

Omaha Flops

David Medel Pizarro

Grado de Ingeniería Informática

Desarrollo aplicaciones dispositivos móviles (Android)

Consultores:

David Escuer Latorre

Jordi Almirall López

Profesor responsable de la asignatura:

Carles Garrigues Olivella

Junio/2024



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

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual [3.0 España de Creative Commons](#)



Esta obra está sujeta a una licencia de Reconocimiento [3.0 España de Creative Commons](#)

B) GNU Free Documentation License (GNU FDL)

Copyright © 2024 - David Medel Pizarro.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free

Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

C) Copyright

© (el autor/a)

Reservados todos los derechos. Está prohibido la reproducción total o parcial de esta obra por cualquier medio o procedimiento, comprendidos la impresión, la reprografía, el microfilme, el tratamiento informático o cualquier otro sistema, así como la distribución de ejemplares mediante alquiler y préstamo, sin la autorización escrita del autor o de los límites que autorice la Ley de Propiedad Intelectual.

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Omaha Flops</i>
Nombre del autor:	<i>David Medel Pizarro</i>
Nombre del consultor/a:	<i>David Escuer Latorre Jordi Almirall López</i>
Nombre del PRA:	<i>Carles Garrigues Olivella</i>
Fecha de entrega:	06/2024
Titulación:	<i>Ingeniería informática</i>
Área del Trabajo Final:	<i>Desarrollo aplicaciones dispositivos móviles (Android)</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>Android, póquer, omaha</i>
Resumen del Trabajo:	
<p>Este proyecto consiste en la creación de una aplicación móvil Android para que los usuarios puedan jugar mediante su dispositivo móvil a una variante del póquer Omaha.</p> <p>La aplicación permite el acceso a todos los usuarios registrados previamente. Una vez se ha accedido a la aplicación está permite la visualización de las reglas, creación de una partida, búsqueda de adversarios, mostrar el listado de las diferentes partidas creadas y por último jugar contra el resto de los jugadores de la partida.</p> <p>Dada la complejidad del proyecto se ha optado por seguir una metodología ágil basada en Sprints. Este tipo de metodología permite realizar iteraciones rápidas, recibir retroalimentación continua y garantizar la entrega de las funcionalidades claves.</p> <p>Para el desarrollo de la aplicación se ha utilizado IDE de Android Studio, para el desarrollo de la API se ha utilizado el framework de Django, la base de datos está hecha con MongoDB, para la conexión entre Django y MongoDB se ha utilizado Djongo. Hay que destacar el uso de la librería <i>phevaluator</i> para el cálculo de la puntuación y del ganador de las manos de póquer. Finalmente hay una parte de pruebas manuales en la aplicación móvil y una parte de pruebas automáticas en la API.</p>	

Como conclusión, se ha obtenido una aplicación funcional, aunque no se han implementado el total de funcionalidades previstas. Esto demuestra la dificultad de desarrollar un proyecto de estas características, por simple que parezca.

Abstract:

The aim of this project is the creation of an Android mobile application so that users can play a variant of Omaha poker using their mobile device.

The application allows access to all previously registered users. Once the application has been accessed, it allows you to view the rules, create a game, search for opponents, show the list of the different games created and finally play against the rest of the players in the game.

Given the complexity of the project, it has been decided to follow an agile methodology based on Sprints. This type of methodology allows for rapid iterations, continuous feedback, and guaranteed delivery of key functionalities.

For the development of the application, the Android Studio IDE was used, the Django framework was used for the development of the API, the database was made with MongoDB, and Django was used for the connection between Django and MongoDB. It is worth highlighting the use of the phevulator library to calculate the score and the winner of the poker hands. Finally, there is a part of manual testing in the mobile application and a part of automatic testing in the API.

In conclusion, a functional application has been obtained, although not all the planned functionalities have been implemented. This demonstrates the difficulty of developing a project of these characteristics, no matter how simple it may seem.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	3
1.2.1 Objetivos funcionales.....	3
1.2.2 Objetivos no funcionales.....	4
1.2.3 Objetivos a futuro.....	4
1.3 Enfoque y método seguido.....	5
1.4 Planificación del Trabajo.....	5
1.5 Breve resumen de productos obtenidos.....	9
1.6 Breve descripción de los otros capítulos de la memoria.....	9
2. Diseño centrado en el usuario.....	11
2.1 Usuarios.....	11
2.1.1 Métodos de indagación.....	11
2.1.1.1 Análisis competitivo (Benchmarking).....	11
2.1.1.2 Encuesta.....	14
2.1.2 Ficha de perfil de usuario por cada uno de los diferentes perfiles ..	18
2.2 Diseño conceptual.....	19
2.2.1 Point of view statements.....	19
2.2.2 Flujo de iteración: estructura general de la aplicación.....	20
2.3 Prototipado.....	21
2.3.1 Sketches escaneados: dibujos a mano alzada.....	21
2.3.2 Prototipado con Figma.....	24
2.4. Evaluación.....	28
2.4.1. Tareas del usuario.....	28
2.4.2 Documento con el feedback de cada tarea.....	30
2.4.3 Listado de puntos de mejora.....	33
3. Diseño técnico.....	35
3.1. Casos de uso.....	35
3.1.1 Diagrama UML para identificar gráficamente actores y flujo.....	35
3.1.2 Listado de casos de uso donde se especifican actores precondiciones, flujo y post condiciones.....	36
3.2. Diseño de la arquitectura.....	43
3.2.1 Diagrama UML de la base de datos.....	43
3.2.2 Diagrama UML del diseño de las entidades y clases.....	44
3.2.3 Diagrama explicativo de la arquitectura del sistema.....	45
4. Implementación.....	46
4.1 Herramientas de trabajo.....	47
4.1.1 Lenguajes de programación.....	47
4.1.2 Base de datos.....	47
4.1.3 Frameworks.....	47
4.1.4 IDEs.....	47
4.1.5 Librerías y APIs.....	47
4.1.6 Otros.....	48
4.2 Patrones y buenas prácticas.....	49
4.3 Desarrollo de la aplicación.....	50

4.4 Estado actual del proyecto	52
5. <i>Testing</i> y pruebas de usuario	53
5.1 Frontend (pruebas manuales)	53
5.1.1 Sign up.....	53
5.1.2 Login	57
5.1.3 Main menu	59
5.1.4 Rules.....	60
5.1.5 Create game	61
5.1.6 InGame	63
5.2 Backend (pruebas automáticas).....	64
6. Conclusiones.....	65
7. Glosario	66
8. Bibliografía	68
9. Anexos	72
9.1 Resultado encuestas.....	72

Lista de figuras

Figura 1. Posición inicial de la partida	2
Figura 2. Posición una vez el jugador ha ordenado sus cartas	2
Figura 3. Revelación de las cartas y lectura de las jugadas.....	3
Figura 4. Tabla del diagrama de Gantt.....	7
Figura 5. Diagrama de Gantt.....	8
Figura 6. PokerStars. Ejemplo de partida. Recuperado de https://pokerstarslearn.online/wp-content/uploads/2023/11/release-notes-vi.jpg	11
Figura 7. 888poker. Ejemplo de partida multitable. Recuperado de https://pokerindustrypro.com/site_media/media/uploads/news/888poker-table-app-android-wm.png	12
Figura 8. Winamax. Ejemplo de Partida. Recuperado de https://operator-front-static-cdn.winamax.fr/img/es/content/betting/2023/20230206_screen_newlogo/20230206_guia-de-uso_7.jpg	12
Figura 9. Pineapple App. Ejemplo de menú	13
Figura 10. Pineapple App. Ejemplo de partida	13
Figura 11. Gráfico encuesta edad	14
Figura 12. Gráfico encuesta sexo.....	14
Figura 13. Gráfico encuesta juego cartas móvil	15
Figura 14. Gráfico encuesta jugar a póquer	15
Figura 15. Gráfico encuesta jugar a Omaha.....	16
Figura 16. Gráfico encuesta jugar a variante de Omaha	16
Figura 17. Gráfico encuesta jugar a póquer en móvil.....	16
Figura 18. Gráfico encuesta orientación pantalla	17
Figura 19. Gráfico encuesta elegir tema cartas.....	17
Figura 20. Gráfico encuesta elegir tema mesa.....	17
Figura 21. Gráfico encuesta valores en una aplicación.....	18
Figura 22. Tabla encuesta aspectos a mejorar	18
Figura 23. Flujo de iteración.....	21
Figura 24. Boceto de la aplicación parte 1	22
Figura 25. Boceto de la aplicación parte 2	23
Figura 26. Página de carga	24
Figura 27. Página de acceso.....	24
Figura 28. Página registrarse	24
Figura 29. Página olvidar contraseña.....	24
Figura 30. Página menú principal.....	25
Figura 31. Página reglas	25
Figura 32. Página perfil usuario.....	25
Figura 33. Página selección parte delantera baraja	25
Figura 34. Página selección parte trasera baraja	26
Figura 35. Página selección fondo	26
Figura 36. Página crear nuevo juego	26
Figura 37. Página juego con 1 jugador, fase 1	26
Figura 38. Página juego con 1 jugador, fase 2.....	27
Figura 39. Página juego con 1 jugador, fase 3.....	27
Figura 40. Página juego con 2 jugadores, fase 1	27

Figura 41. Página juego con 2 jugadores, fase 2	27
Figura 42. Página juego con 2 jugadores, fase 3	28
Figura 43. Diagrama UML con actores y flujo	35
Figura 44. Diagrama UML de la base de datos	43
Figura 45. Diagrama UML de las entidades y clases	44
Figura 46. Diagrama de la arquitectura del sistema	45
Figura 47. Versiones y compatibilidad de los diferentes dispositivos	46
Figura 48. Estructura Android.....	50
Figura 49. Estructura Django.....	51
Figura 50. Sign up	53
Figura 51. Sign up nombre requerido	53
Figura 52. Sign up email válido requerido	53
Figura 53. Sign up email inválido	54
Figura 54. Sign up contraseña requerida	54
Figura 55. Sign up contraseña oculta para el usuario	54
Figura 56. Sign up botón de visualizar la contraseña	55
Figura 57. Sign up contraseñas no coincidentes.....	55
Figura 58. Sign up exitoso.....	55
Figura 59. Usuario creado satisfactoriamente en la base de datos.....	55
Figura 60. Sign up usuario ya existente	56
Figura 61. Sign up email ya existente	56
Figura 62. Sign up nuevo usuario creado correctamente	56
Figura 63. Nuevo usuario creado satisfactoriamente	56
Figura 64. Sign up via google en futura versión	57
Figura 65. Login via google en futura versión.....	58
Figura 66. Login username obligatorio	58
Figura 67. Login contraseña obligatoria	58
Figura 68. Login usuario o contraseña incorrectos.....	58
Figura 69. Login usuario o contraseña incorrectos.....	58
Figura 70. Login satisfactorio	58
Figura 71. Menú principal sin juegos creados	59
Figura 72. Menú principal con dos juegos creados	59
Figura 73. Reglas del juego.....	60
Figura 74. Inicialización de juego, tabla games, tabla players, tabla sets y tabla points.....	62
Figura 75. Inicialización de juego, tabla players_hands y tabla boards.....	62
Figura 76. Ingame contra un jugador	63
Figura 77. Ingame coger y soltar carta	63
Figura 78. Ingame contra dos jugadores.....	63
Figura 79. Cartas juego 2, jugador 1, primera mano	63
Figura 80. Ejemplo de test de url.....	64
Figura 81. Ejemplo de test de vista	64
Figura 82. Ejecución de los tests via terminal	64

1. Introducción

1.1 Contexto y justificación del Trabajo

El póquer es un juego de cartas con apuesta en el que confluyen tanto la suerte como la habilidad estratégica de los jugadores. Aunque sus orígenes están en civilizaciones antiguas como China y Persia, su evolución lo ha convertido en un fenómeno global, presente en varios contextos, desde casinos y salas de juego hasta plataformas en línea y reuniones sociales.

Podemos encontrar una amplia gama de variantes de este juego, cada una con sus propias reglas, dinámicas y estrategias. Quizás el más conocido sea el clásico *Texas Hold'em*, aunque existen otras variantes como *Omaha*, *Omaha High/Low*, *Seven Stud*, entre otros. Mi proyecto se centra en la variante del *Omaha*, pero con unas reglas personalizadas.

Aunque hoy hay muchas aplicaciones y casas de apuesta que ofrecen el servicio de poder jugar a póquer con otros jugadores del mundo. Dado que no se puede competir con estas empresas, surge la necesidad de innovar y ofrecer nuevas experiencias a los aficionados de este mundillo. Es aquí donde esta aplicación móvil entra en escena, como una reinterpretación del clásico *Omaha*.

El Omaha es una variante del póquer en que cada jugador tiene 4 cartas en la mano y habrá 5 cartas comunitarias, obligatoriamente la mejor mano se elige entre 2 cartas de tu mano y 3 de las comunitarias y se rige por las jugadas clásicas del póquer (escalera real > escalera de color > póquer > full > color > escalera > trio > doble pareja > pareja > carta alta). Hay 4 momentos en que hay acción de apuestas: *preflop* (una vez te han repartido las 4 cartas y no hay ninguna carta comunitaria), *flop* (una vez se han visto las 3 primeras cartas comunitarias), *turn* (cuando se muestra la cuarta carta comunitaria) y *river* (cuando se destapa la última carta comunitaria). Normalmente el Omaha se juega *Pot limit* (con las apuestas limitadas al contenido del bote).

Esta nueva variante del juego consiste en repartir no 4, si no 12 cartas a cada jugador (máximo 3 jugadores) y habrá 3 juegos de 5 cartas comunitarias. Inicialmente cada jugador verá sus 12 cartas y el *flop* de cada uno de los juegos. Cada jugador dividirá sus cartas en grupos de 4 para jugar en cada juego. Una vez divididas, se mostrarán las últimas dos cartas comunitarias de cada juego y se procederá con el ganador siguiendo las normas del *Omaha*. De cada uno de los juegos (3 juegos) se utilizan obligatoriamente 2 cartas (de las 4) de la mano y 3 (de las 5) de las comunitarias, eligiendo automáticamente la mejor mano entre los participantes.

A continuación, se muestra un ejemplo del procedimiento con imágenes:

En la primera imagen podemos observar cómo se han repartido 12 cartas al jugador y aparecen 3 juegos diferentes con el *flop* descubierto, faltando por mostrar las dos últimas cartas.

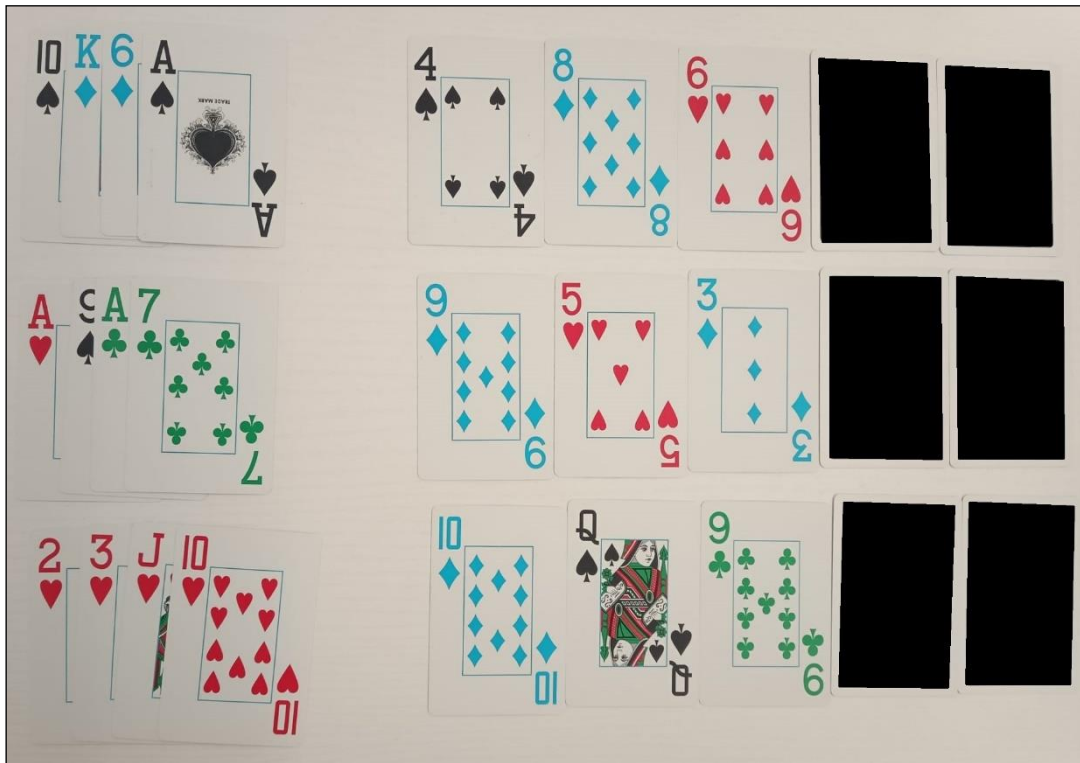


Figura 1. Posición inicial de la partida

En la siguiente imagen podemos observar como el jugador ha reordenado sus cartas.

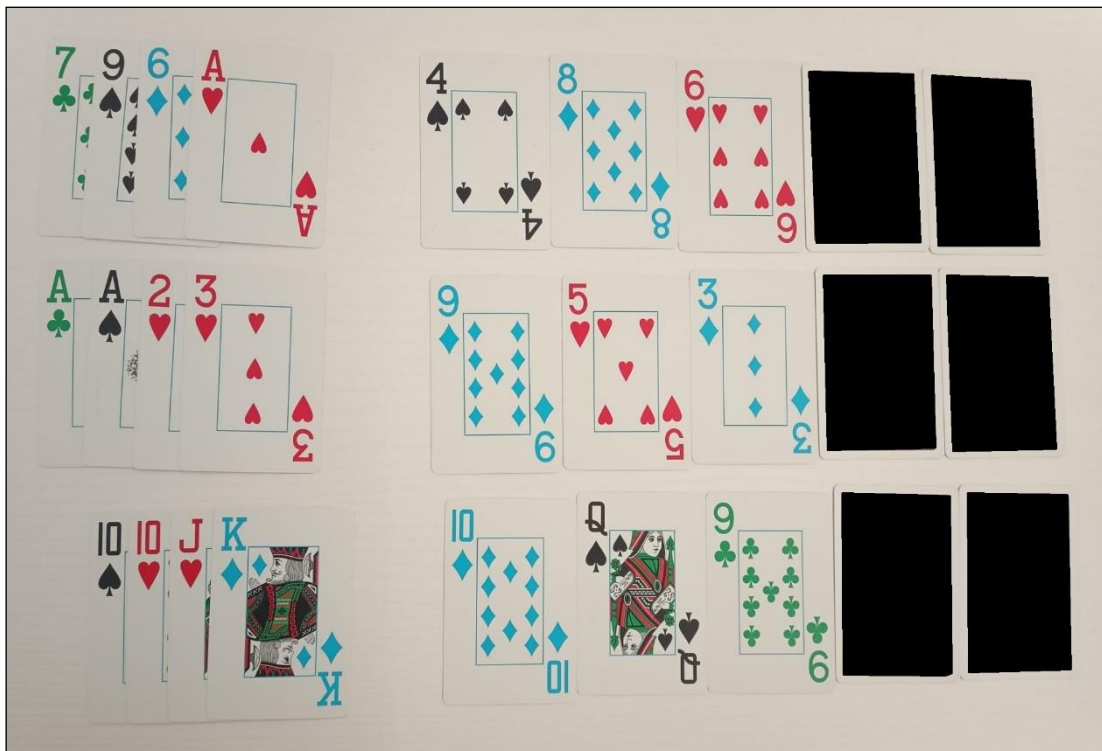


Figura 2. Posición una vez el jugador ha ordenado sus cartas

Por último, la siguiente imagen revela las últimas dos cartas y se procede con la lectura de jugadas, teniendo en cuenta las reglas del *Omaha*. Por ejemplo, en esta mano el jugador tiene pareja de 6 en el primer juego, doble pareja de 3s 2s en el segundo juego y full de 10s 2s en el último juego. Posteriormente, se compara la jugada de cada juego con el resto de los jugadores y se declara al ganador.

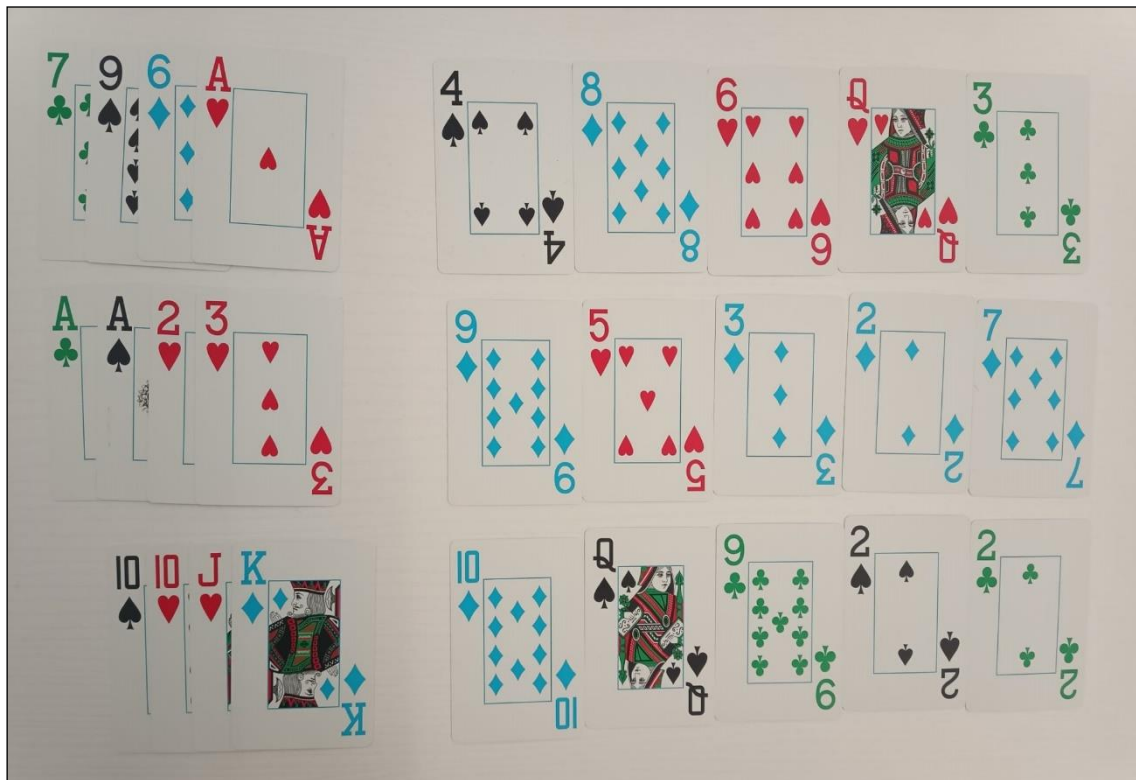


Figura 3. Revelación de las cartas y lectura de las jugadas

Con esta aplicación, por tanto, se busca que el usuario pueda pasárselo bien jugando a cartas con otras personas, sin necesidad de una baraja de cartas y sin tener en cuenta la parte económica, ya que esta aplicación únicamente tiene el objetivo de entretener al usuario y que éste disfrute de una nueva experiencia de juego, con la reinención de un clásico del póquer.

1.2 Objetivos del Trabajo

El objetivo general del proyecto es el desarrollo de una aplicación Android que permita entretener a los usuarios con esta variante del Omaha que tiene reglas personalizadas y ofrecer así, una experiencia única de juego.

1.2.1 Objetivos funcionales

OF1 – Implementar un sistema de gestión de usuarios que permita el acceso a la aplicación, utilizando servicios externos como el Login con cuenta de Google.

OF2 – Permitir a los usuarios crear y editar sus perfiles, guardando información relevante como: el nombre, la foto de perfil, la mejor mano, las partidas jugadas, las partidas ganadas y las partidas perdidas.

OF3 – Integrar una función para agregar amigos, donde cada usuario pueda mantener una lista de contactos.

OF4 – Desarrollar una función para poder crear partidas contra uno o dos amigos.

OF5 – Incluir la opción de crear partidas en línea con otros jugadores.

OF6 – Implementar un sistema de gestión de datos que permita almacenar información de las partidas creadas y su estado.

OF7 - Añadir funciones de seguridad para proteger los datos de los usuarios frente a posibles ataques cibernéticos.

OF8 – Incorporar un sistema de notificaciones para informar a los usuarios sobre invitaciones de amigos, turnos en las partidas y otras actualizaciones importantes.

OF9 – Incluir una opción para visualizar las reglas de este juego.

OF10 - Añadir la opción de activar/desactivar el sonido del juego tanto de efectos como música de fondo.

1.2.2 Objetivos no funcionales

ONF1 – Garantizar una interfaz intuitiva y simple.

ONF2 – Garantizar la estabilidad y fiabilidad de la aplicación mediante pruebas exhaustivas de calidad y corrección de errores.

1.2.3 Objetivos a futuro

Por falta de tiempo y conocimiento en el campo, estos recursos se harán si el resto de los objetivos, tanto funcionales como no funcionales, se han realizado.

OFU1 – Integrar un sistema de chat para que los jugadores puedan comunicarse durante las partidas.

OFU2 – Desarrollar un algoritmo de inteligencia artificial para permitir jugar contra la computadora en modo offline.

OFU3 – Incluir opciones de personalización avanzadas para las mesas de juego, como la selección de fondos, temas y estilos de cartas. En función del número de partidas ganadas.

