

“PARTYUPP”

**PROYECTO DE DESARROLLO DE APP DE OCIO EN GRUPO PARA
DISPOSITIVOS MÓVILES ANDROID**



CICLO FORMATIVO

SUPERIOR

DE GRADO

DESARROLLO DE APLICACIONES MULTIPLATAFORMA

AUTORES:

**RICARDO MARIN
VITALIY POPOVYCH
ELOI GONZALEZ**

Licencia

Esta obra está bajo una licencia Reconocimiento-Compartir bajo la misma licencia 3.0 España de Creative Commons. Para ver una copia de esta licencia, visite

<http://creativecommons.org/licenses/by-sa/3.0/es/> o envíe una carta a Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Disclaimer

El objetivo de "PartyUpp" es ofrecer una experiencia de entretenimiento socialmente responsable y centrada en la diversión compartida entre amigos. Es necesario enfatizar que esta aplicación no promueve ni incita de ninguna manera al consumo excesivo de alcohol ni a comportamientos irresponsables.

Es importante fomentar un ambiente de diversión seguro, donde los usuarios puedan disfrutar de juegos y desafíos sin poner en riesgo su bienestar ni el de los demás. Es por eso que "PartyUpp" está diseñada cuidadosamente para garantizar que sea una plataforma que promueva la interacción social positiva y el disfrute responsable.

DEDICATORIAS/AGRADECIMIENTOS

Los autores del proyecto del proyecto, Ricardo Marín y Vitaliy Popovych Eloi González , quieren agradecer a Isreal Domingo, compañero en el Ciclo Formativo Grado Superior de Desarrollo de Aplicaciones Multiplataforma en el IES Juan de la Cierva (Madrid), su trabajo y colaboración en las primeras fases de desarrollo de la aplicación.

RESUMEN

Autores del proyecto

Este proyecto final del ciclo de Desarrollo de Aplicaciones Multiplataforma (DAM) se centra en la creación y desarrollo de una aplicación de ocio para dispositivos móviles Android llamada "PartyUpp". Vitaliy Popovych, Ricardo Marín, Eloi González alumnos de dicho ciclo en el Instituto Juan de la Cierva (Madrid).

Cobertura de la demanda y público objetivo

El producto está diseñado para ofrecer entretenimiento en grupo a través de una serie de desafíos que los jugadores deben superar para evitar beber demasiado. Busca combinar la diversión de los juegos de fiesta con una plataforma digital de ocio grupal que facilite la interacción y la competición entre amigos.

El proyecto nace de la necesidad detectada de ofrecer una experiencia mejorada de las aplicaciones ya existentes, tanto en lo relativo al diseño de las pruebas y a la experiencia de usuario, como a su disponibilidad en castellano (además de inglés).

Por tanto, "PartyUpp" se presenta como una opción para grupos de jóvenes que desean pasar un gran rato entre amigos, aprovechando así la creciente demanda de este tipo de productos digitales, como demuestra el éxito de aplicaciones como "Guess up" o "Pícolo".

1. Penetración en el Mercado de Entretenimiento Digital:

- Posicionar "PartyUpp" como una opción destacada para reuniones sociales entre aplicaciones de entretenimiento.
- Capturar una cuota significativa del mercado de aplicaciones de juegos para fiestas, ofreciendo una experiencia divertida, social y claramente diferenciada de otras aplicaciones similares.

2. Fidelización de Usuarios:

- Desarrollar mecanismos de fidelización, como pueden ser las opciones de personalización de la experiencia o la posibilidad de almacenar y consultar los resultados de las partidas.

3. Escalabilidad y Expansión:

- Diseñar la aplicación con una arquitectura escalable que permita la incorporación de nuevas funcionalidades y/o maneras de monetizar los contenidos, ya sea a través de publicidad o de opciones de suscripción *freemium*.
- Preparar el lanzamiento en diferentes idiomas para ampliar el alcance a nivel internacional.

Estructura

API

La API constituye el núcleo de la aplicación, gestionando el registro de usuarios, la creación y seguimiento de partidas, y el control de inicio de sesión. Sus características principales incluyen:

- **Registro de Usuarios:**
 - Endpoint para crear nuevos usuarios con validación de datos y almacenamiento seguro de información personal.
 - Implementación de medidas de seguridad como la encriptación de contraseñas.
- **Gestión de Partidas:**
 - Creación de partidas nuevas.
 - Almacenamiento de progreso y resultados en la base de datos de la aplicación.
- **Autenticación:**
 - Uso de tokens de autenticación para sesiones seguras, evitando accesos no autorizados y protegiendo la privacidad del usuario.
 - Mecanismos de renovación de tokens y cierre de sesión seguro.

Pantallas de la Aplicación

- **Pantallas de Login y Registro:**
 - **Pantalla de Login:**
 - Formulario de inicio de sesión que solicita el nombre de usuario, el correo electrónico y la contraseña.
 - Opción de logearse sin registro previo, facilitando un uso más casual de la aplicación para usuarios que así lo decidan.
 - **Pantalla de Registro:**
 - Formulario de registro con campos para nombre, correo electrónico, nombre de usuario y contraseña.
 - Validaciones en tiempo real para asegurar la calidad de los datos ingresados, como la fuerza de la contraseña, la unicidad del nombre de usuario o la validez de la dirección de correo electrónico.
- **Pantalla Inicial:**
 - Compuesta por un menú que permite navegar hacia la galería y los ajustes. Esta pantalla es el punto de partida para los usuarios una vez han iniciado sesión, facilitando el acceso a las diferentes secciones de la aplicación.

- **Pantalla de Selección de Jugadores:**

- Los usuarios pueden seleccionar a los jugadores que participarán en la partida. Esta pantalla incluye opciones para agregar o eliminar jugadores y recibir sugerencias de jugadores para añadir a la partida.

- **Pantalla de Foto Inicial:**

- Antes de comenzar la partida, el grupo se toma una foto que servirá como recuerdo de la sesión. Esta foto se podrá posteriormente guardar en la galería de la aplicación.

- **Pantalla de Pruebas:**

- Aquí se muestran sucesivamente los desafíos que los jugadores deben superar, junto a un botón que les permitirá avanzar a la siguiente una vez hayan completado la actual.

En conjunto, "PartyUpp" es una aplicación diseñada para transformar las reuniones sociales en experiencias donde la diversión está garantizada. La estructura de la aplicación y el desarrollo cuidadoso de su API aseguran que sea segura, fácil de usar y escalable, preparada para adaptarse y crecer según las necesidades del mercado y las expectativas tanto de los desarrolladores como de los usuarios.

ABSTRACT

Market Coverage and Target Audience

This product is designed to offer group entertainment through a series of challenges that players must overcome to avoid drinking too much. It seeks to combine the fun of party games with a digital group leisure platform that facilitates interaction and competition among friends.

The project stems from the identified need to offer an enhanced experience compared to existing applications, both in terms of the design of the challenges and the user experience, as well as its availability in Spanish (in addition to English).

Therefore, "PartyUpp" is presented as an option for groups of young people who want to have a great time with friends, thus taking advantage of the growing demand for this type of digital product, as demonstrated by the success of applications such as "Guess Up" or "Picolo".

1. Penetration in the Digital Entertainment Market:

- Position "PartyUpp" as a prominent option for social gatherings among entertainment applications.
- Capture a significant share of the party game app market by offering a fun, social, and clearly differentiated experience from other similar applications.

2. User Loyalty:

- Develop loyalty mechanisms, such as options to personalize the experience or the ability to store and view game results.

3. Scalability and Expansion:

- Design the application with a scalable architecture that allows the incorporation of new functionalities and/or ways to monetize the content, whether through advertising or freemium subscription options.
- Prepare for launch in different languages to expand the reach internationally.

Structure

API

The API constitutes the core of the application, managing user registration, game creation and tracking, and login control. Its main features include:

- **User Registration:**
 - Endpoint to create new users with data validation and secure storage of personal information.
 - Implementation of security measures such as password encryption.

- **Game Management:**
 - Creation of new games.
 - Storage of progress and results in the application database.
- **Authentication:**
 - Use of authentication tokens for secure sessions, preventing unauthorized access and protecting user privacy.
 - Mechanisms for token renewal and secure logout.

Application Screens

- **Login and Registration Screens:**
 - **Login Screen:**
 - Login form requesting username, email, and password.
 - Option to log in without prior registration, facilitating more casual use of the application for users who choose to do so.
 - **Registration Screen:**
 - Registration form with fields for name, email, username, and password.
 - Real-time validations to ensure the quality of the entered data, such as password strength, username uniqueness, or email address validity.
- **Home Screen:**
 - Composed of a menu that allows navigation to the gallery and settings. This screen is the starting point for users once they have logged in, facilitating access to the different sections of the application.
- **Player Selection Screen:**
 - Users can select the players who will participate in the game. This screen includes options to add or remove players and receive suggestions of players to add to the game.
- **Initial Photo Screen:**
 - Before starting the game, the group takes a photo that will serve as a memory of the session. This photo can later be saved in the application gallery.
- **Challenge Screen:**
 - Here, the challenges that players must overcome are displayed successively, along with a button that allows them to move on to the next one once they have completed the current one.

Overall, "PartyUpp" is an application designed to transform social gatherings into experiences where fun is guaranteed. The structure of the application and the careful development of its API ensure that it is secure, easy to use, and scalable, prepared to adapt and grow according to the market needs and the expectations of both developers and users.

Índice de contenido

Cobertura de la demanda y público objetivo	5
Pantallas de la Aplicación	6
2 INTRODUCCIÓN	14
Propósito del Proyecto	14
2.1.1 Fases de Desarrollo del Proyecto	14
3 NECESIDADES DEL SECTOR PRODUCTIVO	17
3.1 Análisis de la situación actual.....	17
2.2 Necesidades del Cliente y Oportunidad de Negocio	18
2.3 El Nuevo Proyecto: "PartyUpp"	19
Elementos diferenciadores	20
Entorno Específico	20
Herramientas a Utilizar	21
Recursos tanto Humanos y Materiales Necesarios	21
2.3.3 Obligaciones Fiscales, Legales, Laborales y/o de Prevención de Riesgos	21
Por el momento, "PartyUpp" es una aplicación desarrollada en el ámbito académico, y aunque tiene proyección de comercialización, aún no cuenta con planes respectivos a obligaciones fiscales, laborales y/o de Prevención de Riesgos.	21
2.3.4 Ayudas/Subvenciones.....	22
4 DISEÑO DEL PROYECTO	23
4.1 Fases del proyecto	23
4.1.1 Análisis.....	23
Requisitos Funcionales.....	23
Requisitos No Funcionales	24
Alcance del Proyecto	25
Restricciones del Proyecto	26
4.1.2 Diseño	27
Diseño de Requisitos Funcionales.....	27
Estructura de la Aplicación	29
Diseño de Componentes	30
4.1.3 Implementación.....	32
Estructura de la Aplicación	35
Diseño de Componentes	36
4.1.4 Pruebas.....	40
Caso de Prueba 1	40
Caso de Prueba 2	41
Caso de Prueba 3	41
Caso de Prueba 4	41
Caso de Prueba 5	42
Caso de Prueba 6	42
Caso de Prueba 7	43
Caso de Prueba 8	43

Caso de Prueba 9	44
Caso de Prueba 10	44
.....	44
4.2 Objetivos futuros	44
4.2.1.1 Objetivos de Desarrollo Técnico	44
Objetivos de Negocio.....	45
4.2.2 3.3 Previsión de los Recursos Materiales y Humanos Necesarios	46
3.4 Presupuesto Económico	47
4.2.3 Recursos Humanos.....	47
4.2.4 Recursos Materiales.....	48
4.2.5 Otros Gastos	48
4.2.6 Presupuesto Total	48
4.2.7 Planificación de la Ejecución del Proyecto	48
4.2.7.1 Fase de Análisis.....	49
4.2 Fase de Diseño.....	50
4.3 Fase de Implementación.....	50
4.2.7.2 Recursos Humanos	51
Recursos Materiales	52
5 DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN.....	53
5.1.1.1 Procedimiento de Evaluación de las Incidencias	53
4.2 Procedimiento para Gestionar los Posibles Cambios en los Recursos y en las Actividades.....	54
6 INTERFAZ DE USUARIO	56

Índice de figuras

Figura 1 – Interfaz de inicio de sesión.....	56
Figura 2 – Interfaz de registro de usuarios.....	61
Figura 3 – Interfaz de inicio.....	64
Figura 4 – Interfaz de la galería.....	68
Figura 5 – Interfaz del perfil.....	71
Figura 6 – Interfaz de selección de jugadores.....	74
Figura 7 – Interfaz de foto inicial.....	77
Figura 8 – Interfaz de partida.....	79
Figura 9 – Interfaz de final de la partida.....	81
Figura 10 – Interfaz de foto final.....	83

1

INTRODUCCIÓN

Este documento responde a la realización del módulo de Proyecto del CFGS en Desarrollo de Aplicaciones Multiplataforma (DAM). El módulo de Proyecto complementa, la formación establecida para el resto de los módulos profesionales que integran el título en las funciones de análisis del contexto, diseño del proyecto y organización de la ejecución.

