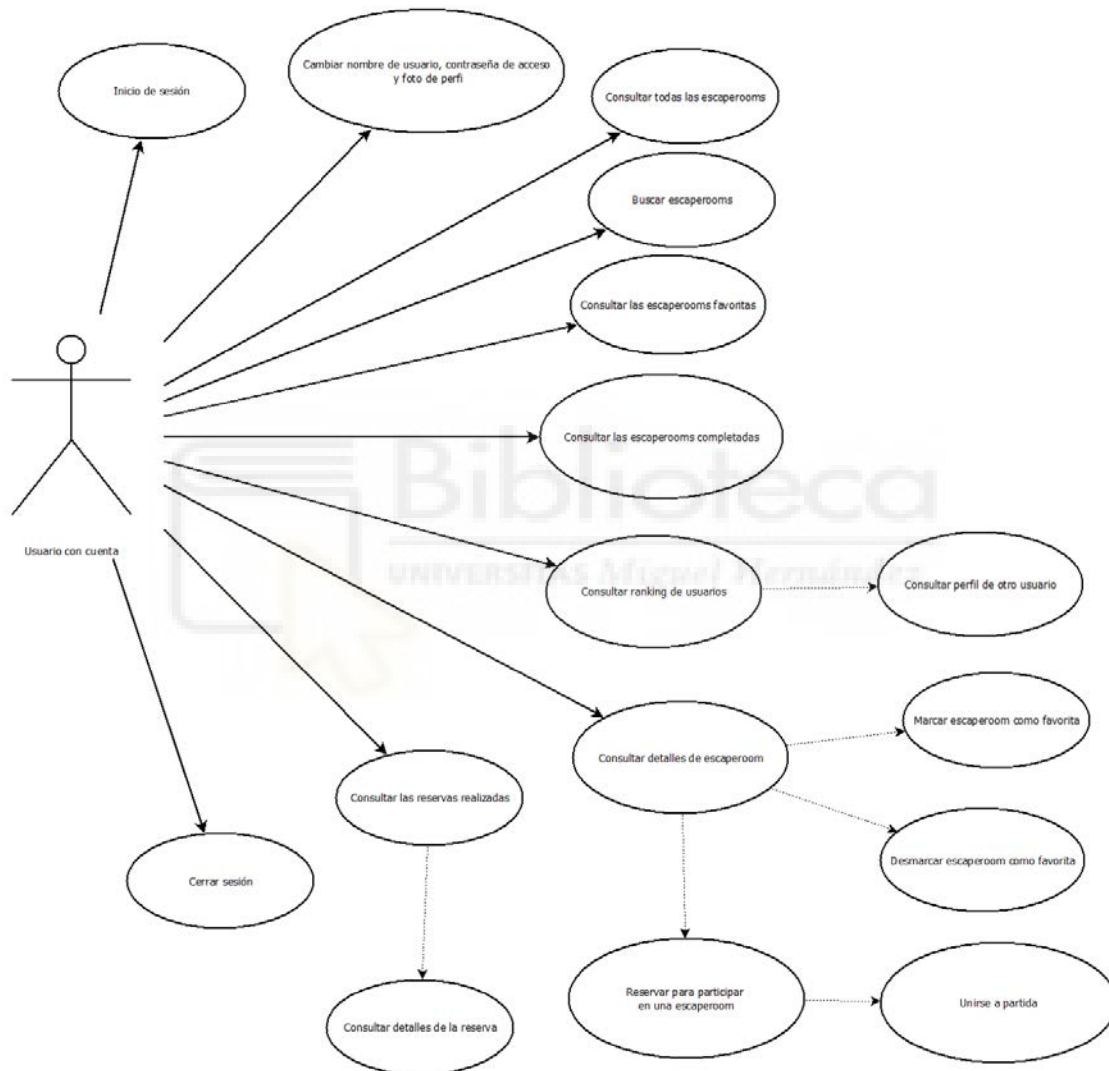


Imagen 49. Diagrama de usuario con cuenta

C.U. 2.1	Inicio de sesión.
Actor/es	Usuario con cuenta.

Descripción	Un usuario que ya se ha registrado quiere acceder a la aplicación para hacer uso de sus funcionalidades.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario tenga una cuenta en la aplicación.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: En la pantalla de inicio de sesión introduce los datos con los que se ha registrado. P4: Una vez introducidos le da al botón de “Iniciar sesión”.
Poscondición	PostC1: El usuario ha iniciado sesión con éxito.
Excepciones	ExC1: El usuario ha introducido un dato erróneo en cualquiera de los campos, a lo que saltará un error notificándole de la errata tras presionar el botón de “Iniciar sesión”.

Tabla 7. Caso de uso 2.1

C.U. 2.2	Buscar escaperooms.
Actor/es	Usuario con cuenta.
Descripción	Un usuario al abrir la aplicación quiere buscar una escaperoom para consultar sus detalles.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Pulsa el buscador en la página principal de la aplicación. P3: Se despliega el teclado y escribe el nombre, la ciudad o la temática de la escaperoom que busca.
Poscondición	PostC1: Aparece la escaperoom que busca el usuario en pantalla.
Excepciones	ExC1: La escaperoom que busca no existe.

	ExC2: Ha introducido los parámetros erróneamente y por lo tanto no muestra el resultado deseado.
--	--

Tabla 8. Caso de uso 2.2

C.U. 2.3	Consultar el perfil de otro usuario.
Actor/es	Usuario con cuenta.
Descripción	Un usuario quiere consultar el perfil de otro usuario de la plataforma.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Arrastra hacia la derecha desde la parte izquierda de la pantalla. P3: Pulsa la opción de "Ranking" P4: Pulsar en uno de los usuarios para ver su perfil.
Poscondición	PostC1: El usuario ha visto el perfil de otro usuario.
Excepciones	No tiene excepciones.

Tabla 9. Caso de uso 2.3

C.U. 2.4	Realizar una reserva.
Actor/es	Usuario con cuenta.
Descripción	Un usuario quiere hacer una reserva en una escaperoom que le ha llamado la atención.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Localiza la escaperoom que quiere hacer la reserva mediante la búsqueda por nombre, dándole al botón de "Ver más" en el carrusel de la página principal, o si se ha marcado como favorita a la lista de escape rooms favoritas.

	<p>P3: Una vez localizada pulsar en la foto o el nombre de la escaperoom.</p> <p>P4: En la página de detalles de la escaperoom pulsar el botón de la esquina inferior izquierda “Reservar”.</p> <p>P5: Rellenar los datos requeridos para la reserva.</p> <p>P6: Pulsa el botón de “Enviar”.</p>
Poscondición	PostC1: El usuario ha realizado una reserva exitosamente.
Excepciones	<p>ExC1: La escaperoom que quiere reservar no existe.</p> <p>ExC2: Ha introducido los parámetros erróneamente y por lo tanto no muestra el resultado deseado.</p> <p>ExC3: Si el día que se ha elegido para reservar no está disponible, no se podrá reservar y deberá elegir otro día.</p> <p>ExC4: No tiene el número mínimo de jugadores así que debería conseguir ese mínimo o unirse a una partida.</p>

Tabla 10. Caso de uso 2.4

C.U. 2.5	Consultar una reserva.
Actor/es	Usuario con cuenta.
Descripción	Un usuario quiere revisar algún dato sobre una reserva en una escaperoom.
Dependencias	C.U. 2.4.
Precondición	<p>PreC1: Que la aplicación esté operativa.</p> <p>PreC2: Que el usuario haya iniciado sesión.</p>
Secuencia normal	<p>P1: El usuario ingresa en la aplicación.</p> <p>P2: Arrastra hacia la derecha desde la parte izquierda de la pantalla.</p> <p>P3: Pulsa la opción de “Mis reservas”.</p> <p>P4: Pulsar en la reserva que el usuario quiera revisar .</p>
Poscondición	PostC1: El usuario ha visualizado la información relativa a la reserva.
Excepciones	No tiene excepciones.

Tabla 11. Caso de uso 2.5

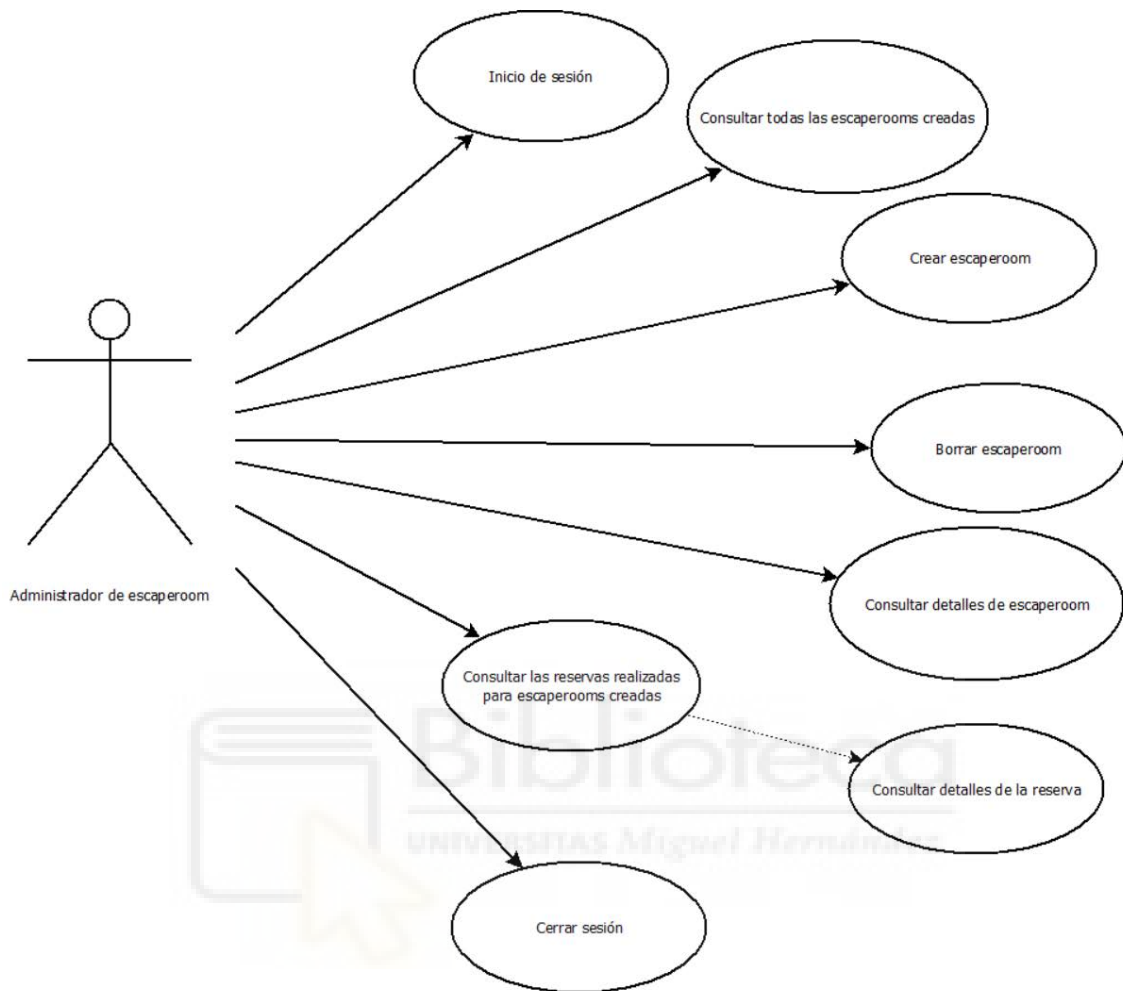


Imagen 50. Diagrama de administrador de escaperoom

C.U. 3.1	Crear escaperoom.
Actor/es	Administrador de escaperoom.
Descripción	Un administrador de escaperoom quiere publicar una escaperoom en la app.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Pulsa en el botón de la parte superior de "Añadir escaperoom".

	P3: Introduce todos los datos que requiere para registrar una escaperoom en la app. P4: Pulsa en el botón de “Enviar”.
Poscondición	PostC1: Se ha añadido correctamente la escaperoom a la base de datos
Excepciones	No hay excepciones.

Tabla 12. Caso de uso 3.1

C.U. 3.2	Consultar las escaperooms creadas.
Actor/es	Administrador de escaperoom.
Descripción	Un administrador de escaperooms quiere consultar las escaperooms que ha creado.
Dependencias	C.U. 3.1.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: En la pantalla principal puede ver un carrusel de algunas de las escaperooms que ha creado. P3: Para ver todas las escaperooms creadas se pulsa el botón de “Ver más”
Poscondición	PostC1: El usuario ha consultado las escaperooms creadas.
Excepciones	No hay excepciones.

Tabla 13. Caso de uso 3.2

C.U. 3.3	Consultar detalles de una reserva en una escaperoom creada.
Actor/es	Administrador de escaperoom.
Descripción	Un administrador de escaperoom quiere revisar la lista de reservas de sus escaperooms.
Dependencias	C.U. 3.1.

Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión. PreC3: Que el administrador haya publicado mínimo una escaperoom.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Arrastra hacia la derecha desde la parte izquierda de la pantalla. P3: Pulsa la opción de “Lista de reservas”. P4: Pulsar en la reserva que el administrador quiera revisar .
Poscondición	PostC1: El usuario ha visualizado la lista de reservas.
Excepciones	ExC1: No hay ninguna reserva, en ese caso se mostrará un mensaje en pantalla que pone “No hay reservas”.

Tabla 14. Caso de uso 3.3

C.U. 3.4	Borrar una escaperoom.
Actor/es	Administrador de escaperoom.
Descripción	Un administrador de escaperooms quiere borrar una escaperoom que ha creado.
Dependencias	C.U. 3.1.
Precondición	PreC1: Que la aplicación esté operativa. PreC2: Que el usuario haya iniciado sesión. PreC3: Que el administrador haya publicado mínimo una escaperoom.
Secuencia normal	P1: El usuario ingresa en la aplicación. P2: Busca la escaperoom que desea eliminar entre todas las escaperooms creadas pulsando el botón de “Ver más”. P3: Localiza la escaperoom que desea eliminar y pulsa en la imagen o nombre. P4: Se mostrará la página de detalles de la escaperoom y pulsará el botón de la esquina inferior izquierda “Borrar”.

Poscondición	PostC1: El usuario administrador ha eliminado la escaperoom.
Excepciones	No tiene excepciones.

Tabla 15. Caso de uso 3.4

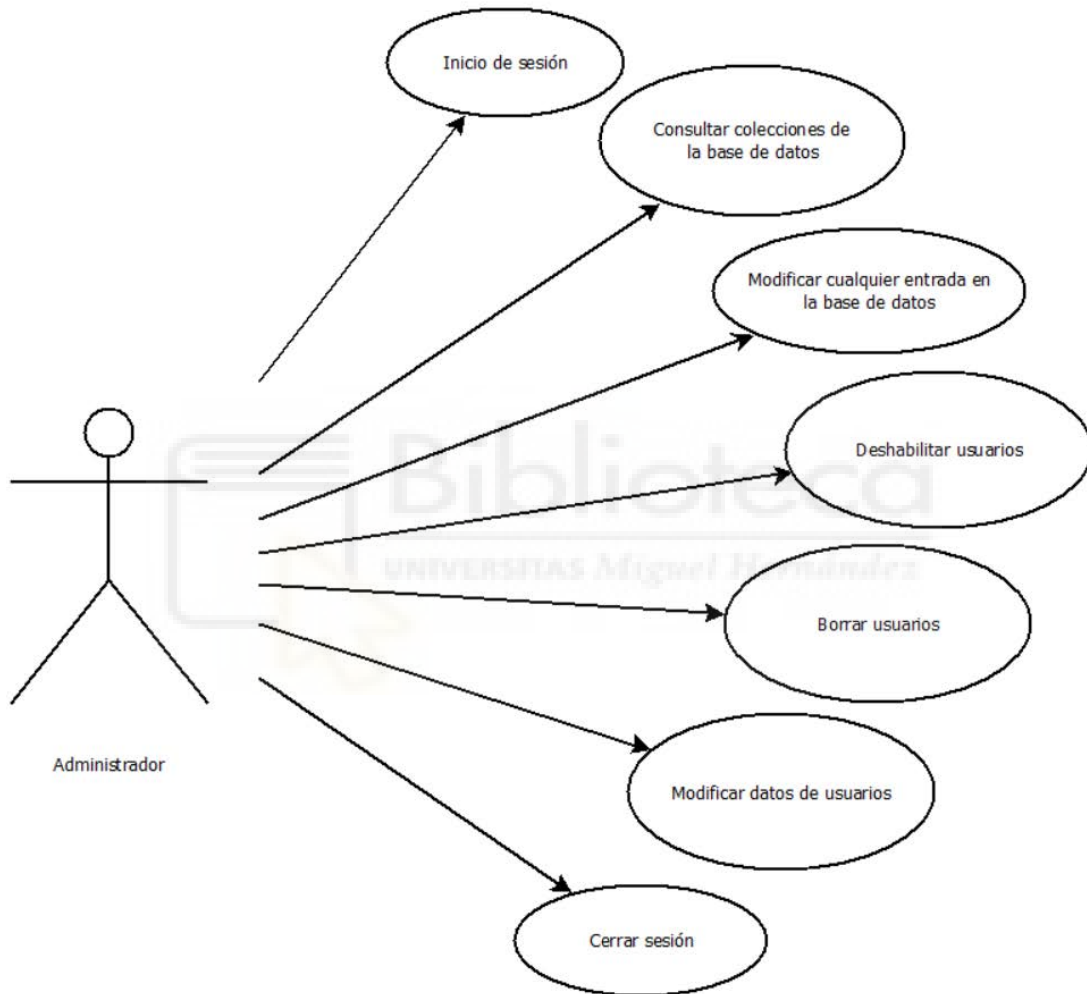


Imagen 51. Diagrama de administrador

C.U. 4.1	Consultar colecciones de la base de datos.
Actor/es	Administrador.
Descripción	Un administrador quiere consultar datos relacionados con la aplicación.
Dependencias	No tiene dependencias.

Precondición	PreC1: Que Firebase esté operativo. PreC2: Que el administrador haya iniciado sesión en Firebase.
Secuencia normal	P1: Despliega la barra lateral izquierda. P2: Le hace click a la opción de "Firestore Database". P3: Se muestra en pantalla una tabla con las colecciones, sus documentos y los campos de cada documento.
Poscondición	PostC1: El administrador ha consultado los datos que deseaba.
Excepciones	No hay excepciones.

Tabla 16. Caso de uso 4.1

C.U. 4.2	Modificar datos de un usuario.
Actor/es	Administrador.
Descripción	Un administrador desea modificar un dato de un usuario.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que Firebase esté operativo. PreC2: Que el administrador haya iniciado sesión en Firebase.
Secuencia normal	P1: Despliega la barra lateral izquierda. P2: Le hace click a la opción de "Firestore Database". P3: Se muestra en pantalla una tabla con las colecciones, clicar la de "users". P4: Busca entre los documentos de la colección el del usuario que quiere modificar y darle click. P5: Una vez localizado clicar el campo o campos que quiera editar y efectuar el cambio.
Poscondición	PostC1: El administrador ha modificado los datos del usuario con éxito.
Excepciones	No hay excepciones.

Tabla 17. Caso de uso 4.2

C.U. 4.3	Deshabilitar un usuario.
Actor/es	Administrador.
Descripción	Un administrador quiere deshabilitar la cuenta de un usuario.
Dependencias	No tiene dependencias.
Precondición	PreC1: Que Firebase esté operativo. PreC2: Que el administrador haya iniciado sesión en Firebase.
Secuencia normal	P1: Despliega la barra lateral izquierda. P2: Le hace click a la opción de "Authentication". P3: Se muestra en pantalla una lista con los usuarios que hay registrados en nuestra aplicación. P4: Identifica el usuario que quiere deshabilitar. P5: Una vez localizado, clicar en el botón de tres puntos a la derecha y seleccionar la opción de deshabilitar. P6: Aparecerá un pop-up de para confirmar la acción, y hace click en la opción de "inhabilitar".
Poscondición	PostC1: El administrador ha inhabilitado al usuario exitosamente.
Excepciones	No hay excepciones.

Tabla 18. Caso de uso 4.3

5.3.3 Diagrama de actividades

A continuación, se exhibirán diagramas que representan algunas de las operaciones que se pueden realizar en la aplicación:

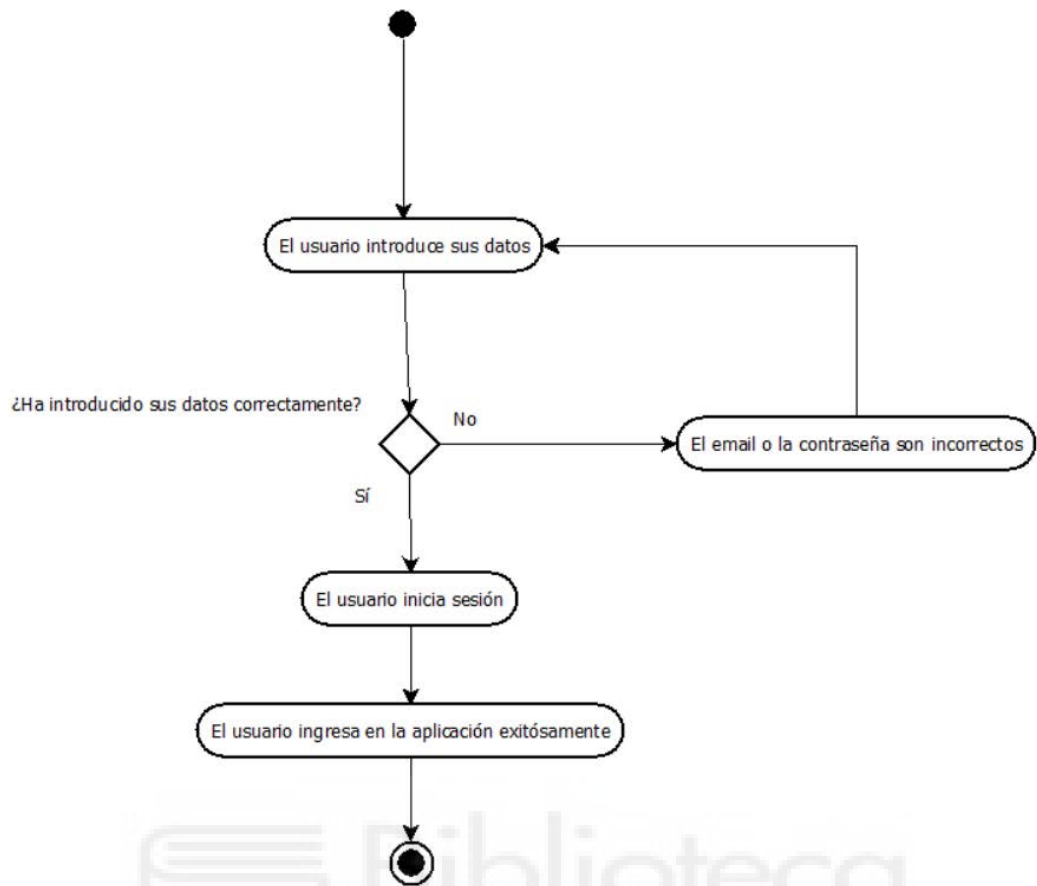


Imagen 52. Diagrama de actividad del inicio de sesión



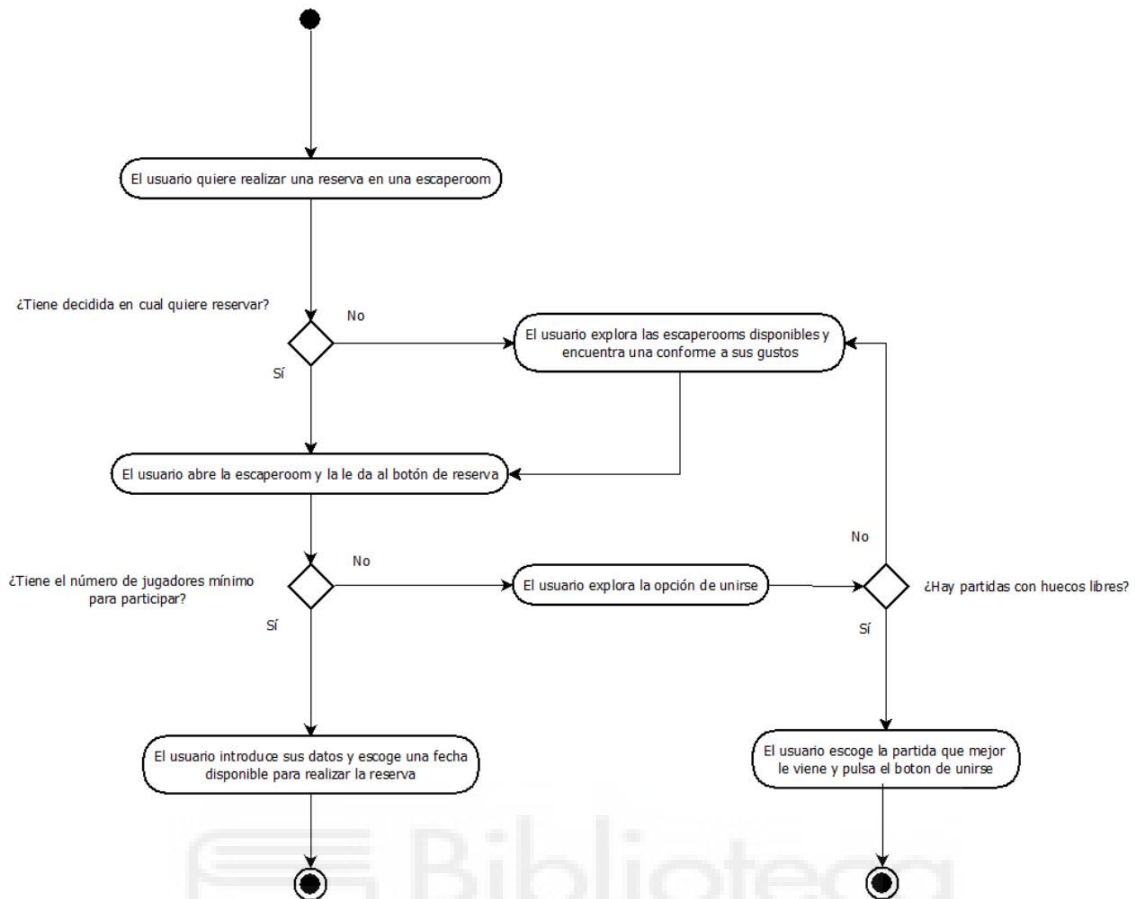


Imagen 53. Diagrama de actividad de reserva en escape room

5.3.4 Diagrama de clases

Los diagramas de clases representan toda la información relativa a las clases y sus relaciones con otras, sirven para abstraerse y tener una visión más clara de las clases que componen el proyecto.