ONFU1 – Asegurar la compatibilidad con la máxima variedad de dispositivos Android y tamaños de pantalla, desde teléfonos pequeños hasta tabletas grandes.

1.3 Enfoque y método seguido

Dado que no hay ninguna aplicación actualmente que cumpla con el propósito del proyecto, se ha decidido partir de cero en la creación de la aplicación. Esto brindará la oportunidad de diseñar y construir cada aspecto del producto según las necesidades y requisitos de nuestro proyecto.

Aprovecharé la experiencia personal como crupier que he adquirido durante estos años, para garantizar que la aplicación refleje con precisión las dinámicas y reglas de este nuevo juego.

La idea de esta variante del Omaha surge de la experiencia personal de jugar con amigos, lo que asegura que se tenga en cuenta la diversión y la interacción social en el diseño de la aplicación.

Se han recopilado y considerado comentarios y sugerencias de compañeros y jugadores de póquer para mejorar y refinar la experiencia de juego de la aplicación.

Para el diseño de la aplicación se realizará un análisis exhaustivo de aplicaciones de póquer ya existentes para identificar características exitosas, así como áreas de mejora, que puedan inspirar el desarrollo de la aplicación.

Además, se utilizará una metodología ágil basada en *sprints* para gestionar el desarrollo de la aplicación de manera eficiente y adaptable a los cambios. Esto permitirá realizar iteraciones rápidas, recibir retroalimentación continua y garantizar la entrega oportuna de funcionalidades clave.

1.4 Planificación del Trabajo

Los recursos necesarios para poder realizar el trabajo son los siguientes:

- **Acceso a Internet:** Se requiere acceso a Internet para realizar investigaciones, acceder a recursos de programación, servicios externos, levantar y alojar la parte del *backend* de la aplicación.

- **Hardware Adecuado:** Es importante contar con un equipo informático adecuado que permita ejecutar el software de desarrollo de manera eficiente.
- **Software Específico:** A continuación, se cita el software con el que se llevará a cabo el desarrollo de la aplicación:
 - **Android Studio:** Un entorno de desarrollo integrado (IDE) utilizado para desarrollar aplicaciones Android, que será utilizado para el *frontend* de la aplicación.
 - **Kotlin:** Un lenguaje de programación moderno y compatible con Java, que será utilizado para escribir el código del *frontend* de la aplicación.
 - **Python con Django:** Un *framework* de desarrollo web que será utilizado para crear el *backend* de la aplicación y gestionar la lógica del servidor.
 - **MongoDB:** Sistema de gestión de bases de datos NoSQL, que almacenará y gestionará los datos de la aplicación.
- **Documentación y Tutoriales:** Dada la naturaleza del proyecto, será importante tener acceso a documentación técnica, tutoriales y recursos de aprendizaje para familiarizarse con las tecnologías utilizadas y resolver problemas durante el desarrollo de la aplicación.

Aparte de los recursos mencionados anteriormente, también es imprescindible los recursos personales, en este caso la persona que lleva a cabo el desarrollo del proyecto, un desarrollador con los conocimientos y habilidades adecuados para poder desarrollar tanto el *frontend* como el *backend*.

Para la realización del proyecto se ha estimado una dedicación media de unas 3 horas al día, todos los días de la semana. Se ha decidido esta estimación teniendo en cuenta que no se trabajara todos los días y que algunos días se dedicarán más horas y otros menos.

A continuación, se muestra la tabla con los datos del diagrama de Gantt y el tiempo aproximado de realización de esa parte del trabajo.

Nombre	Duración (horas)	Inicio	Fin
Proyecto Omaha Flops	363	28/02/2024	27/06/2024
PEC1. Plan de Trabajo	36	28/02/2024	10/03/2024
Contexto y justificación del Trabajo	6	28/02/2024	29/02/2024
Objetivos del Trabajo	6	01/03/2024	02/03/2024
Enfoque y método elegido	6	03/03/2024	04/03/2024
Planificación del Trabajo	6	05/03/2024	06/03/2024
Breve resumen de productos obtenidos	6	07/03/2024	08/03/2024
Breve descripción del resto de capítulos	6	09/03/2024	10/03/2024
App hello world	3	09/03/2024	09/03/2024
PEC2. Diseño y arquitectura	81	12/03/2024	07/04/2024
Diseño Centrado en el Usuario	36	12/03/2024	23/03/2024
Usuarios	9	12/03/2024	14/03/2024
Diseño conceptual	9	15/03/2024	17/03/2024
Prototipado	9	18/03/2024	20/03/2024
Evaluación	9	21/03/2024	23/03/2024
Diseño técnico	42	25/03/2024	07/04/2024
Definición de los casos de uso	21	25/03/2024	31/03/2024
Diseño de la arquitectura	21	01/04/2024	07/04/2024
PEC3. Implementación	126	09/04/2024	20/05/2024
Implementación	96	09/04/2024	10/05/2024
Módulo login	18	09/04/2024	14/04/2024
Módulo perfil usuario	6	15/04/2024	16/04/2024
Módulo agregar amigos	12	17/04/2024	21/04/2024
Módulo crear partida con amigos	15	22/04/2024	27/04/2024
Módulo crear partida con otros jugadores	12	28/04/2024	01/05/2024
Módulo reglas del juego	3	02/05/2024	02/05/2024
Módulo otras opciones	6	03/05/2024	04/05/2024
Módulo de notificaciones	9	05/05/2024	07/05/2024
Segurización	9	08/05/2024	10/05/2024
Testing y pruebas de usuario	30	11/05/2024	20/05/2024
PEC4. Entrega final	60	21/05/2024	09/06/2024
Mejoras y correcciones	15	21/05/2024	25/05/2024
Memoria	15	26/05/2024	30/05/2024
Grabación	3	31/05/2024	31/05/2024
Presentación	6	01/06/2024	02/06/2024
Imprevistos	21	03/06/2024	09/06/2024
Defensa	51	11/06/2024	27/06/2024

Figura 4. Tabla del diagrama de Gantt

En la siguiente página se muestra el diagrama de Gantt con las fechas de entrega aproximadas.

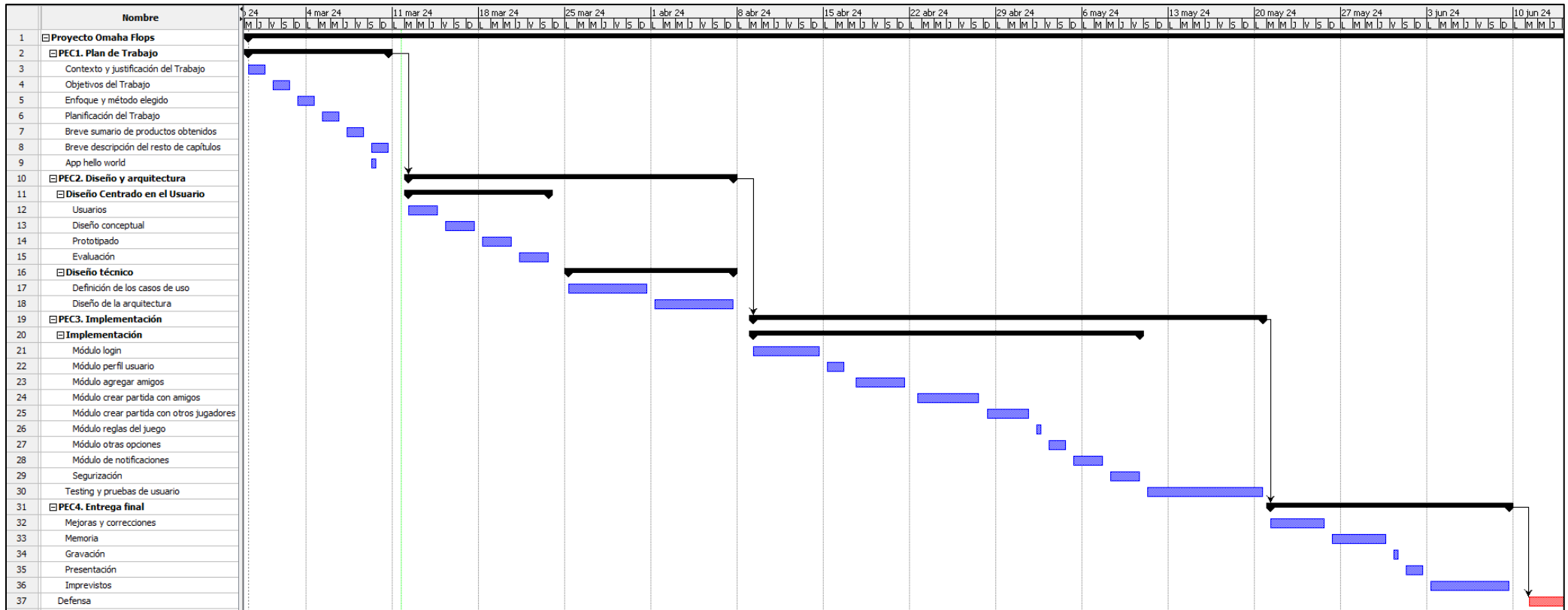


Figura 5. Diagrama de Gantt

1.5 Breve resumen de productos obtenidos

Los productos obtenidos una vez finalizado el proyecto serán los siguientes:

- Archivo de instalación para dispositivos Android con extensión .apk.
- Proyecto completo de Android Studio con el código fuente del *frontend* la aplicación.
- Proyecto completo de Django con el código fuente del *backend* de la aplicación.
- Memoria del proyecto
- Presentación en vídeo.

1.6 Breve descripción de los otros capítulos de la memoria

Capítulo 2. Diseño centrado en el usuario: Dentro de este apartado se detalla el proceso de diseño centrado en el usuario, este proceso se realizará mediante la experiencia previa del autor, además de analizar diferentes aplicaciones ya creadas. Este punto también incluye la creación de prototipos de la interfaz de usuario y el diseño de la experiencia de usuario (UX).

Capítulo 3. Diseño técnico: Este capítulo se centra en la definición de los casos de uso, los aspectos técnicos del desarrollo de la aplicación, incluyendo la arquitectura del sistema, la selección de tecnologías y herramientas, así como la descripción de los componentes y módulos principales de la aplicación.

Capítulo 4. Implementación: En este capítulo se describirá el proceso de implementación de la aplicación, desde la codificación de las funcionalidades principales hasta la integración de los diferentes componentes del sistema.

Capítulo 5. *Testing* y pruebas de usuario: En esta sección se abordará el proceso de *testing* y validación de la aplicación, que incluirá pruebas unitarias, de integración, de sistema y de aceptación. También se describirán las pruebas de usuario realizadas para evaluar la usabilidad, la funcionalidad y la satisfacción del usuario con la aplicación.

Capítulo 6. Conclusiones: Este capítulo se resumirá los principales hallazgos y conclusiones del trabajo realizado, incluyendo los logros alcanzados, los desafíos enfrentados y las lecciones aprendidas durante el desarrollo de la aplicación.

Capítulo 7. Glosario: Este punto incluye las definiciones de términos técnicos y conceptos relevantes utilizados en la memoria para facilitar la comprensión del lector y asegurar una interpretación precisa del contenido.

Capítulo 8. Bibliografía: Lista de las fuentes bibliográficas, artículos, documentos y recursos consultados durante la investigación y el desarrollo del proyecto

Capítulo 9. Anexos: El anexo incluirá el resto de los apartados demasiado extensos para incluirse dentro de la memoria y con carácter autocontenido.

2. Diseño centrado en el usuario

2.1 Usuarios

2.1.1 Métodos de indagación

Al ser un juego de cartas nuevo, me he decantado por los siguientes métodos de indagación.

2.1.1.1 Análisis competitivo (Benchmarking)

He analizado cuidadosamente algunas de las diferentes aplicaciones de póquer móvil para poder identificar sus fortalezas y debilidades. Esto me ha ayudado a comprender que características son populares y cómo puedo hacer para diferenciar mi aplicación de la competencia.

La mayoría de las aplicaciones de póquer pertenecen a una empresa de apuestas, esto implica que haya dinero de por en medio, lo que las hace ideales para las personas que busquen lucrarse a través de la aplicación, en cambio mi aplicación está enfocada en el hecho de poder jugar, pasarlo bien y disfrutar de una nueva experiencia de juego.

Lo que si me ha servido de analizar las diferentes aplicaciones es como estructurar los diferentes componentes, límites de tamaño de las cartas, posición del móvil, diferentes colores, entre otros.

A continuación, se muestran capturas de pantalla de algunas de las aplicaciones que he analizado.

Pokerstars:



Figura 6. PokerStars. Ejemplo de partida. Recuperado de <https://pokerstarslearn.online/wp-content/uploads/2023/11/release-notes-vi.jpg>

Todas estas aplicaciones tienes más o menos los siguientes puntos positivos y negativos:

Positivo: Ofrece versatilidad, multiplataforma (PC, Tablet, móvil), también puedes jugar indiferentemente de la orientación de la pantalla, cambiar color al tapete, diferentes variantes de póquer, otro tipo de apuestas, jugar dinero real, entre muchas otras cosas.

Negativo: Comisiones altas, limitaciones para jugar sin dinero real.

Encuestas en línea: He creado un formulario de Google para recopilar datos sobre posibles usuarios de mi aplicación. Me he centrado en la búsqueda de usuarios que estén relacionados con el ámbito del póquer, aunque no he dejado de banda al resto de usuarios. En esta encuesta he mezclado tanto preguntas de respuesta cerrada como preguntas de respuesta abierta para conocer con más profundidad las necesidades de los usuarios.

Otra aplicación para tener en cuenta y en la que será referente en el momento de diseñar y programar mi aplicación es Pineapple, una aplicación de póquer chino que no gestiona pagos y únicamente sirve para divertirse.

Positivo: Me gusta como está enfocada, el diseño, el juego en sí.

Negativo: La aplicación, al menos la versión de Android falla mucho.

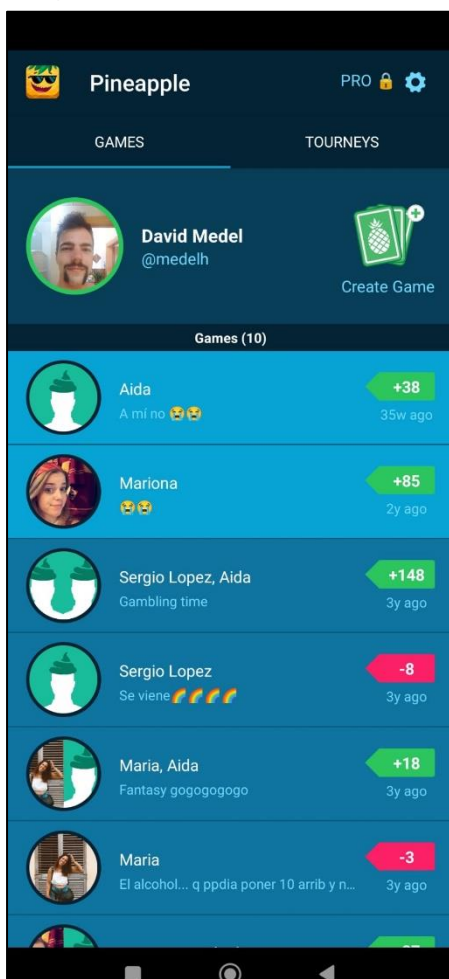


Figura 9. Pineapple App. Ejemplo de menú

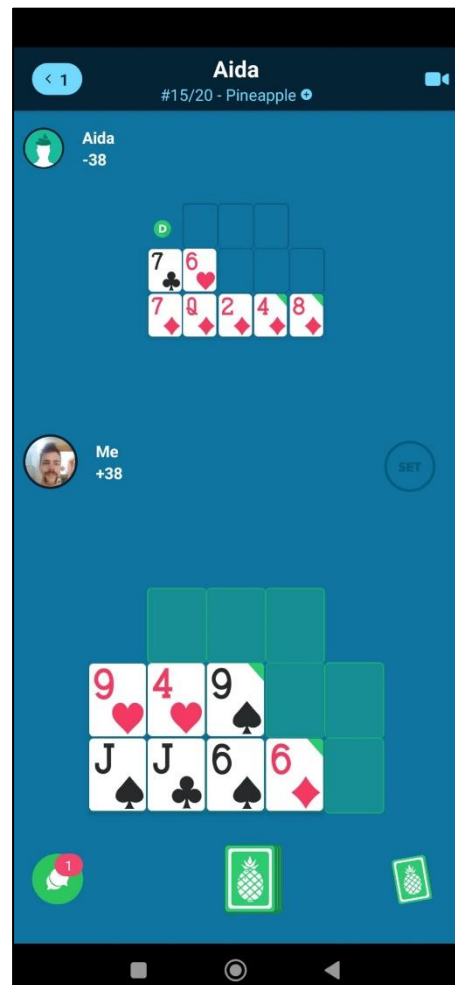


Figura 10. Pineapple App. Ejemplo de partida

2.1.1.2 Encuesta

He procedido a realizar un cuestionario online mediante la plataforma que ofrece Google Form. En este cuestionario se hacía preguntas personales relacionadas con la edad, sexo, demografía. También se han hecho preguntas relacionadas con el conocimiento de póquer que tenía la persona que contestaba el cuestionario. Posteriormente hay preguntas relacionadas con posibles funcionalidades que le gustaría que ofreciera la aplicación. Por último, se añade una pregunta de respuesta abierta para recopilar comentarios adicionales de los entrevistados.

Enlace a la encuesta:

https://docs.google.com/forms/d/e/1FAIpQLSfIRXmZcwrHvRhNRosar2-zSCNNCmNI5O2LSqkL_qQaEa28nw/viewform

Podemos sacar algunos datos de esta encuesta.

Aproximadamente el 90% de los encuestados tiene entre 18 y 40 años.

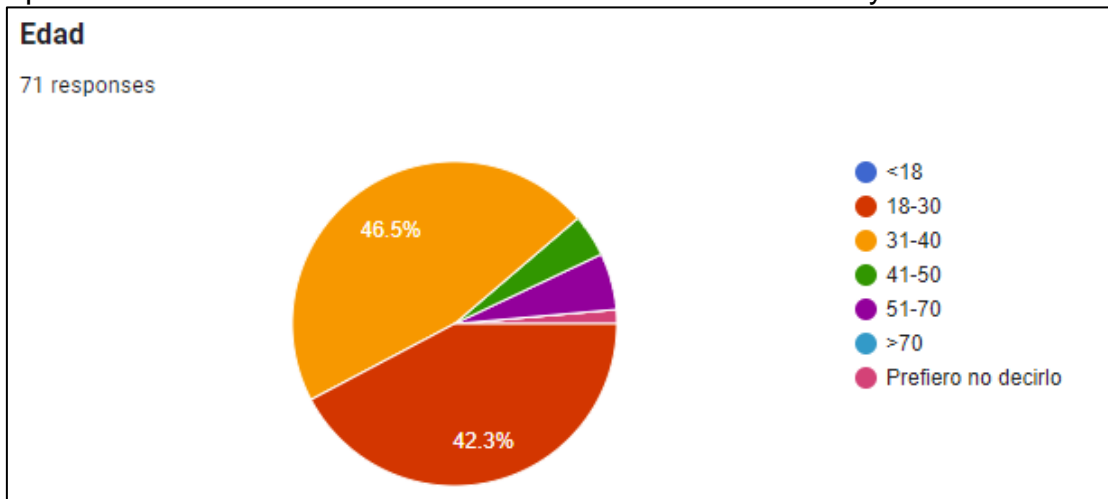


Figura 11. Gráfico encuesta edad

El sexo no ha sido un factor

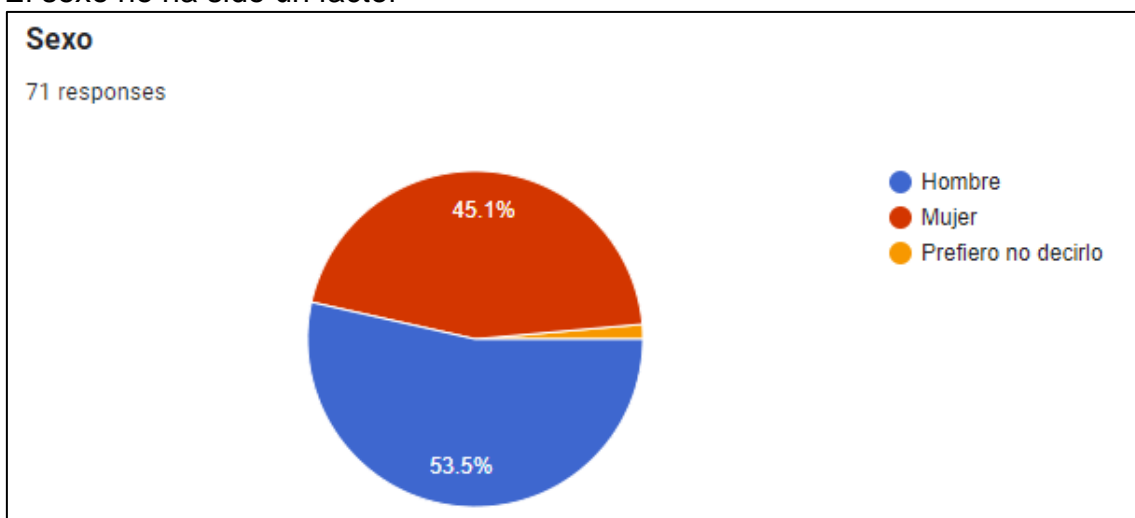


Figura 12. Gráfico encuesta sexo

Al haber hecho las preguntas a personas de mi entorno, es normal que más del 95% de los encuestados sea de origen español.

También se han hecho preguntas relacionadas para poder determinar diferentes perfiles de encuestados, como por ejemplo si la persona tenía experiencia previa con juegos de cartas en el móvil, si había jugado alguna vez a póquer, si conocía el póquer Omaha, si le gustaría probar una nueva variante de este anterior, si juega a póquer en el móvil.



Figura 13. Gráfico encuesta juego cartas móvil



Figura 14. Gráfico encuesta jugar a póquer

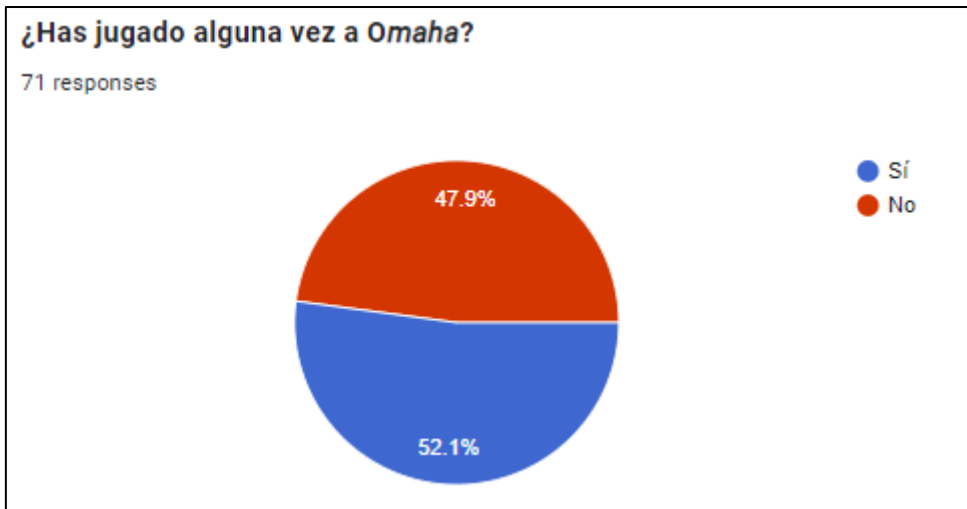


Figura 15. Gráfico encuesta jugar a Omaha

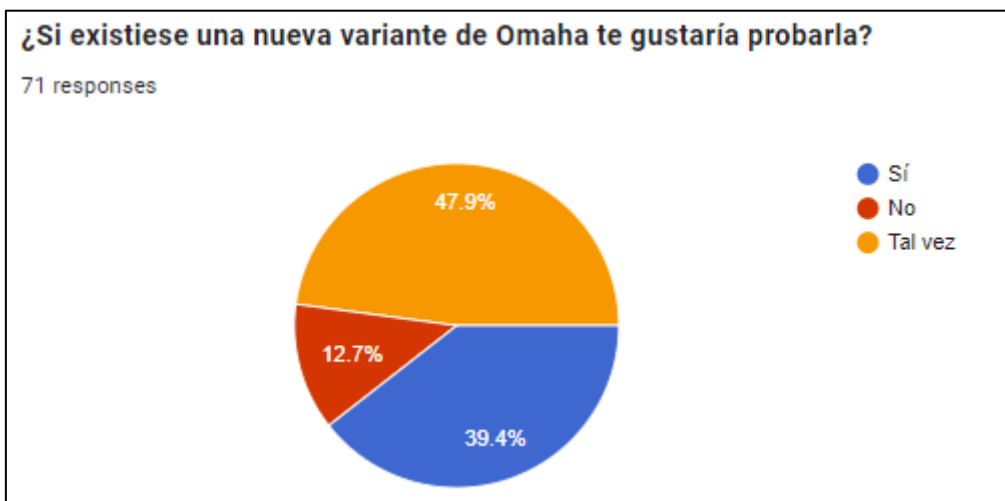


Figura 16. Gráfico encuesta jugar a variante de Omaha

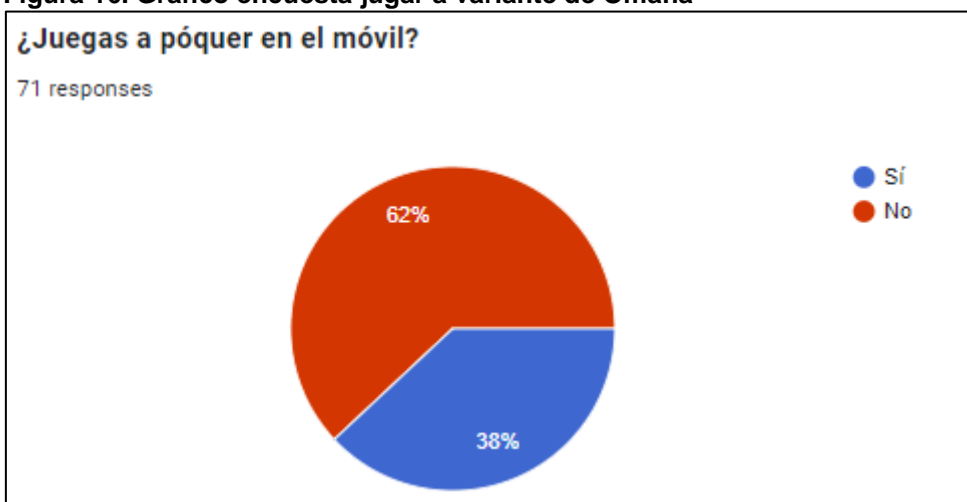


Figura 17. Gráfico encuesta jugar a póquer en móvil

A continuación, se ha procedido con preguntas un poco más relevantes relacionadas con el desarrollo de la aplicación que se llevará a cabo. La orientación de la pantalla, si le gustaría poder elegir el tema de las cartas y la mesa.