Propósito del Proyecto

El propósito de "PartyUpp" es proporcionar una plataforma digital innovadora que transforme las reuniones sociales en experiencias dinámicas y memorables mediante juegos y desafíos interactivos. La aplicación está diseñada para jóvenes que buscan entretenimiento de calidad en un entorno social, facilitando la interacción y la competición amistosa entre amigos.

Al combinar la diversión de los juegos de fiesta con una interfaz intuitiva y personalizable, "PartyUpp" busca cubrir una brecha en el mercado actual de aplicaciones de ocio, especialmente en el segmento de habla hispana. Además, el proyecto tiene como objetivo fomentar la fidelización de usuarios y crear una plataforma escalable que pueda evolucionar y expandirse a nivel internacional.

1.1.1

Fases de Desarrollo del Proyecto

El desarrollo de "PartyUpp" se divide en varias fases clave para asegurar que la aplicación sea funcional, segura y atractiva para los usuarios. A continuación, se enumeran las principales fases de desarrollo del proyecto:

1. Investigación y Análisis:

- **Objetivo:** Comprender las necesidades del mercado y definir los requisitos del proyecto.
- **Tareas:**
 - Estudio de mercado y análisis de la competencia.
 - Encuestas y entrevistas con el público objetivo para identificar preferencias y necesidades.
 - Recolección de datos sobre tendencias tecnológicas y de entretenimiento.

2. Planificación y Diseño:

- **Objetivo:** Establecer la estructura y el diseño de la aplicación.
- **Tareas:**
 - Creación del plan de proyecto y cronograma de desarrollo.
 - Diseño de la arquitectura de la aplicación y definición de las funcionalidades clave.
 - Desarrollo de prototipos de las pantallas principales.
 - Revisión y feedback continuo con los diferentes miembros del proyecto.

3. Desarrollo de la API:

- **Objetivo:** Construir el núcleo de la aplicación que gestionará las funciones esenciales.
- **Tareas:**
 - Implementación del sistema de registro de usuarios con medidas de seguridad.
 - Desarrollo de la funcionalidad de gestión de partidas y almacenamiento de resultados.
 - Configuración de la autenticación mediante tokens y mecanismos de seguridad.

4. Desarrollo Frontend:

- **Objetivo:** Crear la interfaz de usuario interactiva y amigable.
- **Tareas:**
 - Desarrollo de las pantallas de login y registro con validaciones en tiempo real.
 - Implementación de la pantalla inicial y el menú de navegación.
 - Creación de la pantalla de selección de jugadores y la pantalla de pruebas.
 - Integración con la API para asegurar la funcionalidad completa de la aplicación.

5. Mantenimiento y Actualización:

- **Objetivo:** Mantener la aplicación actualizada y mejorar continuamente la experiencia del usuario.
- **Tareas:**
 - Desarrollo inteligente dejando siempre abierta a la escalabilidad de la aplicación.
 - Preparación de la aplicación para su expansión a nuevos mercados e idiomas.

En conjunto, estas fases aseguran que "PartyUpp" no solo sea una aplicación innovadora y divertida, sino también una plataforma robusta, segura y completamente adaptable a las necesidades cambiantes del mercado, así como a las expectativas y preferencias de los usuarios.

Este enfoque integral permite que la aplicación evolucione continuamente, manteniendo su relevancia y atractivo en un entorno dinámico y competitivo, garantizando así una experiencia de usuario de alta calidad y una respuesta eficiente a las demandas emergentes.

2

NECESIDADES DEL SECTOR PRODUCTIVO

A continuación se identifican las necesidades detectadas en el sector productivo que originan la oportunidad de negocio que se detalla en los siguientes puntos.

2.1

Análisis de la situación actual

El mercado del entretenimiento digital ha experimentado un crecimiento significativo en los últimos años, impulsado por el aumento del uso de dispositivos móviles y la búsqueda constante de nuevas formas de entretenimiento. Las aplicaciones de juegos para fiestas han ganado popularidad, especialmente entre los jóvenes, debido a su capacidad para facilitar la interacción social y proporcionar diversión en grupo.

Actualmente, existen varias aplicaciones de juegos de fiesta en el mercado, como "Guess Up" y "Picolo", que han demostrado ser exitosas. Estas aplicaciones han capturado una parte considerable del mercado debido a sus características atractivas y su facilidad de uso. Sin embargo, a pesar de su popularidad, hay áreas de mejora y oportunidades no explotadas que pueden ser aprovechadas por nuevos desarrollos.

Una de las principales deficiencias de las aplicaciones actuales es la falta de personalización y la limitada oferta de contenido en idiomas distintos al inglés. Además, muchas aplicaciones no ofrecen una experiencia de usuario optimizada ni funcionalidades avanzadas que permitan a los usuarios almacenar y consultar sus resultados de las partidas, lo cual podría aumentar el compromiso y la lealtad de los usuarios.

Para comprender mejor el contexto en el que se desarrollará y lanzará "PartyUpp", es crucial tener una perspectiva del mercado de aplicaciones móviles y del sector de entretenimiento digital, especialmente en la región de Madrid y la zona centro de España. Este análisis permitirá identificar oportunidades y desafíos específicos del entorno local.

1. Demografía y Penetración de Dispositivos Móviles:

- En España, el uso de dispositivos móviles es extremadamente alto, con una penetración de smartphones del 94% entre los jóvenes de 16 a 24 años, según datos del Instituto Nacional de Estadística (INE). Este grupo demográfico es el principal objetivo de "PartyUpp", dado su interés en aplicaciones de ocio y entretenimiento.

2. Mercado de Aplicaciones de Entretenimiento:

- El mercado de aplicaciones de entretenimiento y ocio ha mostrado un crecimiento constante. Según un informe de Statista, el segmento de aplicaciones de juegos móviles en España generó ingresos de aproximadamente 1,200 millones de euros en 2022, con una proyección de crecimiento anual del 8% hasta 2027. Este crecimiento subraya la viabilidad económica de proyectos como "PartyUpp".

3. Preferencias del Consumidor:

- Un estudio de hábitos de consumo de aplicaciones móviles realizado por la Asociación para la Investigación de Medios de Comunicación (AIMC) en 2023 reveló que el 68% de los jóvenes entre 18 y 30 años descargan y utilizan aplicaciones de entretenimiento y juegos en sus dispositivos móviles. Además, el 40% de estos usuarios buscan aplicaciones en español, destacando la importancia de la localización del contenido.

4. Competencia y Diferenciación:

- La competencia en el sector de aplicaciones de juegos para fiestas incluye títulos populares como "Guess Up" y "Picolo". Sin embargo, muchas de estas aplicaciones están disponibles principalmente en inglés y carecen de opciones de personalización profunda. "PartyUpp" tiene la oportunidad de diferenciarse al ofrecer una interfaz en español, desafíos personalizados y funcionalidades adicionales como el almacenamiento y consulta de resultados de las partidas.

En resumen, el análisis estadístico revela un mercado prometedor para "PartyUpp", con una alta penetración de dispositivos móviles y una creciente demanda de aplicaciones de entretenimiento en español. A pesar de la presencia de competencia, hay oportunidades claras para diferenciarse y capturar una porción significativa del mercado mediante la personalización y la mejora de la experiencia de usuario. Además, el considerable número de PYMEs en el sector TIC en Madrid ofrece potencial para colaboraciones estratégicas y el fortalecimiento del ecosistema digital local.

2.2 Necesidades del Cliente y Oportunidad de Negocio

Los clientes de aplicaciones de juegos de fiesta buscan principalmente diversión y una manera sencilla de interactuar con sus amigos en un entorno social. Las aplicaciones que mejoran la experiencia de usuario y ofrecen funcionalidades adicionales tienen una mayor probabilidad de captar y retener usuarios. A continuación, se detallan las necesidades específicas de los clientes:

1. Diversión y Entretenimiento:

- Los usuarios desean juegos que sean fáciles de entender y jugar, pero que también ofrezcan desafíos y mantengan el interés durante las reuniones sociales.

2. Interacción Social:

- Las aplicaciones deben facilitar la interacción entre los jugadores, promoviendo la competición amistosa y la cooperación.

3. Personalización:

- Los usuarios valoran la capacidad de personalizar su experiencia, desde la selección de desafíos hasta la apariencia de la aplicación.

4. Contenido Multilingüe:

- Dada la diversidad lingüística de los usuarios, es importante que las aplicaciones estén disponibles en múltiples idiomas, especialmente en castellano además del inglés.

5. Facilidad de Uso:

- La interfaz debe ser intuitiva y accesible, permitiendo a los usuarios navegar y utilizar la aplicación sin complicaciones.

6. Seguridad y Privacidad:

- La protección de los datos personales y la seguridad en las transacciones dentro de la aplicación son esenciales para ganar la confianza de los usuarios.

2.3 El Nuevo Proyecto: "PartyUpp"

El proyecto "PartyUpp" surge como una respuesta a las necesidades detectadas en el mercado de aplicaciones de entretenimiento digital para fiestas. Este proyecto se enfoca en proporcionar una experiencia mejorada en comparación con las aplicaciones existentes, con un enfoque especial en el diseño de pruebas, la experiencia de usuario y la disponibilidad multilingüe. A continuación, se detallan las características y objetivos principales del proyecto:

1. Posicionamiento en el Mercado:

- "PartyUpp" se presenta como una opción destacada para reuniones sociales, buscando capturar una cuota significativa del mercado de aplicaciones de juegos para fiestas. Su objetivo es ofrecer una experiencia divertida y social, claramente diferenciada de otras aplicaciones similares.

2. Fidelización de Usuarios:

- Para asegurar la lealtad de los usuarios, "PartyUpp" incorpora mecanismos de fidelización, como opciones de personalización y la posibilidad de almacenar y consultar los resultados de las partidas. Esto no solo mejora la experiencia del usuario, sino que también fomenta el uso repetido de la aplicación.

3. Escalabilidad y Expansión:

- La aplicación está diseñada con una arquitectura escalable, permitiendo la incorporación de nuevas funcionalidades y la monetización de contenidos a través de publicidad o suscripciones freemium. Además, se preparará para el lanzamiento en diferentes idiomas, ampliando su alcance a nivel internacional.

4. API y Seguridad:

- La API de "PartyUpp" gestiona el registro de usuarios, la creación y seguimiento de partidas, y el control de inicio de sesión. Se han implementado medidas de seguridad, como la encriptación de contraseñas y el uso de tokens de autenticación, para proteger la privacidad de los usuarios.

5. Interfaz de Usuario:

- La aplicación incluye varias pantallas clave, como las de login y registro, selección de jugadores, y pruebas. Cada pantalla está diseñada para ser intuitiva y facilitar la navegación y el uso de la aplicación.

En conjunto, "PartyUpp" tiene como objetivo transformar las reuniones sociales en experiencias entretenidas y memorables, ofreciendo una plataforma segura, fácil de usar y adaptable a las necesidades cambiantes del mercado y de los usuarios.

Elementos diferenciadores

- **Enfoque Multilingüe:** A diferencia de muchas aplicaciones de la competencia, "PartyUpp" ofrecerá soporte completo en castellano desde su lanzamiento, con posibilidad de expansión a otros idiomas.
- **Personalización de Desafíos:** Los usuarios podrán personalizar y ajustar el juego según sus preferencias, lo que no es común en otras aplicaciones similares a "PartyUpp".
- **Almacenamiento de Resultados:** La aplicación permitirá a los usuarios almacenar y consultar los resultados de sus partidas anteriores, fomentando la repetición y la fidelización.
- **Seguridad y Privacidad:** Un fuerte enfoque en la seguridad de los datos y la privacidad del usuario, utilizando encriptación avanzada y tokens de autenticación para un tratamiento adecuado de los datos.

Entorno Específico

El proyecto se desarrollará principalmente para el sistema operativo Android, dado su amplia base de usuarios y la facilidad de distribución a través de plataformas como Google Play Store o la App Store de Apple. La API backend está diseñada para garantizar la escalabilidad y la disponibilidad.

Herramientas a Utilizar

- **Android Studio:** Herramienta oficial para el desarrollo de aplicaciones Android, que proporciona un entorno completo y robusto para la programación en Java.
- **Spring Security:** Para autenticación y base de datos en tiempo real, ofreciendo una solución escalable y segura para gestionar usuarios y almacenar datos de partidas, proporcionando un entorno eficiente y flexible para la implementación de servicios web.
- **Git/GitHub:** Para control de versiones y colaboración, permitiendo una gestión eficiente del código fuente y la coordinación entre los desarrolladores.

Recursos tanto Humanos y Materiales Necesarios

- **Equipo de Desarrollo:**
 - 3 Desarrolladores Full-Stack especializados en Java, Java Spring Boot y Android Studio.
- **Recursos Materiales:**

- Equipos de desarrollo (ordenadores personales).
- Licencias de software (Android Studio, Spring Boot, Visual Studio Code, GitHub).

2.3.3 Obligaciones Fiscales, Legales, Laborales y/o de Prevención de Riesgos

Por el momento, "PartyUpp" es una aplicación desarrollada en el ámbito académico, y aunque tiene proyección de comercialización, aún no cuenta con planes respectivos a obligaciones fiscales, laborales y/o de Prevención de Riesgos.