Los siguientes diagramas representan los atributos y los métodos que tienen las clases relacionadas con las reservas (Imagen 54) y las clases relacionadas con las escape rooms (Imagen 55).

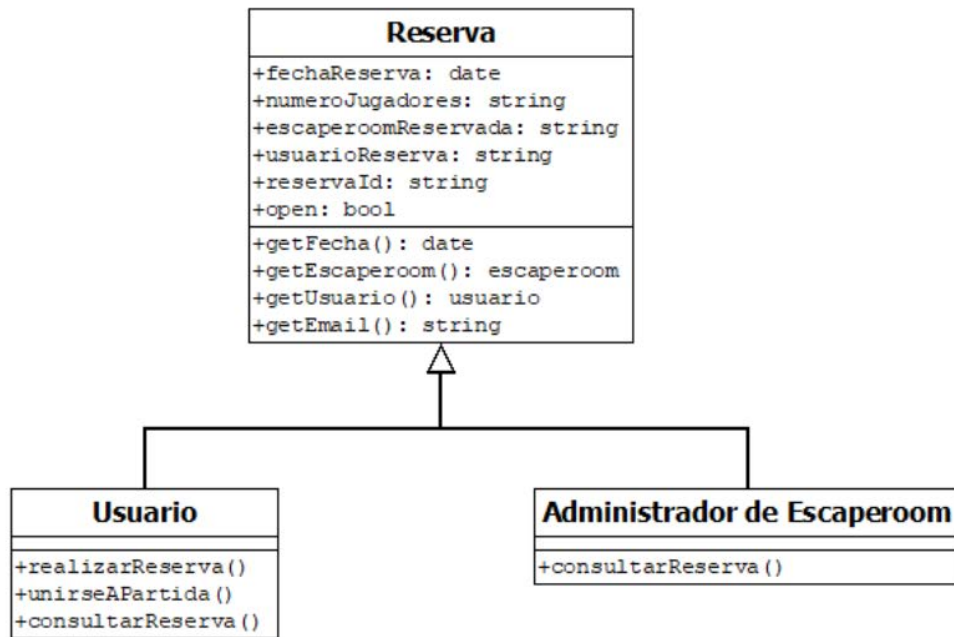


Imagen 54. Diagrama de clases relacionado con las reservas

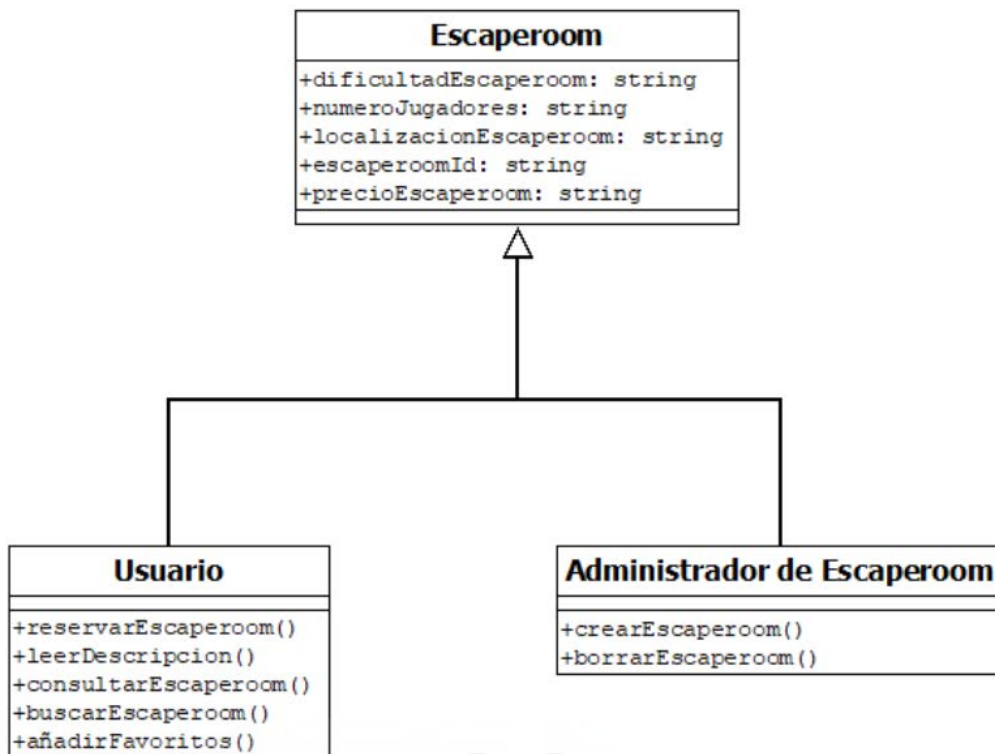


Imagen 55. Diagrama de clases relacionado con las escaperooms

5.4 Descripción de la base de datos

Como se ha mencionado en el apartado 4.1.3.1, Firebase es un servicio de base de datos que tiene una estructura NoSQL, eso significa que no hace uso de elementos como claves primarias, secundarias, no hay tablas ni relación entre estas. En cambio hace uso de unas estructuras llamadas colecciones formadas por documentos que tienen un conjunto de campos y definen el tipo de dato y hacen las veces de tablas. En este proyecto hay tres colecciones distintas:

- users: es la colección que contiene la lista de usuarios que hay registrados en la aplicación y guardan sus datos, que son:
 - nombre: es el nombre que tiene el usuario en la aplicación.

- email: es el email vinculado a la cuenta y con el que se registra, no se puede cambiar.
 - id: es el uid generado en firebase al crear la cuenta.
 - isAdmin: es un booleano que será true si la cuenta es de tipo administrador.
 - favoritas: es un array de strings que guarda los uid de las escaperooms que el usuario marque como favoritas.
 - completadas: es un array de strings que guarda los uid de las escaperooms que el usuario haya completado.
 - opcionesSeleccionadas: es un array de strings que guarda las opciones elegidas en el carrusel antes de registrarse en la aplicación.
 - empresa: el nombre de la empresa a la que pertenece el administrador, exclusivo para cuentas de administrador.
 - urlImagen: es un identificador para acceder a la foto de perfil del usuario en Storage.
- escaperooms: contiene los documentos con los datos relativos a las escaperooms que son los siguientes:
- nombre: es el nombre de la escaperoom.
 - ciudad: es la ciudad en la que se encuentra la escaperoom.
 - descripción: en este campo se guarda la descripción de la escaperoom.
 - dificultad: puede tener tres valores, fácil, medio y difícil.
 - tiempo: es el tiempo que se tarda en completar la escaperoom.
 - jugadoresMax: el máximo de jugadores que pueden participar en la escaperoom.
 - jugadoresMin: el mínimo de jugadores que pueden participar en la escaperoom.
 - etiquetas: es un array de strings con las etiquetas de la escaperoom, sirve para recomendar a los jugadores en función de sus preferencias.
 - precio: es el precio que tiene reservar la escaperoom.

- empresa: el nombre de la empresa a la que pertenece la escaperoom.
 - imagen: es un identificador para acceder a la foto de la escaperoom en Storage.
- reserves: es la colección que tiene los datos de las reservas, los documentos de esta colección tienen los siguientes campos:
- nombre: es el nombre del usuario que ha hecho la reserva.
 - email: es el email del usuario que ha hecho la reserva.
 - erId: es el id de la escaperoom en la que se ha hecho la reserva.
 - userId: es id del usuario que ha hecho la reserva.
 - jugadores: es el número de jugadores que van a participar.
 - fecha: es la fecha para la que se ha hecho la reserva.
 - phonenumber: es el número de teléfono que ha introducido el usuario que ha hecho la reserva.
 - open: es un booleano que, si es true y hay huecos disponibles en la reserva, aparecerá en la lista de partidas a las que unirse, si es false no aparecerá.

Para terminar, aun sin ser una base de datos relacional, se pueden realizar operaciones básicas utilizando los ids de los documentos o los uids de los usuarios registrados en el proyecto, que hacen las veces de claves primarias y que se han puesto a prueba, como se explica en el Anexo II.

5.5 Publicación de la aplicación

La aplicación no está disponible para descargar desde ninguna plataforma de distribución de aplicaciones móviles. Aunque es una aplicación que podría evolucionar y tener éxito en su ámbito, actualmente no dispone de todas las funcionalidades que se espera de una aplicación comercializada.

No obstante, es posible acceder a la aplicación desde Github[15]. Para ello, se debe de clonar el proyecto en una carpeta local y abrirlo desde Android Studio o Visual Studio Code.

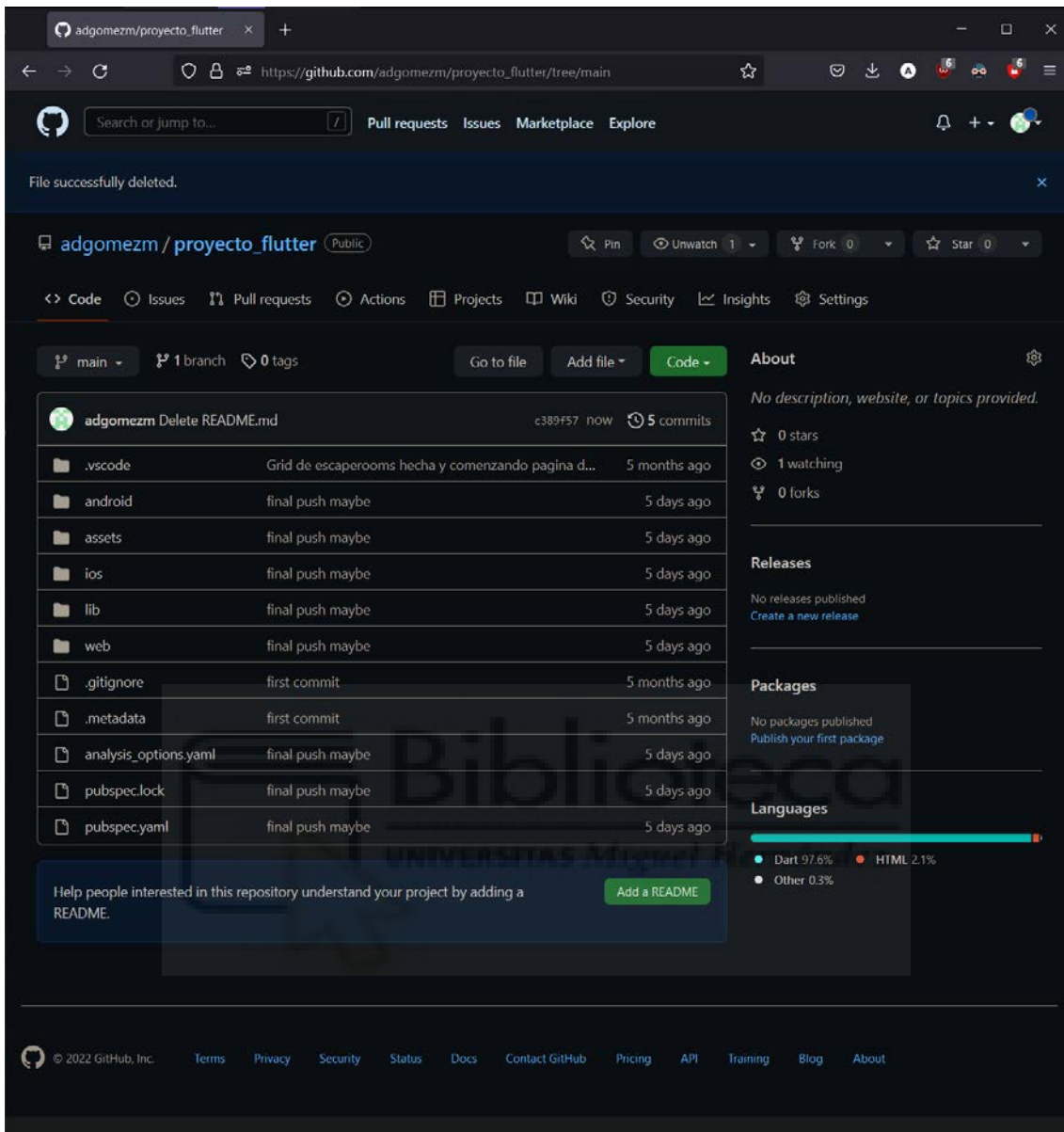


Imagen 56. Repositorio del proyecto en GitHub.