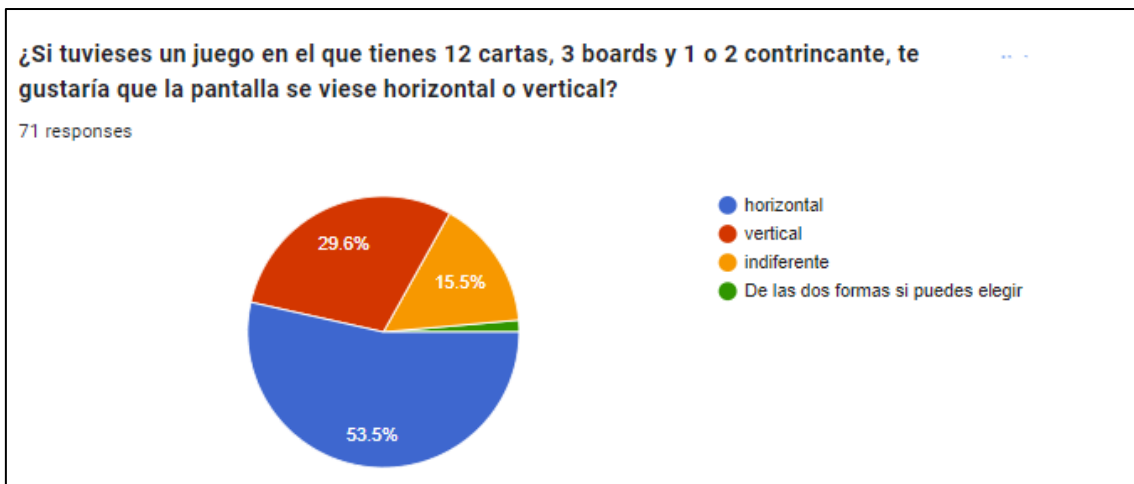


Figura 18. Gráfico encuesta orientación pantalla

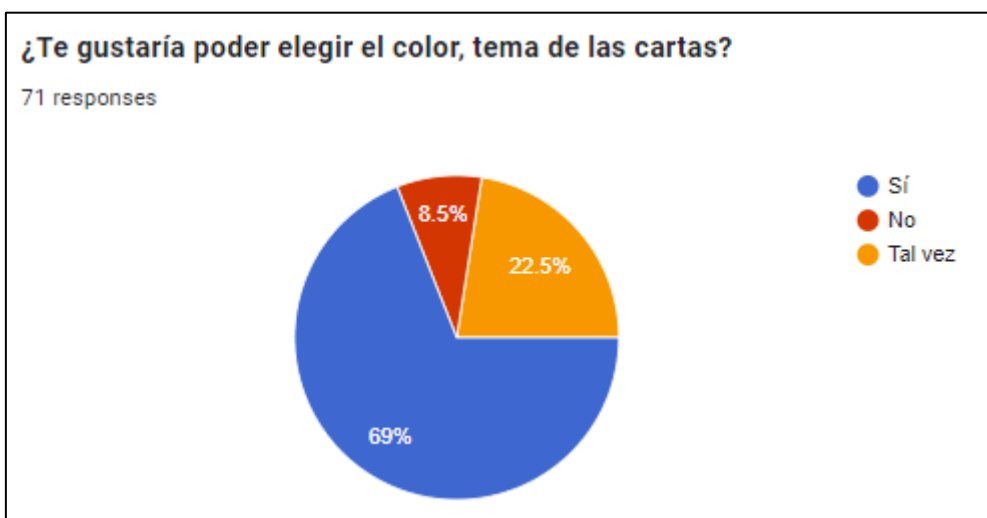


Figura 19. Gráfico encuesta elegir tema cartas

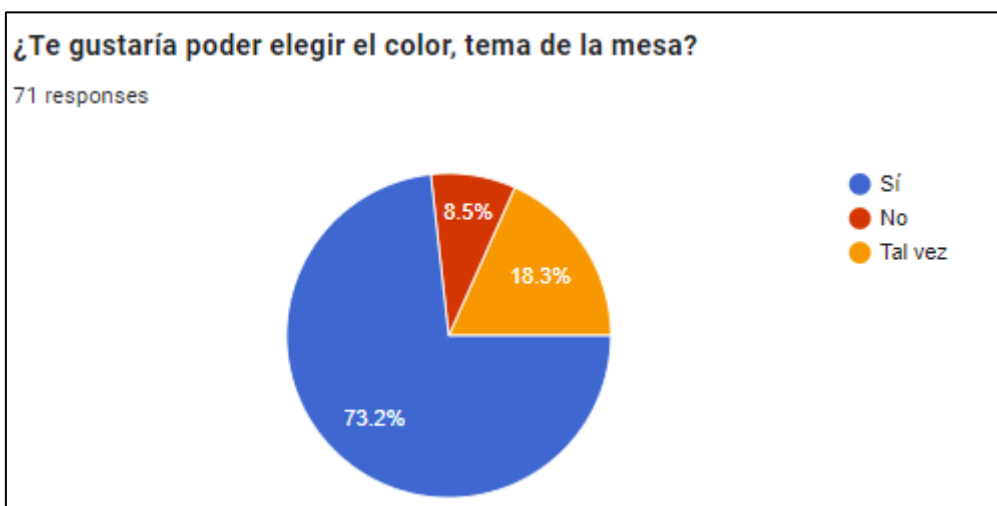


Figura 20. Gráfico encuesta elegir tema mesa

Por último, se ha procedido con preguntas de respuesta abierta que me han servido para conocer ventajas y desventajas de aplicaciones ya existentes. Estas últimas preguntas me han transmitido mucha información respecto las personas a las que se les ha realizado la encuesta.

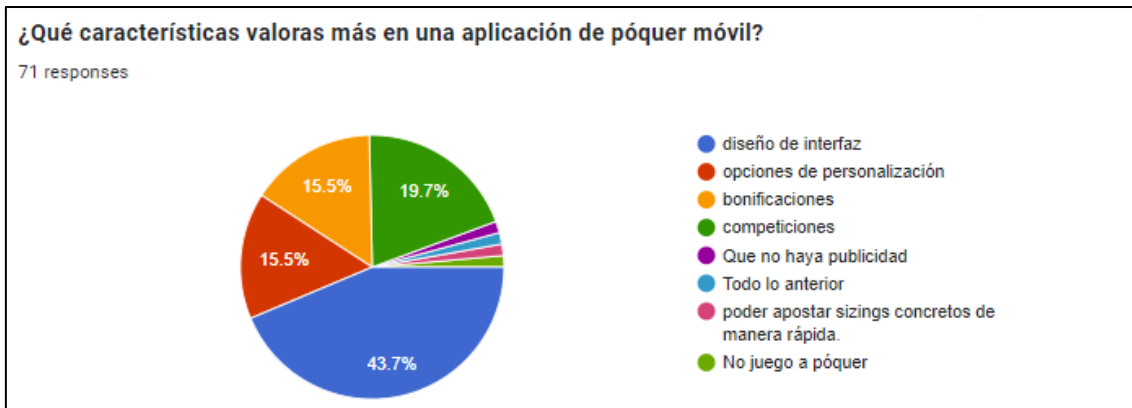


Figura 21. Gráfico encuesta valores en una aplicación

¿Qué aspectos te gustaría mejorar o cambiar en las aplicaciones de póker móvil que has utilizado anteriormente?

rake back mayor
Exceso de publicidad
Personalización y conservación de datos, lo que hace el pío solver por ejemplo
Precisión a la hora de seleccionar la apuesta
Ver al rival
Diseño App
Todas en vertical. Ejemplo app Texas Hold'em de Apple.
Dinamismo en el juego
Los anuncios
Incorporar doble click en acción para evitar error
La adaptabilidad al móvil y facilitar la personalización (show in BBs, sizing bets...)
Poder cambiar la configuración de la mesa sin tener que abandonarla y que a la vez me permita seguir en la partida (no sea complicado reanudar el juego post modificar la configuración)

Figura 22. Tabla encuesta aspectos a mejorar

Como conclusión de este apartado me he dado cuenta de que las encuestas han sido productivas, ya que he podido conocer diferentes puntos de vista de los usuarios, deficiencias y posibles mejoras en relación con otras aplicaciones ya existentes.

2.1.2 Ficha de perfil de usuario por cada uno de los diferentes perfiles

En función de las diferentes respuestas recibidas en la encuesta anterior, he agrupado a los usuarios en 3 perfiles diferentes, cabe mencionar que, dado que la edad predominante es entre 18 y 40, han contestado tanto hombres como mujeres bastante equitativamente y la mayoría de gente que ha respondido la encuesta son hispanos, no podemos clasificar demográficamente los diferentes perfiles.

Perfil: Usuarios con experiencia en póquer online

- Han jugado o juegan póquer en línea regularmente.
- Tienen experiencia en el uso de aplicaciones de póquer móvil.

- Pueden estar familiarizados con una variedad de variantes de póquer, incluida el Omaha.
- Pueden tener preferencias específicas sobre la interfaz y las características de las aplicaciones de póquer móvil, valorando la accesibilidad, la funcionalidad y la seguridad.

Perfil: Usuarios con experiencia en póquer

- Han jugado o juegan póquer.
- Pueden estar familiarizados con una variedad de variantes de póquer, incluida el Omaha.
- Pueden tener opiniones específicas sobre las características y funcionalidades deseadas en una aplicación de póquer móvil.
- Este perfil incluye el perfil anterior, por tanto, este grupo puede tener experiencia o no utilizando aplicaciones de póquer móvil.

Perfil: Usuarios sin experiencia en póquer

- No han jugado o tienen poca experiencia en póquer.
- Pueden estar interesados en probar el juego por primera vez.
- Pueden tener menos familiaridad con las reglas, en concreto del Omaha.
- Es posible que estén más abiertos a explorar diferentes variantes del juego.
- Pueden estar buscando una experiencia de juego fácil de entender y usar, además de que sea gratis en una aplicación de póquer móvil.

2.2 Diseño conceptual

2.2.1 Point of view statements

Perfil: Usuarios con experiencia en póquer online

- Las personas con experiencia en póquer online necesitan una forma de acceder rápidamente a diferentes variantes de póquer en sus aplicaciones móviles porque valoran la diversidad y la variedad de opciones para mantener su interés.
- Las personas con experiencia en póquer online necesitan una forma de personalizar la configuración de la interfaz en las aplicaciones móviles de póquer porque desean adaptar la experiencia de juego a sus preferencias individuales y hábitos de juego.
- Las personas con experiencia en póquer online necesitan una forma de personalizar su experiencia de juego y recibir bonificaciones adicionales porque desean una plataforma que se adapte a sus preferencias individuales y les recompense por su lealtad.

Perfil: Usuarios con experiencia en póquer

- Las personas con experiencia en póquer necesitan una forma de personalizar la configuración de la interfaz y las opciones de juego en las aplicaciones móviles de póquer porque desean una experiencia de juego adaptada a sus preferencias individuales y estilo de juego.

- Las personas con experiencia en póquer necesitan una forma de acceder a estadísticas y análisis detallados sobre su juego dentro de las aplicaciones móviles de póquer porque desean mejorar su rendimiento y tomar decisiones más informadas durante las partidas.
- Las personas con experiencia en póquer necesitan una forma de conectarse con amigos y compañeros de juego a través de funciones sociales integradas en las aplicaciones móviles de póquer porque valoran la interacción con los rivales durante sus sesiones de juego.

Perfil: Usuarios sin experiencia en póquer

- Las personas sin experiencia en póquer necesitan una forma de acceder a tutoriales y guías introductorias en las aplicaciones móviles de póquer porque desean aprender las reglas básicas y las estrategias del juego antes de comenzar a jugar.
- Las personas sin experiencia en póquer necesitan una forma de acceder a una interfaz de usuario intuitiva y fácil de entender en las aplicaciones móviles de póquer porque valoran una experiencia de juego sin complicaciones y sin barreras de entrada.
- Las personas sin experiencia en póquer necesitan una forma de recibir sugerencias y consejos útiles durante el juego dentro de las aplicaciones móviles de póquer porque desean aprender y mejorar mientras juegan.

2.2.2 Flujo de iteración: estructura general de la aplicación

En base a los datos obtenidos y a la idea inicial que tenía de la aplicación, presento un flujo de iteración de las principales funciones de la aplicación. A medida que vaya desarrollando el proyecto, iré adaptando el flujo a las diferentes necesidades que vaya encontrando.

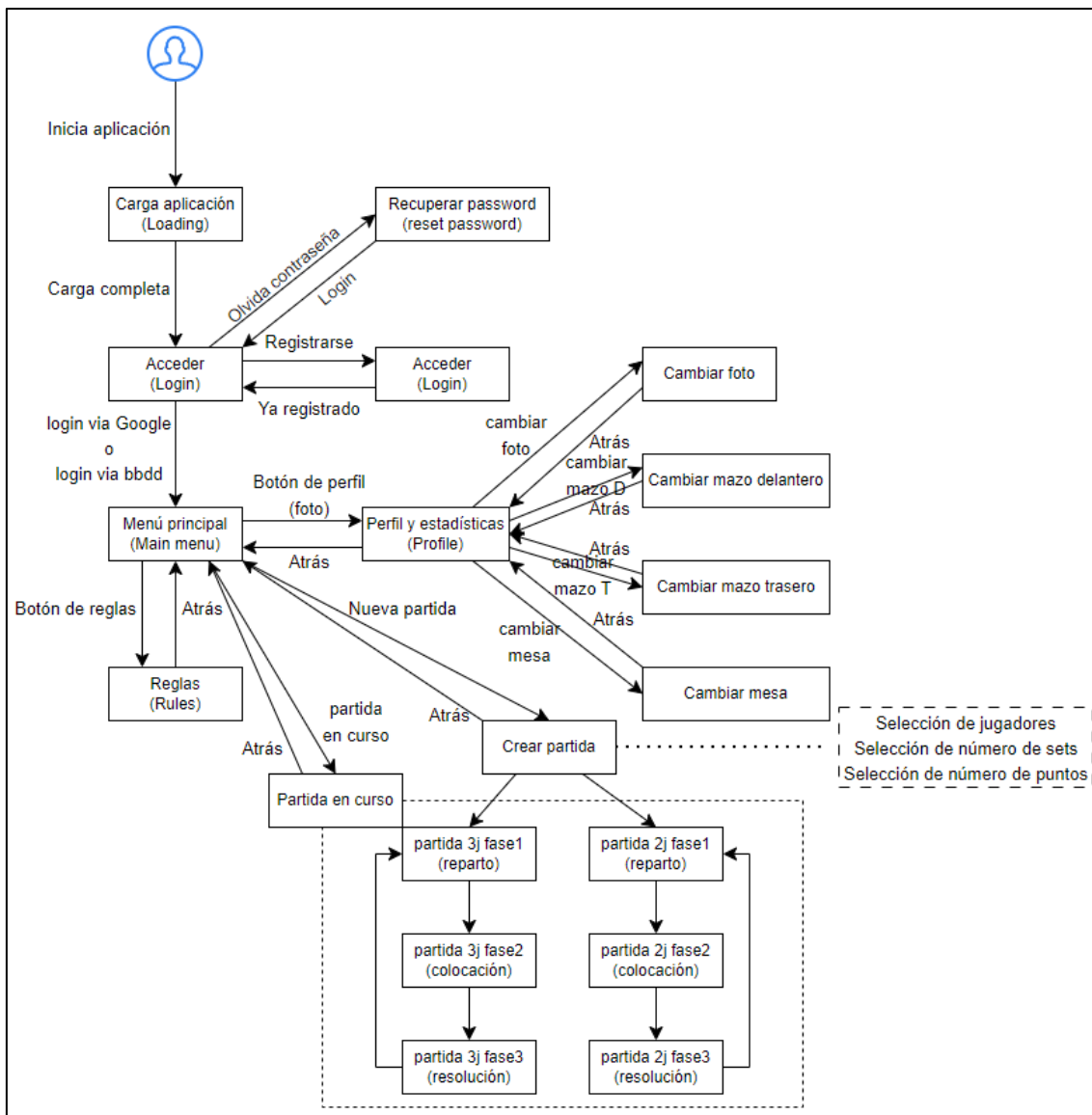


Figura 23. Flujo de iteración

2.3 Prototipado

2.3.1 Sketches escaneados: dibujos a mano alzada

He diseñado un primer boceto de lo que sería la aplicación y la interacción entre las diferentes pantallas.