No obstante, siendo una aplicación de juego en grupo que implica consumo de alcohol, "PartyUpp" quiere aclarar que su objetivo principal es ofrecer una experiencia de entretenimiento socialmente responsable y centrada en la diversión compartida entre amigos. La aplicación no promueve ni incita de ninguna manera al consumo excesivo de alcohol ni a comportamientos irresponsables, temerarios o nocivos.

Es importante fomentar un ambiente de diversión seguro, donde los usuarios puedan disfrutar de juegos y desafíos sin poner en riesgo su bienestar ni el de los demás. Es por eso que cada aspecto de "PartyUpp" está diseñado cuidadosamente para garantizar que sea una plataforma que promueva la interacción social positiva y el disfrute responsable.

2.3.4 Ayudas/Subvenciones

Para la puesta en marcha del proyecto "PartyUpp", se pueden considerar varias ayudas y subvenciones disponibles a nivel local, regional y nacional:

- **Ayudas del Instituto de Crédito Oficial (ICO):** Programas de financiación para emprendedores y empresas tecnológicas.
- **Subvenciones del Ministerio de Industria, Comercio y Turismo:** Apoyo a proyectos innovadores y tecnológicos.
- **Programas de la Comunidad de Madrid:** Subvenciones para startups y PYMEs en el sector TIC.
- **Fondos Europeos (FEDER):** Apoyo a proyectos de innovación y desarrollo tecnológico.
- **Ayudas de la Fundación EOI (Escuela de Organización Industrial):** Subvenciones para proyectos tecnológicos y de digitalización.

- **Programas de Innovación de Red.es:** Apoyo para la digitalización y el desarrollo de nuevas tecnologías.

La combinación de estas ayudas y subvenciones puede proporcionar un respaldo financiero significativo para la comercialización de "PartyUpp", facilitando su lanzamiento y expansión en el mercado.

3

DISEÑO DEL PROYECTO

Dando por hecho la viabilidad del proyecto, en este apartado se concretarán las fases necesarias para llevarlo a cabo, y cumplir con los objetivos que se establezcan, teniendo en cuenta los recursos necesarios.

3.1

Fases del proyecto

El desarrollo de este proyecto se llevará a cabo en cuatro fases: análisis, diseño, implementación y pruebas, que pasan a detallarse a continuación.

3.1.1

Análisis

En la fase de análisis se determinaron los requisitos del proyecto, diferenciando entre requisitos funcionales y no funcionales. Además, se define claramente el alcance del proyecto y sus posibles restricciones.

Requisitos Funcionales

Los requisitos funcionales describen lo que la aplicación "PartyUpp" debe hacer para cumplir con las expectativas de los usuarios y los objetivos del proyecto. A continuación, se detallan algunos de los principales requisitos funcionales identificados:

1. Registro de Usuarios:

- **Creación de Nuevos Usuarios:** La aplicación debe permitir a los usuarios registrarse mediante un formulario que incluya nombre, correo electrónico, nombre de usuario y contraseña.
- **Validación de Datos:** Se debe implementar la validación de los datos ingresados para asegurar que la información proporcionada sea válida y segura.
- **Seguridad en el Almacenamiento de Datos:** Los datos personales deben almacenarse de manera segura, utilizando técnicas de encriptación para proteger la información sensible.

2. Inicio de Sesión:

- **Autenticación de Usuarios:** Los usuarios deben poder iniciar sesión utilizando su nombre de usuario y contraseña.
- **Sesiones Seguras:** Se deben utilizar tokens de autenticación para mantener sesiones seguras y evitar accesos no autorizados.

3. Gestión de Partidas:

- **Creación de Partidas:** Los usuarios deben poder crear nuevas partidas, especificando los participantes .
- **Seguimiento de Partidas:** La aplicación debe permitir el seguimiento del progreso de las partidas en tiempo real, almacenando el estado actual del juego.
- **Almacenamiento de Resultados:** Los resultados de las partidas deben almacenarse en la base de datos de la aplicación para permitir su consulta posterior.

4. Interacción Social:

- **Selección de Jugadores:** Los usuarios deben poder seleccionar a los jugadores que participarán en la partida, con opciones para agregar o eliminar jugadores.
- **Foto Inicial:** Antes de comenzar la partida, el grupo debe poder tomarse una foto que sirva como recuerdo de la sesión y que pueda guardarse en la galería de la aplicación.

5. Pantallas de Desafíos:

- **Desafíos Secuenciales:** Los desafíos deben presentarse de manera secuencial, con un botón para avanzar al siguiente desafío una vez completado el actual.
- **Registro de Resultados:** La aplicación debe registrar si los desafíos han sido completados, y almacenar estos resultados en la base de datos si se cumplen los requisitos para ello.

Requisitos No Funcionales

Los requisitos no funcionales describen las propiedades y cualidades que la aplicación "PartyUpp" debe cumplir para asegurar su eficacia, eficiencia y satisfacción del usuario. A continuación, se enumeran algunos de los requisitos no funcionales clave para este proyecto:

1. Diseño Atractivo:

- La aplicación debe tener una interfaz de usuario atractiva y moderna que sea fácil de usar y visualmente agradable. Esto incluye el uso de colores, tipografía y diseño gráfico que resuene con el público objetivo y transmita adecuadamente la identidad de la aplicación.

2. Incremento de la Fiabilidad:

- La aplicación debe ser confiable y robusta, con mecanismos de autenticación y gestión de datos deben ser seguros y eficientes, garantizando la adecuada protección de los mismos..

3. Mejora del Tiempo de Respuesta:

- La aplicación debe ser rápida y responder de manera eficiente a las acciones de los usuarios. Los tiempos de carga y respuesta deben mantenerse mínimos para asegurar una experiencia fluida.

4. Escalabilidad:

- La arquitectura de la aplicación debe ser escalable, permitiendo la adición de nuevas funcionalidades y la expansión a nuevos mercados sin comprometer el rendimiento.

5. Compatibilidad Multiplataforma:

- Aunque el desarrollo inicial se centra en Android, la aplicación debe diseñarse de manera que, en medida de lo posible, no se comprometa su posible adaptación futura sin excesiva complicación a otras plataformas como podría ser iOS.

6. Seguridad y Privacidad:

- La seguridad de los datos de los usuarios debe ser una prioridad. La aplicación debe utilizar encriptación avanzada y cumplir con las normativas de protección de datos, asegurando la privacidad y seguridad de la información personal.

Alcance del Proyecto

El alcance del proyecto define los límites y la extensión del desarrollo de "PartyUpp". Incluye los siguientes puntos clave:

1. Funcionalidades Básicas:

- Desarrollo y despliegue de las funcionalidades principales descritas en los requisitos funcionales, como el registro de usuarios, la gestión de partidas, y la autenticación segura.

2. Interfaz de Usuario:

- Diseño y desarrollo de una interfaz de usuario intuitiva y atractiva, que facilite la navegación y el uso de la aplicación.

3. Soporte Multilingüe:

- Implementación de soporte para múltiples idiomas, comenzando con español e inglés, con posibilidad de expandirse a otros idiomas en el futuro.

4. Mecanismos de Fidelización:

- Desarrollo de opciones de personalización y almacenamiento de resultados para fomentar la repetición y fidelización de los usuarios.

5. Preparación para Escalabilidad:

- Diseño de la arquitectura de la aplicación para permitir la incorporación de nuevas funcionalidades y la expansión a nivel internacional.

Restricciones del Proyecto

Las restricciones son limitaciones que pueden afectar el desarrollo y despliegue del proyecto. Algunas de las restricciones identificadas para "PartyUpp" incluyen:

1. Recursos Limitados:

- Disponibilidad de recursos humanos y materiales, incluyendo desarrolladores, equipos y software necesario para el desarrollo del proyecto.

2. Presupuesto:

- Limitaciones presupuestarias que puedan afectar la cantidad de funcionalidades desarrolladas o la calidad de los materiales utilizados.

3. Plazos de Entrega:

- Tiempos de desarrollo ajustados que requieren una gestión eficiente del proyecto para cumplir con los plazos establecidos.

4. Cumplimiento de Normativas:

- Necesidad de cumplir con las normativas locales e internacionales de protección de datos y seguridad de la información.

5. Compatibilidad Técnica:

- Asegurar la compatibilidad de la aplicación con diferentes versiones del sistema operativo Android y futuros planes de expansión a iOS.

3.1.2 Diseño

En la fase de diseño del proyecto "PartyUpp", se realiza una aproximación detallada al diseño tecnológico de la solución. Esta fase incluye la descripción de cómo se implementarán los requisitos establecidos en la fase de análisis y la definición de la estructura de la aplicación, así como el diseño de los componentes necesarios (base de datos, servidor web, clientes, etc.). A continuación, se aborda cada uno de estos aspectos en profundidad.

Diseño de Requisitos Funcionales

1. Registro de Usuarios:

- **Formulario de Registro:**

- Diseño de una interfaz intuitiva para el formulario de registro, con campos para nombre, correo electrónico, nombre de usuario y contraseña.
- Implementación de validaciones en el frontend a través de mensajes de confirmación para asegurar que los datos ingresados cumplen con los formatos esperados.

- **Backend de Registro:**

- Desarrollo de un endpoint RESTful en el servidor que reciba las solicitudes de registro.
- Validación adicional en el servidor para asegurar la integridad y seguridad de los datos.
- Almacenamiento de los datos del usuario en la base de datos, utilizando técnicas habituales de encriptación para mayor seguridad de las contraseñas.

2. Inicio de Sesión:

- **Formulario de Inicio de Sesión:**

- Diseño de una pantalla de inicio de sesión sencilla y segura.
- Implementación de validaciones de entrada para asegurar que los campos no estén vacíos y contengan los datos correctos.

- **Backend de Autenticación:**

- Desarrollo de un endpoint RESTful para el inicio de sesión que autentique al usuario verificando sus credenciales.
- Uso de tokens JWT (JSON Web Tokens) para la autenticación y gestión de sesiones seguras.

3. Gestión de Partidas:

- **Interfaz de Creación de Partidas:**

- Diseño de una pantalla donde los usuarios puedan crear nuevas partidas, especificando los participantes y las reglas.
- Inclusión de una interfaz intuitiva para la selección de jugadores.

- **Backend de Partidas:**

- Desarrollo de endpoints RESTful para la creación y gestión de partidas.
- Almacenamiento del estado de las partidas en la base de datos, incluyendo los detalles de los jugadores, la fecha y las imágenes de la partida.

4. Interacción Social:

- **Selección de Jugadores:**

- Diseño de una interfaz que permita a los usuarios seleccionar a los jugadores para las partidas, con opciones para agregar y eliminar participantes.

- **Foto Inicial:**

- Implementación de una funcionalidad que permita a los usuarios tomarse una foto grupal al inicio de la partida, utilizando la cámara del dispositivo.
- Almacenamiento y gestión de las fotos en la base de datos o en un servicio de almacenamiento en la nube.

5. Pantallas de Desafíos:

- **Desafíos Secuenciales:**

- Diseño de una interfaz de usuario que muestre los desafíos de manera secuencial, con un botón para avanzar al siguiente desafío.

- **Registro de Resultados:**

- Implementación de funcionalidades que permitan registrar la partida si los desafíos alcanzan un determinado nivel de compleción.
- Almacenamiento de los resultados en la base de datos para permitir su consulta posterior.

Estructura de la Aplicación

1. Arquitectura de la Aplicación:

- **Cliente-Servidor:**

- La aplicación seguirá una arquitectura cliente-servidor, donde el cliente será la aplicación móvil (inicialmente para Android) y el servidor manejará la lógica de negocio y la persistencia de datos.

- **Microservicios:**

- Consideración de una arquitectura de microservicios para mejorar la escalabilidad y mantenibilidad del sistema. Cada funcionalidad principal (registro de usuarios, gestión de partidas, autenticación) puede ser un microservicio independiente.

2. Base de Datos (BBDD):

- **Elección del Sistema de Gestión de Base de Datos:**

- Se utilizará una base de datos relacional con base SQL debido a su robustez y capacidad para manejar transacciones complejas.

- **Diseño del Esquema de la Base de Datos:**

- Tablas principales incluirán: Usuarios, Partidas, Desafíos, Resultados, Fotos.
- Relaciones entre las tablas para asegurar integridad referencial, como la relación entre usuarios y partidas.

3. Servidor Web:

- **Tecnologías del Servidor:**

- El servidor será implementado utilizando las tecnologías Spring Boot para manejar las solicitudes HTTP y proporcionar los endpoints RESTful.

- **Gestión de Sesiones y Autenticación:**

- Implementación de middleware para la autenticación a través de tokens.
- Protección de rutas sensibles para asegurar que solo usuarios autenticados puedan acceder a ciertas funcionalidades.

4. Clientes:

- **Aplicación Móvil Android:**

- Desarrollo utilizando Java.
- Uso de Retrofit para la comunicación con el servidor.