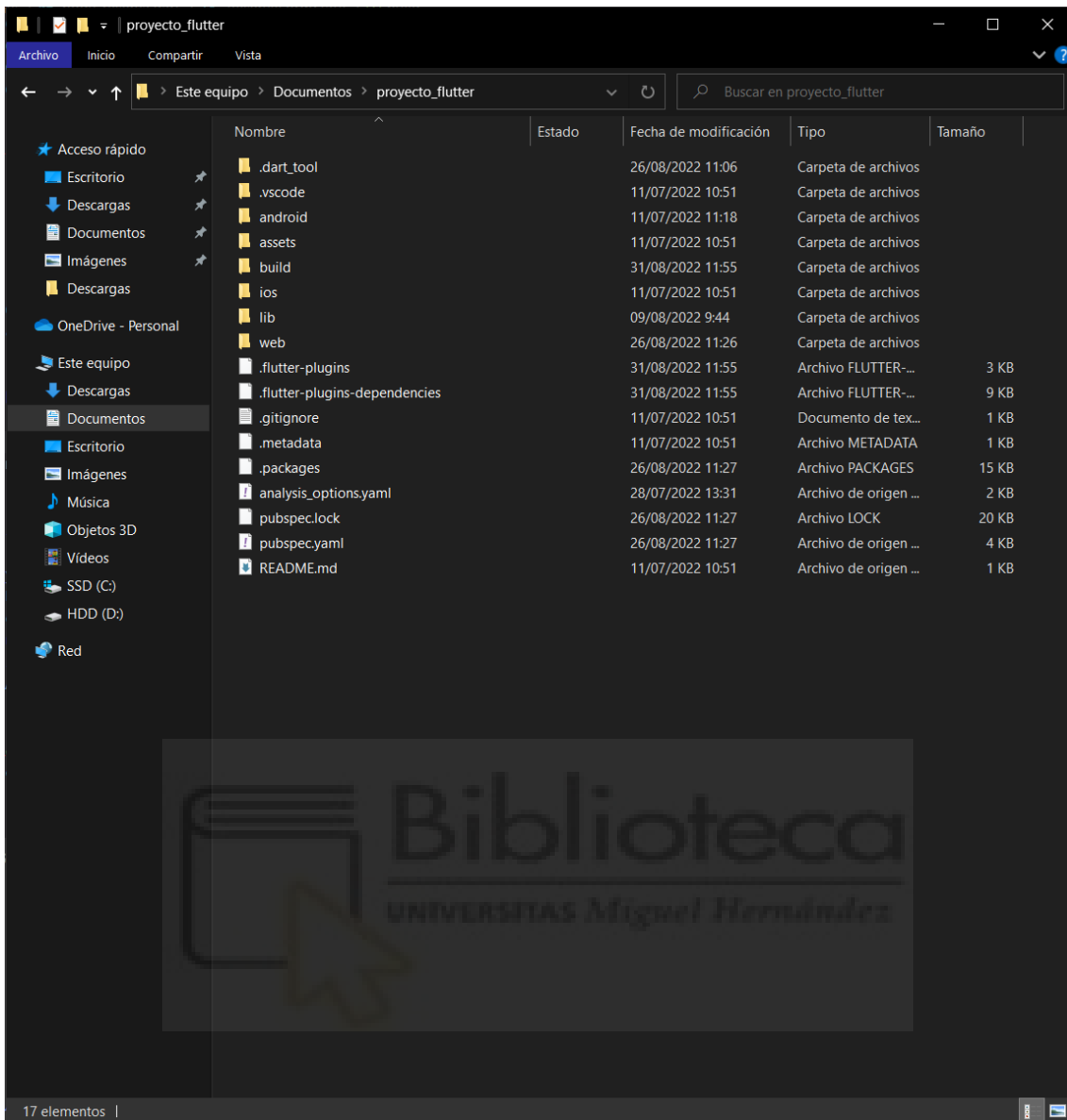


Imagen 57. Carpeta del proyecto en local

En el desarrollo de este trabajo se ha abierto desde Visual Studio Code.

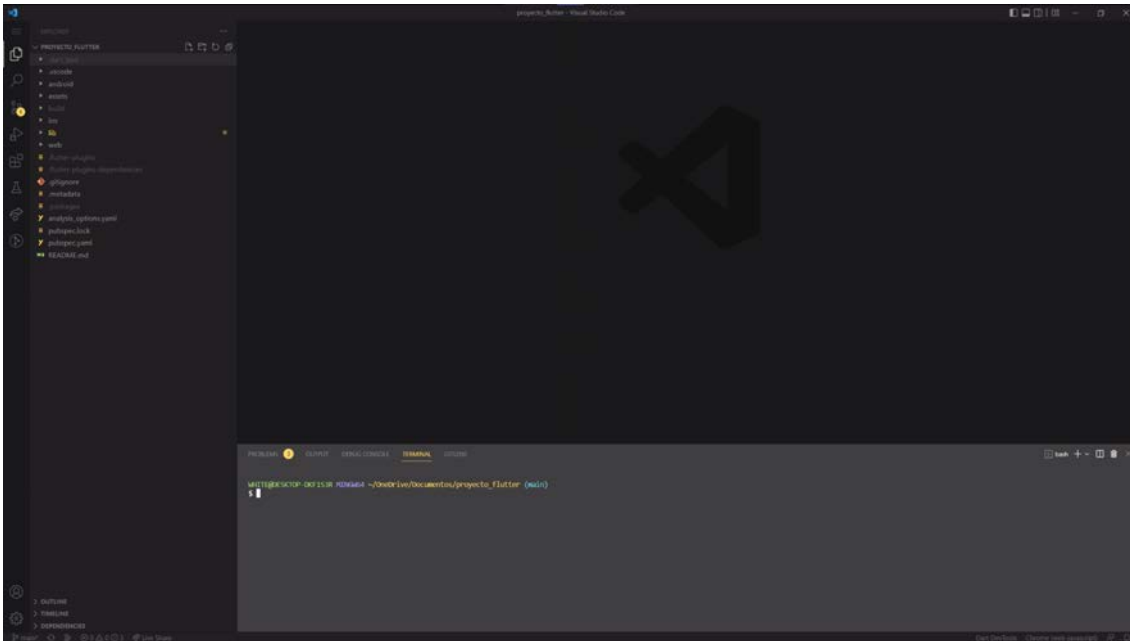


Imagen 58. Proyecto abierto en Visual Studio Code

Para poder poner en marcha la aplicación se necesita instalar el SDK de Android ya que es lo que permite ejecutar los comandos necesarios para compilar y ejecutar la aplicación. Para toda la instalación existe una guía proporcionada por Flutter que detalla cada paso de manera clara [16].

Una vez instalado el SDK ya se puede utilizar el terminal integrado de Visual Studio Code. Primero se debe de comprobar que se tiene la última versión de Flutter entre otras cosas, para ello si se ha hecho una correcta instalación del SDK se escribe el siguiente comando: `flutter doctor`

```
WHITE@DESKTOP-DKF1S3R MINGW64 ~/OneDrive/Documentos/proyecto_flutter (main)
$ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.0.5, on Microsoft Windows [Version 10.0.19044.1889], locale es-ES)
[✓] Android toolchain - develop for Android devices (Android SDK version 33.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.2.5)
[✓] Android Studio (version 2021.2)
[✓] VS Code (version 1.70.2)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

• No issues found!
```

Imagen 59. Resultado del comando `flutter doctor`

En caso de tener todo correcto se mostrará una respuesta como la de la imagen, si hay algo que no cumple los requisitos especificará cual es el problema.

Tras comprobar que todo está en regla y comprobar que el directorio es en el que está el proyecto, se escribe el comando `flutter run` que se encarga de compilar, instalar y ejecutar la aplicación en el dispositivo, en este caso es un emulador de Android Studio.

```
WHITEDESKTOP-DK153R MIKAK4 ~/OneDrive/Documentos/proyecto_flutter (main)
$ flutter run
Using hardware rendering with device Android SDK built for x86. If you notice graphics artifacts, consider enabling software rendering with "--enable-software-rendering".
Launching lib\main.dart on Android SDK built for x86 in debug mode...
Running Gradle task 'assembleDebug'... 53,3s
✓ Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app.apk... 10,9s
W/FlutterActivityAndFragmentDelegate( 6632): A splash screen was provided to Flutter, but this is deprecated. See flutter.dev/go/android-splash-migration for migration steps.
Syncing files to device Android SDK built for x86... 181ms

flutter run key commands.
r Hot reload.
R Hot restart.
h List all available interactive commands.
d Detach (terminate "flutter run" but leave application running).
e Clear the screen.
q Quit (terminate the application on the device).

Running with unsound null safety
For more information see https://dart.dev/null-safety/unsound-null-safety

An Observatory debugger and profiler on Android SDK built for x86 is available at: http://127.0.0.1:51440/hChWwqDjGI=/
The Flutter DevTools debugger and profiler on Android SDK built for x86 is available at: http://127.0.0.1:9101?uri=http://127.0.0.1:51440/hChWwqDjGI=
```

Imagen 60. Resultado del comando `flutter run`

Y con esto ya se encuentra la aplicación funcionando.

CAPÍTULO 6. CONCLUSIONES Y PROPUESTAS

6.1 Conclusiones

Llegados a este punto, el objetivo del proyecto, que consistía en facilitar la realización de reservas en escaperooms, ha sido alcanzado.

Este proyecto me ha supuesto un reto personal, al tener que enfrentarme a una tecnología que no conocía tan en profundidad como otras trabajadas en la carrera. He hecho uso de la documentación y tutoriales para entender algunas de las funcionalidades de Flutter. Y he tenido la obligación de ser autodidacta que ha mejorado mi capacidad de aprendizaje al tener que enfrentarme a esta situación.

También comentar que en el momento de elegir herramientas y tecnologías he podido conocer muchas otras que en las que podría haber desarrollado el proyecto.

6.2 Trabajo futuro.

En el caso de querer seguir con este proyecto, ya sea para comercializar o como proyecto personal, se podrían implementar las funcionalidades que se han comentado en el apartado 1.4.

La primera que implementaría sería la funcionalidad del chat entre usuarios. Pienso que ayudaría bastante a crear una comunidad porque habría un contacto más directo entre los usuarios, que también provocaría la organización de eventos para la que añadiría otra de las funcionalidades que he comentado anteriormente. El apartado de noticias es otra funcionalidad de la que harían uso los administradores de escaperoom para anunciar ofertas, eventos y promociones, y para los usuarios sería una manera de mantenerse informado.

Y por último, se podría añadir una sección de comentarios para cada escaperoom. De esa manera los usuarios podrán dar sus opiniones para otros usuarios que se planteen participar en las escaperooms.



Bibliografía

Listado de referencias

[1] Artículo sobre escaperooms

<https://www.theguardian.com/games/2019/apr/01/get-out-how-escape-rooms-be-came-a-global-craze>

[2] Artículo sobre escaperooms

https://www.elconfidencial.com/tecnologia/2020-05-15/boom-cierre-escape-room-covid-coronavirus_2595927/

[3] Artículo sobre escaperooms

https://cincodias.elpais.com/cincodias/2018/08/24/companias/1535124533_144897.html

[4] Artículo sobre escaperooms

<https://escapeup.es/blog/estudio-mercado-sector-escape-rooms-2021>

[5] Artículo sobre escaperooms

https://www.eldiario.es/catalunya/escape-rooms-no-encuentran-salida-crisis-covid-19-25-salas-no-volveran-abrir_1_6186912.html

[6] Artículo sobre escaperooms

https://www.segre.com/es/noticias/guia/2021/05/09/las_escape_rooms_renacen_134130_1111.html

[7] Documentación de Visual Studio Code.

<https://code.visualstudio.com/docs>

[8] Documentación de Flutter

<https://docs.flutter.dev/>

[9] Documentación de Dart.

<https://dart.dev/overview>

[10] Artículo sobre Flutter.

<https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>

[11] Documentación de Firebase.

<https://firebase.google.com/docs?hl=es>

[12] Artículo sobre Firebase.

<https://www.nts-solutions.com/blog/firebase-que-es.html>

[13] Documentación de Cloud Firestore

<https://firebase.google.com/docs/firestore?hl=es>

[14] Apuntes de la asignatura de I.S. (autor: Juan Pedro Martinez Gallar)

https://docs.google.com/presentation/d/1p_bf_y6xlzuHzJ6W_H38PppKrNpNqE_OR07OzHdJto1M/edit#slide=id.i0

[15] Repositorio de Github.

https://github.com/adgomezm/proyecto_flutter

[16] Manual de instalación de Flutter.