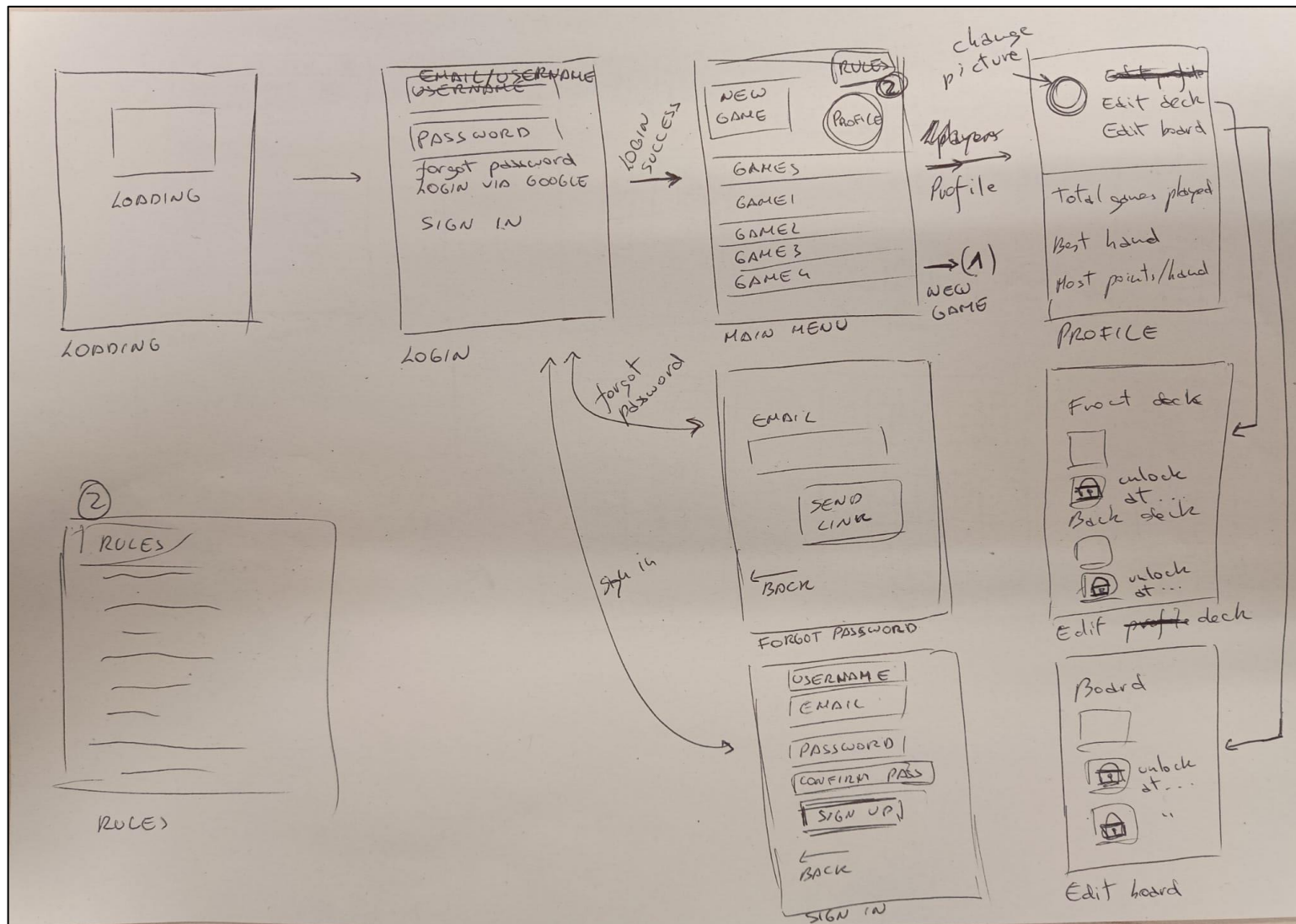


Figura 24. Boceto de la aplicación parte 1

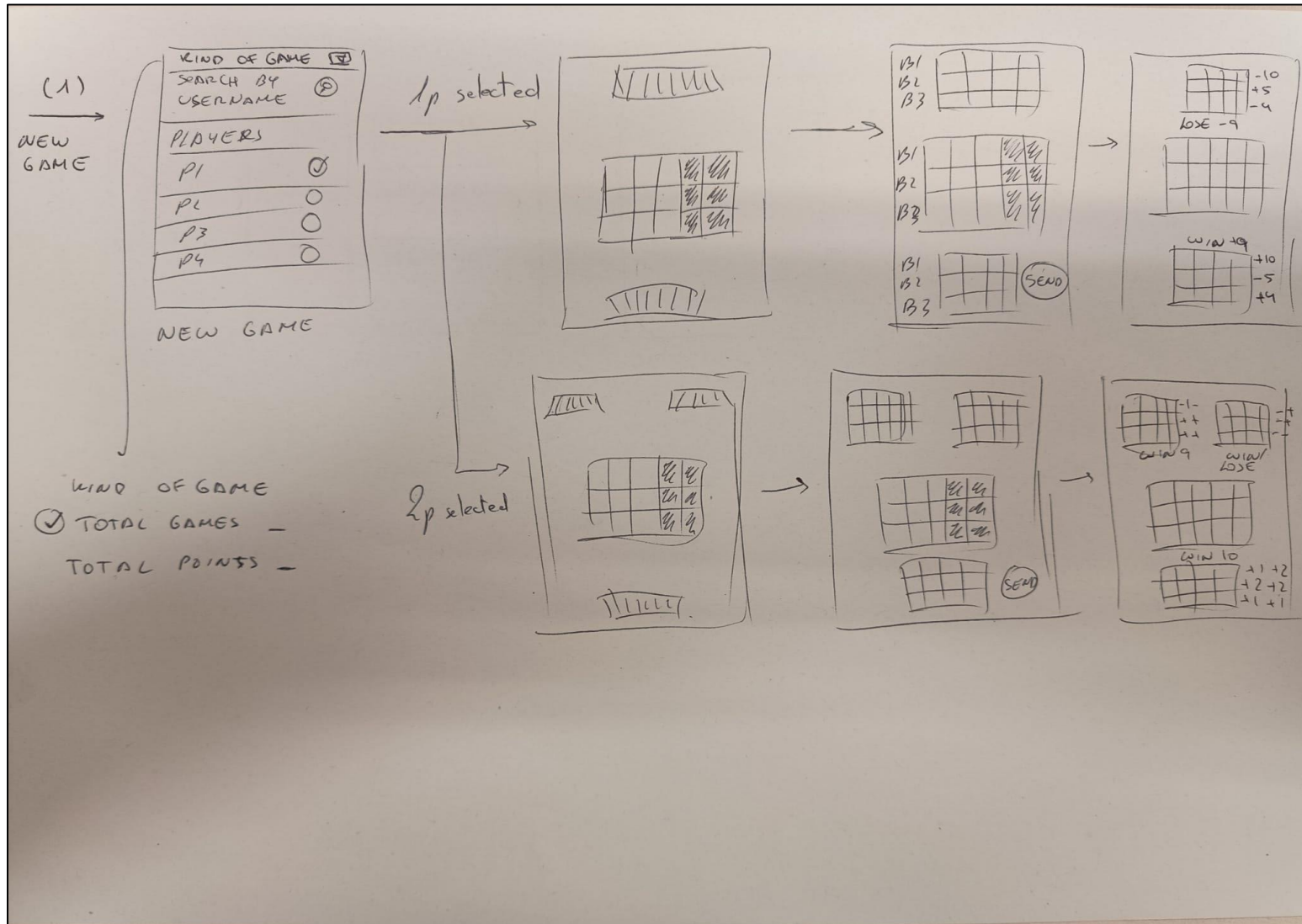


Figura 25. Boceto de la aplicación parte 2

2.3.2 Prototipado con Figma

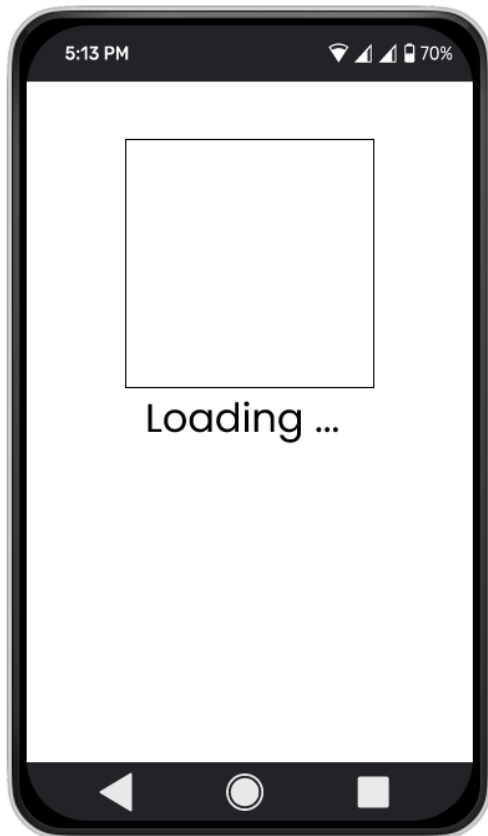


Figura 26. Página de carga

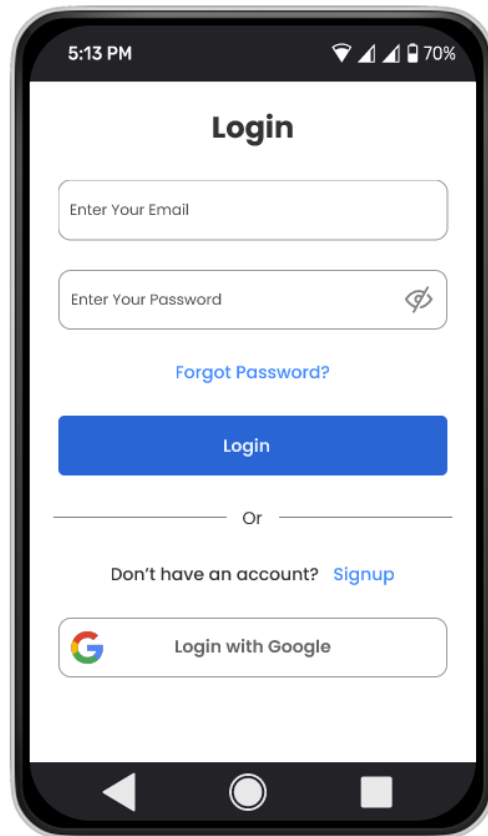


Figura 27. Página de acceso

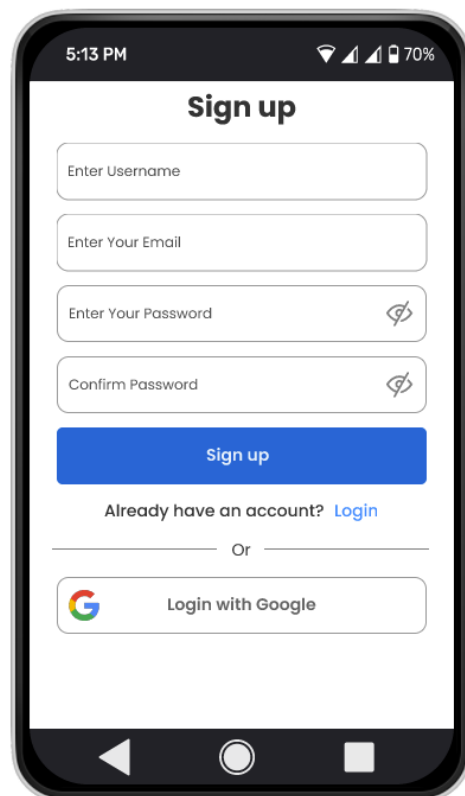


Figura 28. Página registrarse

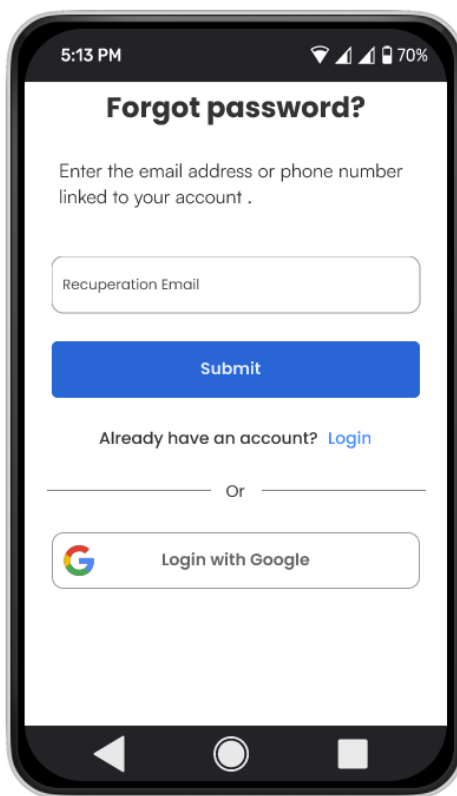


Figura 29. Página olvidar contraseña

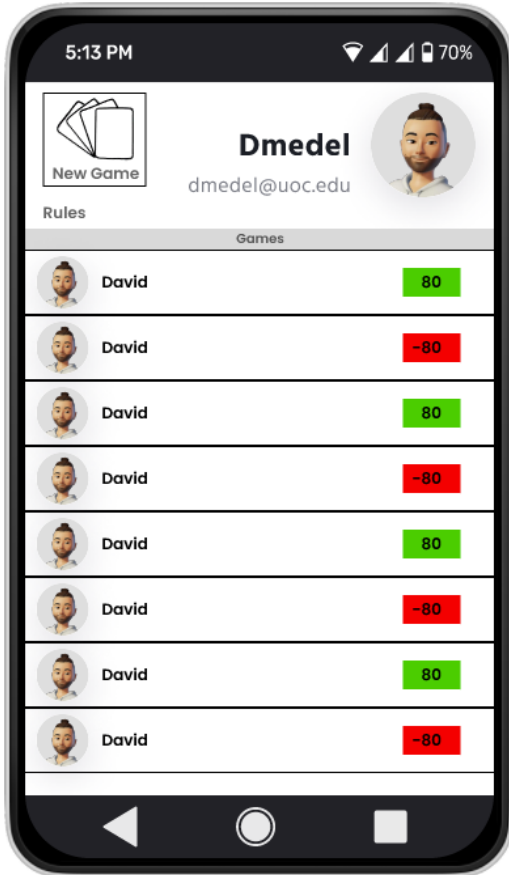


Figura 30. Página menú principal

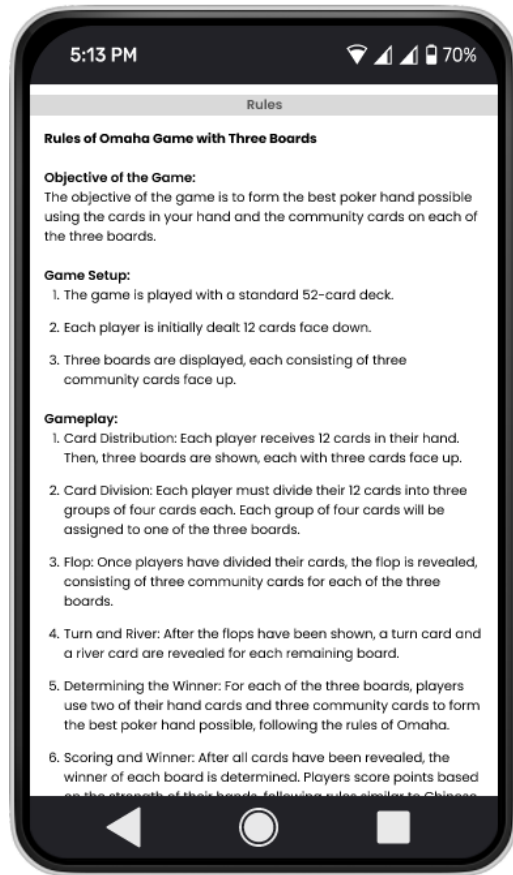


Figura 31. Página reglas

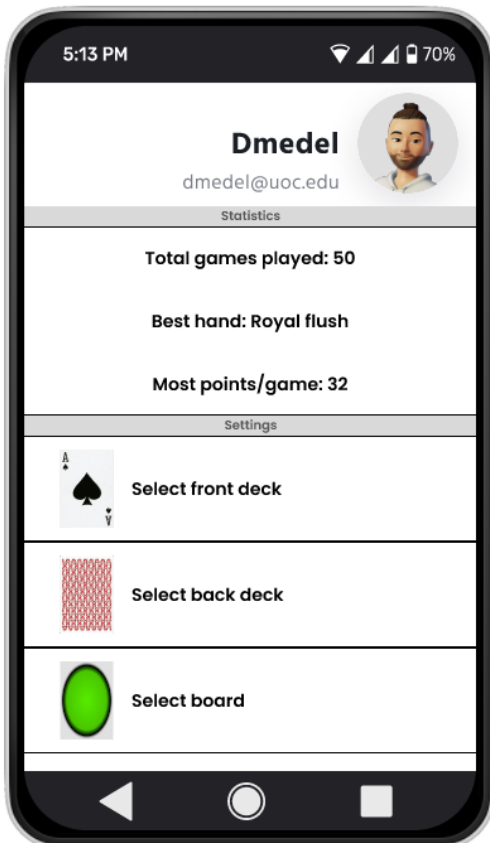


Figura 32. Página perfil usuario



Figura 33. Página selección parte delantera baraja

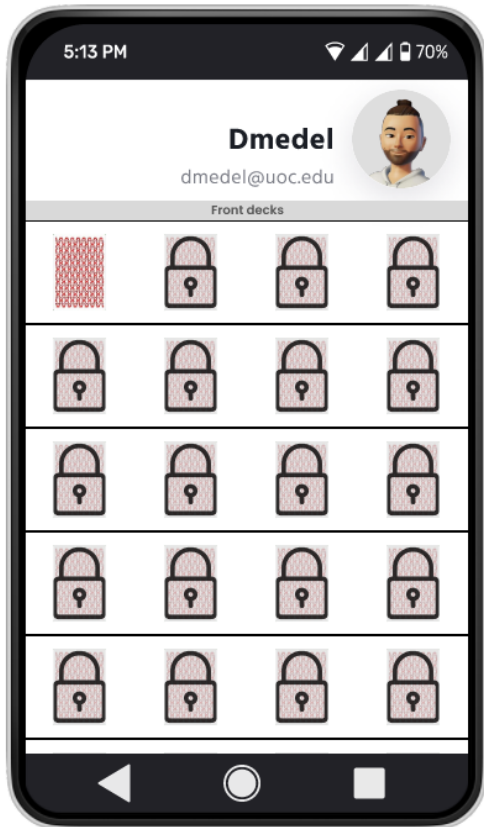


Figura 34. Página selección parte trasera baraja

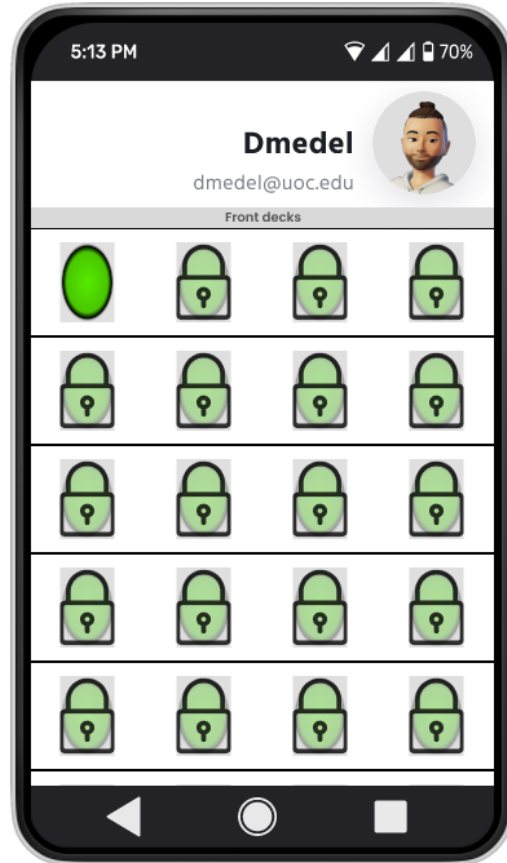


Figura 35. Página selección fondo

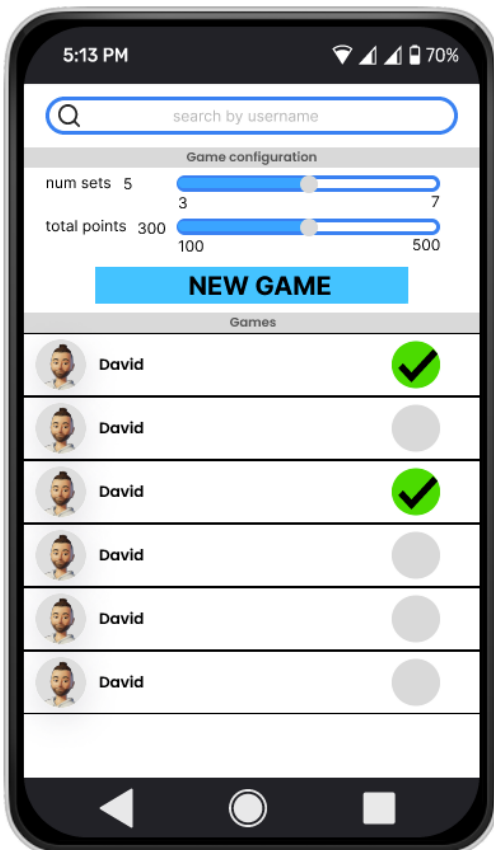


Figura 36. Página crear nuevo juego

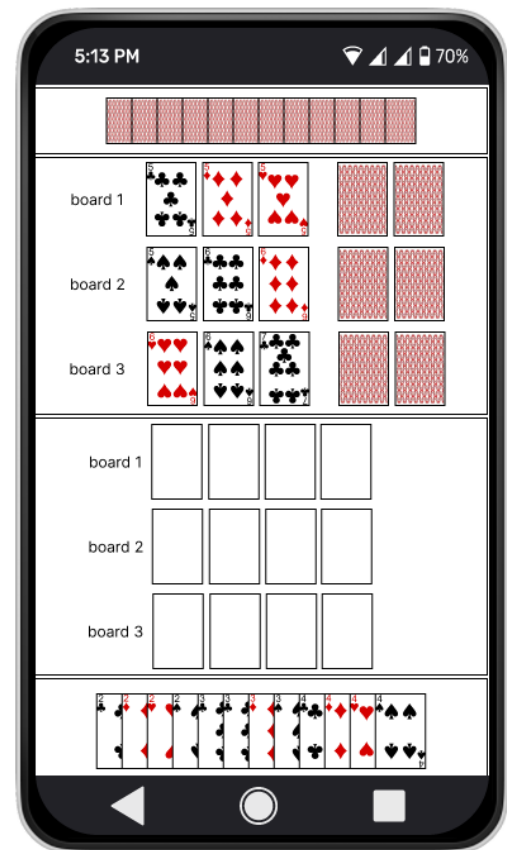


Figura 37. Página juego con 1 jugador, fase 1

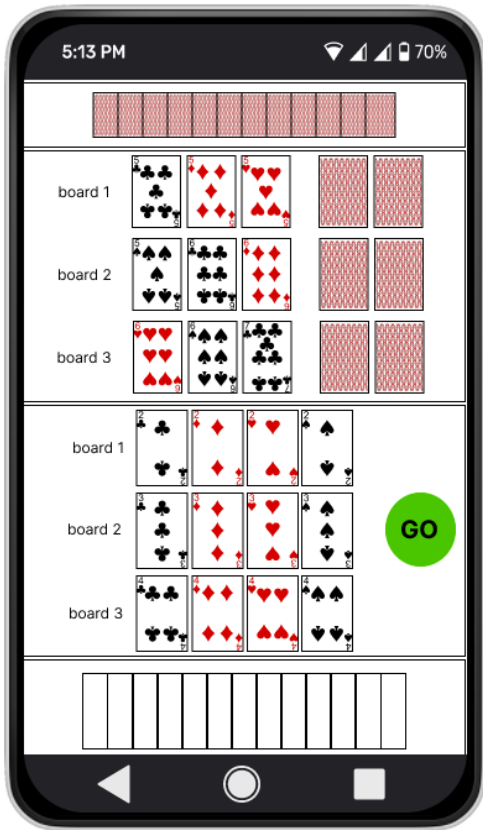


Figura 38. Página juego con 1 jugador, fase 2

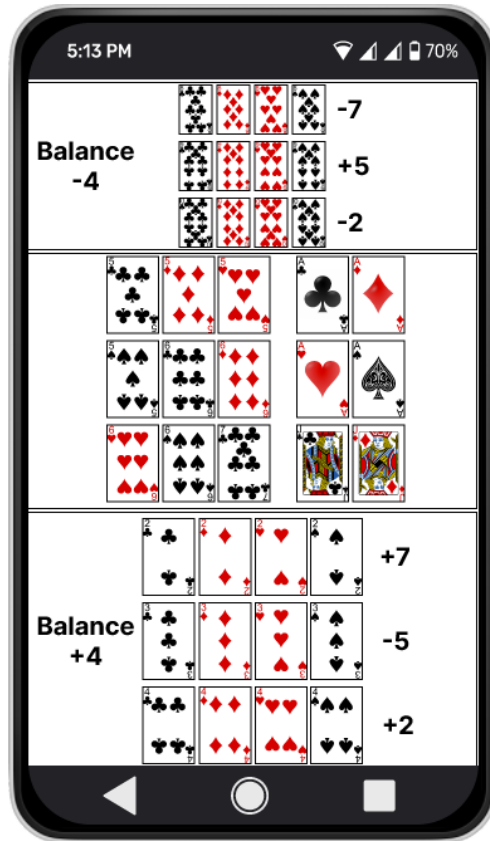


Figura 39. Página juego con 1 jugador, fase 3

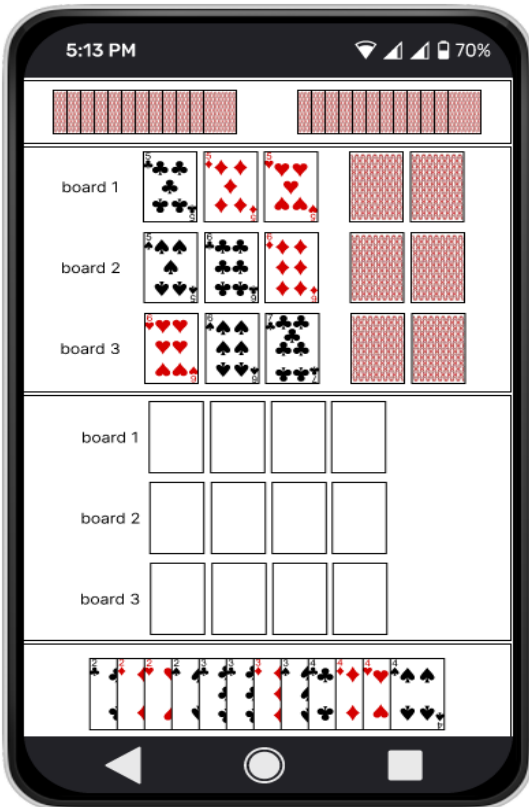


Figura 40. Página juego con 2 jugadores, fase 1

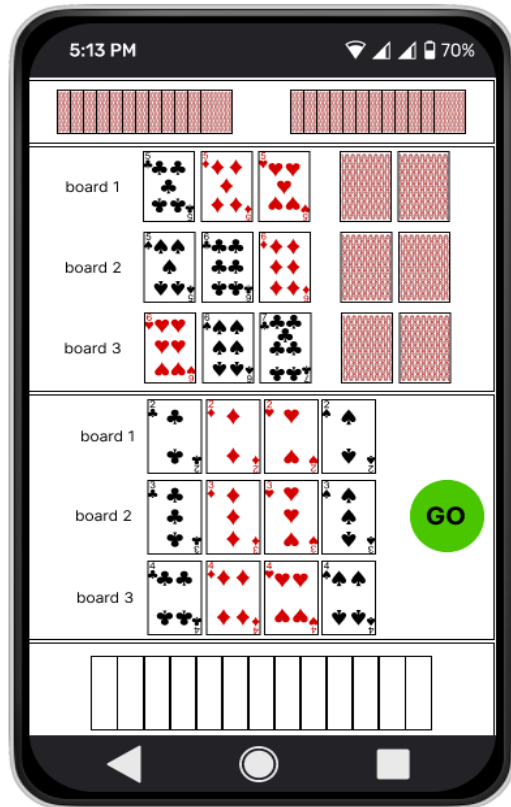


Figura 41. Página juego con 2 jugadores, fase 2

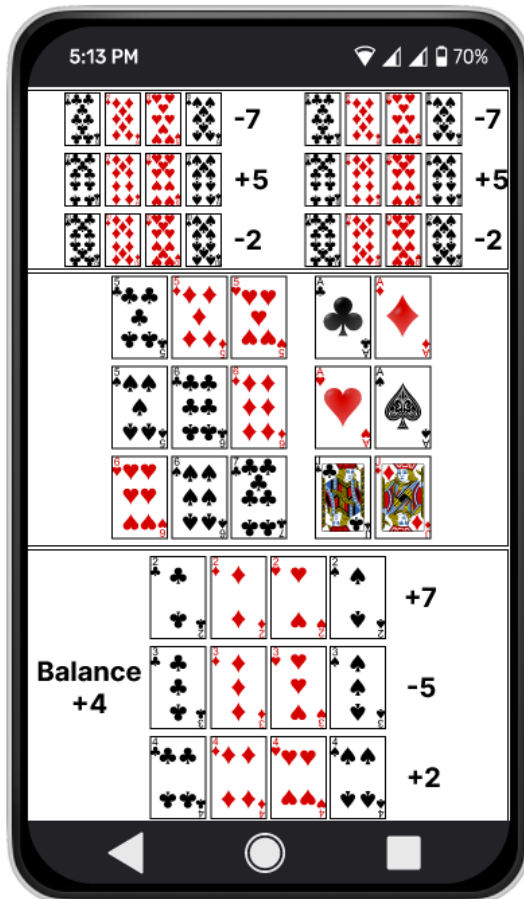


Figura 42. Página juego con 2 jugadores, fase 3

2.4. Evaluación

2.4.1. Tareas del usuario.

Para llevar a cabo esta parte se ha decidido realizar el estudio de dos usuarios diferentes, un usuario que tiene experiencia con aplicaciones de póquer online y otro usuario que no tiene experiencia en póquer, aunque el perfil demográfico sea parecido.

Preguntas sobre el usuario:

Para realizar esta parte, recuperaremos las preguntas o a un usuario que respondió al formulario de Google que hicimos para el apartado 2.1.1.2.

- Edad (<18, 18-30, 31-40, 41-50, 51-70, >70, Prefiero no decirlo)
- Sexo (Hombre, Mujer, Prefiero no decirlo, Otro)
- ¿De dónde eres? (Pregunta abierta)
- ¿Has jugado/juegas a juegos de cartas en el móvil? (Sí, No)
- ¿Has jugado/juegas a póquer alguna vez? (Sí, No)
- ¿Has jugado alguna vez a Omaha? (Sí, No)

- ¿Si existiese una nueva variante de Omaha te gustaría probarla? (Sí, No, Tal vez)
- ¿Juegas a póquer en el móvil? (Sí, No)
- ¿Si tuvieses un juego en el que tienes 12 cartas, 3 boards y 1 o 2 contrincante, te gustaría que la pantalla se viese horizontal o vertical? (Pregunta abierta)
- ¿Te gustaría poder elegir el color, tema de las cartas? (Sí, No, Tal vez)
- ¿Te gustaría poder elegir el color, tema de la mesa? (Sí, No, Tal vez)
- ¿Qué características valoras más en una aplicación de póquer móvil? (diseño de interfaz, opciones de personalización, bonificaciones, competiciones, otras)
- ¿Qué aspectos te gustaría mejorar o cambiar en las aplicaciones de póker móvil que has utilizado anteriormente? (Pregunta abierta)
- ¿Algún comentario adicional que quieras añadir? (Pregunta abierta)

Tareas que se realizarán sobre el prototipo:

Dado que el prototipo está en fase experimental y no es completamente funcional, aparte de seguir las instrucciones que presentaremos a continuación, se le pedirá al usuario que es lo que haría en alguna parte, para ver si la interfaz es intuitiva o no.

- Tarea 1: Desde el menú de *login* que harías para crear una cuenta nueva
- Tarea 2: Nos damos cuenta de que nos hemos olvidado la contraseña introducida, ¿qué harías?
- Tarea 3: Vuelve al menú de *login*, accede a la aplicación mediante el formulario.
- Tarea 4: ¿Qué crees que es el menú actual?
- Tarea 5: ¿A primera vista, que acciones crees que puedes realizar?
- Tarea 6: ¿Cómo accederías a las reglas?
- Tarea 7: ¿Crees que las reglas hacen una buena explicación del juego?
- Tarea 8: ¿Dónde le darías si quisieras cambiar el tema de las cartas, y la foto de perfil?
- Tarea 9: Si quisieras retomar una partida con un adversario, ¿cómo procederías?
- Tarea 10: Necesitamos crear una partida contra 1 jugador, de 3 sets y 300 puntos por set ¿Cómo lo harías?
- Tarea 11: Una vez en partida, ¿te parece una interfaz intuitiva?, ¿se ven bien las cartas?
- Tarea 12: Finaliza tu jugada y envíala al adversario.
- Tarea 13: Referente al último estado de juego, ¿se ve claro quién gana en esta pantalla?
- Tarea 14: ¿Dónde crees que debería redirigir esta última pantalla?