- Implementación de un diseño llamativo para asegurar una experiencia de usuario coherente y atractiva.
- **Consideraciones para Futuras Expansiones:**
 - Diseño de la arquitectura del cliente de manera modular para facilitar la expansión a iOS en el futuro.

Diseño de Componentes

1. Interfaz de Usuario (UI):

- **Componentes UI:**
 - Formularios de registro e inicio de sesión.
 - Pantallas de creación y gestión de partidas.
 - Pantallas de desafíos secuenciales con botones de navegación.
 - Galería de fotos y resultados y perfil de usuario con opciones de personalización.

2. Backend y API:

Para el backend de la aplicación **se ha** usado Spring Boot con una base de datos **asociada**. Para realizar la autenticación en el API existen varias opciones:

Para el backend de la aplicación **se ha** usado Spring Boot con una base de datos **asociada**. Para realizar la autenticación en el API existen varias opciones que se detallan a continuación:

- **Autenticación HTTP básica:**
 - El nombre de usuario y contraseña se envían junto con la solicitud en un header.
- **Autenticación por token opaco:**
 - Cuando el usuario inicia sesión, se genera un token aleatorio y se almacena en la base de datos asociado a un usuario y otra información como la fecha de caducidad. El usuario envía el token en lugar de sus credenciales con su solicitud.
- **Autenticación por token JWT:**

- Los tokens JWT contienen la identidad del usuario, fecha de caducidad y cualquier información que el servidor quiera guardar. La ventaja es que el servidor no debe almacenar los tokens y no es necesario realizar ninguna petición a la base de datos para verificar la autenticidad del token. El token está firmado con una clave secreta en el servidor, evitando que el usuario pueda crear tokens fraudulentos.

Finalmente se decide recurrir a la última opción debido a su fiabilidad y seguridad.

3. Base de Datos:

- **Tablas y Relaciones:**

- Usuarios: id, nombre, correo electrónico, nombre de usuario, contraseña.
- Partidas: id, id_creador, fecha_creacion, jugadores.
- Desafíos: id, texto en español, texto en inglés, categoría.

4. Gestión de Fotos:

- **Almacenamiento en la Nube:**

- Uso de servicios web para almacenar las fotos de los usuarios de manera segura y escalable.

- **Integración con la Aplicación:**

- Implementación de funcionalidades en la aplicación para capturar y cargar fotos a la nube.

3.1.3

Implementación

1. Registro de Usuarios:

- **Formulario de Registro:**

- Para el diseño gráfico de la interfaz se recurre a la librería `com.google.android.material.textfield.TextInputLayout` de Android, que proporciona una vista agradable, perfecta para este tipo de funcionalidades.
- Implementación de validaciones utilizando el elemento `AlertDialog` para asegurar que los datos ingresados cumplen con los formatos esperados.
- El usuario puede elegir entre realizar un login, al presionar el botón “Iniciar sesión” una vez haya introducido sus datos personales, o bien usar la aplicación de manera rápida mediante el botón “Continuar como invitado”.

- También se ofrece la posibilidad de redirigirse a la pantalla de registro, mediante un tercer botón.

- **Backend de Registro:**

- Al presionar el botón “Registrarse” se hace la llamada correspondiente a la API para su almacenaje en la base de datos .

2. Inicio de Sesión:

- **Formulario de Inicio de Sesión:**

- De manera similar a la pantalla de login, se utiliza la librería `com.google.android.material.textfield.TextInputLayout`.
- Validación de nuevo con `AlertDialog`.

- **Backend de Autenticación:**

- Se realiza una llamada a la API para comprobar la validez de los datos de inicio de sesión al presionar el botón “Inicio de sesión”, procediendo a recuperar la token del usuario y comprobar su existencia o validez.

3. Gestión de Partidas:

- **Menú de inicio:**

- La pantalla principal de la aplicación ofrece un menú de navegación mediante el recurso gráfico de Android `com.google.android.material.bottomnavigation.BottomNavigationView`. Este menú permite al usuario desplazarse entre los tres Fragments que componen la pantalla principal (`FragmentHome`, `FragmentGallery` y `FragmentSettings`).
- El componente `HomeFragment` contiene dos botones principales:
 - “Play”: principal elemento, ya que permite a los usuarios iniciar la partida, desplazando el flujo de la aplicación a la pantalla de selección de usuarios.
 - “Tutorial”: breve tutorial destinado a familiarizar al usuario con la aplicación.
- El componente `GalleryFragment` tiene su acceso restringido; es únicamente accesible para usuarios registrados. En caso de que un usuario no registrado tratase de acceder, se desplegará un `AlertDialog` ofreciendo al posibilidad de iniciar sesión o de regresar al fragment anterior. Los principales elementos visuales de esta pantalla son:
 - Un `RecyclerView` que permite la visualización de las partidas que haya completado el usuario.

- Una barra superior con el nombre del perfil y la imagen asociada (si ha sido previamente cargada a la aplicación por el usuario). Esta imagen también actúa como enlace al fragment que contiene el perfil.
 - El componente SettingsFragment permite al usuario consultar sus preferencias de juego (sonidos del juego, música, vibración, etc.), así como actualizar su imagen de perfil (esto último sólo en caso de usuarios registrados). También contiene un botón para cerrar/iniciar sesión.
- **Interfaz de Selección de Jugadores:**
 - La interfaz de selección de jugadores se realiza mediante el uso de un componente RecyclerView. Cada elemento del RecyclerView permite editar el nombre del jugador, así como borrarlo de la partida gracias a un botón de borrado incluido en la vista.
 - Mediante un FloatingActionButton se permite añadir rápidamente jugadores al RecyclerView.
 - La inserción de jugadores puede realizarse también a través de un segundo RecyclerView auxiliar que muestra sugerencias de nombres, facilitando una interacción rápida mediante click.
 - Una vez finalizada la selección, y siempre que se cumplan los requisitos especificados (longitud de nombre, número de jugadores), se habilita el botón “Continuar”, que envía al usuario a la pantalla “InitialPhotoActivity”.
 - **Interfaz de subida de foto inicial de la partida:**
 - En esta pantalla se le ofrece al usuario la posibilidad de guardar una foto del grupo de juego antes de que dé comienzo la partida, si así lo desea.
 - **Backend de Partidas:**
 - La API de PartyUpp genera un elemento almacenable similar a un historial de partidas, que después permite mostrarlas en la galería. Este elemento guarda los nombres de los jugadores, la fecha de la partida y dos fotos (una tomada al inicio de la partida y una segunda al final de la misma).
 - La interfaz de la aplicación está configurada para registrar al momento la foto como un mapa de bits mediante la clase de java Bitmap (android.graphics.Bitmap). Esta imagen es almacenada en el servidor, además de vincularse automáticamente al historial de partidas del usuario en caso de que este haya iniciado sesión.

4. Pantallas de Desafíos:

- **Desafíos Secuenciales:**

- La aplicación cuenta con una interfaz de usuario que muestra los desafíos de manera secuencial, con un botón para avanzar al siguiente desafío. El diseño está basado en la simplicidad, haciendo que el usuario no tenga distracciones y pueda centrarse claramente en cada una de las pruebas que se le van presentando. La sucesión de desafíos se genera dentro de una única actividad (Activity) que a su vez aloja un contenedor de fragmentos (FragmentManager). Dicho contenedor de fragmentos (Fragment) se encarga de renovar su vista con un nuevo desafío cada vez que se acciona el botón “Siguiente”.
- **Backend de Desafíos:**
 - La aplicación realiza una única petición a la API para cargar las partidas que haya alojadas en ese momento. Por tanto, estas se no se encuentran “hardcodeadas” en la propia aplicación, sino que se accede a ellas cada vez que se genera una nueva partida. Esta característica, entre otras ventajas, permite la completa actualización, revisión y añadido de nuevas pruebas sin obligar al usuario a tener que actualizar el ejecutable.
 - Así mismo, soporta diversas funcionalidades que permiten registrar los desafíos cuando son completados, estableciéndose un número determinado de desafíos tras los cuales la propia aplicación determina el final de la partida. Tras ello, se procede a realizar el almacenamiento de los resultados en la base de datos para permitir su consulta posterior.

5. Pantallas de gestión de final de la partida

- **Pantalla de aviso de final de la partida:**

Una vez alcanzado el número máximo de pruebas determinadas por la aplicación, se redirecciona al usuario a esta actividad y se informa de ello al usuario.

- **Pantalla de foto final:**

- A su vez, la pantalla anterior permite navegar hacia esta pantalla. En ella, de manera similar a la pantalla de foto inicial, se le ofrece al usuario la posibilidad de almacenar una segunda foto de la partida mediante su cámara o galería del dispositivo, haciendo más claro y palpable el efecto de las pruebas en los participantes. Una vez se presiona en botón de “Guardar y salir”, se envía de nuevo al usuario a la pantalla de inicio, completando el flujo circular de la aplicación.

Estructura de la Aplicación

1. Arquitectura de la Aplicación:

- **Cliente-Servidor:**

- La aplicación sigue una arquitectura cliente-servidor, donde el cliente será la aplicación móvil (inicialmente para Android) y el servidor maneja la lógica de negocio y la persistencia de datos.

- **Microservicios:**

- Consideración de una arquitectura de microservicios para mejorar la escalabilidad y mantenibilidad del sistema. Cada funcionalidad principal (registro de usuarios, gestión de partidas, autenticación) es un microservicio independiente.

2. Base de Datos (BBDD):

- **Elección del Sistema de Gestión de Base de Datos:**

- Se utilizará una base de datos relacional MySQL, potenciada mediante las herramientas Spring Boot, JPA y Hibernate, debido a su facilidad de configuración y despliegue, así como su robustez y capacidad para manejar transacciones complejas.

- **Diseño del Esquema de la Base de Datos:**

- Las tablas principales incluirán: Usuarios, Partidas y Desafíos.
- Relaciones entre las tablas para asegurar integridad referencial, como la relación entre usuarios y partidas. Desafíos es una tabla independiente que no necesita ser relacionada con las demás tablas en ningún caso.

3. Servidor Web:

- **Tecnologías del Servidor:**

- El servidor será implementado utilizando las tecnologías Spring Boot y RESTful, que permiten gestionar las peticiones HTTP.

- **Gestión de Sesiones y Autenticación:**

- Implementación de middleware para la autenticación con JWT.
- Protección de rutas sensibles para asegurar que solo usuarios autenticados puedan acceder a ciertas funcionalidades (más detalles en el apartado "Backend API").

4. Clientes:

- **Aplicación Móvil Android:**

- Desarrollo utilizando Java.
- Uso de Retrofit para la comunicación con el servidor.
- Implementación de un diseño basado en las herramientas gráficas que proporciona Android Studio.

- **Consideraciones para Futuras Expansiones:**

- Diseño de la arquitectura del cliente de manera modular para facilitar la expansión a iOS en el futuro.

Diseño de Componentes

1. Interfaz de Usuario (UI):

- **Componentes UI:**

- Formularios de registro e inicio de sesión.
- Pantallas de inicio, galería y perfil navegables a través de un menú inferior:
 - Galería de fotos iniciales y resultados.
 - Menú principal de la aplicación con los botones de acceso a las partidas y al tutorial
 - perfil de usuario con opciones de personalización de la experiencia y de inicio/clausura de sesión.
- Pantallas de creación y gestión de partidas (selección de jugadores y guardado de foto inicial).
- Pantallas de gestión del final de la partida.
- Pantallas de desafíos secuenciales con botones de navegación.

2. Backend y API:

Para la implementación final se usan tokens JWT, ya que se trata de una opción más escalable que los tokens opacos (no requieren almacenamiento adicional) y son más seguros que la autenticación básica.

La autenticación corre a cargo de Spring Security, una opción que permite evitar fallos de seguridad en la implementación y facilita algunas características comunes a la mayoría de proyectos. Para llevar a cabo la autenticación con Spring Security, es necesario implementar algunos métodos e interfaces:

- **UserDetails:**

- Interfaz implementada en la propia entidad de usuario. Necesita el método “getUsername” que devuelve el identificador único del usuario. En éste caso devuelve el correo electrónico a pesar de su nombre. También necesita otros métodos que proporcionan información sobre caducidad de contraseñas y bloqueo de cuentas. La aplicación no usa estas características, así que los métodos devuelven siempre “true”.
- **UserDetailsService:**
 - Contiene un método “loadUserByName” que obtiene un objeto UserDetails a partir del identificador del usuario (su correo electrónico). Está implementado usando el repositorio de usuarios.
- **AuthenticationProvider:**
 - **Especifica qué objeto UserDetailsService y que codificador de contraseñas va a usar Spring Security.**

La autenticación se realiza en un filtro de solicitud. El filtro extrae el token del header Authentication y valida su autenticidad. Si no es correcto o está expirado, devuelve un estado 401.

Si la autenticación ha sido un éxito, guarda el email del usuario en un componente de Spring “AuthData”, que es instanciado al comienzo de la solicitud y destruido al acabar la misma. El controlador o servicio que se ejecute a continuación puede automontar este componente y obtener fácilmente el usuario que ha enviado la petición.