<https://docs.flutter.dev/get-started/install>



ANEXO I

A continuación se hará un recorrido completo por la aplicación.

Para empezar, al abrir la aplicación se muestra la siguiente splash screen.



Imagen 61. Splash screen

Se supone que se abre la aplicación por primera vez, por lo que aparece un carrusel con la elección de las preferencias que se puede saltar dándole a “Saltar” en la esquina inferior izquierda.



Imagen 62, 63, 64 y 65. Carrusel de preferencias

A continuación, el formulario de registro y de inicio de sesión.

En el formulario de registro se puede elegir registrarse como “admin” pulsando el checkbox, lo que habilita un campo extra para el nombre de la empresa.

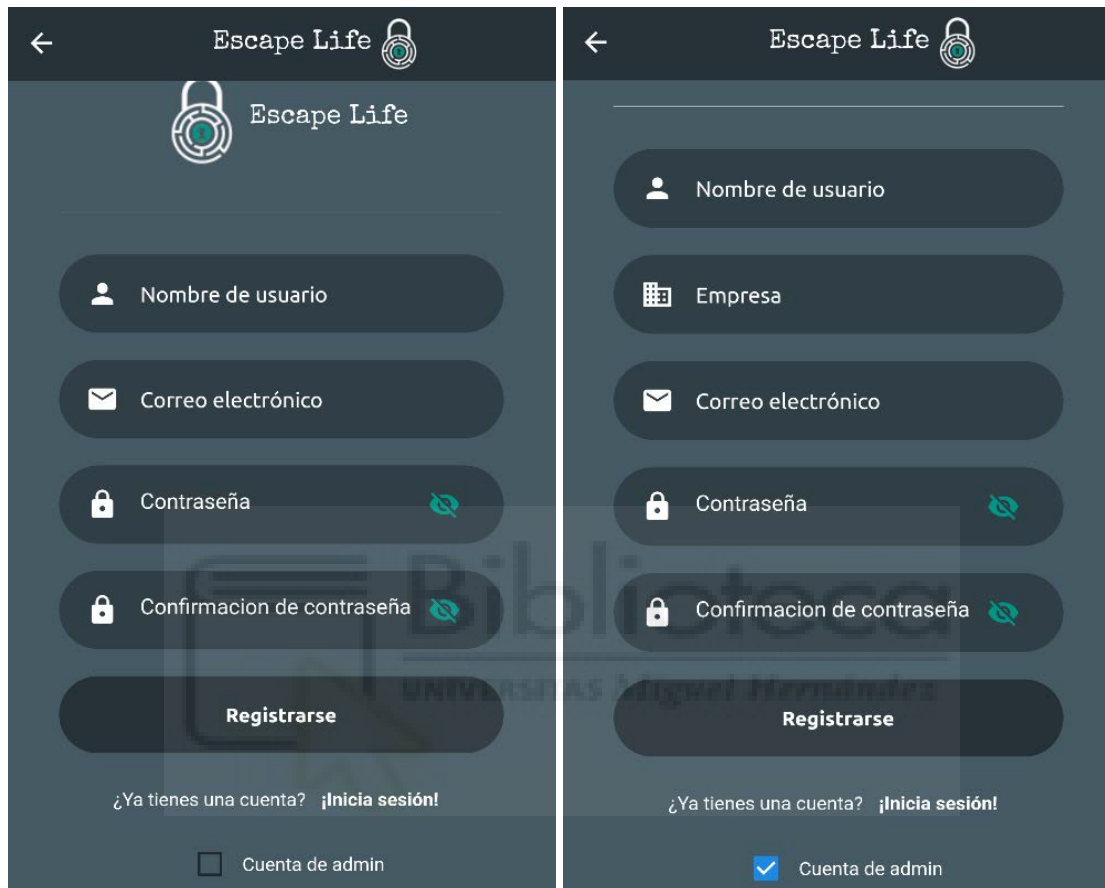


Imagen 66 y 67 . Formulario de registro

El formulario de inicio de sesión, donde el usuario en caso de tener cuenta escribirá su email y su contraseña para acceder. En caso de introducir los datos erróneamente te notifica.

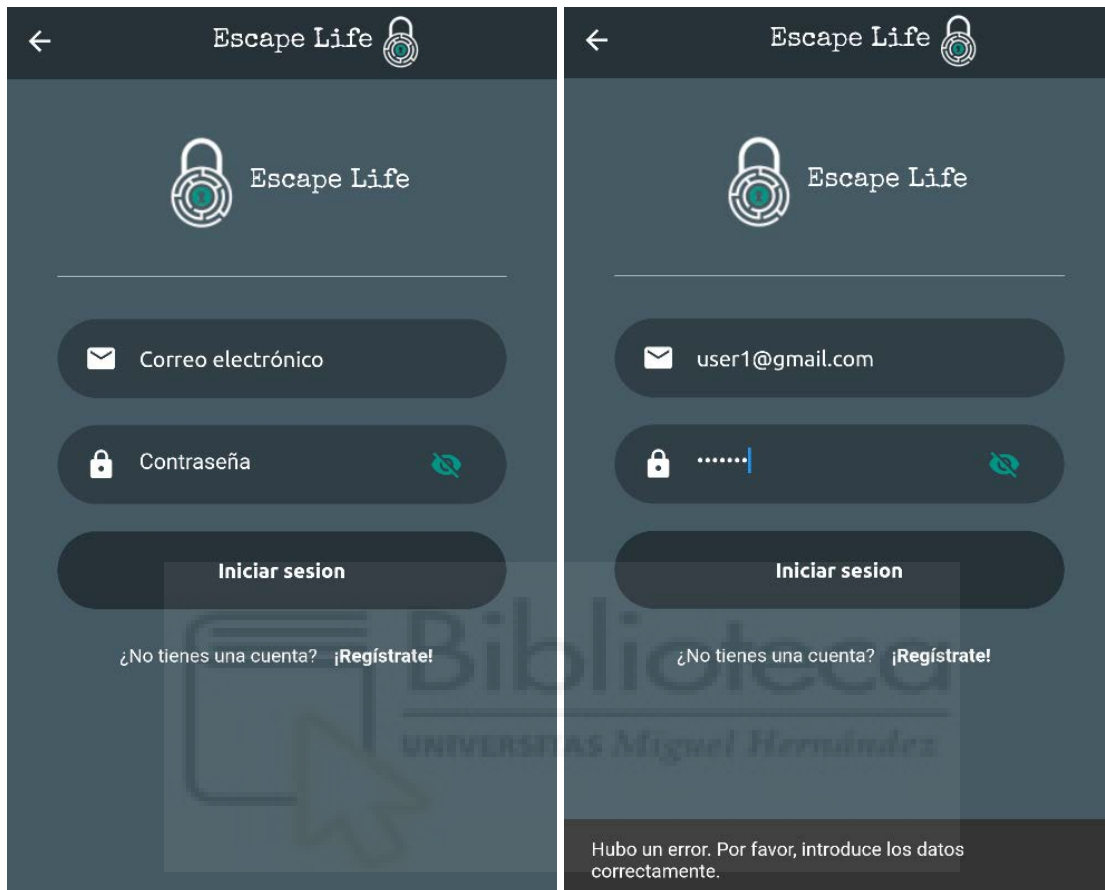


Imagen 68 y 69. Formulario de inicio de sesión

Desde el punto de vista de un usuario normal, una vez iniciada la sesión se mostrará la pantalla principal de la aplicación.

Se visualiza un slider con las mejor valoradas y abajo las que coinciden con las preferencias que se eligieron en el carrusel del principio. También se puede utilizar el buscador para localizar escape rooms por nombre, ciudad, o tipo.

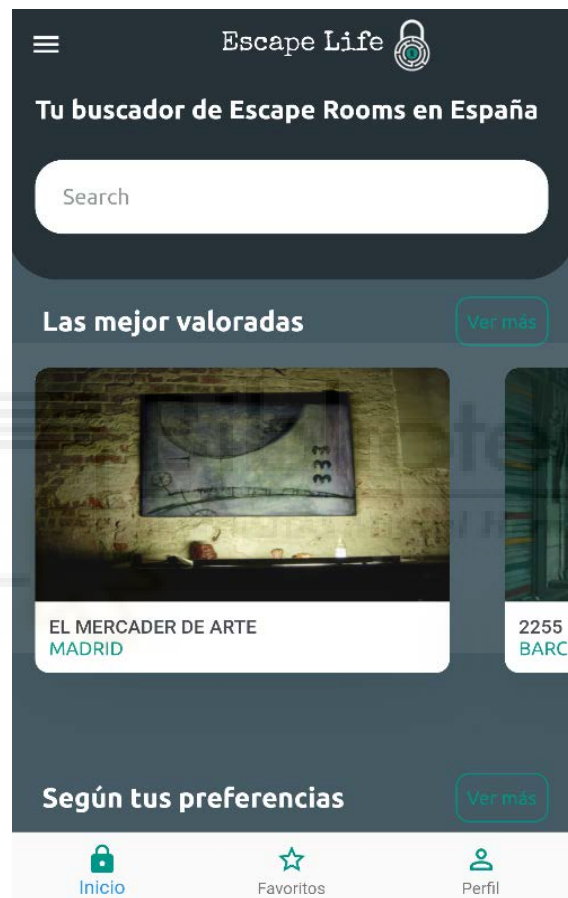


Imagen 70 y 71. Pantalla principal

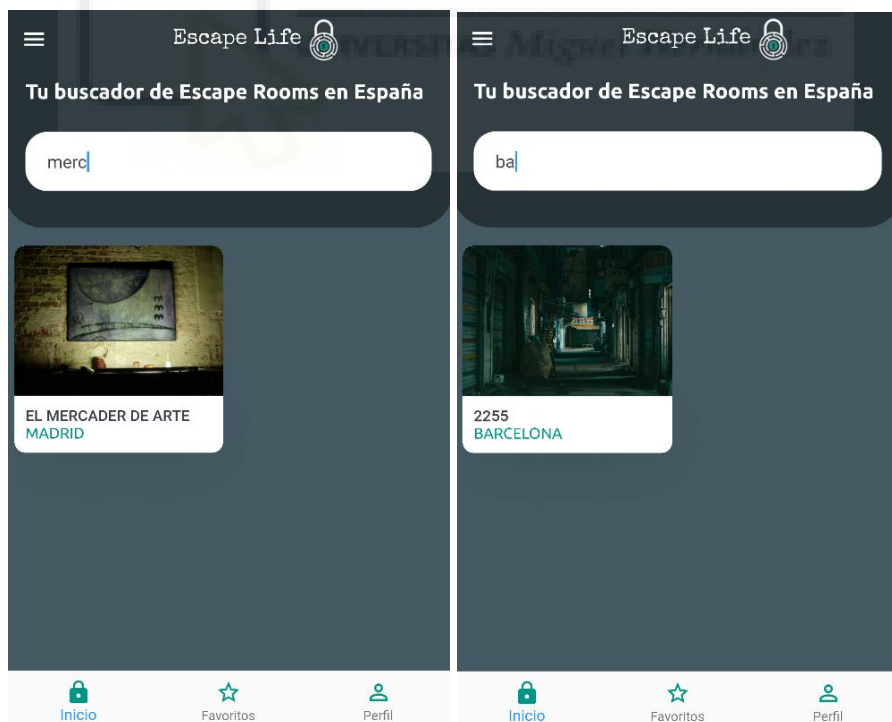
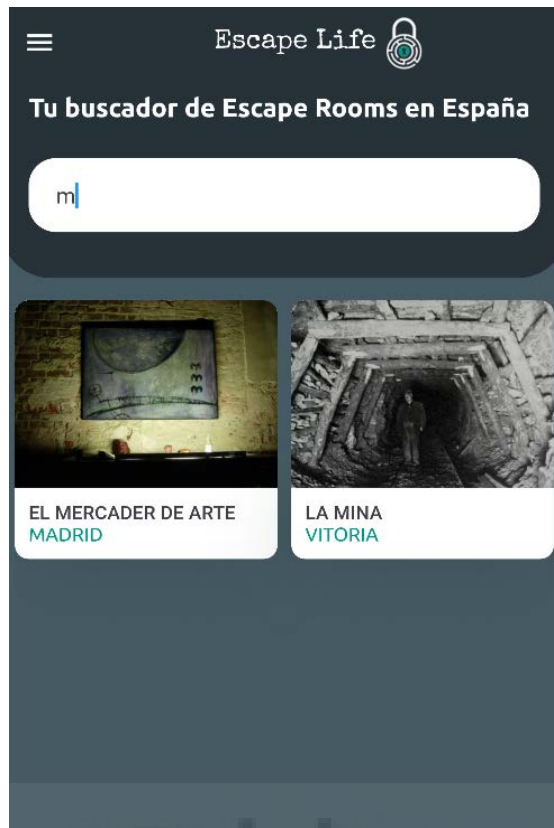


Imagen 72, 73 y 74. Resultados de búsqueda

Al pulsar en una escaperoom se muestra la página de detalles de la misma.