Preguntas referentes a la tarea.

- Pregunta 1: ¿Encuentras que la navegación dentro de la aplicación es fluida y fácil de entender?
- Pregunta 2: ¿Te resultó sencillo encontrar las funciones o características que necesitabas durante la realización de la tarea?
- Pregunta 3: ¿Hubo algún elemento de la interfaz que te resultara confuso o poco claro?
- Pregunta 4: ¿Te gustaría ver alguna función adicional que facilite la realización de la tarea?
- Pregunta 5: ¿Hay algún aspecto de la aplicación que destacarías como especialmente positivo o negativo durante la realización de la tarea?
- Pregunta 6: Otros

2.4.2 Documento con el feedback de cada tarea

He decidido analizar a dos usuarios diferenciados, aunque demográficamente sean parecidos. Uno de los usuarios es jugador profesional de póquer y el otro es una persona que no ha jugado demasiado al póquer, es decir, más inexperto.

La decisión de porque he elegido a estos dos perfiles de usuario es porque un usuario experimentado puede dar un punto de vista más profesional en cuanto al tema de la interfaz, jugadas y otras cosas. Y el otro usuario puede proporcionar información referente a la usabilidad de la aplicación, como por ejemplo si es intuitiva.

	Mariona Mujer española de 29 años actualmente es jugadora profesional de póquer, es decir, se gana la vida jugando a póquer. Actualmente dedica más tiempo a jugar a póquer online que en vivo ya que puede optimizar mejor su tiempo y jugar más manos por hora. Es un usuario experimentado y que además ha probado otras variantes de póquer.	
	Inputs positivos	Inputs negativos
Tarea 1	Ha accedido a sign up correctamente y comenta que tiene que rellenar el formulario para registrarse	-
Tarea 2	Comenta que ve fácilmente el botón de forgot your password, introduciría el correo de recuperación y le daría a submit.	-
Tarea 3	En vez de acceder con el formulario, el usuario decide acceder mediante el login de Google.	-
Tarea 4	Comenta que ve el menú principal del usuario.	-
Tarea 5	Crear una partida nueva Acceder a una partida ya creada Cree que dándole al avatar puede cambiar la imagen/nombre del jugador	Debido a que la pantalla desde donde visualiza el contenido tiene una rallada, no ha visto claramente que hay un botón de Rules.

Tarea 6	Apretando el botón de Rules	-
Tarea 7	Las reglas están bien estructuradas.	Podría haber imágenes para que fuese una explicación más clara. En la regla número 3 comenta que el flop ya se muestra en el punto anterior.
Tarea 8	Al avatar de la persona y dentro del menú que accede, le daría al avatar para cambiar la imagen. Comenta que se ve claramente las opciones de cambiar el tema de las cartas.	El usuario comenta que, en las aplicaciones de póquer, en vez de poner select board, pone table theme. También comenta que quizá se puede eliminar la palabra select de la pantalla de configuración de usuario.
Tarea 9	Le daría al nombre de la partida y accedería a aquella partida	No está prototipada esta función.
Tarea 10	Accede a New Game, buscaría el usuario contra quien quiere jugar, en la barra de los sets la desplazaría hasta el valor 3 y el total de puntos lo desplazaría hasta el valor 300. Finalmente le daría a new game	Esta página no es funcional actualmente.
Tarea 11	Las cartas en sí se ven bien (desde el ordenador). La interfaz parece intuitiva, movería las cartas de abajo donde los huecos que faltan por rellenar.	La distribución de las imágenes de las cartas no es habitual (ejemplo 7 de tréboles, los tréboles no están bien distribuidos en la superficie de la carta) Seguramente desde el móvil no se vean tan nítidas.
Tarea 12	Clicaría sobre el botón GO para enviar la jugada al adversario	-
Tarea 13	Se ve claramente quien gana.	Comenta que podría haber una animación (iluminando de las cartas de la jugada ganadora) además de que apareciese el nombre de la jugada (full, póquer, ...)
Tarea 14	Debería redirigir a la siguiente mano. A no ser que el juego se haya acabado, es decir que haya ganado los sets correspondientes.	-

Pregunta 1	Si, el usuario comenta que no se le cargaba el prototipo en el dispositivo móvil (la carga de las diferentes imágenes).
Pregunta 2	Si, se veía todo correctamente, no está sobrecargado y se ve claramente lo que se tiene que ver.
Pregunta 3	La aplicación parece clara, aunque en cuanto a las cartas, destacaría el número y palo.
Pregunta 4	Añadir algún tipo de ordenación de las cartas de tu mano. Espaciar las cartas en función del número de cartas libres de tu mano.
Pregunta 5	Como aspectos positivos destacaría el login con Google y que la aplicación es bastante intuitiva. Como aspecto negativo está el tema

	de la carga de las imágenes y el tema de las cartas comentado anteriormente.
Pregunta 6	Las preguntas realizadas estaban bien buscadas y por lo tanto no tengo ningún comentario adicional.

	Aroa Mujer española de 31 años, a pesar de su relativa inexperiencia, ha jugado a juegos de cartas en su dispositivo móvil en el pasado, lo que indica su predisposición a probar nuevas formas de diversión digital. Ha tenido la oportunidad de jugar al póquer antes, aunque no en su dispositivo móvil, además no ha explorado otras variantes como Omaha.	
	Inputs positivos	Inputs negativos
Tarea 1	Le daría a Sign up o login with Google. En el caso de que le dé a sign up rellenaría el formulario y le daría al botón de Sign up, también está la opción de login with Google.	-
Tarea 2	Le daría al botón forgot password, introduciría mi correo electrónico y me supongo que me enviarían al correo electrónico algún formulario para cambiar la contraseña.	-
Tarea 3	Si no accedo con Google, pondría mi correo y contraseña y le daría al botón de login.	-
Tarea 4	Parece el inicio del perfil, aparecen las partidas que has jugado.	-
Tarea 5	Ver las reglas Ver las puntuaciones de cada partida Ver mi perfil Comenzar una partida nueva	Pondría el tema de la puntuación de las partidas en otra pestaña y la personalización del perfil en la principal.
Tarea 6	Le daría al botón de Rules	-
Tarea 7	Si, todo correcto.	-
Tarea 8	Le daría al avatar del usuario, en el caso de querer cambiar la foto de perfil le daría otra vez a la imagen del usuario. En el caso de que quisiera cambiar el tema de las cartas o del tablero le daría a alguna de las otras opciones, están claramente indicadas.	-
Tarea 9	New game, buscaría el nombre del adversario y otra vez nuevo juego.	-
Tarea 10	Le daría a new game, desplazaría las configuraciones	-

	de juego a mis preferencias de partida y le daría a nuevo juego.	
Tarea 11	Basándome en las cartas que hay en la mesa, arrastraría las cartas que hay abajo hacia los huecos que están por rellenar. Las cartas se ven correctamente.	-
Tarea 12	Cuando tengo todas las cartas puestas en los huecos, le daría al botón de GO. Entiendo que aparece el botón cuando están todas las cartas colocadas en los huecos.	-
Tarea 13	Si se ve claramente, en este caso el jugador de abajo.	Las cartas del contrario son muy pequeñas y casi no se ven.
Tarea 14	Debería aparecer una pantalla que te redirija al inicio o nuevo juego.	-

Pregunta 1	Si, creo que es una aplicación usable
Pregunta 2	Si, ya que me recordaba a otras aplicaciones de juegos (solitario)
Pregunta 3	En el prototipo al solo aparecer el nombre de David como usuario agregado, me ha resultado un poco confuso, no me ha quedado claro cómo podía elegir el adversario.
Pregunta 4	No, todas las funciones existentes están bien creadas para llevar a cabo la realización de las diferentes tareas.
Pregunta 5	La personalización del perfil.
Pregunta 6	A mi parecer, se podría modificar el menú principal para que las partidas ya realizadas apareciesen en otro menú.

2.4.3 Listado de puntos de mejora.

A continuación, se exponen las posibles mejoras y/o modificaciones rescatadas de las tareas y feedbacks del apartado anterior:

Se tendrá en cuenta la modificación de las reglas para añadir imágenes y modificación de las reglas para que no haya posibilidad de confusión. Me parece una mejora en cuanto a la claridad de las reglas además de que no creo que comporte demasiado tiempo.

Se tendrá en cuenta la modificación del texto que aparecerá en los botones de selección del tema de las cartas y de la mesa: table theme, front deck, back deck. Es un cambio que propone un usuario que tiene más experiencia en el campo y no creo que comporte demasiado tiempo.

Se tendrá en cuenta la búsqueda de temas de cartas que sean mejores al seleccionado, hay cartas que no tienen el mismo tema que otras. Es un cambio importante de cara a la claridad del juego, aunque seguramente me lleve tiempo en la búsqueda de imágenes y la edición de estas.

Se tendrá en cuenta la creación de una animación iluminando la jugada ganadora, además de hacer aparecer el nombre de la jugada (full, póquer, ...). Es una mejora de cara a la claridad del juego, aunque dada mi inexperiencia en el campo del desarrollo móvil no sé cuánto tiempo me va a llevar.

Se tendrá en cuenta añadir una funcionalidad que ordene las cartas de tu mano (por palo, por valor). Es una ayuda de cara al jugador para poder analizar mejor su mano, además que no creo que lleve demasiado tiempo ya que entiendo que será únicamente ordenar una lista.

En función del tiempo se puede añadir una funcionalidad para espaciar las cartas en función del número de cartas libres de tu mano. Aunque no lo considero una funcionalidad relevante, puede ser de ayuda para el usuario el ver las cartas con mayor claridad.

En función del tiempo se puede añadir una funcionalidad donde en un apartado/pestaña estén las partidas finalizadas. Aunque no lo considero una funcionalidad relevante, puede ser de ayuda para el usuario el poder distinguir con mayor claridad las partidas en curso de las ya finalizadas.

No se llevará a cabo la funcionalidad de añadir el tema de la personalización del perfil en la página principal. Ya que, creo que el perfil debería estar situado donde está actualmente.

Se tendrá en cuenta ampliar las cartas del adversario para ver su jugada, ya que no se aprecian.

3. Diseño técnico

3.1. Casos de uso

3.1.1 Diagrama UML para identificar gráficamente actores y flujo

Mediante el siguiente diagrama UML identificamos que existen 2 tipos de usuario, el registrado y el no registrado. Se ha resaltado con diferentes colores las acciones que puede realizar los diferentes usuarios.

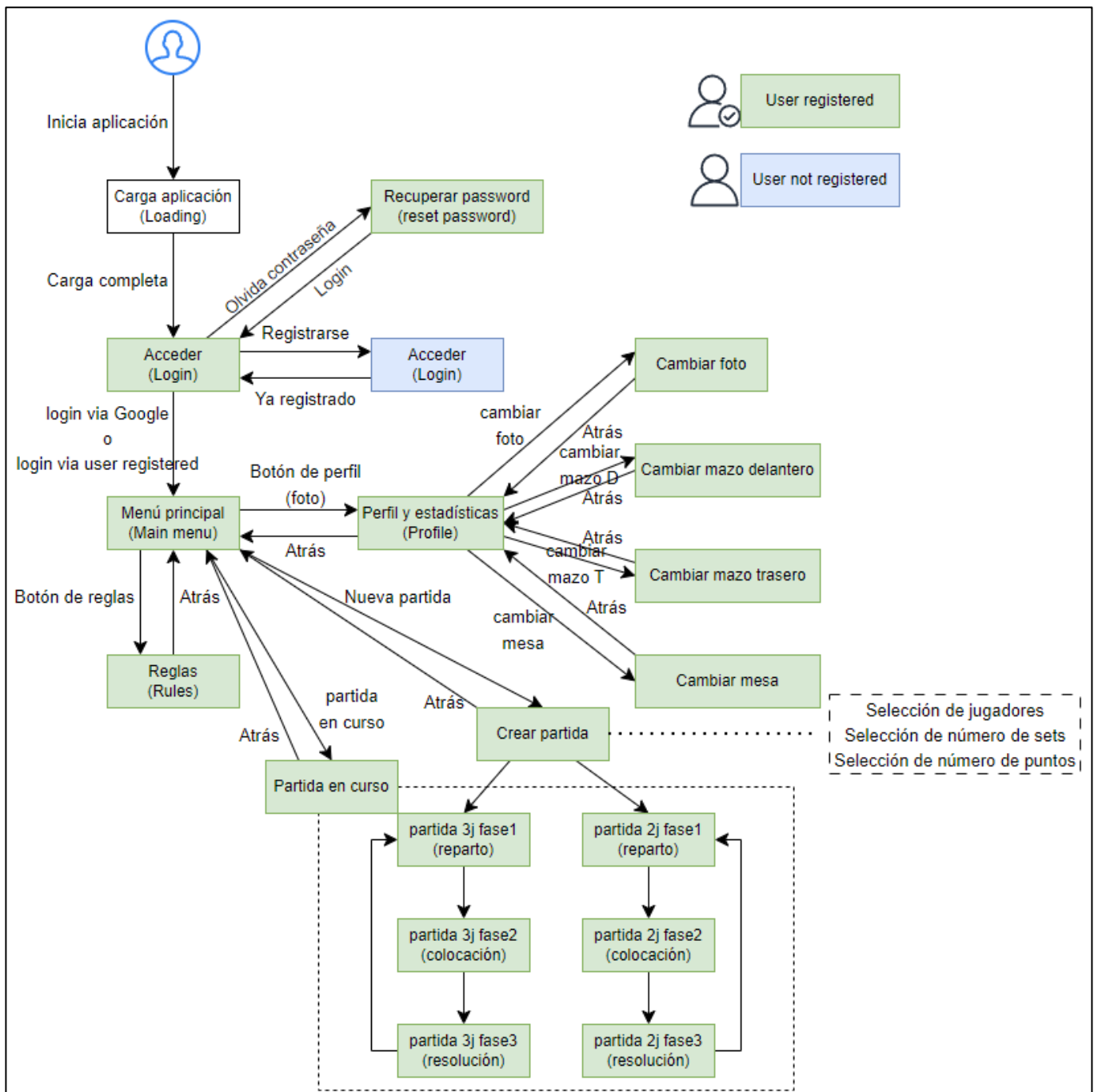


Figura 43. Diagrama UML con actores y flujo

3.1.2 Listado de casos de uso donde se especifican actores precondiciones, flujo y post condiciones

CU00 – PÁGINA DE CARGA	
Prioridad	Baja
Descripción	Animación de carga
Actores	Todos
Precondiciones	-
Iniciado por	Siempre que este cargando la aplicación (normalmente cuando se inicia la aplicación)
Flujo	<ol style="list-style-type: none"> 1. Aparece una animación de carga de la aplicación 2. Redirige a la vista adecuada
Postcondiciones	Se redirige a la página adecuada
Notas	

CU01 – REGISTRAR USUARIO	
Prioridad	Media
Descripción	Se crea un perfil de usuario para poder acceder a la aplicación
Actores	Usuario no registrado
Precondiciones	El usuario no se ha registrado previamente
Iniciado por	El usuario no registrado accede a la aplicación, desde la página de login pulsa en el enlace que dice forgot sign up
Flujo	<ol style="list-style-type: none"> 3. Introduce el nombre de usuario <ol style="list-style-type: none"> a. Si el campo está vacío, aparecerá un error 4. Introduce el correo electrónico <ol style="list-style-type: none"> a. Si el campo está vacío, aparecerá un error b. Si no es un correo electrónico válido, aparecerá un error 5. Introduce la contraseña <ol style="list-style-type: none"> a. Si el campo está vacío, aparecerá un error b. Si la contraseña no tiene un mínimo de 8 caracteres aparecerá un error c. Si la contraseña no tiene un mínimo de una mayúscula, aparecerá un error d. Si la contraseña no tiene un mínimo de una minúscula, aparecerá un error e. Si la contraseña no contiene un carácter especial aparecerá un error 6. Introduce la misma contraseña que en el punto anterior <ol style="list-style-type: none"> a. Si la contraseña introducida no coincide con la anterior, aparecerá un error 7. Clic en el botón de Sign up <ol style="list-style-type: none"> a. Si el nombre de usuario ya está registrado en la aplicación, la aplicación devolverá un error b. Si el correo electrónico ya está registrado en la aplicación, la aplicación devolverá un error 8. Se registra al usuario en el sistema

	9. Se redirige a la página de login
Postcondiciones	El usuario queda registrado en la aplicación Se redirige a la página de login
Notas	

CU02 – CONTRASEÑA OLVIDADA	
Prioridad	Baja
Descripción	Se envía un correo electrónico para reiniciar la contraseña
Actores	Usuario registrado
Precondiciones	El usuario está registrado en el sistema
Iniciado por	El usuario registrado accede a la aplicación, desde la página de login pulsa en el enlace que dice forgot password
Flujo	<ol style="list-style-type: none"> 1. Introduce el correo electrónico <ol style="list-style-type: none"> a. Si no es un correo electrónico válido, aparecerá un error. 2. Se le da al botón de Submit <ol style="list-style-type: none"> a. Si el correo electrónico no existe en la base de datos, devuelve un error 3. Se envía un correo electrónico al correo introducido con información de cómo proceder para cambiar la contraseña
Postcondiciones	Se envía un correo electrónico al correo introducido
Notas	

CU03 – ACCEDER VIA FORMULARIO	
Prioridad	Alta
Descripción	Se accede a la página principal de la aplicación mediante el formulario del login
Actores	Usuario registrado
Precondiciones	Usuario registrado
Iniciado por	Usuario registrado, inicializa la aplicación y está en la página de acceso
Flujo	<ol style="list-style-type: none"> 1. Introduce el correo electrónico <ol style="list-style-type: none"> a. Si no es un correo electrónico válido, aparecerá un error. 2. Introduce a contraseña 3. Se le da al botón de login <ol style="list-style-type: none"> a. Si el usuario y la contraseña no coinciden con la base de datos, se devuelve un error 4. Se accede correctamente a la página principal.
Postcondiciones	Se redirige a la página principal
Notas	

CU04 – ACCEDER VIA GOOGLE	
Prioridad	Media
Descripción	Se accede a la página principal de la aplicación mediante el formulario de Google
Actores	Usuario registrado

Precondiciones	Usuario registrado
Iniciado por	Usuario registrado, inicializa la aplicación y está en la página de acceso
Flujo	<ol style="list-style-type: none"> 1. Clic en el botón de Login with Google 2. Introduce los datos de la cuenta de Google 3. Se accede correctamente a la página principal.
Postcondiciones	Se redirige a la página principal
Notas	

CU05 – RULES	
Prioridad	Baja
Descripción	Se accede a la página que muestra las reglas del juego
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	Usuario registrado, desde el menú principal clic en el botón de reglas
Flujo	<ol style="list-style-type: none"> 1. Clic en el botón de reglas desde el menú principal 2. Visualización de las reglas
Postcondiciones	Visualización de las reglas
Notas	

CU06 – CAMBIAR IMAGEN DE PERFIL	
Prioridad	Baja
Descripción	Se cambia la imagen de perfil que tiene el usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario en la página de perfil del usuario le da clic a la imagen
Flujo	<ol style="list-style-type: none"> 1. El usuario le da clic a su imagen de perfil 2. El usuario selecciona una imagen que tenga en el dispositivo 3. Le da clic a cambiar la imagen <ol style="list-style-type: none"> a. Si la imagen supera los 4 MB, devolverá un error 4. Se guarda la imagen en la base de datos
Postcondiciones	La imagen se guarda en la base de datos
Notas	

CU07 – CAMBIAR TEMA FRONTAL DE LAS CARTAS	
Prioridad	Baja
Descripción	Se cambia el tema frontal de las cartas que tiene el usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario en la página de perfil del usuario le da clic front deck
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el tema del frontal de las cartas entre los disponibles

	<ol style="list-style-type: none"> a. Se muestran temas de las cartas que no están disponibles hasta que no se jueguen X partidas u otras condiciones 2. Cuando el usuario selecciona el tema que quiere, se guarda en sus preferencias locales.
Postcondiciones	El tema seleccionado se guarda en sus preferencias locales
Notas	

CU08 – CAMBIAR TEMA POSTERIOR DE LAS CARTAS

Prioridad	Baja
Descripción	Se cambia el tema posterior de las cartas que tiene el usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario en la página de perfil del usuario le da clic back deck
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el tema del posterior de las cartas entre los disponibles <ol style="list-style-type: none"> a. Se muestran temas de las cartas que no están disponibles hasta que no se jueguen X partidas u otras condiciones 2. Cuando el usuario selecciona el tema que quiere, se guarda en sus preferencias locales.
Postcondiciones	El tema seleccionado se guarda en sus preferencias locales
Notas	

CU09 – CAMBIAR TEMA DE LA MESA

Prioridad	Baja
Descripción	Se cambia el tema de la mesa que tiene el usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario en la página de perfil del usuario le da clic table theme
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el tema de la mesa entre los disponibles <ol style="list-style-type: none"> a. Se muestran temas de la mesa que no están disponibles hasta que no se jueguen X partidas u otras condiciones 2. Cuando el usuario selecciona el tema que quiere, se guarda en sus preferencias locales.
Postcondiciones	El tema seleccionado se guarda en sus preferencias locales
Notas	

CU10 – HISTORIAL DE PARTIDAS ACTUALES

Prioridad	Media
Descripción	Se muestran las partidas en curso del usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario está en la página principal

Flujo	1. Se recuperan todas las partidas que no están acabadas del usuario de la base de datos
Postcondiciones	Se muestran las partidas que están en curso
Notas	

CU11 – HISTORIAL DE PARTIDAS FINALIZADAS	
Prioridad	Baja
Descripción	Se muestran el resultado de las partidas finalizadas del usuario
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario hace clic en historial de partidas
Flujo	1. Se recuperan el resultado de todas las partidas finalizadas del usuario de la base de datos
Postcondiciones	Se muestran los resultados de las partidas que están finalizadas
Notas	

CU12 – CREAR UNA PARTIDA	
Prioridad	Alta
Descripción	Se crea una partida
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	El usuario está en el menú principal y le da al botón NEW GAME
Flujo	<ol style="list-style-type: none"> 1. El usuario selecciona el número de sets 2. El usuario selecciona el total de puntos del juego 3. El usuario selecciona el/los jugadores contra los que quiere jugar <ol style="list-style-type: none"> a. En el caso de que el/los jugador/es no estén en la lista de jugadores, el jugador puede buscar los jugadores mediante una barra de búsqueda de usuario (CU12) 4. El usuario le da a NEW GAME 5. Se crea una partida en la base de datos con las características seleccionadas
Postcondiciones	Se crea una partida en la base de datos
Notas	Este caso de uso depende directamente de CU13

CU13 – AGREGAR Y BUSCAR USUARIO	
Prioridad	Alta
Descripción	Se busca y agrega un usuario registrado
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado, otros usuarios registrados
Iniciado por	El usuario introduce un nombre de usuario en la barra de búsqueda, pasado unos segundos aparecerá posibles resultados
Flujo	1. El usuario escribe un nombre o parte del nombre en la barra de búsqueda

	<ol style="list-style-type: none"> 2. Se hace una búsqueda en la base de datos 3. Se devuelve nombres que coincidan con la búsqueda <ol style="list-style-type: none"> a. Puede no devolver ningún nombre si la búsqueda no coincide 4. Cuando el usuario selecciona el nombre del usuario de su interés 5. Se envía una petición a la base de datos para agregar a ese usuario como favoritos y que aparezca para posibles partidas
Postcondiciones	Se agrega al nuevo usuario en la lista de amigos del usuario actual
Notas	

CU14 – REPARTIR MANO	
Prioridad	Alta
Descripción	Se reparte una mano a cada jugador y se muestran las primeras cartas comunes
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	Se ha creado un juego nuevo con 1 o dos jugadores
Flujo	<ol style="list-style-type: none"> 1. Cada jugador tiene un total de 12 cartas 2. Se muestran 9 cartas comunes (3 de cada board)
Postcondiciones	Se muestra el número de cartas correctas, las 12 propias y las 9 comunes
Notas	
CU15 – HACER UNA JUGADA	
Prioridad	Alta
Descripción	Realizar una jugada para poder enviarla al rival
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado
Iniciado por	Se ha creado un juego nuevo con 1 o dos jugadores y se ha repartido una mano
Flujo	<ol style="list-style-type: none"> 1. El jugador coloca cada una de las 12 cartas que ha recibido en los huecos disponibles para ello (4 cartas por cada board) 2. Le da al botón de GO 3. Se envía la mano a la base de datos para proceder con el ganador
Postcondiciones	Se envía la mano a la base de datos
Notas	

CU15 – GANADOR DE LA MANO	
Prioridad	Alta
Descripción	Se procede con el ganador de la mano
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado, todos los jugadores han realizado la jugada
Iniciado por	Se ha creado un juego nuevo con 1 o dos jugadores y se ha repartido una mano y se ha realizado la jugada

Flujo	<ol style="list-style-type: none"> 1. Se muestra los resultados de cada mano, tanto la jugada que ha hecho cada jugador como el balance de puntos 2. Se muestra el balance total de puntos 3. Se declara al ganador 4.1 En el caso de que se haya ganado el total de sets se declara el juego como finalizado 4.2 En el caso de que se haya ganado un set, pero no el total de sets, se aumentará el número de sets ganados del jugador correspondiente y se procederá con poner a 0 los puntos para iniciar un nuevo set en el mismo juego 4.3 En el caso de que no se haya ganado el set, se ajustará el número de puntos de los jugadores y se procederá una nueva partida en el mismo set y juego 5. Redirigir a una nueva partida o menú principal en función de si se ha acabado el juego o no
Postcondiciones	<p>Recalculo de puntos Recalculo de sets Finalización de la partida</p>
Notas	

CU16 – ORDENAR CARTAS	
Prioridad	Baja
Descripción	Funcionalidad que ordena las cartas de la mano
Actores	Usuario registrado
Precondiciones	Usuario registrado y logeado, se está jugando una partida nueva y aun no se han colocado todas las cartas
Iniciado por	Se ha creado un juego nuevo con 1 o dos jugadores y se ha repartido una mano y no se han colocado todas las cartas
Flujo	<ol style="list-style-type: none"> 1. El jugador le da a SORT 2. Las cartas que quedan por colocar se ordenan de mayor a menor y después por palo 3. El jugador le da a SORT 4. Las cartas que quedan por colocar se ordenan por palo y después de mayor a menor 5. El jugador le da a SORT vuelve a 2
Postcondiciones	Ordenación de las cartas que restan por colocar, la ordenación puede ser de mayor a menor y después por palo o por palo y después de mayor a menor
Notas	

3.2. Diseño de la arquitectura

3.2.1 Diagrama UML de la base de datos

En el siguiente diagrama se representan las principales clases y como se relacionan en nuestra base de datos. Podemos observar que este tipo de diagramas, aunque se trate de una base de datos NoSQL es muy similar al de una base de datos SQL.