- **Endpoints RESTful:**
 - La aplicación escucha en el puerto 8080.
 - Los endpoints ubicados en /api/v1/noAuth no requieren autenticación para su uso.
 - Los endpoints ubicados en /api/v1/user requieren estar autenticado con un usuario cualquiera.
 - Los endpoints ubicados en /api/v1/admin requieren estar autenticado con un usuario administrador.
- **Uso de los endpoints:**

- Para crear un usuario, se usa el endpoint POST `"/api/v1/noAuth/register"` y se pasan los datos por el cuerpo de la petición. Para crear un usuario administrador, se usa en su lugar el endpoint POST `"/api/v1/admin/registerAdmin"`, que requiere un usuario administrador. Un ejemplo de uso sería:

```
{
  "name": "admin",
  "email": "admin@gmail.com",
  "password": "contraAdmin"
}
```

- Si el registro ha tenido éxito, se devolverá un token de autenticación. Para obtener otro token, se usa el endpoint GET `"/api/v1/noAuth/authenticate"` y se pasan las credenciales por el cuerpo de la petición, por ejemplo:

```
{
  "email": "test@gmail.com",
  "password": "contraseña"
}
```

- Para autenticarse, hay que proporcionar el token junto con la petición usando el prefijo "Bearer".
- Al comenzar la aplicación, se cargan las pruebas iniciales (localizadas en `"src/main/resources/static/init/challenges.json"`).
- Si se quieren añadir pruebas nuevas, se puede usar la instrucción POST `"/api/v1/admin/postChallenge"`, que admite una sola prueba en el cuerpo.
- Para obtener todas las pruebas, se usa el endpoint GET `"/api/v1/noAuth/getChallenges"`.
- Para subir imágenes se utiliza el endpoint POST `"/api/v1/user/uploadImage"`.
- Para obtener imágenes existe el endpoint `"/api/v1/user/getImage/{id}"`. Para que funcione adecuadamente, el usuario autenticado debe ser el mismo que subió la imagen.

- **Middleware de Seguridad:**

- Implementación de autenticación y autorización para proteger los endpoints.
- Sistema de tokens para garantizar la integridad de los accesos a la aplicación.

3. Base de Datos:

- **Tablas y Relaciones:**

- Usuarios: id, nombre, correo electrónico, nombre de usuario, contraseña.
- Partidas: id, id_creador, fecha_creacion, estado.
- Desafíos: id, descripcion, id_partida.

4. Gestión de Fotos:

- **Almacenamiento en la Nube:**

- Almacenado las fotos de los usuarios de manera segura y escalable.

- **Integración con la Aplicación:**

- Implementación de funcionalidades en la aplicación para capturar y cargar fotos a la nube.

En resumen, la fase de diseño de "PartyUpp" establece un plan claro y detallado para la implementación de la solución tecnológica. Cada requisito funcional se aborda con un diseño específico, asegurando que la aplicación sea robusta, escalable y fácil de usar. La estructura de la aplicación y el diseño de los componentes clave proporcionan una base sólida para las fases de implementación y pruebas que seguirán.

3.1.4

Pruebas

Caso de Prueba 1

Característica Probada: Registro de Usuario

Descripción: Validar que el usuario puede registrarse correctamente con datos válidos.

Pasos:

1. Abrir la aplicación y navegar a la pantalla de registro.
2. Introducir nombre, correo electrónico, nombre de usuario y contraseña.
3. Presionar el botón "Registrarse".

Resultado Esperado: Usuario registrado exitosamente y redirigido a la pantalla principal.

Fecha de Prueba: 10 de abril de 2024

Resultado Obtenido: Usuario registrado y redirigido a la pantalla principal sin errores.

Caso de Prueba 2

Característica Probada: Inicio de Sesión

Descripción: Validar que el usuario puede iniciar sesión con credenciales correctas y están vigentes.

Pasos:

1. Abrir la aplicación y navegar a la pantalla de inicio de sesión.
2. Introducir correo electrónico y contraseña válidos.
3. Presionar el botón "Iniciar sesión".

Resultado Esperado: Usuario autenticado y redirigido a la pantalla principal.

Fecha de Prueba: 15 de abril de 2024

Resultado Obtenido: Usuario autenticado exitosamente y redirigido a la pantalla principal.

Caso de Prueba 3

Característica Probada: Inicio de Sesión con Credenciales Incorrectas

Descripción: Validar que el usuario no puede iniciar sesión con credenciales incorrectas y se le deniega el acceso.

Pasos:

1. Abrir la aplicación y navegar a la pantalla de inicio de sesión.
2. Introducir correo electrónico y contraseña incorrectos.
3. Presionar el botón "Iniciar sesión".

Resultado Esperado: Impedir el inicio y mostrar un mensaje al usuario informando de la situación mediante un AlertDialog con una descripción del error.

Fecha de Prueba: 20 de febrero de 2024

Resultado Obtenido: Se muestra un AlertDialog con un mensaje de error.

Caso de Prueba 4

Característica Probada: Acceso a Galería sin Registro

Descripción: Validar que un usuario no registrado no puede acceder a la galería.

Pasos:

1. Iniciar la aplicación como invitado.
2. Intentar acceder a la galería desde el menú principal.

Resultado Esperado: Mostrar un AlertDialog ofreciendo iniciar sesión o regresar al inicio.

Fecha de Prueba: 25 de mayo de 2024

Resultado Obtenido: AlertDialog mostrado correctamente.

Caso de Prueba 5

Característica Probada: Navegación en el Menú Principal

Descripción: Validar la navegación entre los Fragments del menú principal.

Pasos:

1. Iniciar sesión en la aplicación.
2. Navegar entre HomeFragment, GalleryFragment y SettingsFragment.

Resultado Esperado: Se navega correctamente entre las pantallas sin errores.

Fecha de Prueba: 1 de marzo de 2024

Resultado Obtenido: Navegación fluida y sin errores.

Caso de Prueba 6

Característica Probada: Adición de Jugadores en Selección de Jugadores

Descripción: Validar la adición de jugadores en la pantalla de selección de jugadores.

Pasos:

1. Iniciar una nueva partida.
2. Añadir jugadores usando el FloatingActionButton.
3. Verificar que los jugadores se agreguen al RecyclerView.

Resultado Esperado: Jugadores añadidos y mostrados en el RecyclerView.

Fecha de Prueba: 5 de marzo de 2024

Resultado Obtenido: Jugadores añadidos y visibles en el RecyclerView.

Caso de Prueba 7

Característica Probada: Subida de Foto Inicial de la Partida

Descripción: Validar que el usuario puede subir una foto inicial de la partida.

Pasos:

1. Iniciar una nueva partida.
2. Navegar a la pantalla de foto inicial.
3. Capturar y subir una foto.

Resultado Esperado: Foto subida y almacenada correctamente.

Fecha de Prueba: 13 de mayo de 2024

Resultado Obtenido: Foto subida y almacenada sin errores.

Caso de Prueba 8

Característica Probada: Secuencia de Desafíos

Descripción: Validar la correcta secuencia de desafíos durante una partida.

Pasos:

1. Iniciar una nueva partida.
2. Completar varios desafíos.
3. Navegar entre desafíos usando el botón “Siguiente”.

Resultado Esperado: Desafíos presentados en secuencia sin interrupciones.

Fecha de Prueba: 25 de febrero de 2024

Resultado Obtenido: Secuencia de desafíos fluida y sin interrupciones.

Caso de Prueba 9

Característica Probada: Guardado de Resultados al Final de la Partida

Descripción: Validar que los resultados de la partida se guardan correctamente al finalizar.

Pasos:

1. Completar una partida hasta el final.
2. Capturar y subir la foto final.
3. Verificar que los resultados se guardan en la base de datos.

Resultado Esperado: Resultados y fotos almacenados correctamente.

Fecha de Prueba: 20 de marzo de 2024

Resultado Obtenido: Resultados y fotos guardados sin errores.

Caso de Prueba 10

Característica Probada: Gestión de Preferencias de Usuario en SettingsFragment

Descripción: Validar que el usuario puede actualizar sus preferencias de juego. **Pasos:**

1. Iniciar sesión en la aplicación.
2. Navegar a SettingsFragment.
3. Actualizar preferencias de sonido, música y vibración.
4. Guardar cambios.

Resultado Esperado: Preferencias actualizadas y reflejadas en la configuración de la aplicación.

Fecha de Prueba: 25 de marzo de 2024

Resultado Obtenido: Preferencias actualizadas y guardadas correctamente.

3.2

Objetivos futuros

El proyecto "PartyUpp" se plantea con una serie de objetivos claros y alcanzables que se dividen en dos grandes áreas: desarrollo técnico y objetivos de negocio. A continuación, se detallan estos objetivos con un enfoque en establecerse en el mercado de las aplicaciones de ocio y comenzar su comercialización de manera modesta pero efectiva.

3.2.1.1

Objetivos de Desarrollo Técnico

1. Implementación de Nuevas Metodologías y Estándares:

- Introducir y consolidar el uso de metodologías ágiles en el desarrollo del proyecto para mejorar la eficiencia y la colaboración dentro del equipo. Para ello formarán una parte decisiva el uso de tecnologías de control de versiones y trabajo colaborativo como Git, que ya se han mostrado cruciales en los primeros pasos de "PartyUpp".
- Adoptar e ir implementando cada vez más estándares de codificación modernos y prácticas de desarrollo como TDD (Test Driven Development) y CI/CD (Integración Continua/Despliegue Continuo) para asegurar la calidad y la entrega continua del software.

2. Cumplimiento de Requisitos Técnicos:

- Asegurar que todas las funcionalidades de la aplicación continúan cumpliendo con los requisitos especificados en el diseño inicial, incluyendo la gestión de usuarios, autenticación, manejo de partidas y desafíos.
- Garantizar la seguridad y escalabilidad del backend mediante el uso de tecnologías robustas como Spring Boot, JWT para autenticación y una arquitectura de microservicios.

3. Despliegue y Mantenimiento de la Aplicación:

- Realizar el despliegue inicial de la aplicación en el mercado Android y mantener un ciclo regular de actualizaciones para mejorar la funcionalidad y corregir errores y ampliar el número de experiencias de usuario ofrecidas por la app.
- Implementar un sistema de monitoreo y logging para detectar y resolver problemas en tiempo real.

Objetivos de Negocio

1. Establecerse en el Mercado de Apps de Ocio:

- Posicionar "PartyUpp" como una aplicación innovadora y divertida en el sector de las aplicaciones de ocio, destacándose por su interfaz amigable y funcionalidades únicas como la gestión de desafíos y partidas, así como su oferta en diversos idiomas.
- Participar en ferias y eventos de tecnología y ocio para aumentar la visibilidad de la aplicación y atraer a los primeros usuarios.

2. Comercialización Inicial:

- Realizar una campaña de lanzamiento modesta, enfocándose en redes sociales y marketing digital para atraer a los primeros usuarios sin incurrir en altos costes.
- Ofrecer incentivos iniciales como desafíos exclusivos o funciones premium temporales para atraer y retener usuarios.

3. Satisfacción del Cliente con Valor Añadido:

- Asegurar que la aplicación no solo cumpla con los requisitos básicos de los usuarios, sino que también ofrezca un valor añadido a través de una experiencia de usuario fluida y características adicionales más allá de tutoriales integrados y personalización de perfiles.
- Recopilar y analizar feedback de los usuarios para realizar mejoras continuas y adaptarse a las necesidades del mercado.

4. Objetivos Financieros Modestos:

- Establecer metas financieras alcanzables, como facturar 1.000,00 € en el primer año y atraer a 1.000 clientes en el mismo período, basando estas metas en un análisis realista del mercado y las capacidades del equipo.
- Utilizar una estrategia de monetización diversificada que incluya publicidad, compras dentro de la aplicación y suscripciones premium.

5. Apertura del Mercado Internacional:

- Preparar la aplicación para su internacionalización desde una fase temprana, permitiendo su expansión a mercados fuera del país de origen y en idiomas más allá del castellano y del inglés.
- Realizar estudios de mercado para identificar los países con mayor potencial de crecimiento y adaptar la aplicación y las estrategias de marketing a estos mercados.

Al enfocarse en estos objetivos, "PartyUpp" busca no solo introducirse en el competitivo mercado de las aplicaciones de ocio, sino también establecer una base sólida para el crecimiento futuro, ofreciendo un producto de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

3.2.2 y Humanos Necesarios

3.3 Previsión de los Recursos Materiales

Para llevar a cabo el desarrollo del proyecto "PartyUpp", es fundamental prever y asignar adecuadamente los recursos tanto materiales como humanos. Esto incluye la selección de herramientas adecuadas, la formación del equipo y la planificación del tiempo necesario para completar cada actividad del proyecto.

- **Servicios en la Nube:** Amazon Web Services (AWS) o Google Cloud Platform (GCP) para alojamiento y escalabilidad.

2. Formación:

- **Cursos de Actualización:** Cursos en tecnologías emergentes y metodologías ágiles para el equipo de desarrollo.
- **Workshops:** Talleres de UX/UI y prácticas de pruebas para diseñadores y testers.
- **Certificaciones:** Certificaciones en gestión de proyectos ágiles para el Project Manager y el Scrum Master.