Se pueden ver datos como el tiempo de completado, el número de jugadores y la localización entre otros.

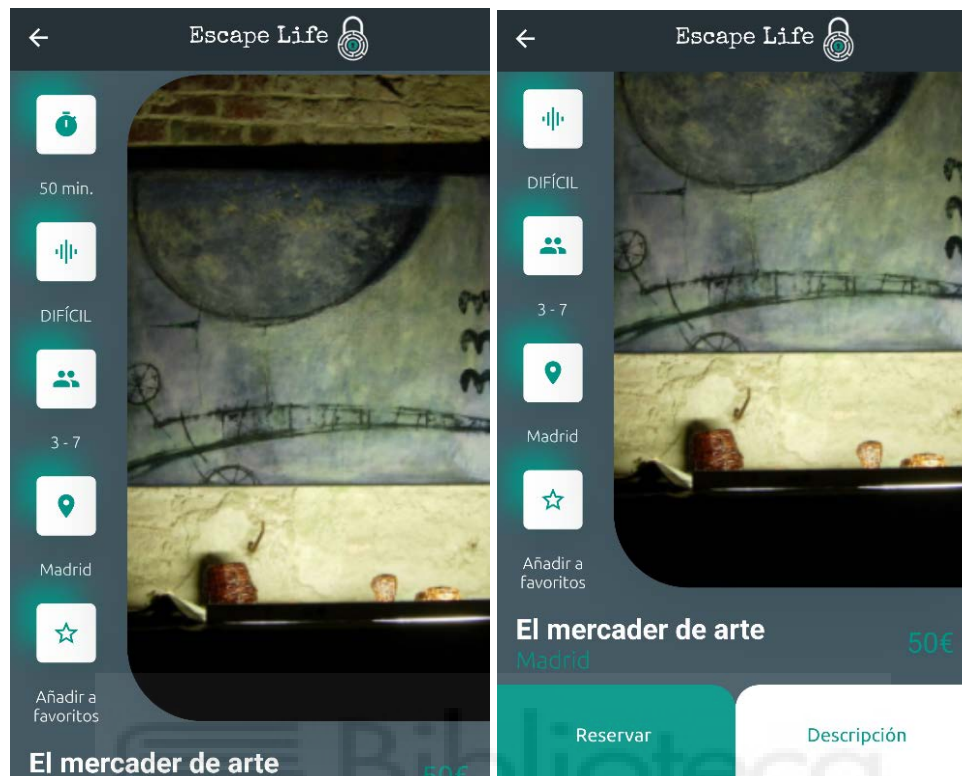


Imagen 75 y 76. Detalles de escaperoom

Al pulsar en el botón de descripción llevará a una pantalla en la que se muestra una descripción de la escaperoom.

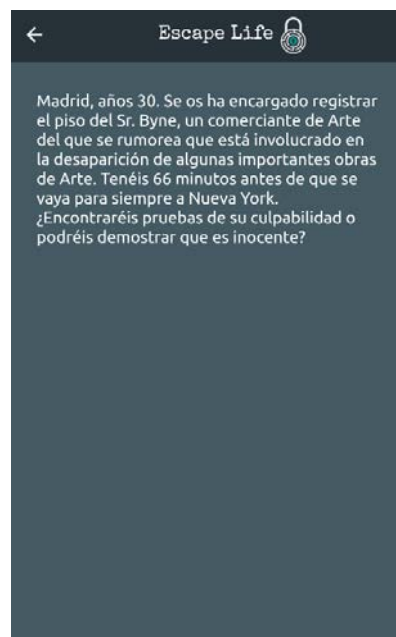


Imagen 77. Descripción de escaperoom

Y al pulsar en reservar se mostrará el formulario para realizar una reserva.



The image shows a mobile application interface for 'Escape Life'. At the top, there is a navigation bar with a back arrow, the text 'Escape Life' with a padlock icon, and two buttons: 'Reservar' (highlighted with a blue underline) and 'Unirse'. Below the navigation bar, the title 'Reservar en 'El mercader de arte'' is displayed. The form consists of several rounded rectangular input fields: 'Nombre y apellidos' (with a person icon), 'Introduzca su email' (with an envelope icon), 'Introduzca un numero de telefono' (with a telephone icon), 'Jugadores' (with a group of people icon), and 'Introduzca una fecha' (with a calendar icon). Below these fields is a checkbox labeled 'Permitir a la gente unirse'. At the bottom of the form is a large 'Enviar' button. A faint watermark 'Biblioteca' is visible in the background.

Imagen 78. Formulario de reserva

En caso de no tener el mínimo de jugadores se puede presionar la opción de “Unirse” en la parte superior que mostrará una lista de reservas hechas en la escaperoom.

Al presionar en una de las opciones disponibles se muestra una pantalla con datos de la reserva y un campo de texto para especificar el número de jugadores que quiere unirse. En caso de que exceda el máximo de jugadores permitido no se efectuará la unión.

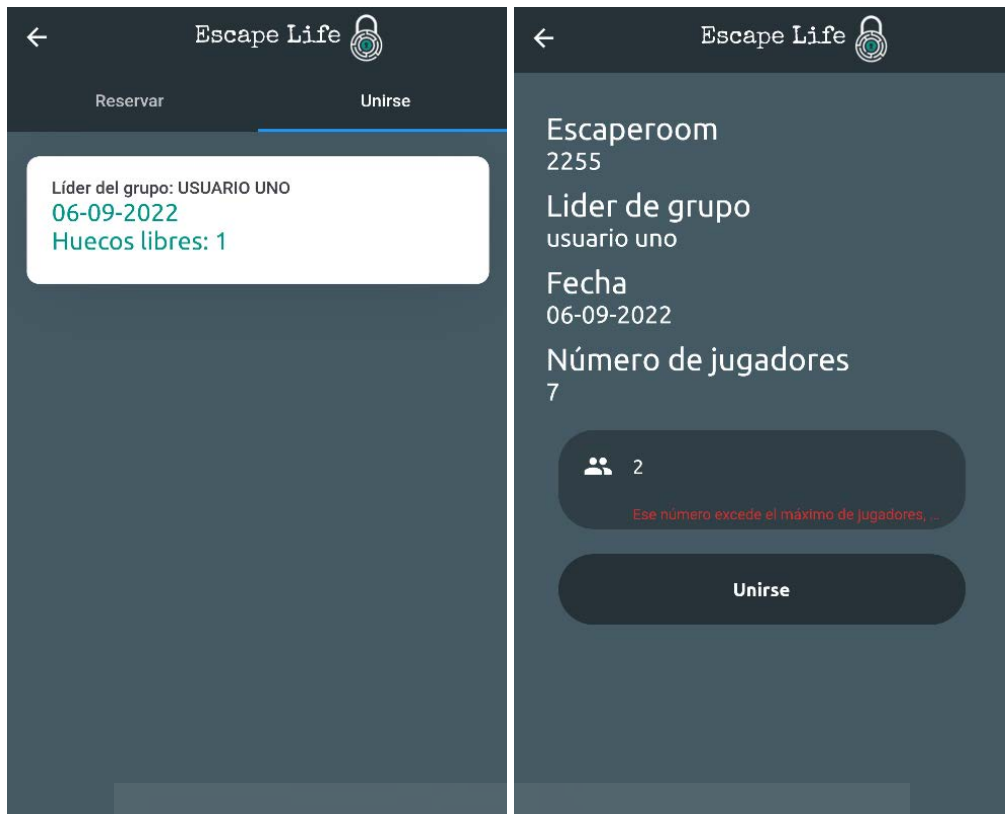


Imagen 79. Lista de partidas disponibles para unirse

Imagen 80. Formulario para unirse a la partida

Si el número de jugadores a unirse está permitido en la parte inferior se mostrará el siguiente mensaje.



Imagen 81. Notificación de unión con éxito

En el submenú al seleccionar la opción de “Favoritos” se muestra el perfil público con una lista de algunas de las salas marcadas como favoritas por el usuario.

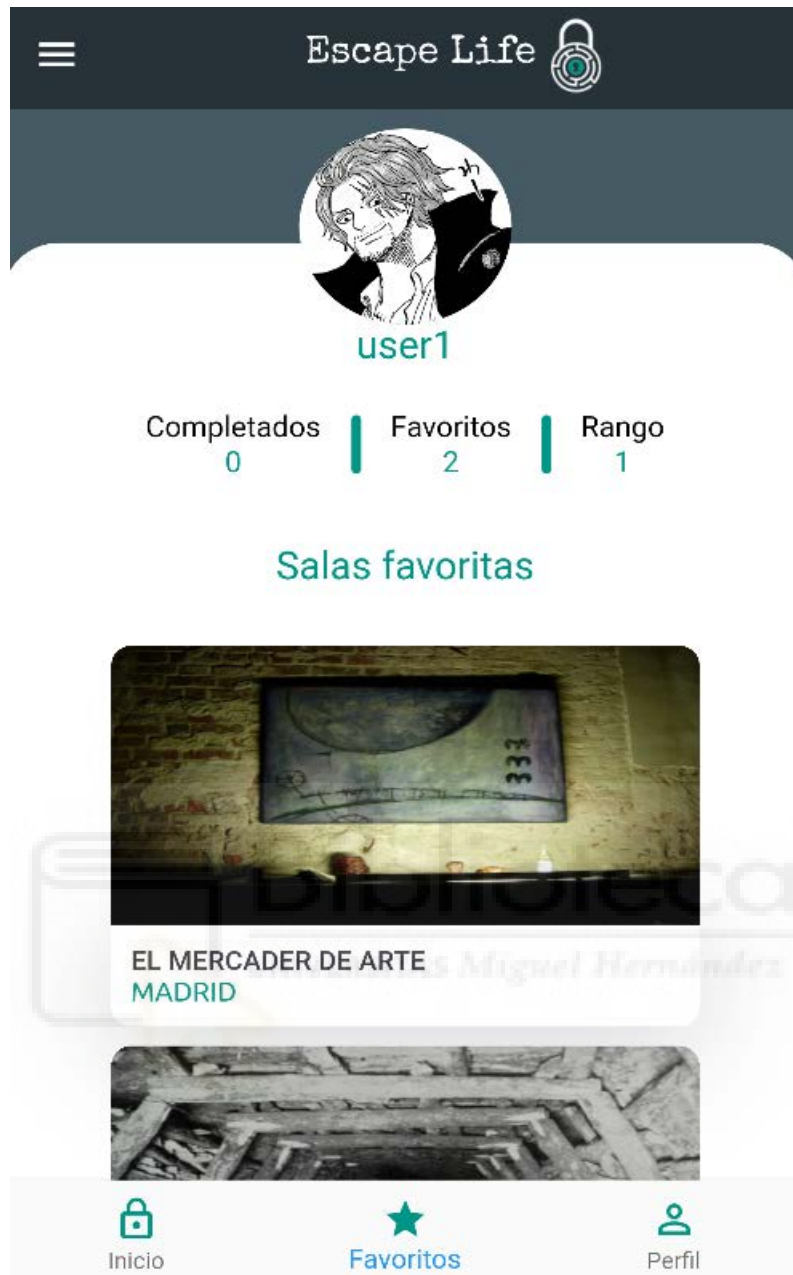


Imagen 82. Perfil público del usuario

En la última opción del submenú, en la que se muestra el perfil privado del usuario. Se puede cambiar la foto de perfil pulsando el botón al lado de la misma que abrirá el navegador de documentos del dispositivo para seleccionar una de las imágenes disponibles.

Al pulsar en el botón de “Salas favoritas” muestra una lista con las salas marcadas como favoritas por el usuario.