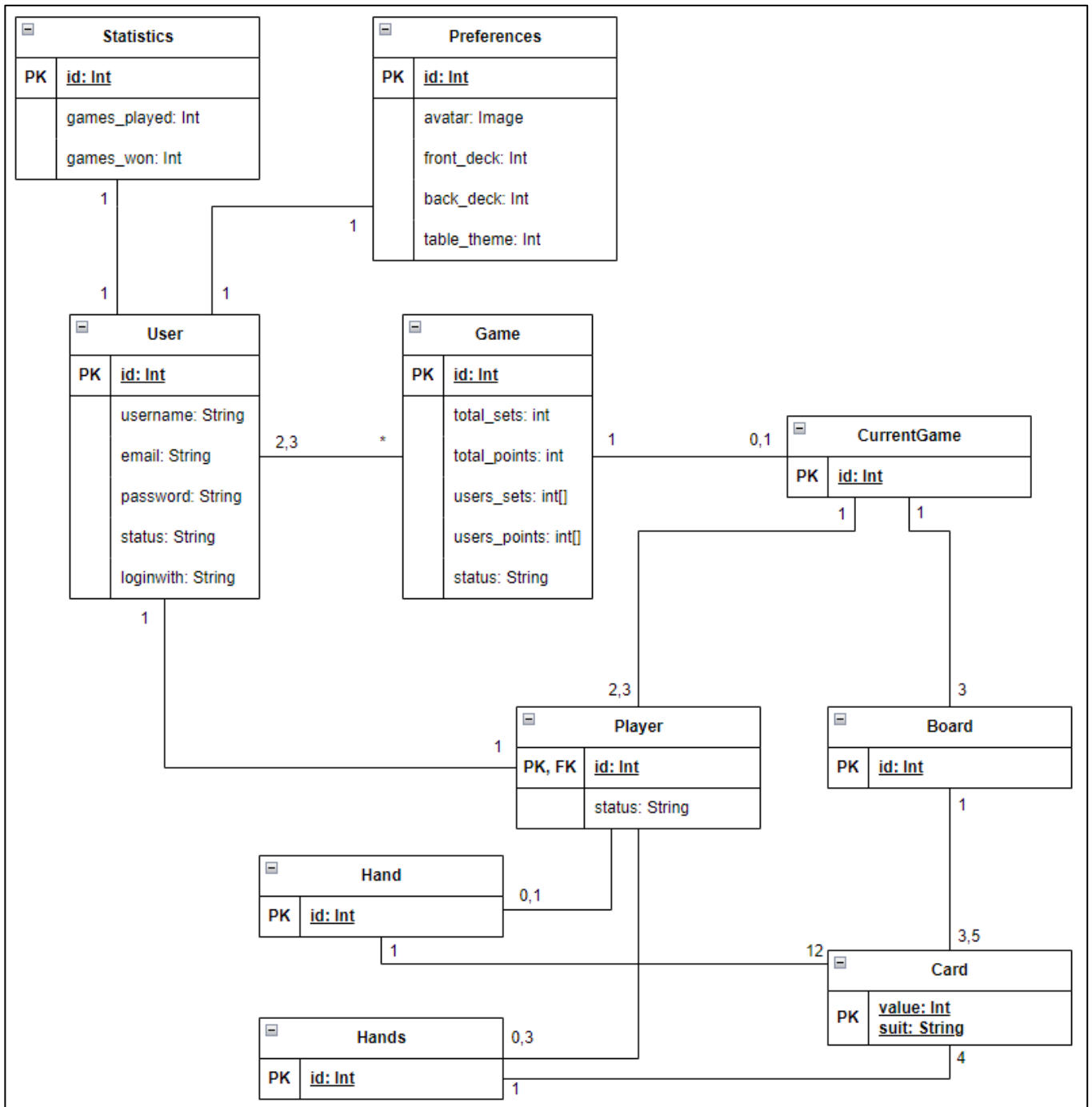


Figura 44. Diagrama UML de la base de datos

3.2.2 Diagrama UML del diseño de las entidades y clases

En el siguiente diagrama se representan las diferentes entidades y clases, destacar que en el diagrama no está implementada la lógica de la resolución de las jugadas, tampoco está implementada la conexión con la API de Google para realizar el login, no está implementado la conexión con la base de datos y tampoco está implementada la gestión de los tokens y seguridad de la aplicación.

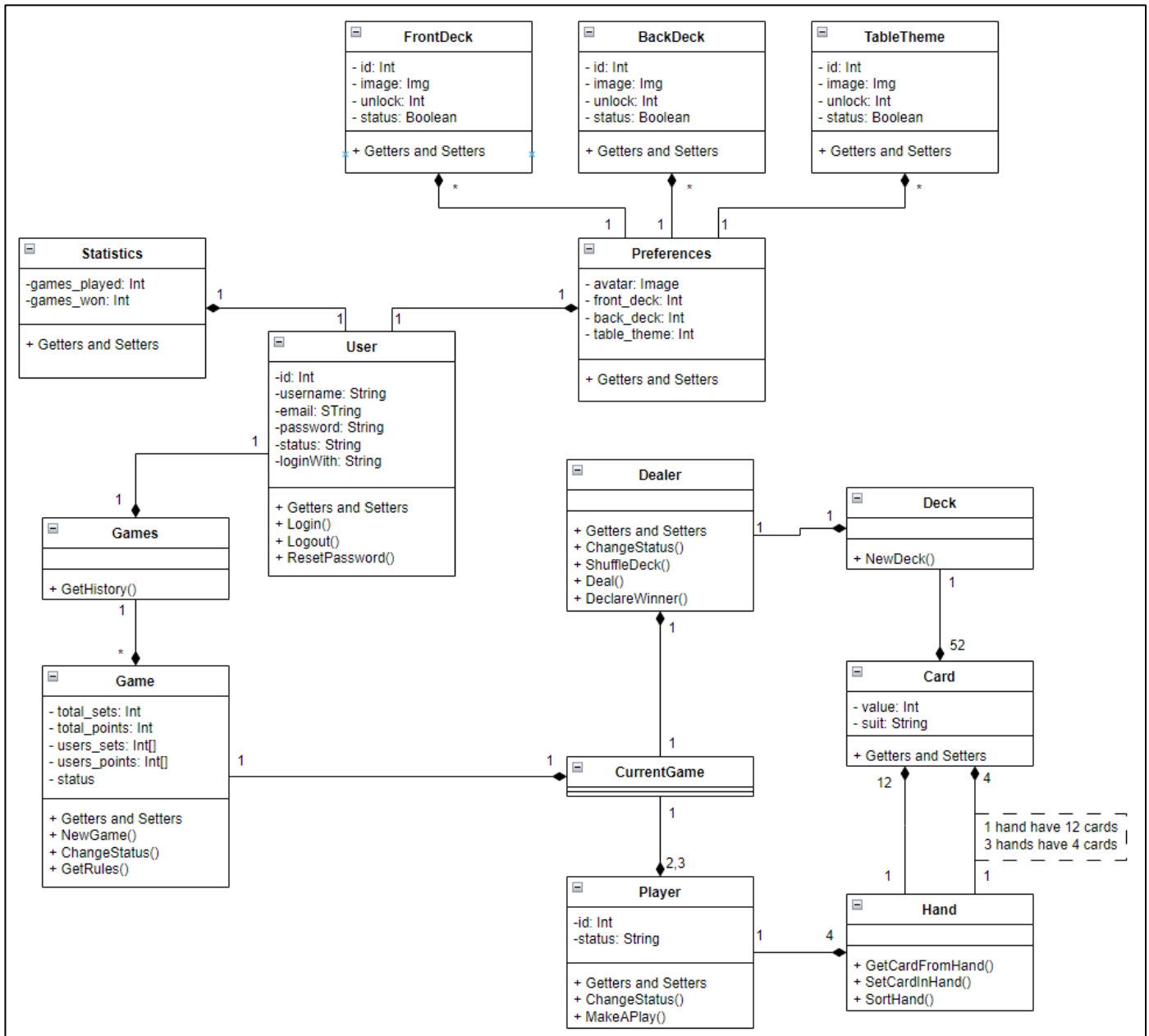


Figura 45. Diagrama UML de las entidades y clases

3.2.3 Diagrama explicativo de la arquitectura del sistema

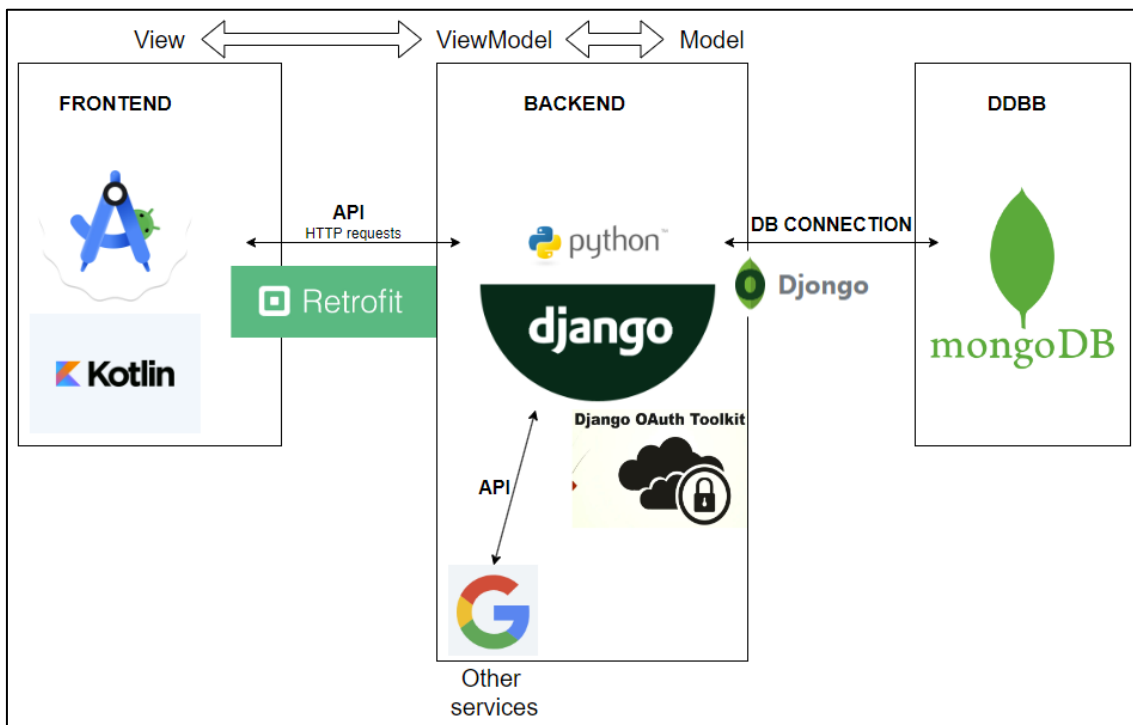


Figura 46. Diagrama de la arquitectura del sistema

Voy a implementar una arquitectura MVVM (Model, View, ViewModel), he elegido esta arquitectura debido a que la mayoría de las aplicaciones implementadas con Android siguen este tipo de arquitectura, aunque en mi caso adaptaré la parte del ViewModel y Model que se implementa normalmente.

La Vista de mi aplicación consistirá en las actividades y fragmentos de la aplicación Android, esta parte será la responsable de mostrar la interfaz de usuario al usuario final. Además, se implementará retrofit para realizar las conexiones con el servicio web.

El Modelo-Vista actuará como intermediario entre la Vista y el Modelo, estará implementado en la parte del backend de Django. Se encargará de preparar y proporcionar los datos necesarios para la interfaz de usuario de la aplicación. El Modelo definirá la estructura de los datos de la aplicación y proporcionará la lógica para interactuar con la base de datos. Esta conexión se realizará con Djongo ya que la base de datos se implementará con mongoDB.

Cabe mencionar que en la parte del backend implementaremos Django OAuth Toolkit para securizar la aplicación y además se quiere implementar un servicio de autenticación mediante la API de Google.

4. Implementación

Una vez establecidos los objetivos de la aplicación y definidas todas las especificaciones de diseño centradas en el usuario, técnicas y arquitectónicas, se procede con la implementación de la aplicación móvil. Se ha intentado que la aplicación sea lo más robusta y escalable posible. Para ello he utilizado todos los conocimientos adquiridos en la carrera.

Por una parte, el frontend se ha desarrollado con Android Studio, el software oficial de desarrollo de las aplicaciones Android, esto hace que tengamos a nuestra disposición las diferentes herramientas y funcionalidades propias del entorno. He utilizado la versión Oreo 9.1 que es compatible con el 91.8% de los dispositivos.

Por otra parte, me he ayudado del IDE de VisualStudioCode para implementar el backend, el backend está hecho en Django, que es un framework que utiliza Python como lenguaje de programación.

Por último, la base de datos está implementada con mongodb. Para visualizar los datos de la base de datos me he ayudado de MongoDB Compass.

A continuación, se detallan las diferentes tecnologías y herramientas utilizadas.

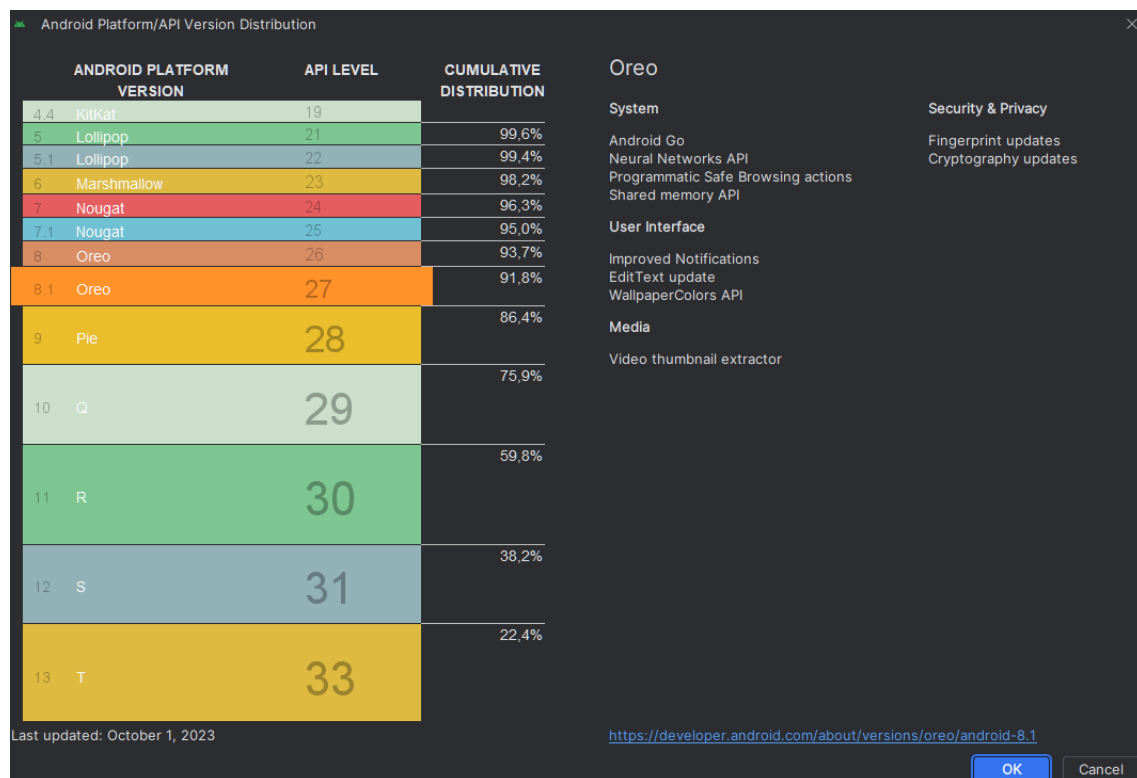


Figura 47. Versiones y compatibilidad de los diferentes dispositivos

4.1 Herramientas de trabajo

4.1.1 Lenguajes de programación

Para el desarrollo del frontend de la aplicación móvil, se ha utilizado Kotlin ya que es un lenguaje bastante moderno y está bastante de moda entre los nuevos programadores. Es un lenguaje promovido por Google, ofrece características avanzadas que facilitan un desarrollo más seguro y eficiente en Android. Para el backend, se ha optado por Python3, aprovechando su simplicidad y robustez, así como su vasta colección de bibliotecas y frameworks.

4.1.2 Base de datos

Se ha utilizado MongoDB como base de datos, elegida por su flexibilidad y escalabilidad, lo que permite manejar grandes volúmenes de datos de manera eficiente. MongoDB es una base de datos NoSQL que se adapta perfectamente a las necesidades de aplicaciones modernas que requieren almacenamiento de datos en esquemas dinámicos. Cabe mencionar que, dado que he utilizado djongo para la integración de Django con MongoDB, djongo no hace uso de MongoDB como si fuese una base de datos NoSQL, si no que lo implementa como si fuese una base de datos SQL.

4.1.3 Frameworks

Para el desarrollo del backend, se ha empleado Django, un framework de alto nivel que facilita la creación rápida de aplicaciones web seguras y mantenibles. Django se integra perfectamente con MongoDB a través de Djongo, permitiendo aprovechar las ventajas de ambos mundos.

4.1.4 IDEs

Para el desarrollo del frontend en Android, se ha utilizado Android Studio, el entorno de desarrollo integrado oficial para la plataforma Android, que proporciona potentes herramientas de edición y depuración. Para el backend, se ha empleado Visual Studio Code, un editor de código fuente altamente configurable y ligero que soporta una amplia variedad de lenguajes de programación y herramientas.

4.1.5 Librerías y APIs

A continuación, se mencionan las librerías más relevantes de las cuales he hecho uso para poder desarrollar mi aplicación:

Frontend:

- **Picasso:** Picasso es una librería de código que facilita la manipulación y el manejo de imágenes con operaciones como la carga de imágenes

desde URLs, su almacenamiento en caché, y su transformación (cortes, escalados, etc.).

- **Retrofit:** Retrofit es una potente librería de cliente HTTP para Android y Java. Permite la interacción con APIs RESTful de una manera más sencilla y organizada, transformando automáticamente las respuestas JSON en objetos Java mediante el uso de convertidores como Gson.
- **JUnit:** JUnit se emplea para realizar pruebas unitarias, garantizando que las diferentes partes del código funcionen correctamente. La integración de JUnit en el flujo de trabajo de desarrollo permite identificar y corregir errores en las etapas iniciales, asegurando la calidad y la estabilidad del código antes de su despliegue.

Backend:

- **Django-OAuth-Toolkit:** Esta es una librería extensible para Django que implementa el protocolo OAuth2, el cual es esencial para la autenticación segura de usuarios. OAuth2 permite a la aplicación delegar el proceso de autenticación a un servidor de autorización, mejorando la seguridad al no manejar directamente las credenciales del usuario.
- **Djongo:** Djongo es una herramienta que permite la integración de Django con MongoDB, una base de datos NoSQL. Djongo traduce automáticamente las operaciones ORM de Django en comandos de MongoDB, facilitando el uso de las potentes características de Django en un entorno NoSQL. Aunque si miramos la documentación oficial, pone que Djongo se utiliza para aquellos que quieran hacer migraciones de bases de datos SQL a Mongo.
- **NumPy:** NumPy es una librería fundamental para la computación científica en Python. Es una dependencia de la librería que menciono a continuación.
- **Phevaluator:** Phevaluator es una librería específica para la evaluación de manos de poker. En el contexto de esta aplicación, Phevaluator se ha utilizado para implementar la lógica del juego de poker, evaluando y comparando las manos de los jugadores para determinar el ganador. La librería ha sido adaptada a las necesidades específicas de la aplicación, asegurando un rendimiento óptimo y una evaluación precisa.

4.1.6 Otros

Git: Se ha utilizado GitLab como plataforma de control de versiones y gestión de repositorios. GitLab facilita la colaboración entre desarrolladores y ofrece un conjunto completo de herramientas para la integración y el despliegue continuos (CI/CD).

MongoDB Compass: Herramienta gráfica para gestionar y visualizar los datos almacenados en MongoDB, facilitando la administración y el análisis de la base de datos.

Postman: Utilizado para probar y documentar las APIs desarrolladas, asegurando que cumplan con las especificaciones requeridas y que funcionen correctamente antes de ser integradas en la aplicación.

4.2 Patrones y buenas prácticas

En mi aplicación, he implementado en la medida de lo posible las buenas prácticas de los principios SOLID, adaptándolos tanto en el frontend como en el backend, con la particularidad de estar condicionado por el uso de frameworks: Android Studio con su arquitectura MVVM y Django con arquitectura MVT.

Se tiene en cuenta el principio **Single Responsibility Principle (SRP)** en el que cada clase tiene una única responsabilidad. En el caso de Android estudio, la vista se encarga de la interfaz del usuario, el viewmodel maneja la lógica de la aplicación y los modelos gestionan los datos. En el caso de Django, los modelos representan la estructura de los datos, la vista representa la lógica de la aplicación y los templates se encargan del diseño de la interfaz del usuario, en nuestro caso concreto no tendremos templates, ya que se utiliza Django como un servicio.

En cuanto al **Open/Closed Principle (OCP)**, en el caso de Android, se pueden añadir clases y componentes sin la necesidad de modificar el código existente. En cuanto a Django, este promueve la extensión mediante la creación de nuevos componentes, manteniendo el código original cerrado a modificaciones, pero abierto a extensiones.

El **Liskov Substitution Principle (LSP)** se asegura mediante la creación de clases derivadas que pueden ser usadas en lugar de sus clases base sin alterar el correcto funcionamiento del programa. En Android Studio, esto significa que cualquier ViewModel derivado debe funcionar como el ViewModel base, permitiendo su intercambio sin problemas. En Django, cualquier subclase de un modelo o vista debe comportarse de manera consistente con su superclase, garantizando así la correcta sustitución.

El **Interface Segregation Principle (ISP)** se aplica al diseñar interfaces específicas para cada cliente, evitando que las clases implementen métodos que no utilizan. En Android Studio, esto implica que las interfaces para eventos de usuario sean específicas y granulares, permitiendo que cada actividad o fragmento implemente solo lo que necesita. En Django, las interfaces de servicios se separan para asegurar que cada componente solo tenga acceso a los métodos necesarios para su funcionalidad específica.

El **Dependency Inversion Principle (DIP)** se pone en práctica mediante la inversión de las dependencias, permitiendo que las clases de alto nivel no dependan de clases de bajo nivel, sino de abstracciones. En Android Studio, esto se logra mediante la inyección de dependencias, donde el ViewModel no depende directamente de los modelos sino de interfaces. En Django, el uso de servicios y repositorios para acceder a los datos en lugar de depender directamente de los modelos ayuda a desacoplar las dependencias, mejorando la flexibilidad y extensibilidad del código.

4.3 Desarrollo de la aplicación

Android Studio

En la imagen se pueden apreciar la organización de los diferentes directorios.

AndroidManifest.xml: Archivo que declara las diferentes actividades de la aplicación.

build.gradle.kts: Contiene las configuraciones necesarias para compilar y construir la aplicación, incluyendo las dependencias, configuraciones del SDK de Android, y tareas de construcción.

/network: contiene todos los archivos relacionados con la comunicación de red en la aplicación.

- **ApiClient:** Clase responsable de configurar y proporcionar el cliente de red
- **ApiService:** Interfaz que define los endpoints de la API y sus respectivas solicitudes HTTP
- El resto de los archivos son los modelos que se espera enviar y recibir al servidor.

/utils: contiene clases de utilidad y funciones de ayuda que son reutilizables, en mi caso el Toast y las preferencias compartidas.

Activities: las actividades representan la lógica de los layouts (pantallas que se muestran al usuario)

/res: contiene todos los recursos de la aplicación.