3. Tiempo:

- **Duración del Proyecto:** Estimación de 12 a 18 meses desde el inicio hasta el lanzamiento oficial.
- **Sprints de Desarrollo:** Ciclos de desarrollo de 2-4 semanas según la metodología Scrum.
- **Reuniones de Seguimiento:** Reuniones diarias (daily stand-ups) y revisiones de sprint cada dos semanas.

3.4 Presupuesto Económico

La estimación del coste económico de los recursos necesarios para el desarrollo del proyecto "PartyUpp" se detalla a continuación. Esto incluye costos de personal, hardware, software, formación y otros gastos asociados.

3.2.3

Recursos Humanos

1. Salarios:

- **Desarrolladores:** 35.000 € por año x 4 = 140.000 €
- **Diseñadores UX/UI:** 30.000 € por año x 2 = 60.000 €
- **Tester/QAs:** 27.500 € por año x 2 = 55.000 €
- **Project Manager:** 45.000 € por año x 1 = 45.000 €
- **Scrum Master:** 42.500 € por año x 1 = 42.500 €
- **Especialista en Marketing Digital:** 32.500 € por año x 1 = 32.500 €
- **Analista de Mercado:** 30.000 € por año x 1 = 30.000 €
- **Técnico de Soporte:** 25.000 € por año x 1 = 25.000 €
- **Administrativo:** 20.000 € por año x 1 = 20.000 €

Total de Salarios: 450.000 €

3.2.4 Recursos Materiales

1. Hardware:

- **Servidores:** 5.000 €
- **Estaciones de Trabajo:** 1.000 € x 10 = 10.000 €
- **Dispositivos Móviles:** 2.500 €

Total de Hardware: 17.500 €

2. Software:

- **Licencias de Software de Desarrollo:** 2.500 €
- **Herramientas de Gestión de Proyectos:** 1.500 €
- **Plataformas de Colaboración:** 1.000 €
- **Servicios en la Nube:** 7.500 €

Total de Software: 12.500 €

3. Formación:

- **Cursos de Actualización:** 5.000 €
- **Workshops:** 2.500 €
- **Certificaciones:** 2.500 €

Total de Formación: 10.000 €

3.2.5 Otros Gastos

- **Marketing y Publicidad:** 25.000 €
- **Gastos Administrativos y de Oficina:** 10.000 €

Total de Otros Gastos: 35.000 €

3.2.6 Presupuesto Total

Total Estimado del Proyecto: 450.000 € (Salarios) + 17.500 € (Hardware) + 12.500 € (Software) + 10.000 € (Formación) + 35.000 € (Otros Gastos) = **525.000 €**

3.2.7 Planificación de la Ejecución del Proyecto

La planificación de la ejecución del proyecto se estructura en varias fases, cada una con actividades y tareas específicas que deben realizarse para asegurar el éxito del proyecto.

3.2.7.1 Fase de Análisis

1. Estudio de las Necesidades a Cubrir:

- Identificación de las necesidades y expectativas de los usuarios finales.
- Definición de los problemas que "PartyUpp" pretende resolver.

2. Estudio de la Situación Actual:

- Análisis de las aplicaciones de ocio existentes en el mercado.
- Identificación de brechas y oportunidades de mejora.

3. Establecimiento de los Requisitos del Proyecto:

- Recopilación de requisitos funcionales y no funcionales.
- Elaboración de un documento de requisitos detallado.

4. Valoración Comparativa de las Posibles Soluciones:

- Evaluación de diferentes enfoques técnicos y arquitectónicos.
- Selección de la solución más adecuada basada en criterios de viabilidad y costo-beneficio.

5. Identificación de las Necesidades que Implica el Nuevo Proyecto en la Empresa:

- Evaluación de la necesidad de contratar personal adicional.
- Planificación de la formación necesaria para el equipo existente.
- Identificación de herramientas y equipos adicionales necesarios.

6. Estudio de Viabilidad de la Solución Elegida:

- Análisis de viabilidad técnica, operativa y económica.
- Identificación de riesgos y su mitigación.

7. Corrección de Posibles Errores:

- Revisión y corrección de cualquier error detectado en la fase de análisis.
- Aseguramiento de que todos los requisitos estén claros y completos.

4.2 Fase de Diseño

1. Preparación del Entorno de Diseño:

- Configuración de las herramientas de diseño y desarrollo.
- Creación de entornos de trabajo colaborativos.

2. Diseño de la Arquitectura:

- Definición de la arquitectura del sistema, incluyendo el cliente, el servidor y la base de datos.
- Diseño de la estructura de microservicios.

3. Diseño de los Interfaces:

- Creación de wireframes y prototipos de las interfaces de usuario.
- Aseguramiento de la usabilidad y la experiencia del usuario.

4. Diseño de los Datos:

- Diseño del esquema de la base de datos.
- Definición de las relaciones entre tablas y estructuras de datos.

5. Diseño de los Procedimientos:

- Documentación de los flujos de trabajo y procesos de negocio.
- Definición de procedimientos para el manejo de datos y la comunicación entre servicios.

6. Corrección de Posibles Errores:

- Revisión y validación del diseño.
- Corrección de errores y ajustes necesarios antes de la implementación.

4.3 Fase de Implementación

1. Preparación del Entorno de Implementación:

- Configuración de entornos de desarrollo y pruebas.
- Despliegue de herramientas de CI/CD.

2. Desarrollo de la Arquitectura:

- Implementación de la arquitectura definida.
- Configuración de servidores y servicios en la nube.

3. Desarrollo de los Interfaces, los Datos y los Procedimientos:

- Desarrollo del frontend y backend.
- Implementación de la base de datos y APIs.
- Integración de todos los componentes del sistema.

3.2.7.2**Recursos Humanos****1. Equipo de Desarrollo:**

- **Desarrolladores de Frontend y Backend:** Un equipo de 4-6 desarrolladores con experiencia en Java, Android Studio, Spring Boot y RESTful APIs.
- **Diseñadores UX/UI:** Dos diseñadores encargados de la interfaz de usuario y la experiencia de usuario para asegurar que la aplicación sea intuitiva y agradable.
- **Tester/QAs:** Dos especialistas en aseguramiento de calidad para llevar a cabo pruebas exhaustivas y garantizar que la aplicación funcione sin errores.

2. Equipo de Gestión:

- **Project Manager:** Un gerente de proyecto para coordinar todas las actividades, asegurar que se cumplan los plazos y gestionar los recursos.
- **Scrum Master:** Un facilitador para las metodologías ágiles, asegurando que el equipo siga las prácticas Scrum.

3. Especialistas en Marketing:

- **Especialista en Marketing Digital:** Responsable de las campañas de marketing online, SEO y la gestión de redes sociales.
- **Analista de Mercado:** Para estudiar el mercado y ayudar en la estrategia de expansión internacional.

4. Equipo de Soporte:

- **Técnico de Soporte:** Proveer soporte técnico a los usuarios y resolver problemas relacionados con la aplicación.
- **Administrativo:** Gestionar tareas administrativas y de coordinación entre los equipos.

Recursos Materiales**4. Hardware:**

- **Servidores:** Servidores para hosting del backend y bases de datos.
- **Estaciones de Trabajo:** Computadoras y laptops de alto rendimiento para los desarrolladores y diseñadores.
- **Dispositivos Móviles:** Varios modelos de smartphones y tablets para pruebas de compatibilidad.

5. Software:

- **Licencias de Software de Desarrollo:** Android Studio, IntelliJ IDEA, herramientas de diseño gráfico (Adobe XD, Figma).

- **Herramientas de Gestión de Proyectos:** JIRA, Trello, Confluence para la gestión ágil de proyectos.
- **Plataformas de Colaboración:** Slack, Zoom para comunicación y colaboración entre el equipo.

4 CONTROL Y EVALUACIÓN

DEFINICIÓN DE PROCEDIMIENTOS DE

Para asegurar el éxito del proyecto "PartyUpp" y garantizar que se cumplan los objetivos establecidos, es fundamental implementar procedimientos de control y evaluación rigurosos. Estos procedimientos permitirán gestionar eficazmente las incidencias y los cambios que puedan surgir durante el desarrollo del proyecto.

4.1.1.1

Procedimiento de Evaluación de las Incidencias

1. Identificación de Incidencias:

- Establecer un sistema de registro para documentar todas las incidencias que se presenten durante la realización de las actividades.
- Utilizar una plataforma de gestión de incidencias (por ejemplo, Jira) para asegurar que todas las incidencias sean documentadas y rastreadas adecuadamente.

2. Clasificación y Priorización:

- Clasificar las incidencias según su gravedad (baja, media, alta, crítica).
- Priorizar las incidencias críticas y de alta gravedad para su resolución inmediata.

3. Análisis de Incidencias:

- Realizar un análisis detallado de cada incidencia para identificar la causa raíz.
- Determinar el impacto de la incidencia en el proyecto y en los entregables asociados.

4. Resolución de Incidencias:

- Asignar un responsable para la resolución de cada incidencia.
- Definir un plan de acción específico para corregir la incidencia y prevenir su recurrencia.
- Establecer un plazo para la resolución de cada incidencia según su prioridad.

5. Monitoreo y Seguimiento:

- Monitorear el progreso en la resolución de las incidencias.

- Realizar revisiones periódicas para asegurar que las incidencias se están gestionando adecuadamente.
- Actualizar el registro de incidencias con el estado de resolución y cualquier información relevante.

6. Evaluación Post-Incidencia:

- Evaluar la efectividad de las acciones tomadas para resolver la incidencia.
- Documentar las lecciones aprendidas y mejorar los procedimientos para futuras incidencias.

4.2 Procedimiento para Gestionar los Posibles Cambios en los Recursos y en las Actividades

1. Solicitud de Cambio:

- Definir un formato estándar para la solicitud de cambios que incluya detalles como la descripción del cambio, la justificación, el impacto esperado y los recursos necesarios.
- Establecer un canal oficial para la recepción de solicitudes de cambio (por ejemplo, a través de un sistema de gestión de proyectos).

2. Evaluación del Cambio:

- Formar un comité de evaluación de cambios compuesto por miembros clave del equipo del proyecto (Project Manager, Scrum Master, líderes técnicos, etc.).
- Evaluar el impacto del cambio en términos de costos, tiempo, alcance y calidad.
- Determinar la viabilidad del cambio y su alineación con los objetivos del proyecto.

3. Aprobación del Cambio:

- Someter el cambio a la aprobación del comité de evaluación de cambios.
- Documentar la decisión del comité y comunicarla a todas las partes interesadas.
- Si el cambio es aprobado, actualizar el plan del proyecto y los recursos asignados según sea necesario.

4. Implementación del Cambio:

- Asignar recursos y responsabilidades para la implementación del cambio.
- Definir un cronograma claro para la ejecución del cambio.
- Asegurar que todos los miembros del equipo estén informados y alineados con el cambio.

5. Monitoreo y Control del Cambio:

- Supervisar la implementación del cambio para asegurar que se ejecute conforme al plan.
- Evaluar el progreso y realizar ajustes si es necesario.
- Documentar cualquier incidencia o desviación durante la implementación del cambio.

6. Evaluación Post-Cambio:

- Evaluar el impacto del cambio una vez implementado.
- Medir los resultados contra las expectativas iniciales y determinar si se han logrado los objetivos del cambio.
- Documentar las lecciones aprendidas y ajustar los procedimientos de gestión de cambios para mejorar la eficacia en futuros proyectos.

Estos procedimientos de control y evaluación son esenciales para mantener la calidad y la coherencia durante todo el ciclo de vida del proyecto "PartyUpp". Asegurarán que las incidencias se gestionen de manera eficiente y que los cambios necesarios se implementen sin comprometer los objetivos del proyecto.

5**INTERFAZ DE USUARIO**

A continuación se describen las principales interfaces de usuario con las que cuenta la aplicación:

1. Interfaz de inicio de sesión:



Figura 1: Interfaz de inicio de sesión

Esta interfaz define una pantalla de inicio de sesión (login), utilizando un `LinearLayout` con orientación vertical. El contenido se centra horizontalmente en la pantalla. A continuación, se detalla cada componente y su disposición:

1. **`LinearLayout` principal:**

- Ancho y alto establecidos en `match_parent`, ocupando toda la pantalla.

- La gravedad está configurada en center, centrando los elementos horizontalmente.
- Orientación vertical.
- Fondo configurado con un drawable.
- Contexto de herramientas definido para la actividad LoginActivity.

2. Espacio inicial:

- Un espacio de 104dp de alto para crear un margen superior.

3. TextView para el título de inicio de sesión:

- Ancho y alto ajustados automáticamente (wrap_content).
- Margen inferior de 10dp.
- Texto configurado con un recurso de cadena @string/log_in_tag.
- Tamaño del texto de 30sp y estilo en negrita (bold).

4. Campo de texto para el correo electrónico:

- Utiliza TextInputLayout y TextInputEditText de la biblioteca de Material Design.
- Ancho match_parent y alto wrap_content.
- Márgenes laterales (inicio y fin) de 20dp y margen inferior de 10dp.
- El TextInputEditText tiene un id editTextEmailLogin, un placeholder configurado con @string/email_placeholder_login, entrada de texto (text), una línea máxima (maxLines="1"), y opciones de teclado configuradas para actionDone.