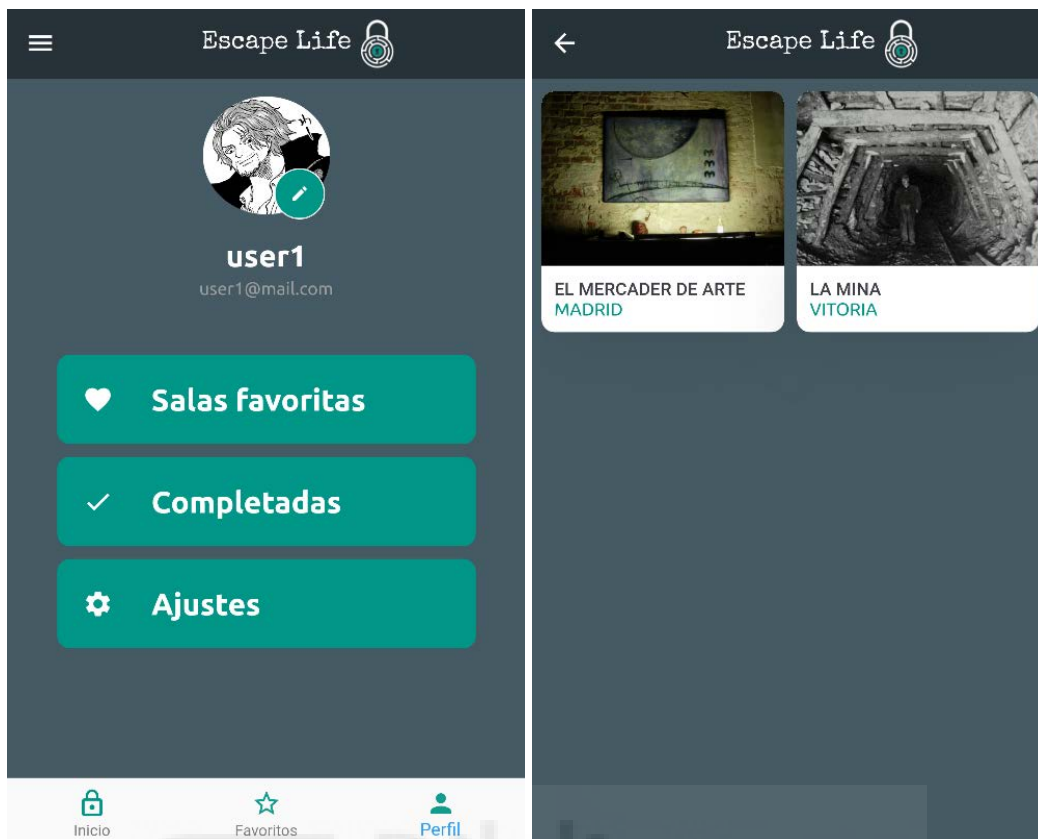


Imagen 83. Perfil privado del usuario

Imagen 84. Lista de salas favoritas

Lo mismo ocurre con el botón de “Completadas” que muestra las escape rooms que ha completado el usuario, en este caso está vacía, la única manera de insertarla es desde Firebase.

Y por último, en cuanto a las opciones del perfil, el botón de “Ajustes” que permite cambiar el nombre de usuario y la contraseña.

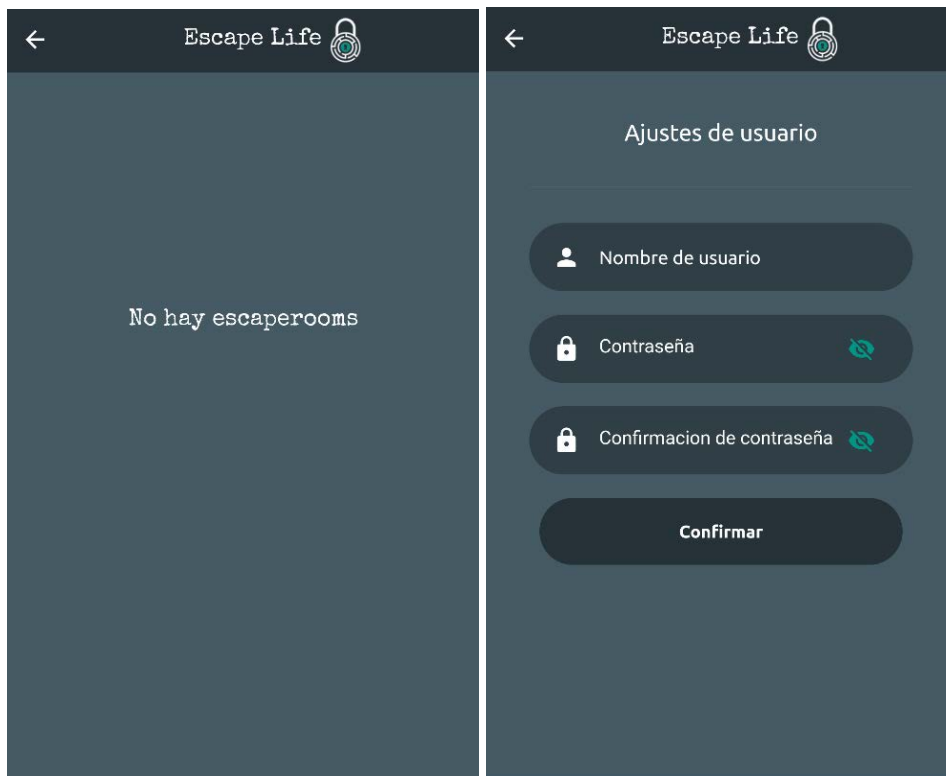


Imagen 85. Lista de salas completadas

Imagen 86. Formulario para cambiar nombre de usuario y contraseña.

El menú hamburguesa tiene algunas opciones que se van a explorar a continuación.

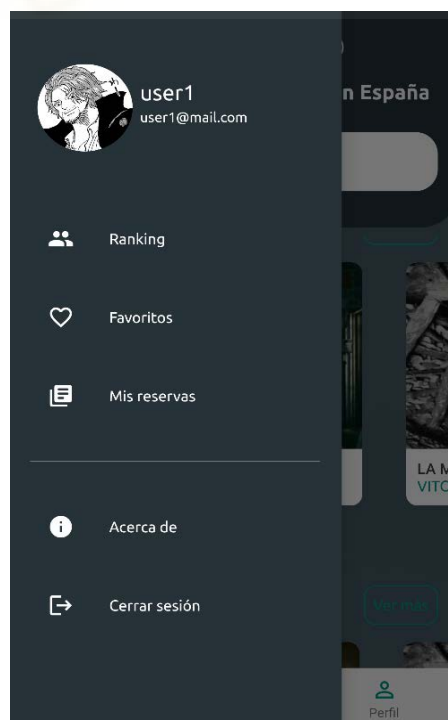


Imagen 87. Opciones del menú hamburguesa

Primero la opción de “Ranking”, se muestra una lista de usuarios que al pulsar enseña el perfil público del usuario.

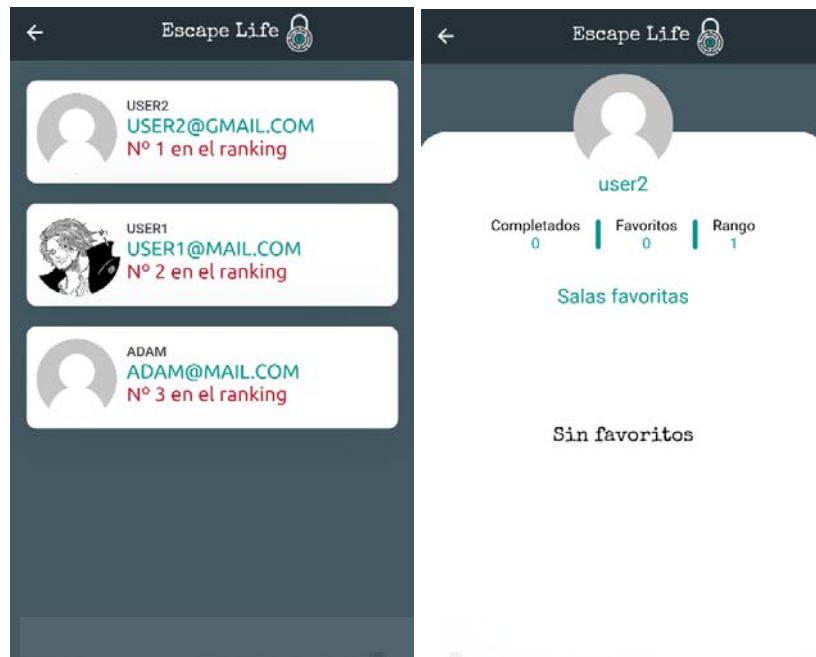


Imagen 88. Ranking de usuarios

Imagen 89. Perfil de otro usuario

La siguiente opción, “Mis reservas”, muestra una lista con las reservas hechas por el usuario, que al pulsar revela los detalles de la reserva.

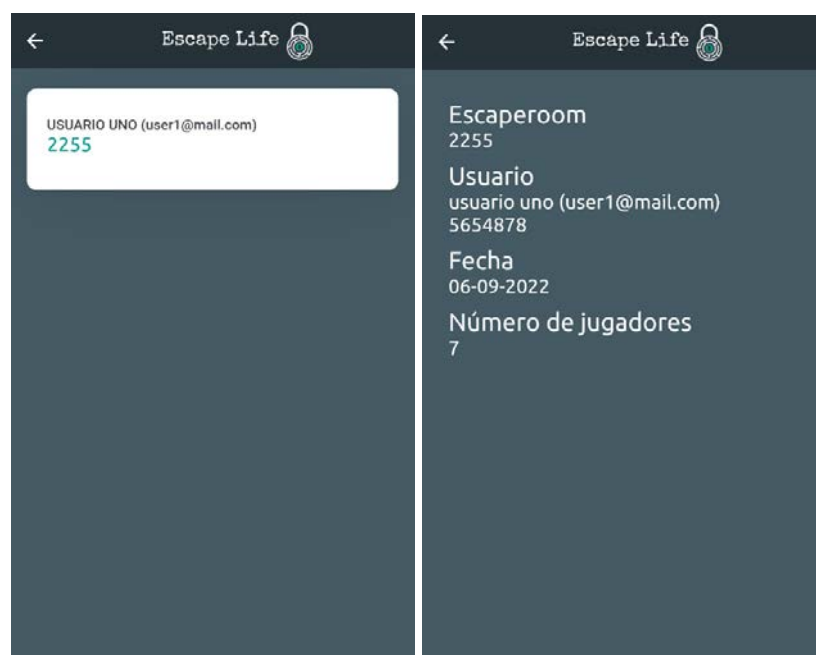


Imagen 90. Lista de reservas

Imagen 91. Detalles de reserva

Para acabar se encuentran las opciones de “Acerca de” que muestra un texto hablando sobre la aplicación. Y la opción de “Cerrar sesión” que cierra la sesión del usuario y se redirige a la pantalla de inicio de sesión.

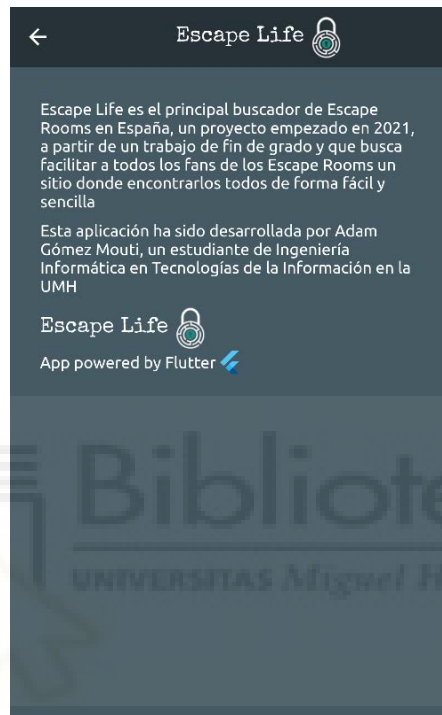


Imagen 92. Pantalla con información sobre la aplicación

Ahora para el punto de vista del administrador de la escaperoom.

La pantalla principal muestra un botón para añadir una escaperoom y un slider con las escaperooms creadas.

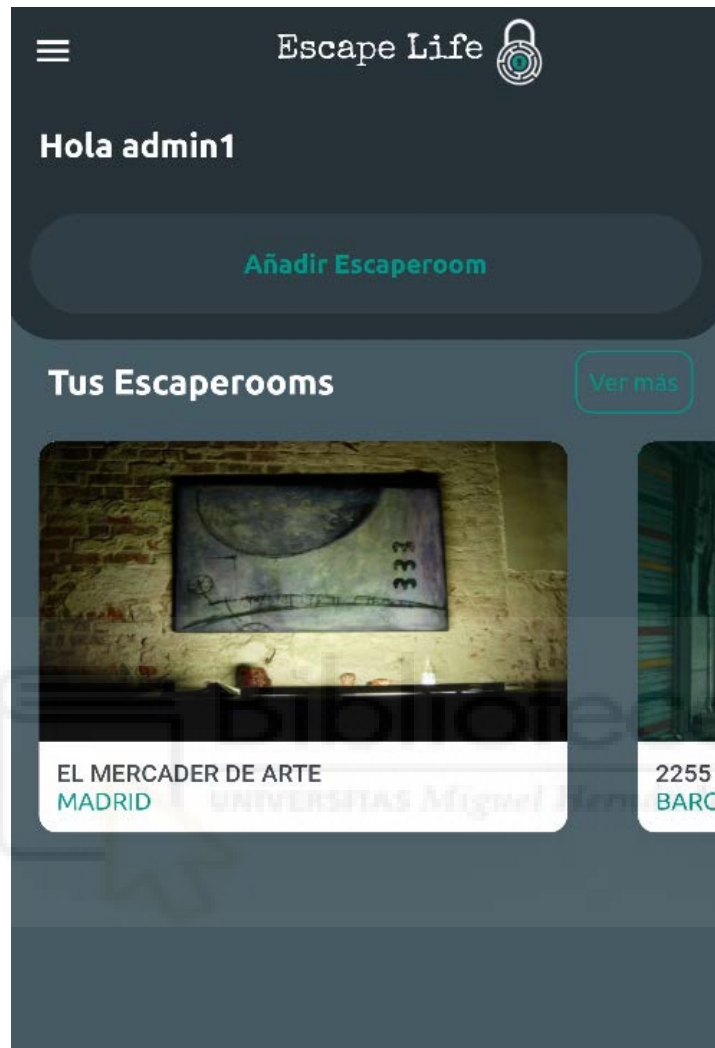
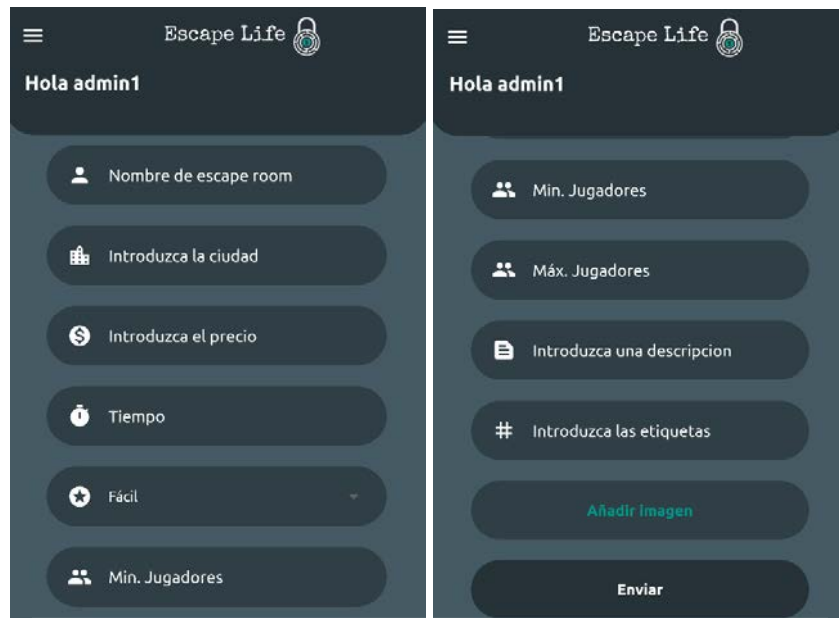