- **/drawable:** contiene todos los archivos gráficos y recursos de imagen que utiliza la aplicación.
- **/layout:** contiene los archivos de diseño XML que definen la disposición de la UI para las actividades.
- **/values:** define otros recursos como los textos (strings.xml), estilos (styles.xml), colores (colors.xml), entre otros.

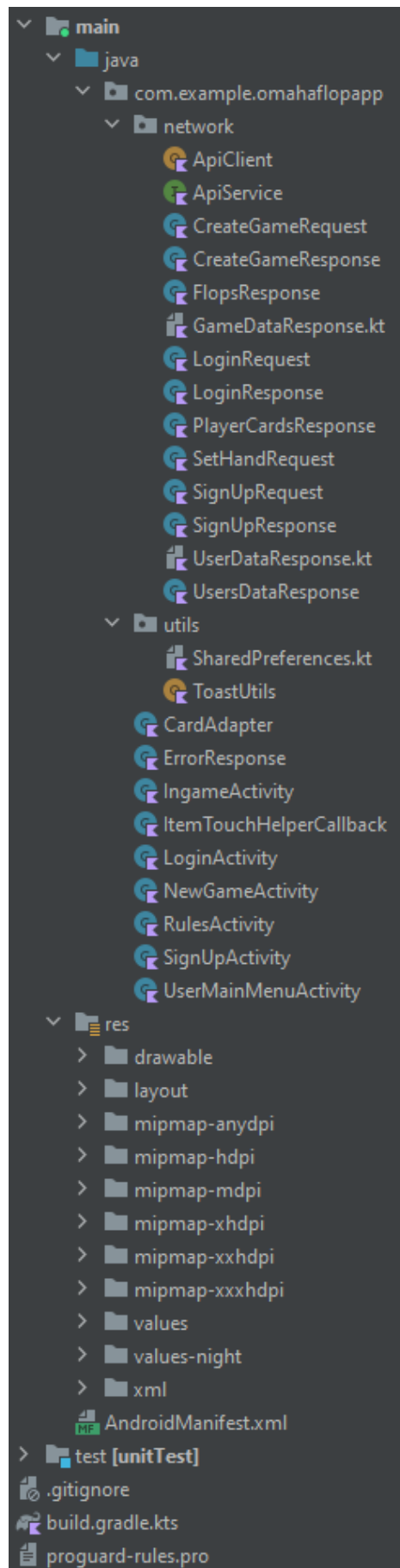


Figura 48. Estructura Android

Django

/OmahaFlopAPI: Aplicación principal

/UsersApp: Aplicación que gestiona la lógica relacionada con los usuarios

/GamesApp: Aplicación que gestiona la lógica relacionada con los juegos.

/PokerUtils: Subdirectorio dentro de GamesApp que contiene utilidades específicas para el juego de póker.

manage.py: Es un archivo que facilita la gestión del proyecto Django, se utiliza para ejecutar comandos administrativos, como iniciar el servidor de desarrollo, aplicar migraciones y crear aplicaciones.

settings.py: Es el archivo de configuración principal del proyecto, contiene entre otros la configuración de la base de datos.

urls.py: Archivo que define el mapeo de URLs a vistas correspondientes en el proyecto Django. Gestiona las rutas de las diferentes vistas y aplicaciones.

views.py: Archivo donde se definen las vistas de la aplicación. Una vista es una función o clase que recibe una solicitud web y devuelve una respuesta web.

aux_functions.py: Archivo opcional que contiene funciones auxiliares de las views.py.

serializers.py: Archivo utilizado en aplicaciones que implementan Django REST Framework. Contiene definiciones de serializadores, que son responsables de transformar datos complejos (como instancias de modelos) en tipos de datos nativos de Python, que luego pueden ser fácilmente renderizados en JSON o XML.

models.py: Archivo que define los modelos de la aplicación. Un modelo es una clase que representa una tabla en la base de datos.

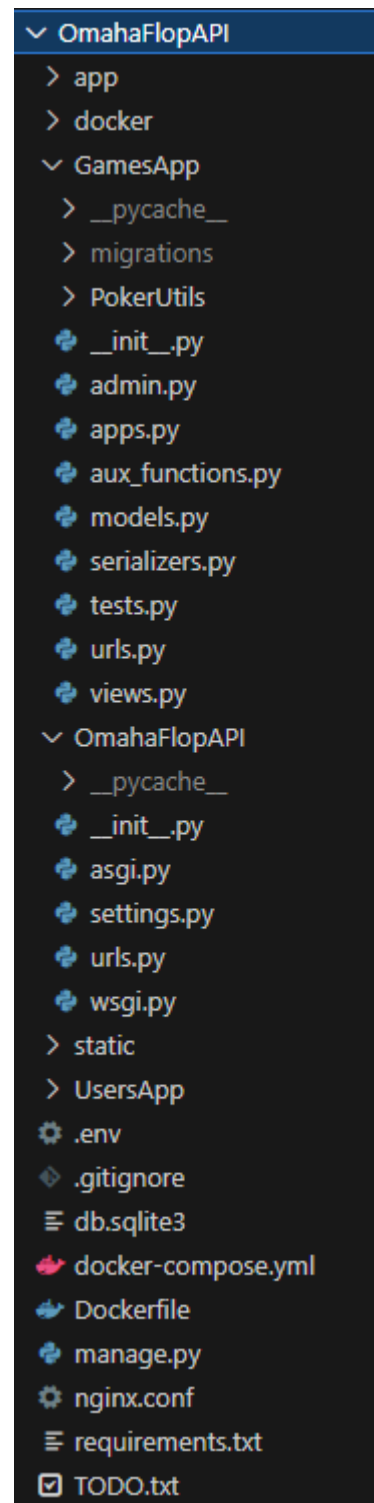


Figura 49. Estructura Django

4.4 Estado actual del proyecto

Respecto al diagrama de Gantt inicialmente propuesto, se han querido implementar diferentes funcionalidades que no estaban incluidas en el diagrama. Sin embargo, debido a la longitud y dificultad del proyecto, la falta de conocimiento en diversas tecnologías, además que mi proyecto no incluía únicamente el desarrollo del front, si no también de la creación de una API desde 0 y otros problemas derivados se ha optado por crear un mínimo entregable.

He dedicado muchas más horas de lo inicialmente previsto, pero aun así no he implementado muchas de las funcionalidades previstas. A fecha 20 de mayo del 2024 lo que está desarrollado es lo siguiente:

- CU00. Página de carga: **KO**
- CU01. Registrar usuario: **OK**, pero con margen de mejora de enviar un email al usuario que se acaba de registrar para validar el correo electrónico.
- CU02. Contraseña olvidada: **KO**
- CU03. Acceder via formulario: **OK**
- CU04. Acceder via Google: **KO**
- CU05. Reglas: **OK**
- CU06. Cambiar imagen de perfil: **KO**
- CU07. Cambiar tema frontal de las cartas: **KO**
- CU08. Cambiar tema posterior de las cartas: **KO**
- CU09. Cambiar tema de la mesa de las cartas: **KO**
- CU10. Historial de partidas actuales: **OK**
- CU11. Historial de partidas finalizadas: **OK**, se muestran todas las partidas, da igual el estado de esta
- CU12. Crear una partida: **OK**
- CU13. Agregar y buscar usuario: **OK**, no se agrega usuarios, pero si se pueden buscar
- CU14. Repartir mano: **OK**
- CU15. Ganador de la mano: **OK**, implementado el backend, pero no el front
- CU16. Ordenar cartas: **KO**, dado que en la versión actual de la aplicación las cartas ya están puestas por defecto en la tabla, creo que esta funcionalidad no es necesaria

De cara a la última entrega, además de lo que se requiera, quiero terminar de agregar el tema del ganador de la mano (CU15) y que se continúe con la siguiente partida. Esta funcionalidad está implementada en el backend pero no en el frontend. Al completar esta funcionalidad, aunque se deje un poco de lado el tema UI/UX, se podrá jugar, cumpliendo así con un objetivo clave del proyecto.

Como conclusión del estado actual del proyecto, podemos decir que he hecho una mala estimación de las horas asignadas a cada funcionalidad, además de que en el diagrama de Gantt inicial no existen muchas de estas funcionalidades adicionales.

5. Testing y pruebas de usuario

Dado que he tenido muchos problemas con las dependencias del proyecto y que el proyecto no únicamente ha sido el desarrollo de la parte del cliente si no también del desarrollo de la parte servidora (API), he decidido implementar una serie de tests manuales que intentarán abarcar el contenido actual del proyecto.

Para poder realizar estas pruebas, hemos borrado la base de datos del proyecto y levantaremos una base de datos nueva sin ningún dato añadido. Las siguientes pruebas se han realizado tanto sobre un entorno virtual (Pixel 3a) como en un dispositivo físico (POCO M4 Pro, versión 13 de Android)

5.1 Frontend (pruebas manuales)

5.1.1 Sign up

En las siguientes imágenes podemos ver como esta testeado la interfaz de Sign up.

En la primera imagen podemos ver el formulario sin que haya ninguna interacción. En la segunda imagen vemos que después de darle al botón de Sign Up sin introducir ningún dato en el formulario, nos aparece un mensaje diciendo que el nombre es un campo obligatorio. En la tercera imagen vemos que, si únicamente introducimos el nombre y le damos al botón de Sign Up, nos aparece un mensaje diciendo que es necesario introducir un correo electrónico válido.

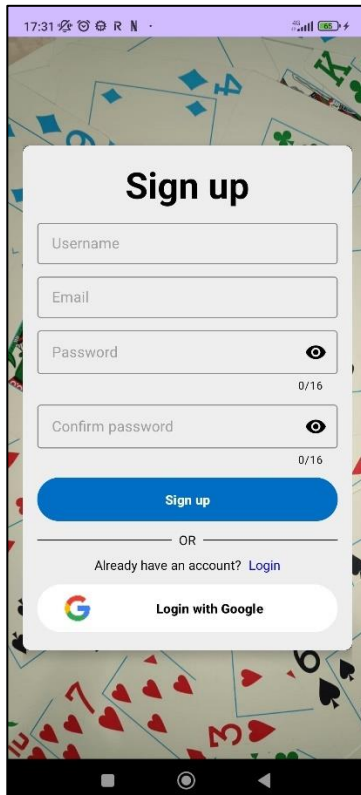


Figura 50. Sign up

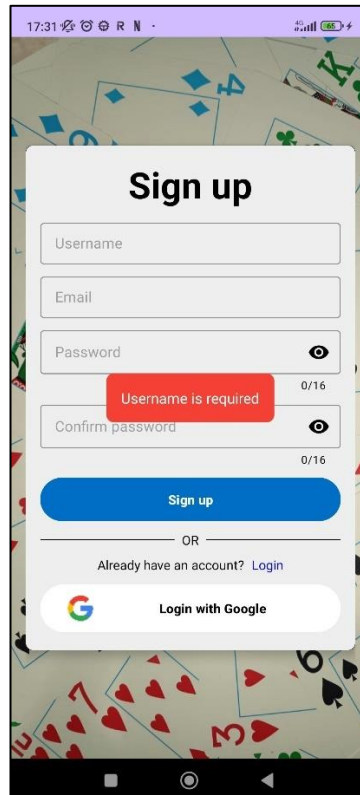


Figura 51. Sign up nombre requerido

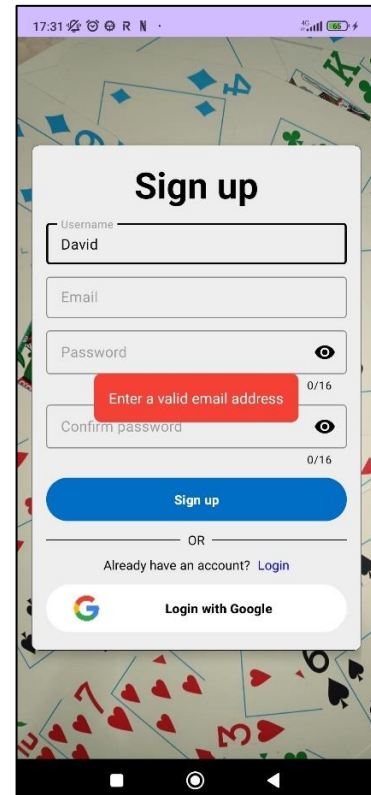


Figura 52. Sign up email válido requerido

En la primera imagen podemos ver que, aunque introduzcamos un campo alfanumérico en el campo email, si no lo detecta como un correo electrónico válido, nos aparecerá un mensaje diciendo que debemos introducir un correo electrónico válido.

En la segunda imagen podemos ver el caso de que hayamos introducido un nombre y un correo electrónico válido y le demos a Sign up, nos aparecerá un mensaje reclamándonos la introducción de una contraseña.

En la tercera imagen podemos ver que, si introducimos un campo en la contraseña, este no será visible por el usuario.

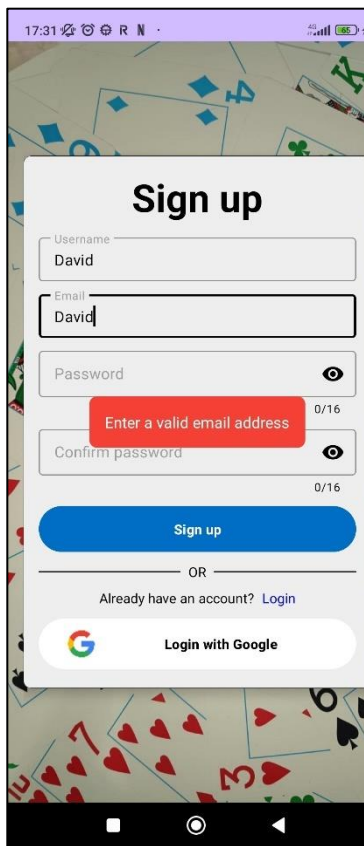


Figura 53. Sign up email inválido

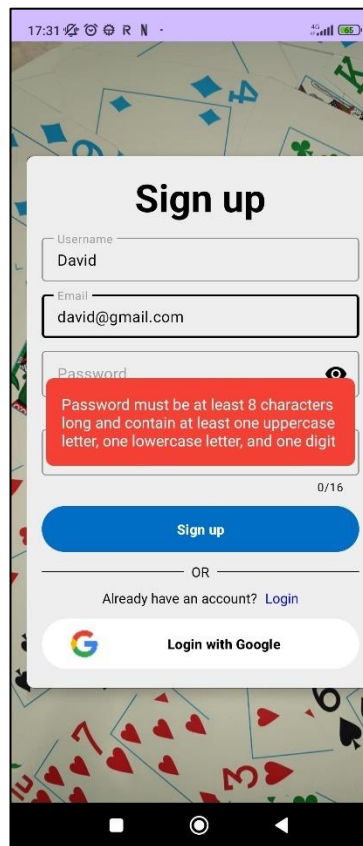


Figura 54. Sign up contraseña requerida

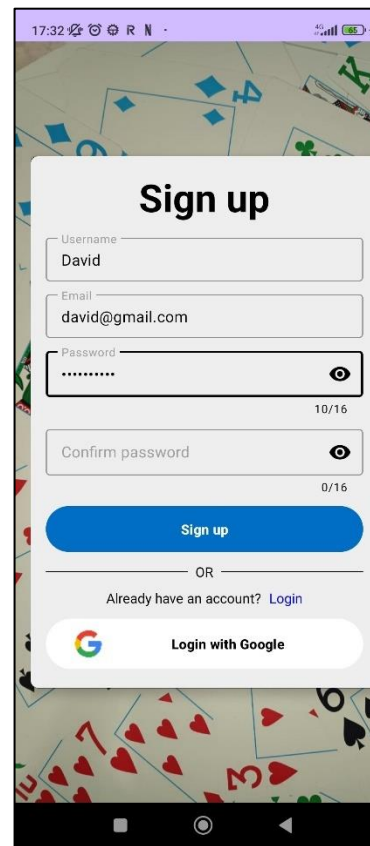


Figura 55. Sign up contraseña oculta para el usuario

En la primera imagen podemos ver que, si le damos al botón del ojo, se nos muestra la contraseña introducida, esto pasa tanto para el campo password como el confirm password. El password necesita cumplir unos criterios de seguridad.

En la segunda imagen podemos ver que si la contraseña introducida en password y en confirm password no coinciden aparecerá un mensaje en el que nos informa de que las contraseñas no coinciden.

En la tercera imagen podemos ver que en el caso de que introduzcamos todos los datos correctamente, se creará un usuario y se redirigirá a la pantalla de Login.

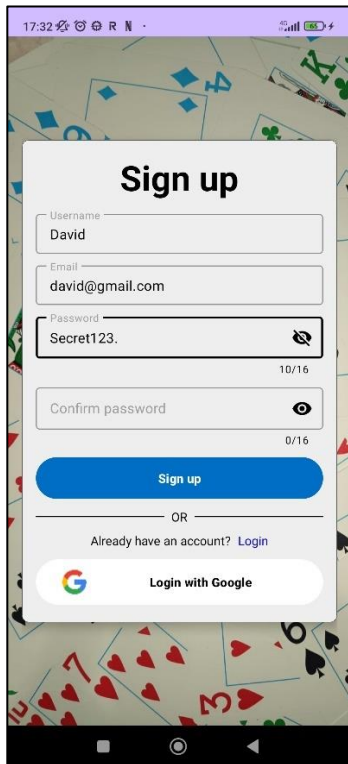


Figura 56. Sign up botón de visualizar la contraseña

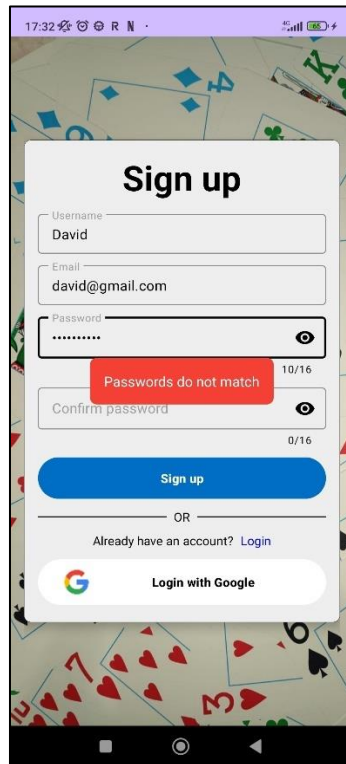


Figura 57. Sign up contraseñas no coincidentes

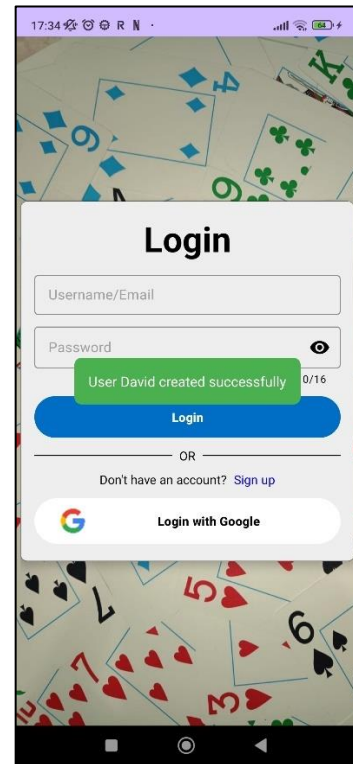


Figura 58. Sign up exitoso

Si ahora miramos en la base de datos, veremos que efectivamente el usuario se ha creado correctamente.

```
_id: ObjectId('664a1c16a6cc574684da7bf1')
id: 1
password: "pbkdf2_sha256$390000$rrHq4dYU38Md08iA44XKClx$Dsi0gvRFBwY/eth1Uq6XfU3IJH..."
last_login: null
is_superuser: false
username: "David"
first_name: ""
last_name: ""
email: "david@gmail.com"
is_staff: false
is_active: true
date_joined: 2024-05-19T15:34:46.941+00:00
```

Figura 59. Usuario creado satisfactoriamente en la base de datos

Continuamos con las validaciones del sign up. En la primera imagen podemos ver que, si intentamos crear un usuario con un nombre ya existente en la base de datos, nos aparece un mensaje informándonos de que ya hay un usuario con ese nombre en la base de datos.

En la segunda imagen podemos ver que si intentamos crear un usuario con un nombre diferente pero un correo electrónico que ya existe en la base de datos nos aparecerá un mensaje informándonos de que el correo electrónico introducido ya existe en la base de datos.

En la tercera imagen podemos observar que, si introducimos un nombre y un correo electrónico diferentes a los existentes en la base de datos, se creará correctamente el usuario.

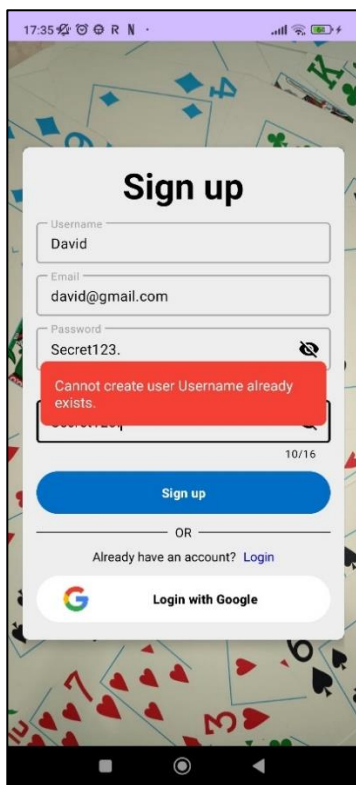


Figura 60. Sign up usuario ya existente

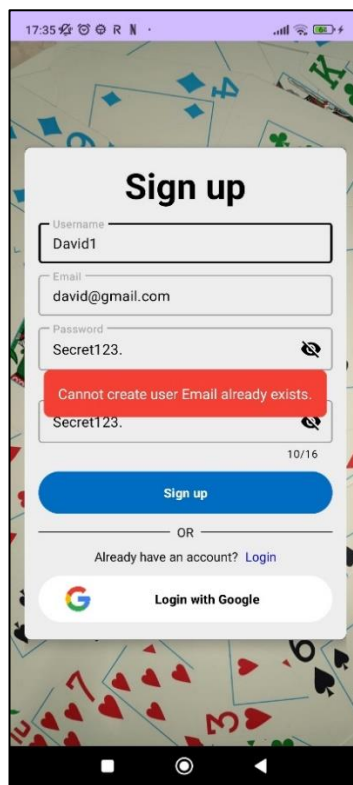


Figura 61. Sign up email ya existente



Figura 62. Sign up nuevo usuario creado correctamente

```
_id: ObjectId('664a1c35a6cc574684da7bf5')
id: 2
password: "pbkdf2_sha256$390000$CKXYdH35XIeP8PCAByVeMf$5ixISZpDoixy4V7EIuyFeK5gs2..."
last_login: null
is_superuser: false
username: "David1"
first_name: ""
last_name: ""
email: "david1@gmail.com"
is_staff: false
is_active: true
date_joined: 2024-05-19T15:35:17.323+00:00
```

Figura 63. Nuevo usuario creado satisfactoriamente

Está última imagen del Sign up nos muestra que hay un botón para hacer login a través de Google, esta función actualmente no está implementada y aparece un mensaje conforme es una funcionalidad que estará disponible en futuras versiones.

Por último, quiero comentar que hay un enlace que te redirige a la interfaz de Login.

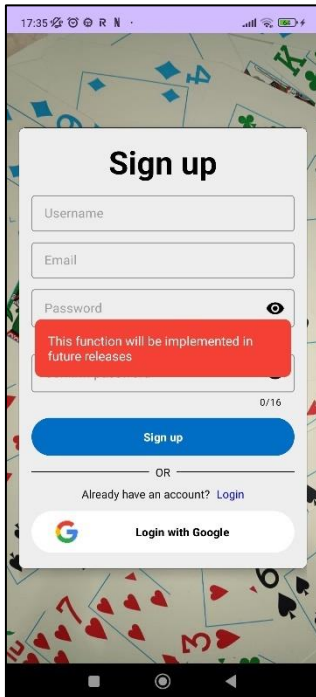


Figura 64. Sign up via google en futura versión

5.1.2 Login

En las siguientes imágenes podemos ver como esta testeado la interfaz de Login.

En la primera imagen, al igual que en el sign up, podemos ver que hay un botón de hacer login via Google, el cual, si lo pulsamos, nos aparecerá un mensaje anunciando que se implementará en siguientes versiones.

En la segunda imagen podemos ver como que, si no introducimos ningún campo, nos aparecerá un mensaje diciendo que el campo username es obligatorio.

En la tercera imagen podemos observar como que, si no se ha introducido una contraseña con cierto nivel de seguridad, aparecerá un mensaje anunciando que la contraseña tiene que seguir un cierto patrón. Este mensaje lo he dejado igual que en el signup para dar una pista al usuario de la contraseña que haya podido introducir para hacer registrarse.