5. Campo de texto para la contraseña:

- Configurado de manera similar al campo de correo electrónico.
- El TextInputEditText tiene un id editTextPasswordLogin, un placeholder configurado con @string/password_placeholder_login, entrada de texto de tipo contraseña (textPassword), una línea máxima, y opciones de teclado configuradas para actionDone.

6. TextView para mostrar errores de inicio de sesión:

- Ancho y alto ajustados automáticamente (wrap_content).
- Margen inferior de 20dp.
- Color de texto rojo configurado con @color/red.
- Texto configurado con @string/login_error.
- Estilo de texto en negrita.
- Visibilidad inicial configurada como gone (invisible).

7. ProgressBar:

- Ancho y alto ajustados automáticamente.
- Id configurado como progressBarLogin.
- Margen inferior de 20dp.
- Visibilidad inicial configurada como gone.

8. Espacio inferior medio:

- Un espacio de 94dp de alto para crear separación.

9. Botón de envío de inicio de sesión:

- Ancho y alto ajustados automáticamente (wrap_content).
- Id configurado como buttonSubmitLogin.
- Color de fondo negro configurado con @color/black.
- Elevación de 30dp.
- Sin animador de estado (stateListAnimator="@null").
- Texto configurado con @string/button_submit_login, tamaño de texto de 26sp, y una elevación adicional de 2dp (translationZ).

10. Espacio inferior adicional:

- Un espacio de 50dp de alto para crear separación.

11. TextView para registro:

- Ancho y alto ajustados automáticamente.
- Id configurado como textViewRegisterLogin.
- Texto configurado con @string/button_register_login.
- Tamaño del texto de 16sp y estilo en negrita.

12. Espacio inferior adicional:

- Un espacio de 50dp de alto para crear separación.

13. Botón para continuar como invitado:

- Ancho y alto ajustados automáticamente (wrap_content y 37dp respectivamente).
- Id configurado como buttonContinueAsGuest.
- Color de fondo negro configurado con @color/black.
- Texto configurado con @string/button_continue_as_guest, color de texto blanco configurado con @color/white, tamaño de texto de 12sp.

- Elevación de 10dp.
- Sin animador de estado (stateListAnimator="@null").
- Elevación adicional de 2dp (translationZ).

En resumen, el layout está diseñado para una pantalla de inicio de sesión con elementos centrados vertical y horizontalmente, incluyendo campos de texto para el correo electrónico y la contraseña, botones para iniciar sesión y continuar como invitado, y espacios para separación visual.

2. Interfaz de registro de usuarios

Esta interfaz define una pantalla de registro, usando un `LinearLayout` con orientación vertical. Los elementos se centran horizontalmente en la pantalla.



Figura 2: Interfaz de registro de usuarios

A continuación, se describe cada componente y su disposición:

1. Contenedor principal (LinearLayout):

- Ancho y alto configurados para ocupar toda la pantalla (match_parent).
- Contenido centrado horizontalmente (gravity="center").

- Orientación vertical.
- Fondo configurado con un drawable.
- Contexto de herramientas establecido para RegistrationActivity.

2. Texto para el título de registro:

- Ancho y alto ajustados automáticamente (wrap_content).
- Margen inferior de 10dp.
- Texto configurado con un recurso de cadena @string/sign_up_tag.
- Tamaño de texto de 30sp y estilo en negrita (bold).

3. Campo de texto para el nombre de usuario:

- Utiliza TextInputLayout y TextInputEditText de la biblioteca de Material Design.
- Ancho match_parent y alto wrap_content.
- Márgenes laterales (inicio y fin) de 20dp y margen inferior de 10dp.
- El TextInputEditText tiene un id editTextUsernameRegistration, un placeholder configurado con @string/username_placeholder_login, entrada de texto (text), una línea máxima (maxLines="1"), y opciones de teclado configuradas para actionDone.

4. Campo de texto para el correo electrónico:

- Configurado de manera similar al campo de nombre de usuario.
- El TextInputEditText tiene un id editTextEmailRegistration, un placeholder configurado con @string/email_placeholder_login, entrada de texto (text), una línea máxima, y opciones de teclado configuradas para actionDone.

5. Campo de texto para la contraseña:

- Configurado de manera similar a los campos anteriores.
- El TextInputEditText tiene un id editTextPasswordRegistration, un placeholder configurado con @string/password_placeholder_login, entrada de texto de tipo contraseña (textPassword), una línea máxima, y opciones de teclado configuradas para actionDone.

6. Campo de texto para repetir la contraseña:

- Configurado de manera similar a los campos anteriores.
- El TextInputEditText tiene un id editTextRepeatPassword, un placeholder configurado con @string/repeat_password_placeholder_login, entrada de texto de tipo contraseña, una línea máxima, y opciones de teclado configuradas para actionDone.

7. Botón para enviar el registro:

- Ancho y alto ajustados automáticamente (wrap_content).
- Id configurado como buttonSubmitRegistration.
- Texto configurado con @string/button_submit_registration.
- Tamaño del texto de 26sp.
- Margen superior de 20dp.
- Color de fondo negro configurado con @color/black.

En resumen, la pantalla de registro está diseñada con elementos centrados vertical y horizontalmente, incluyendo campos de texto para nombre de usuario, correo electrónico, contraseña, y repetición de contraseña, y un botón para enviar el formulario de registro.

3. Interfaz de inicio

Esta pantalla define una interfaz para la actividad principal usando un ConstraintLayout.



Figura 3: Interfaz de inicio

A continuación se describen los componentes y su disposición:

1. Contenedor principal (ConstraintLayout):

- Ancho y alto configurados para ocupar toda la pantalla (match_parent).
- Fondo configurado con un drawable.

- Tintado de fondo y primer plano configurados como transparentes.
- Contexto de herramientas establecido para MainActivity.

2. BottomNavigationView:

- Utilizado para la navegación inferior en la aplicación.
- Ancho configurado para ajustarse al ancho del contenedor (0dp) y alto de 80dp.
- Id configurado como navigationViewMain.
- Fondo y tintado del primer plano configurados en rojo (#FF0000).
- Tintado de fondo configurado en un color gris (#706666).
- Fondo de los ítems configurado en negro (@color/black).
- Tintado de los íconos de los ítems configurado en blanco (@color/white).
- Color de ripple de los ítems configurado en rojo (@color/red).
- Color de texto de los ítems configurado en blanco (#FFFFFF).
- Posicionamiento mediante restricciones:
 - La parte inferior del componente está alineada con la parte inferior del contenedor padre.
 - Los lados izquierdo y derecho están alineados con los lados izquierdo y derecho del contenedor padre.
 - Sesgo horizontal de 0.5 para centrar horizontalmente.

3. FragmentContainerView:

- Utilizado para contener un fragmento de navegación.
- Ancho y alto configurados para ajustarse al tamaño disponible (0dp).
- Nombre del fragmento configurado como androidx.navigation.fragment.NavHostFragment.
- Configurado como el host de navegación predeterminado (app:defaultNavHost="true").
- Gráfico de navegación configurado con @navigation/my_nav.
- Posicionamiento mediante restricciones:
 - La parte inferior del componente está alineada con la parte superior de navigationViewMain.
 - Los lados izquierdo y derecho están alineados con los lados izquierdo y derecho del contenedor padre.
 - La parte superior del componente está alineada con la parte superior del contenedor padre.
 - Sesgo horizontal de 0.0 (alineado a la izquierda) y sesgo vertical de 1.0 (alineado a la parte inferior).
- Esta vista carga automáticamente el siguiente fragmento principal (HomeFragment):

- Ancho y alto configurados para ocupar toda la pantalla (match_parent).
- Fondo configurado con un drawable.
- Contexto de herramientas establecido para HomeFragment.
- Ancho configurado para ocupar toda la pantalla (match_parent) y alto de 81dp.
- Orientación horizontal.
- Gravedad del contenido centrada verticalmente.
- **Espacio vacío (izquierda):**
 - Ancho de 285dp y alto para ajustar al padre (match_parent).
 - Utiliza peso (layout_weight) de 1 para distribuir espacio proporcionalmente.
- **Imagen de perfil (ShapeableImageView):**
 - Id configurado como imgViewProfile.
 - Ancho de 50dp y alto de 52dp.
 - Gravedad configurada a la derecha y centrada verticalmente.
 - Imagen de origen configurada con @drawable/profile.
 - Apariencia circular mediante @style/circularShape.
 - Color del borde blanco y ancho del borde de 1dp.
- **Espacio vacío (derecha):**
 - Ancho de 30dp y alto para ajustar al padre (match_parent).
 - Utiliza peso de 1 para distribuir espacio proporcionalmente.
- **Botón de tutorial:**
 - Id configurado como buttonTutorial.
 - Estilo de botón elevado de Material Design.
 - Ancho de 250dp y alto de 70dp.
 - Fondo tintado en color cian (#26C6DA).
 - Gravedad del primer plano centrada.
 - Texto configurado con @string/main_tutorial_button, alineado al centro, color del texto en blanco (#F3F3F3), tamaño del texto de 24sp.
 - Elevación de 10dp y translación en el eje Z de 2dp.
 - Animador de estado configurado como nulo (@null).
 - Posicionamiento mediante restricciones:
 - La parte inferior del botón está alineada con la parte superior del botón de jugar.
 - Los lados izquierdo y derecho están alineados con los lados izquierdo y derecho del contenedor padre.
 - Sesgo horizontal de 0.496 para centrar horizontalmente.

- Sesgo vertical de 0.597 para posicionar verticalmente.
- **Botón de jugar:**
 - Id configurado como buttonPlay.
 - Estilo de botón elevado de Material Design.
 - Ancho de 300dp y alto de 130dp.
 - Fondo tintado en color azul (#42A5F5).
 - Gravedad del primer plano centrada.
 - Texto configurado con @string/main_play_button, alineado al centro, color del texto en blanco (#F3F3F3), tamaño del texto de 48sp.
 - Elevación de 10dp y translación en el eje Z de 2dp.
 - Animador de estado configurado como nulo (@null).
 - Posicionamiento mediante restricciones:
 - La parte inferior del botón está alineada con la parte inferior del contenedor padre.
 - Los lados izquierdo y derecho están alineados con los lados izquierdo y derecho del contenedor padre.
 - Sesgo horizontal de 0.495 para centrar horizontalmente.
 - Sesgo vertical de 0.52 para posicionar verticalmente.

4. Interfaz de la galería

Esta pantalla define una interfaz de usuario para la galería utilizando un FrameLayout.

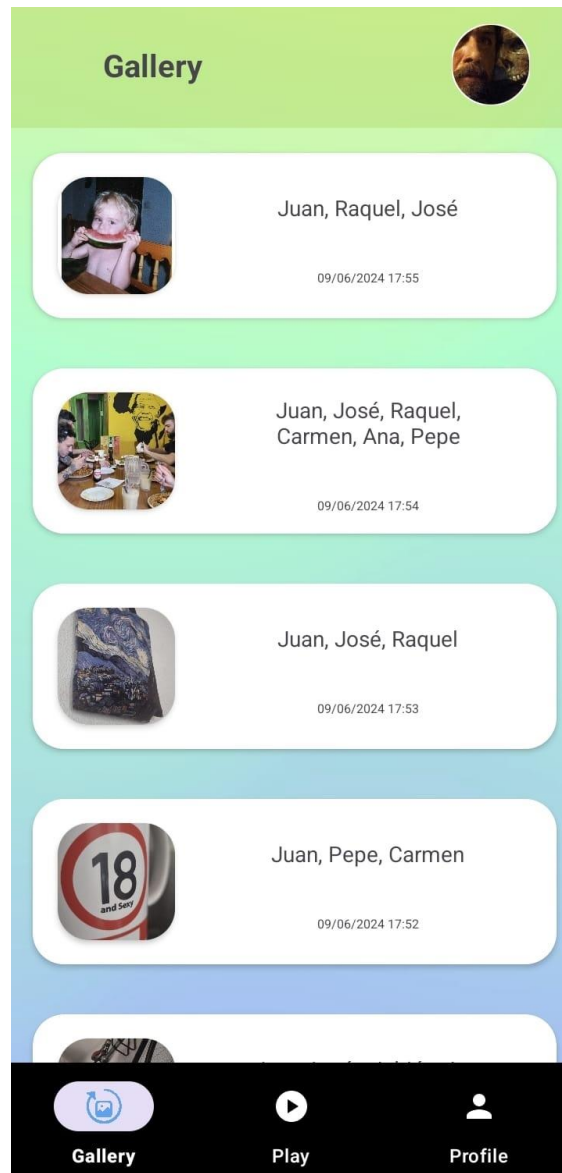


Figura 4: Interfaz de la galería

A continuación se describen los componentes y su disposición:

1. Contenedor principal (FrameLayout):

- Tiene un ancho y alto que ocupan toda la pantalla.
- Utiliza un fondo configurado con un drawable.
- Está destinado a un contexto llamado GalleryFragment.

2. LinearLayout vertical:

- Su ancho y alto también ocupan toda la pantalla.
- La orientación es vertical.