Imagen 93. Pantalla principal para administradores

Al pulsar el botón de “Añadir escaperoom” se revela un formulario que se debe rellenar para añadir una escaperoom a la base de datos.



The image shows two side-by-side screenshots of the 'Escape Life' app's admin interface. Both screens display 'Hola admin1' at the top. The left screenshot shows a vertical list of input fields: 'Nombre de escape room', 'Introduzca la ciudad', 'Introduzca el precio', 'Tiempo', 'Fácil' (with a dropdown arrow), and 'Min. Jugadores'. The right screenshot shows: 'Min. Jugadores', 'Máx. Jugadores', 'Introduzca una descripción', 'Introduzca las etiquetas', an 'Añadir imagen' button, and an 'Enviar' button at the bottom.

Imagen 94 y 95. Formulario de creación de una escaperoom

Una vez creada se mostrará en la pantalla principal y al pulsar sobre la escaperoom mostrará sus detalles. Al pulsar “Borrar” se borra la escaperoom.

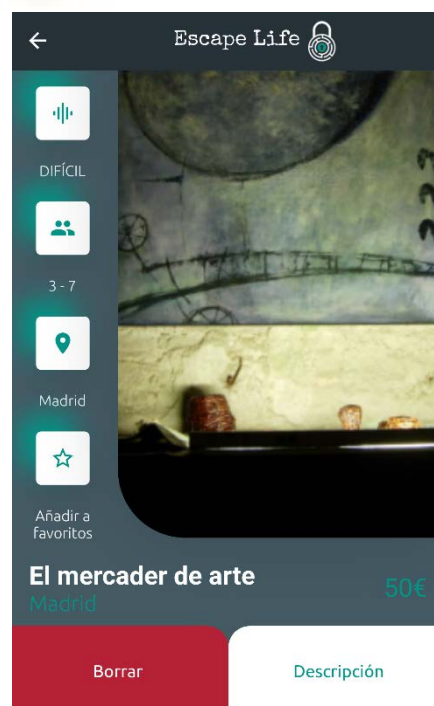


Imagen 96. Detalles de escaperoom para administradores

Y en el menú de hamburguesa, para el administrador se muestran 3 opciones. La única particular del administrador es la de “Lista de reservas” ya que las otras opciones són idénticas a las del usuario.

La siguiente opción, “Lista de reservas”, muestra una lista con las reservas hechas por el usuario, que al pulsar revela los detalles de la reserva.

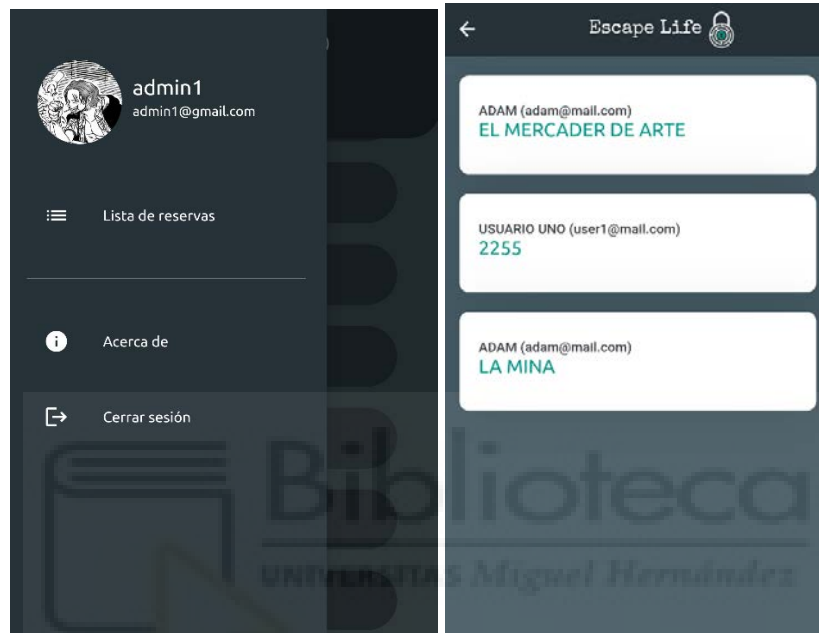


Imagen 97. Menú hamburguesa para administradores

Imagen 98. Lista de reservas para administradores

Imagen 99. Datos de reserva

ANEXO II

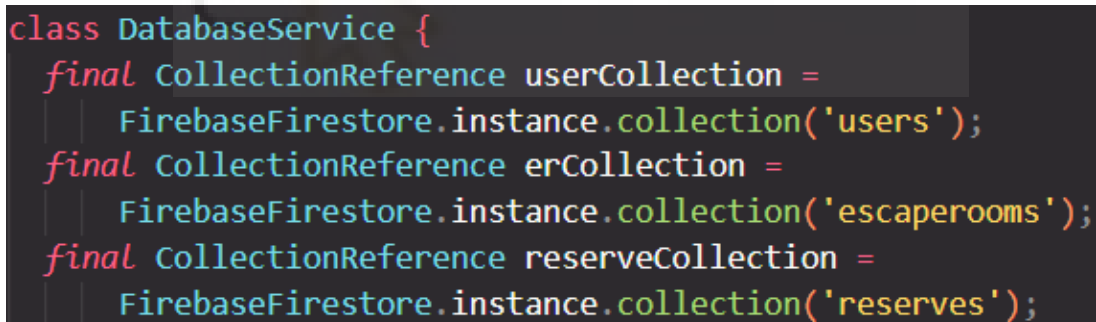
En este anexo se van a explorar algunas de las funcionalidades que tienen cierto grado de complejidad y son piezas clave de la aplicación.

La primera funcionalidad de la que se va a hablar son los Providers.

Para la recogida de datos desde Firebase se han utilizado interfaces para el usuario, las escaperooms y las reservas. En este ejemplo se va a mostrar la recogida de datos del usuario.

Para empezar en la clase DatabaseService, que hace de servicio al que acudir para hacer uso de las funciones que se comunican con la base de datos.

Como se muestra en la siguiente imagen se declaran tres variables del tipo CollectionReference que guardan la referencia de cada una de las colecciones que se van a utilizar en la aplicación.



```
class DatabaseService {  
    final CollectionReference userCollection =  
        FirebaseFirestore.instance.collection('users');  
    final CollectionReference erCollection =  
        FirebaseFirestore.instance.collection('escaperooms');  
    final CollectionReference reserveCollection =  
        FirebaseFirestore.instance.collection('reserves');
```

Imagen 100. Declaración de referencias a las colecciones

El usuario como se ha comentado anteriormente, tiene una serie de campos como el uid, el nombre, el correo, etc. que se usan para identificar a los usuarios en el ranking o en las reservas, por ejemplo.

Por lo tanto, se crea una interfaz para darle forma a esta estructura y junto a esto un método de tipo “factory” que devuelve una instancia de la clase pero

que no necesariamente crea una sino que hace uso de alguna instancia creada previamente.

En esta se recoge un parámetro de tipo DocumentSnapshot con datos de la base de datos y asignándoles a la instancia de un usuario.

```
class Usuario {
    String id;
    String uid;
    String nombre;
    String email;
    String urlImagen;
    String empresa;
    List<String> opcionesSeleccionadas;
    List<String> favoritas;
    List<String> completadas;
    bool isAdmin;
    Usuario({
        @required this.id,
        @required this.nombre,
        @required this.email,
        this.urlImagen,
        this.opcionesSeleccionadas,
        this.favoritas,
        this.completadas,
        this.isAdmin,
        this.empresa,
        this.uid,
    });

    factory Usuario.fromFirestore(DocumentSnapshot doc) {
        Map data = doc.data();

        return Usuario(
            uid: doc.id,
            nombre: data['nombre'] ?? '',
            email: data['email'] ?? '',
            id: data['id'] ?? 0,
            urlImagen: data['urlImagen'] ?? '',
            empresa: data['empresa'] ?? '',
            isAdmin: data['isAdmin'] ?? '',
            opcionesSeleccionadas: data['opcionesSeleccionadas'].cast<String>() ?? [],
            favoritas: data['favoritas'].cast<String>() ?? [],
            completadas: data['completadas'].cast<String>() ?? [],
        );
    }
}
```

Imagen 101. Interfaz de la clase usuario

En este método de tipo get se hace uso de la referencia de la colección, de usuarios en este caso, y se mapean los datos para devolverlos en forma de lista. Se utiliza un tipo Stream para que se puedan ver reflejados los cambios en tiempo real.

```
Stream<List<Usuario>> get usuarios {  
    return userCollection.snapshots().map(  
        (list) => list.docs.map((doc) => Usuario.fromFirestore(doc)).toList());  
}
```

Imagen 102. Función get usuarios

En el fichero main se utiliza un MultiProvider para tener acceso a todos los datos en los contextos que se ramifican. Por lo tanto, se añaden StreamProviders que cargan los datos y los cambios en tiempo real.

Como se puede ver en el StreamProvider se llama al método get de DatabaseService, que recoge la lista de usuarios, escaperooms y reservas.

También se ha de explicar que hay un StreamProvider que recoge un valor desde AuthService, que equivale al usuario que tiene la sesión iniciada.

```
return MultiProvider(  
    providers: [  
        StreamProvider<List<Escaperoom>>.value(  
            initialData: null,  
            value: DatabaseService().escaperooms,  
        ), // StreamProvider.value  
        StreamProvider<List<Usuario>>.value(  
            initialData: null,  
            value: DatabaseService().usuarios,  
        ), // StreamProvider.value  
        StreamProvider<List<Reserve>>.value(  
            initialData: null,  
            value: DatabaseService().reserves,  
        ), // StreamProvider.value  
        StreamProvider<Usuario>.value(  
            initialData: null,  
            value: AuthService().user,  
        ), // StreamProvider.value  
    ],
```

Imagen 103. Declaración de multiproviders en main

Por último, en la siguiente imagen se muestra la asignación a variables de los datos que se han ido recogiendo durante todo el proceso y que se utilizarán para mostrar datos en pantalla del usuario, de las escaperooms y las reservas.

Provider recoge el tipo de dato que se especifique añadiendo el contexto, ya que solo tendrá acceso a los datos que se recojan en niveles superiores del árbol de widgets.

Hay también en la imagen una función setState, en la que utilizando el uid del usuario con la sesión iniciada, recogida desde AuthService, se filtra por id en la lista de usuarios ya que la información recogida por el servicio de autenticación no tiene todos los datos del usuario.

```
Usuario user;
final escaperooms = Provider.of<List<Escaperoom>>(context);
final usuarios = Provider.of<List<Usuario>>(context, listen: false) ?? [];
final usuario = Provider.of<Usuario>(context, listen: false);
setState(() {
  user =
    usuarios.singleWhere((i) => i.id == usuario.id, orElse: () => null);
});
```

Imagen 104. Instanciación de variables con los datos de los providers

La siguiente imagen es un ejemplo del uso de la variable user que contiene la instancia del usuario que ha iniciado sesión con todos los datos y en este caso se pretende mostrar el nombre de usuario para la página principal.

```
"Hola ${user.nombre}",
```

Imagen 105. Ejemplo de uso de variable tipo usuario