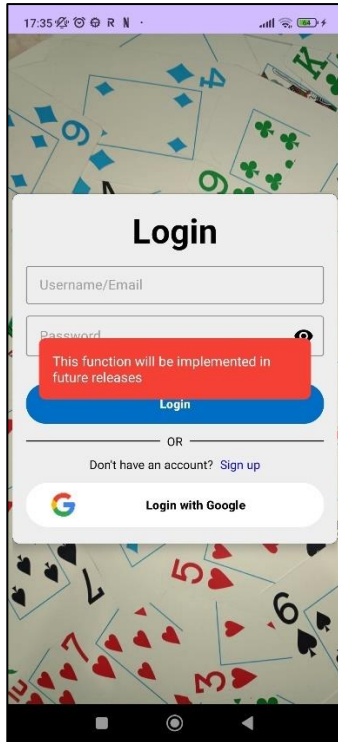


Figura 65. Login via google en futura versión

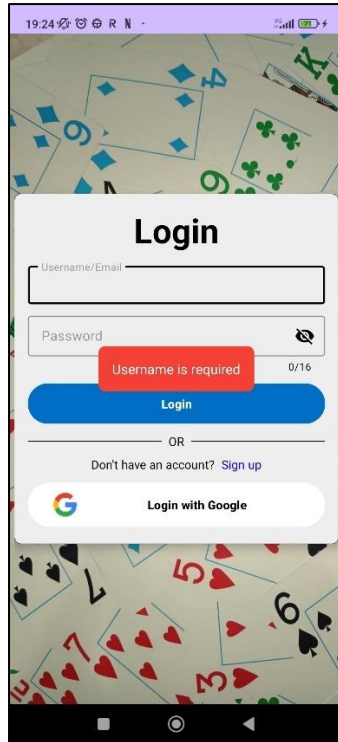


Figura 66. Login username obligatorio

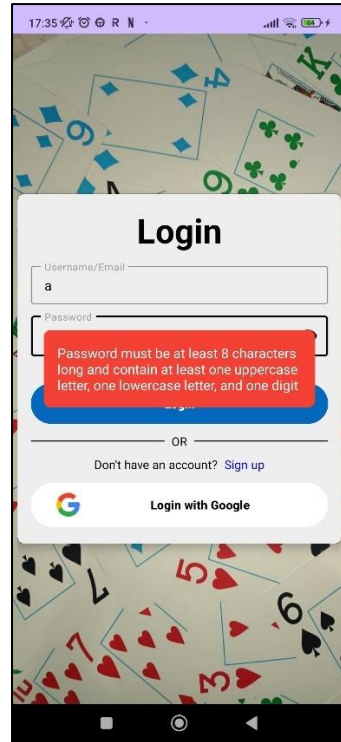


Figura 67. Login contraseña obligatoria

En las siguientes imágenes (primera y segunda) podemos ver que si se introduce un usuario o contraseña incorrectos aparecerá un error. En la tercera imagen podemos ver que si los datos introducidos son correctos se procederá con el login.



Figura 68. Login usuario o contraseña incorrectos



Figura 69. Login usuario o contraseña incorrectos



Figura 70. Login satisfactorio

5.1.3 Main menu

En la primera imagen podemos ver el menú principal del usuario. A simple vista parece un poco simple, pero nos damos cuenta de que al lado de la imagen de perfil nos aparece el nombre del usuario que se ha logeado. También podemos ver que hay 2 botones, uno para crear un juego nuevo y otro para ver las reglas. Posteriormente cuando creamos juegos nuevos irán apareciendo en esta vista.

En la segunda imagen podemos ver que en la base de datos ya se han registrado 2 juegos nuevos, también aparecen ciertos datos que contienen estos juegos como el/los adversarios, el número de sets y puntos actuales, el número de sets y puntos que se tienen que alcanzar y un botón para jugar esa partida.



Figura 71. Menú principal sin juegos creados

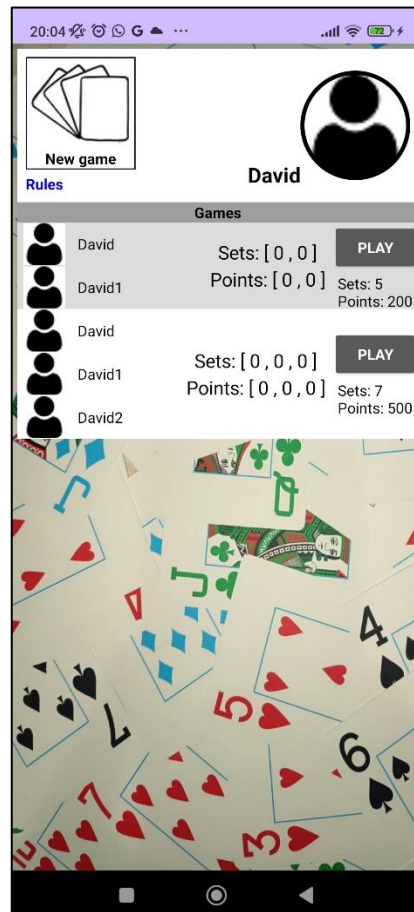


Figura 72. Menú principal con dos juegos creados

5.1.4 Rules

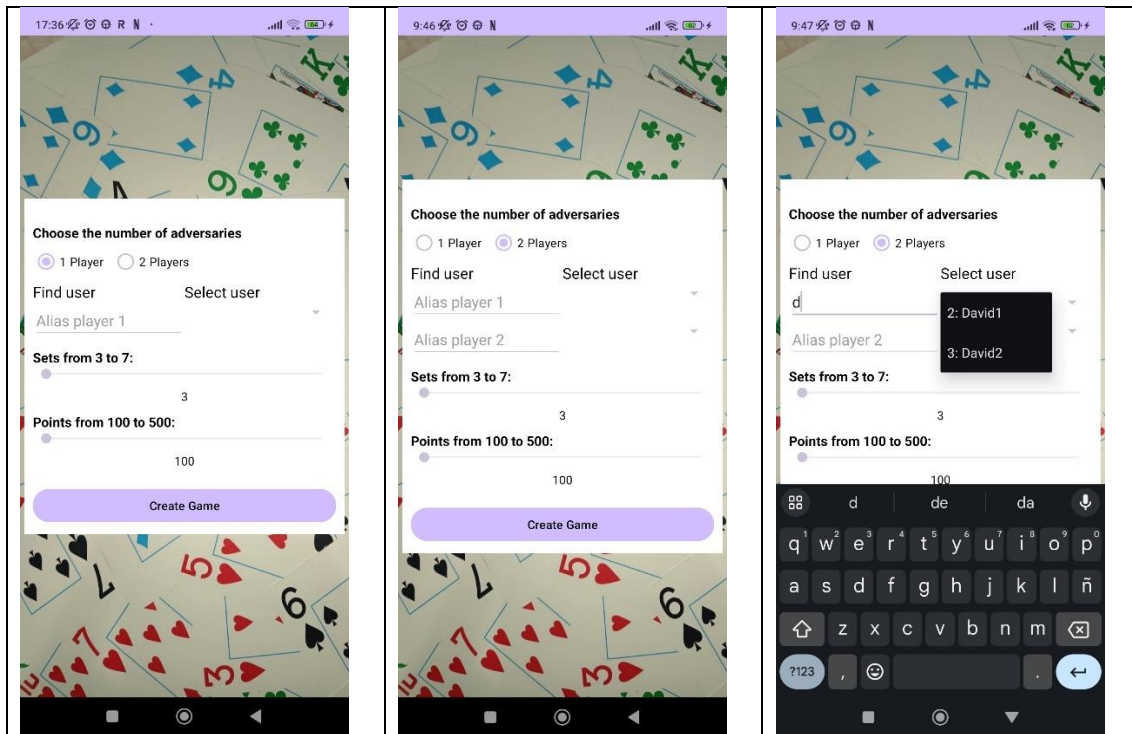
Esta vista únicamente muestra las reglas del juego. Únicamente hay que mencionar que se ha añadido un elemento scroll ya que el contenido de las reglas es superior a la longitud del dispositivo.



Figura 73. Reglas del juego

5.1.5 Create game

En la imagen de la izquierda vemos el formulario para crear un juego, si seleccionamos la opción de 2 players, automáticamente nos añadirá una fila pidiendo el alias del nuevo jugador. En la tercera imagen podemos observar que a medida que introducimos un nombre en la entrada de texto de find user, cuando hacemos click en el desplegable de select user veremos todos los usuarios que coinciden con parte del nombre de nuestra búsqueda. Finalmente, el usuario que elijamos en este desplegable será el usuario contra el que jugaremos.



Crear un juego desencadena una serie de acciones en el backend.

1. Añade un juego en la tabla games con alguno de los datos introducidos en el formulario.
2. Inicializa los jugadores que participaran en este juego, es decir, en la tabla players, se añadirán los jugadores del juego, tanto el que crea la partida como los adversarios.
3. Inicializa los sets de este juego, crea en la tabla sets un nuevo registro con el identificador del juego y los sets iniciados en 0.
4. Inicializa los puntos de este juego, crea en la tabla points un nuevo registro con el identificador del juego y los puntos iniciados en 0.
5. Inicializa la mano
 - 5.1 Se inicializa las manos de los jugadores, a cada jugador se le reparten 12 cartas. Esto se guarda en la tabla players_hands
 - 5.2 Se inicializa las cartas comunitarias, se añaden las 15 cartas comunitarias en la tabla boards.

<code>_id: 1</code>	<code>_id: ObjectId('664a3edea6cc574684da7c01')</code>	<code>_id: ObjectId('664a3edea6cc574684da7c02')</code>	<code>_id: ObjectId('664a3edea6cc574684da7c03')</code>
<code>num_hand: 1</code>	<code>id: 1</code>	<code>game_id: 1</code>	<code>game_id: 1</code>
<code>num_players: 2</code>	<code>game_id: 1</code>	<code>sets_p1: 0</code>	<code>points_p1: 0</code>
<code>num_sets: 5</code>	<code>p1_id: 1</code>	<code>sets_p2: 0</code>	<code>points_p2: 0</code>
<code>num_points: 200</code>	<code>p2_id: 2</code>	<code>sets_p3: 0</code>	<code>points_p3: 0</code>
<code>status: "running"</code>	<code>p3_id: null</code>		
<code>winner: 0</code>			

Figura 74. Inicialización de juego, tabla games, tabla players, tabla sets y tabla points

<code>_id: ObjectId('664a3edea6cc574684da7c01')</code>	<code>_id: ObjectId('664a3edea6cc574684da7c02')</code>
<code>game_id: 1</code>	<code>id: 1</code>
<code>player_id: 1</code>	<code>game_id: 1</code>
<code>num_hand: 1</code>	<code>num_hand: 1</code>
<code>status: "PCH"</code>	<code>flop1: "JC,KC,4C"</code>
<code>hand: "JH,2C,4D,3H,7H,7C,9S,AH,3D,4H,9D,AC"</code>	<code>river1: "JS,2S"</code>
<code>hand1: ""</code>	<code>flop2: "5S,2H,8D"</code>
<code>hand2: ""</code>	<code>river2: "8C,6S"</code>
<code>hand3: ""</code>	<code>flop3: "KS,6D,JD"</code>
	<code>river3: "AS,4S"</code>
<hr/>	
<code>_id: ObjectId('664a3edea6cc574684da7c02')</code>	
<code>game_id: 1</code>	
<code>player_id: 2</code>	
<code>num_hand: 1</code>	
<code>status: "PCH"</code>	
<code>hand: "2D,7D,QH,7S,6H,3C,AD,TC,8H,QD,9C,5H"</code>	
<code>hand1: ""</code>	
<code>hand2: ""</code>	
<code>hand3: ""</code>	

Figura 75. Inicialización de juego, tabla players_hands y tabla boards

5.1.6 InGame

En las siguientes imágenes podemos ver como se ve la pantalla de cuando estas dentro de una partida, las dos imágenes de la izquierda son imágenes de la primera partida, la imagen de la derecha es de la segunda partida, en la parte superior aparece el estado del adversario, si está jugando o esperando a que hagamos nuestra jugada. Las cartas de en medio son las comunitarias y las que hay abajo son nuestras cartas, en la segunda imagen se puede observar que estas imágenes tienen un drag and drop para colocarlas donde queramos, así pues, nuestras cartas superiores valdrán para las comunitarias superiores, y así respectivamente. Por último, el botón de play enviará nuestra jugada al backend, guardando nuestra mano y cambiando el estado de nuestro juego.



Figura 76. Ingame contra un jugador

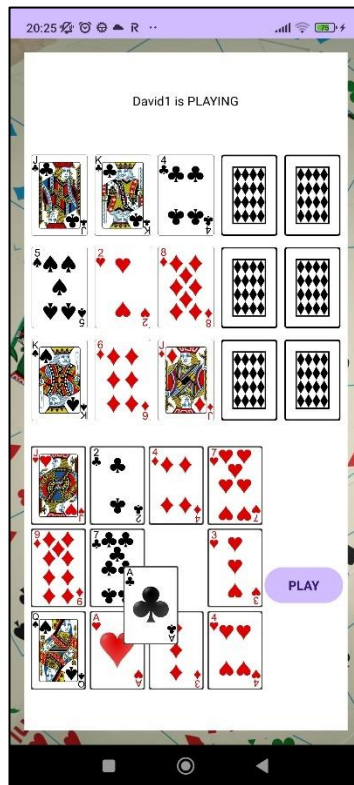


Figura 77. Ingame coger y soltar carta



Figura 78. Ingame contra dos jugadores

En la siguiente imagen podemos ver que las cartas obtenidas para el juego de 2 jugadores son efectivamente las que hay en la base de datos.

```
_id: ObjectId('664a3efea6cc574684da7c07')
game_id: 2
player_id: 1
num_hand: 1
status: "PCH"
hand: "7C,2D,9D,5D,TS,4H,QS,7S,TD,JH,4S,TH"
hand1: ""
hand2: ""
hand3: ""
```

Figura 79. Cartas juego 2, jugador 1, primera mano

5.2 Backend (pruebas automáticas)

Para la realización de las pruebas automatizadas de la API se ha utilizado el mismo framework Django. Este permite la realización y ejecución de tests automáticos en la aplicación.

Las pruebas implementadas son de dos tipos principalmente:

- Tests de urls: Estas pruebas consisten en verificar que las llamadas que se realizan a la API se redirijan correctamente a la vista adecuada.
- Tests de vistas: Estas pruebas consisten en verificar que los datos recibidos en las peticiones se procesen de la manera adecuada. Para poder realizar estas pruebas se han tenido que generar datos de muestra para simular las peticiones y los resultados.

```
class TestUsersUrls(SimpleTestCase):  
  
    def test_signup_url(self):  
        url = reverse('signup')  
        self.assertEqual(resolve(url).func, signup)
```

Figura 80. Ejemplo de test de url

```
##### sign up test #####  
def test_signup_success_view(self):  
    url = reverse('signup')  
    data = {  
        'username': self.new_username,  
        'email': self.new_email,  
        'password': self.new_password  
    }  
    response = self.client.post(url, data)  
    self.assertEqual(response.status_code, 201)  
    self.assertIn('username', response.data)  
    self.assertEqual(response.data['username'], self.new_username)
```

Figura 81. Ejemplo de test de vista

En la imagen podemos ver como se ha ejecutado un total de 35 tests satisfactoriamente.

```
(omahaflop-env) C:\uoc\2023_02\TFG\project\OmahaFlopAPI>py manage.py test  
Found 35 test(s).  
Creating test database for alias 'default'...  
This version of djongo does not support "NULL, NOT NULL column validation check" fully. Visit https://nesdis.github.io/djongo/support/  
This version of djongo does not support "schema validation using CONSTRAINT" fully. Visit https://nesdis.github.io/djongo/support/  
This version of djongo does not support "schema validation using KEY" fully. Visit https://nesdis.github.io/djongo/support/  
This version of djongo does not support "schema validation using REFERENCES" fully. Visit https://nesdis.github.io/djongo/support/  
This version of djongo does not support "COLUMN DROP NOT NULL " fully. Visit https://nesdis.github.io/djongo/support/  
This version of djongo does not support "DROP CASCADE" fully. Visit https://nesdis.github.io/djongo/support/  
System check identified no issues (0 silenced).  
.....  
-----  
Ran 35 tests in 20.261s  
OK  
Destroying test database for alias 'default'...
```

Figura 82. Ejecución de los tests via terminal

6. Conclusiones

Desarrollar esta aplicación móvil ha sido una experiencia desafiante pero gratificante. A lo largo del proceso, han surgido varios puntos clave, tanto positivos como negativos.

Uno de los principales retos fue la falta de experiencia previa en el desarrollo de aplicaciones móviles y la escasa información disponible sobre el uso de Django con MongoDB a través de Djongo. Esto requirió una investigación y aprendizaje extensivos.

La gestión de los diferentes estados del juego, saber quién ha jugado, recuperar el estado del juego, mantener el número de mano correcto, recuperar la resolución de las diferentes manos correctamente ha resultado una de las tareas más complejas a las que me he enfrentado.

Aunque se ha seguido la planificación prevista, ha habido una desviación significativa respecto a la planificación inicial. Gracias a la metodología utilizada, se ha podido priorizar las funcionalidades más relevantes y descartar las que menos valor aportaban a la aplicación. Pudiendo así obtener un resultado satisfactorio.

A pesar de estos desafíos, se ha logrado una parte del objetivo inicial, resultando en una aplicación funcional. Este es un hito significativo y demuestra un progreso hacia el objetivo general.

El conocimiento y la experiencia adquiridas al investigar las diferentes tecnologías y librerías ha sido una experiencia beneficiosa para futuros proyectos y al desarrollo personal.

Este proyecto ha puesto de manifiesto la dificultad inherente en el desarrollo de cualquier proyecto, por pequeño que sea, cuando es llevado a cabo por una única persona.

7. Glosario

Android: Sistema operativo móvil desarrollado por Google, ampliamente utilizado en dispositivos móviles como teléfonos inteligentes y tabletas.

Backend: Parte de un sistema informático que se encarga de procesar los datos y la lógica de negocio de una aplicación.

Cartas comunitarias: Cartas que se reparten boca arriba en el centro de la mesa y que pueden ser utilizadas por todos los jugadores en una partida de póquer.

Django: Es un framework web de alto nivel basado en Python que facilita el desarrollo rápido y seguro de aplicaciones web, ofreciendo una arquitectura MVT y una amplia gama de características integradas.

Djongo: Es una herramienta de integración para Django que permite el uso de MongoDB como base de datos en lugar de las bases de datos relacionales tradicionales.

Django OAuth Toolkit: Es una biblioteca de Django que facilita la implementación de autenticación y autorización OAuth2 en aplicaciones web.

Flop: Tres primeras cartas comunitarias que se reparten boca arriba en la mesa, después de la primera ronda de apuestas.

Frontend: La parte de un sistema informático que interactúa directamente con el usuario final.

Gantt: Tipo de diagrama utilizado en la gestión de proyectos para representar visualmente el cronograma de tareas y actividades planificadas.

Kotlin: Lenguaje de programación desarrollado por JetBrains y adoptado oficialmente por Google para el desarrollo de aplicaciones Android.

Mano: Combinación de cartas que un jugador tiene en su posesión y que utiliza para hacer jugadas durante una partida de póquer.

Modelo: Representa los datos y la lógica de negocio de la aplicación.

MongoDB: Es una base de datos NoSQL que almacena datos en documentos JSON flexibles, siendo altamente escalable y adaptable para diversas aplicaciones.

Omaha: Variante del póquer en la que cada jugador recibe cuatro cartas boca abajo y debe utilizar dos de ellas junto con tres cartas comunitarias para formar la mejor mano posible.

Palo: Una de las cuatro categorías en las que se dividen las cartas de la baraja (picas, corazones, diamantes y tréboles).

Póquer: Juego de cartas en el que los jugadores apuestan sobre la base de las combinaciones de cartas que poseen, intentando ganar el bote.

Pot limit: Estructura de apuestas en la que los jugadores pueden apostar una cantidad máxima igual al tamaño del bote en un momento dado.

Preflop: Período de una mano de póquer que ocurre antes de que se repartan las cartas comunitarias en la mesa.

Retrofit: Es una biblioteca de cliente HTTP para Android que simplifica la comunicación con servicios web RESTful, permitiendo realizar solicitudes de manera sencilla y eficiente.

River: Quinta y última carta comunitaria repartida en la mesa en una partida de póquer, después del turn, y seguida de la última ronda de apuestas.

Sprint: Intervalo de tiempo definido en la metodología ágil durante el que se desarrolla y entrega de un conjunto específico de funcionalidades o características en un proyecto.

Turn: Cuarta carta comunitaria repartida en la mesa en una partida de póquer, después del flop y antes del river, seguida de una ronda de apuestas.

Vista: Es la interfaz de usuario que muestra los datos al usuario y recibe sus interacciones.

Vista-Modelo: Actúa como intermediario entre la Vista y el Modelo, preparando los datos para su presentación en la Vista y gestionando las interacciones del usuario.

8. Bibliografía

Alonso Martínez, Noelia. «Desarrollo de aplicación móvil Android: NutriKNow!»
<https://openaccess.uoc.edu/bitstream/10609/126967/13/nmartinezaloTFG0121memoria.pdf>

[Consultada 04/03/2024]

Parra Zurita, Fernando. «Mi Equipo SNP»
<https://openaccess.uoc.edu/bitstream/10609/106626/8/chamarretaTFG0120memoria.pdf>

[Consultada 04/03/2024]

«Póquer»
<https://es.wikipedia.org/wiki/P%C3%B3quer>

[Consultada 04/03/2024]

[1] «El póker: Origen y curiosidades»
<https://elrethistorico.com/el-poker-origen-y-curiosidades/>

[Consultada 07/03/2024]

<https://www.figma.com/community/file/923140611594993345>

[Consultada 22/04/2024]

<https://help.figma.com/hc/en-us/articles/4405269899287--Beginner-1-Explore-ideas>

[Consultada 22/04/2024]

[https://www.figma.com/file/bXghb9aqO24QsWgdeVg5tt/Login-%26-Signup-Form-Android-By-VSL-\(Community\)?type=design&node-id=0-1&mode=design&t=C0ALxc82VyWnhFmd-0](https://www.figma.com/file/bXghb9aqO24QsWgdeVg5tt/Login-%26-Signup-Form-Android-By-VSL-(Community)?type=design&node-id=0-1&mode=design&t=C0ALxc82VyWnhFmd-0)

[Consultada 24/04/2024]

[https://www.figma.com/file/JttVop2PW7eRR374joVm8/Android-13-Status-Bar-\(Community\)?type=design&mode=design&t=C0ALxc82VyWnhFmd-0](https://www.figma.com/file/JttVop2PW7eRR374joVm8/Android-13-Status-Bar-(Community)?type=design&mode=design&t=C0ALxc82VyWnhFmd-0)

[https://www.figma.com/file/pLL4cF3bZ89utUwsfeGrLK/Price-Range-Prototype-\(Community\)?type=design&mode=design&t=C0ALxc82VyWnhFmd-0](https://www.figma.com/file/pLL4cF3bZ89utUwsfeGrLK/Price-Range-Prototype-(Community)?type=design&mode=design&t=C0ALxc82VyWnhFmd-0)

[Consultada 25/04/2024]

<https://cursokotlin.com/mvvm-en-android-con-kotlin-livedata-y-view-binding-android-architecture-components/>

[Consultada 06/04/2024]

<https://alainnicolastello.medium.com/cual-es-la-mejor-arquitectura-para-android-con-kotlin-477b30c20e2a>
[Consultada 06/04/2024]

<https://square.github.io/retrofit/>
[Consultada 06/04/2024]

<https://pypi.org/project/djongo/>
[Consultada 06/04/2024]

<https://github.com/rishi772001/Django-MongoDb-CRUD-RESTAPI>
[Consultada 09/04/2024]

<https://stackoverflow.com/questions/28854468/docker-compose-to-run-django-with-mongodb>
[Consultada 09/04/2024]

<https://www.youtube.com/watch?v=MIS6y9kffBs>
[Consultada 10/04/2024]

<https://charlascylon.com/2013-07-18-tutorial-mongodb-operaciones-de-actualizaci%C3%B3n-de>
[Consultada 10/04/2024]

<https://dev.to/shlokabhalgat/connecting-django-with-mongodb-all-the-versioning-shenanigans-dissected-2022-edition-13km>
[Consultada 10/04/2024]

<https://www.redeszone.net/tutoriales/servidores/como-crear-clave-ssh-windows-10/>
[Consultada 14/04/2024]

<https://www.djongomapper.com/get-started/>
[Consultada 16/04/2024]

<https://www.django-rest-framework.org/api-guide/serializers/>
[Consultada 17/04/2024]

<https://docs.djangoproject.com/en/5.0/topics/serialization/>
[Consultada 17/04/2024]

<https://www.geeksforgeeks.org/serializers-django-rest-framework/>
[Consultada 17/04/2024]

<https://www.mongodb.com/resources/products/compatibilities/mongodb-and-django>

[Consultada 25/04/2024]

<https://github.com/HenryRLee/PokerHandEvaluator>

[Consultada 26/04/2024]

<https://pypi.org/project/phevaluator/>

[Consultada 26/04/2024]

<https://github.com/docker/awesome-compose>

[Consultada 28/04/2024]

<https://developer.android.com/codelabs/basic-android-kotlin-compose-first-app?hl=es-419#0>

[Consultada 06/05/2024]

<https://github.com/hdodenhof/CircleImageView>

[Consultada 06/05/2024]

<https://www.geeksforgeeks.org/how-to-create-a-circularimageview-in-android-using-hdodenhof-library/>

[Consultada 06/05/2024]

<https://square.github.io/retrofit/>

[Consultada 08/05/2024]

<https://square.github.io/picasso/>

[Consultada 11/05/2024]

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

[Consultada 13/05/2024]

<https://docs.djangoproject.com/en/5.0/topics/testing/>

[Consultada 20/05/2024]

<https://www.youtube.com/@thedumbfounds767>

[Consultada 20/05/2024]

<https://www.veryicon.com/icons>

[Consultada 05/06/2024]

9. Anexos

9.1 Resultado encuestas.

Este apartado engloba los resultados de las encuestas realizadas en el apartado de 2.1.1.2 Encuestas del Diseño Centrado en el Usuario.