Dentro de este LinearLayout vertical se encuentran dos componentes principales:

a. LinearLayout horizontal (cabecera):

- Ocupa todo el ancho y tiene una altura de 81dp.
- Tiene un fondo de color semitransparente con el código hexadecimal #77A6DC5E.
- La gravedad y la gravedad del primer plano están configuradas para llenar horizontal y verticalmente.
- La orientación es horizontal.

Dentro de este LinearLayout horizontal se encuentran los siguientes elementos:

- Espacio vacío (izquierda):
 - Tiene un ancho de 30dp y un alto que ajusta al padre.
 - Utiliza peso (layout_weight) de 1 para distribuir espacio proporcionalmente.
- Título de la galería (TextView):
 - Su id es textViewGalleryTitle.
 - Tiene un ancho de 223dp y un alto de 81dp.
 - Gravedad configurada a la izquierda y centrada verticalmente.
 - Texto configurado con @string/gallery_menu_ttl, alineado al inicio del texto, tamaño del texto de 20sp, estilo de texto en negrita.
- Imagen de usuario (ShapeableImageView):
 - Su id es imageViewUserPhoto.
 - Tiene un ancho de 50dp y un alto de 52dp.
 - Gravedad configurada a la derecha y centrada verticalmente.
 - Imagen de origen configurada con @drawable/profile, con forma circular, borde blanco y ancho de trazo de 1dp.
- Espacio vacío (derecha):
 - Tiene un ancho de 30dp y un alto que ajusta al padre.
 - Utiliza peso (layout_weight) de 1 para distribuir espacio proporcionalmente.

b. RelativeLayout (contenido principal):

- Ocupa todo el ancho y alto restantes.
- Dentro de este RelativeLayout hay los siguientes elementos:
- RecyclerView:
 - Su id es recyclerView.
 - Ocupa todo el espacio disponible.
- TextView (mensaje sin juegos jugados):
 - Su id es textViewNoGamesPlayed.
 - Tiene un ancho de 196dp y un alto de 132dp.
 - Márgenes configurados para centrarlo en el RelativeLayout.

- Texto configurado con `@string/no_games_played_msg`, color de texto negro, estilo en negrita y gravedad centrada.

5. Interfaz del perfil

Esta pantalla contiene el fragmento de configuración utilizando `ConstraintLayout` como contenedor principal.



Figura 5: Interfaz del perfil

A continuación se detallan los componentes y su disposición:

1. ConstraintLayout:

- Este es el contenedor principal que abarca toda la pantalla.
- Tiene un fondo configurado con un drawable.

- Está asociado al contexto SettingsFragment.

2. **LinearLayout vertical:**

- Tiene un ancho de 298dp y una altura de 613dp.
- Su gravedad está configurada como central.
- Orientación vertical.
- Está anclado a los bordes del ConstraintLayout.

Dentro de este LinearLayout vertical se encuentran varios componentes:

a. **ShapeableImageView:**

- Muestra una imagen, posiblemente un perfil de usuario.
- Tiene un ancho de 150dp y una altura de 155dp.
- Tiene un margen superior de 44dp.
- El fondo es negro.
- Tiene un borde blanco con un ancho de 3dp y una forma circular.
- La imagen que muestra es @drawable/vector_profile.
- Escala la imagen para ajustarse al tamaño definido.

b. **Espacio:**

- Tiene un alto de 15dp.
- Se utiliza para crear separación entre los elementos.

c. **TextView:**

- Muestra el nombre de usuario.
- Configurado para centrar el texto.
- El tamaño del texto es de 20sp y en negrita.

d. **Switches:**

- Varios interruptores para configurar diferentes opciones (sonidos, vibración, música, notificaciones).
- Cada uno tiene un ancho de 200dp, una altura de 45dp y un margen superior de 12dp.
- El texto asociado a cada interruptor es en negrita y tamaño de 18sp.

e. **Button:**

- Se utiliza para iniciar o cerrar sesión.
- Tiene un ancho de 189dp, una altura de 43dp y un margen superior de 76dp.
- El texto del botón es @string/button_submit_registration.
- El color de fondo está configurado como negro.
- El texto es de color blanco y centrado.

6. Interfaz de selección de jugadores

La interfaz de usuario para la actividad de selección de jugadores es la siguiente:

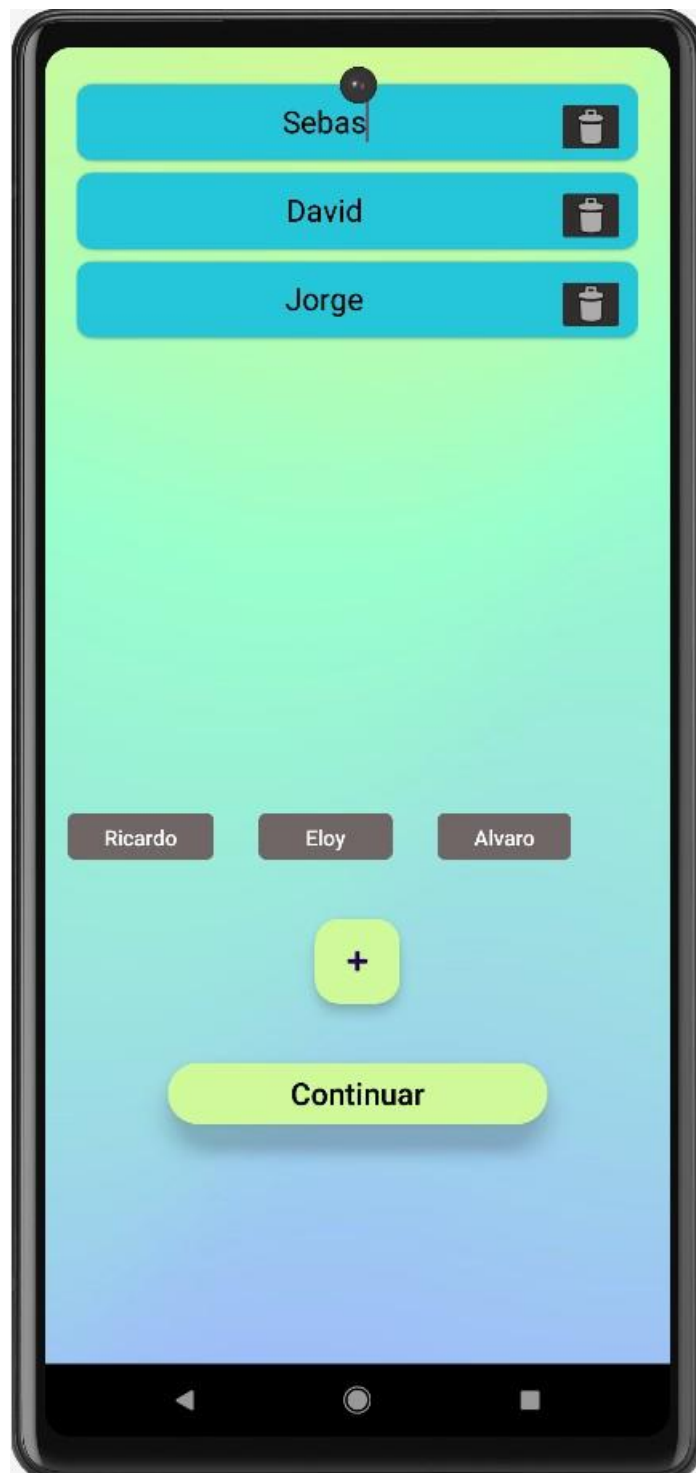


Figura 6: Interfaz de selección de jugadores

Aquí está la descripción de los componentes y su disposición:

1. ConstraintLayout:

- Este es el contenedor principal que abarca toda la pantalla.
- Tiene un fondo configurado con un drawable.
- El color de fondo se ajusta a @color/white.
- Asociado al contexto PlayerSelectionActivity.

2. **LinearLayout vertical:**

- Tiene un ancho y alto que ocupan toda la pantalla.
- Orientación vertical.

Dentro de este LinearLayout vertical se encuentran varios componentes:

a. **Espacio:**

- Tiene un alto de 20dp para crear un espacio entre la parte superior de la pantalla y el RecyclerView.

b. **RecyclerView de selección de jugadores:**

- Identificado por @+id/recyclerViewPlayerSelection.
- Ancho de 370dp y altura de 470dp.
- Centrado horizontalmente.
- Fondo transparente.

c. **Espacio:**

- Tiene un alto de 10dp para crear un espacio entre el RecyclerView de selección de jugadores y el RecyclerView de jugadores sugeridos.

d. **RecyclerView de jugadores sugeridos:**

- Identificado por @+id/recyclerViewSuggestedPlayers.
- Ancho y altura configurados para que se ajusten al contenido.
- Orientación horizontal para que los jugadores sugeridos se desplacen horizontalmente.
- Visible solo si hay jugadores sugeridos.

e. **Espacio:**

- Tiene un alto de 30dp para crear un espacio entre el RecyclerView de jugadores sugeridos y el botón flotante de agregar jugador.

f. **FloatingActionButton para agregar jugador:**

- Identificado por @+id/floatingActionButtonAddPlayer.
- Ancho de 409dp y altura ajustada al contenido.
- Centrado horizontalmente.
- Color de fondo #7B1FA2.
- Color de fondo del botón flotante #D0F999.
- Icono de agregar jugador.

g. **Espacio:**

- Tiene un alto de 35dp para crear un espacio entre el botón flotante de agregar jugador y el botón de iniciar juego.

h. Button para iniciar juego:

- Identificado por @+id/buttonStartGame.
- Ancho de 250dp y altura ajustada al contenido.
- Centrado horizontalmente.
- Texto "Continuar".
- Color de fondo #D0F999.
- Color de texto #000000.
- Tamaño de texto de 20sp.
- Elevación y traducción Z configuradas para dar una apariencia elevada al botón.

7. Interfaz de foto inicial

La interfaz de usuario para la actividad de la foto inicial está definida así:



Figura 7: Interfaz de foto inicial

Aquí está la descripción de los componentes y su disposición:

1. ConstraintLayout:

- Este es el contenedor principal que abarca toda la pantalla.
- Tiene un fondo configurado con un drawable.

- Asociado al contexto InitialPhotoActivity.

2. **ImageButton:**

- Identificado por @+id/imageButtonCamera.
- Ancho de 275dp y altura de 290dp.
- Escala de tipo de imagen configurada para fitCenter.
- Margen superior de 100dp y margen inferior de 100dp.
- Se coloca debajo del TextView textViewInitialPhotoMessage y encima del Button buttonLaunchGame.

3. **TextView:**

- Identificado por @+id/textViewInitialPhotoMessage.
- Ancho de 235dp y altura de 110dp.
- Margen superior de 60dp.
- Alineado al centro horizontalmente.
- Texto "Texto superior de la foto inicial" configurado en negrita y tamaño de texto de 20sp.

4. **Button:**

- Identificado por @+id/buttonLaunchGame.
- Ancho de 300dp y altura de 60dp.
- Margen inferior de 120dp.
- Color de fondo configurado a #D0F999.
- Alineado al centro horizontalmente.
- Se coloca debajo del ImageButton imageButtonCamera.

Todos los elementos están anclados a los bordes del ConstraintLayout y están bien distribuidos verticalmente en la pantalla. Los elementos tienen alturas y anchuras definidas y están centrados horizontalmente.

8. **Interfaz de partida**

La interfaz de usuario para un fragmento de desafío simple es la siguiente:



Figura 8: Interfaz de partida

Aquí está la descripción de los elementos y su disposición:

1. **ConstraintLayout:**

- Este es el contenedor principal que abarca toda la pantalla.
- Tiene un identificador @+id/frameLayout.
- Asociado al contexto SimpleChallengeFragment.

2. **TextView:**

- Identificado por @+id/textViewSimpleChallenge.
- Ancho de 329dp y altura de 327dp.
- Fuente configurada como "Alata".
- Texto predeterminado.
- Tamaño de texto de 24sp.

- Alineado al centro horizontal y verticalmente.
- Se coloca en la parte superior de `Button buttonNextSimpleChallenge`.

3. **Button:**

- Identificado por `@+id/buttonNextSimpleChallenge`.
- Ancho de 140dp y altura de 81dp.
- Margen inferior de 160dp.
- Color de fondo configurado a `#D0F999`.
- Texto "Siguiendo desafío".
- Tamaño de texto de 20sp.
- Alineado al centro horizontalmente.
- Se coloca en la parte inferior del `ConstraintLayout`.

Ambos elementos están anclados a los bordes del `ConstraintLayout` y están bien distribuidos verticalmente en la pantalla. El `TextView` tiene un tamaño grande y ocupa la mayor parte de la pantalla, mientras que el `Button` está ubicado en la parte inferior con un margen adecuado.

9. Interfaz de final de partida

La interfaz de usuario para la actividad que se muestra al finalizar una partida es:

Esta es la descripción de los elementos y su disposición:

1. **ConstraintLayout:**

- Este es el contenedor principal que abarca toda la pantalla.

2. **Button:**

- Identificado por @+id/buttonEndGameContinue.

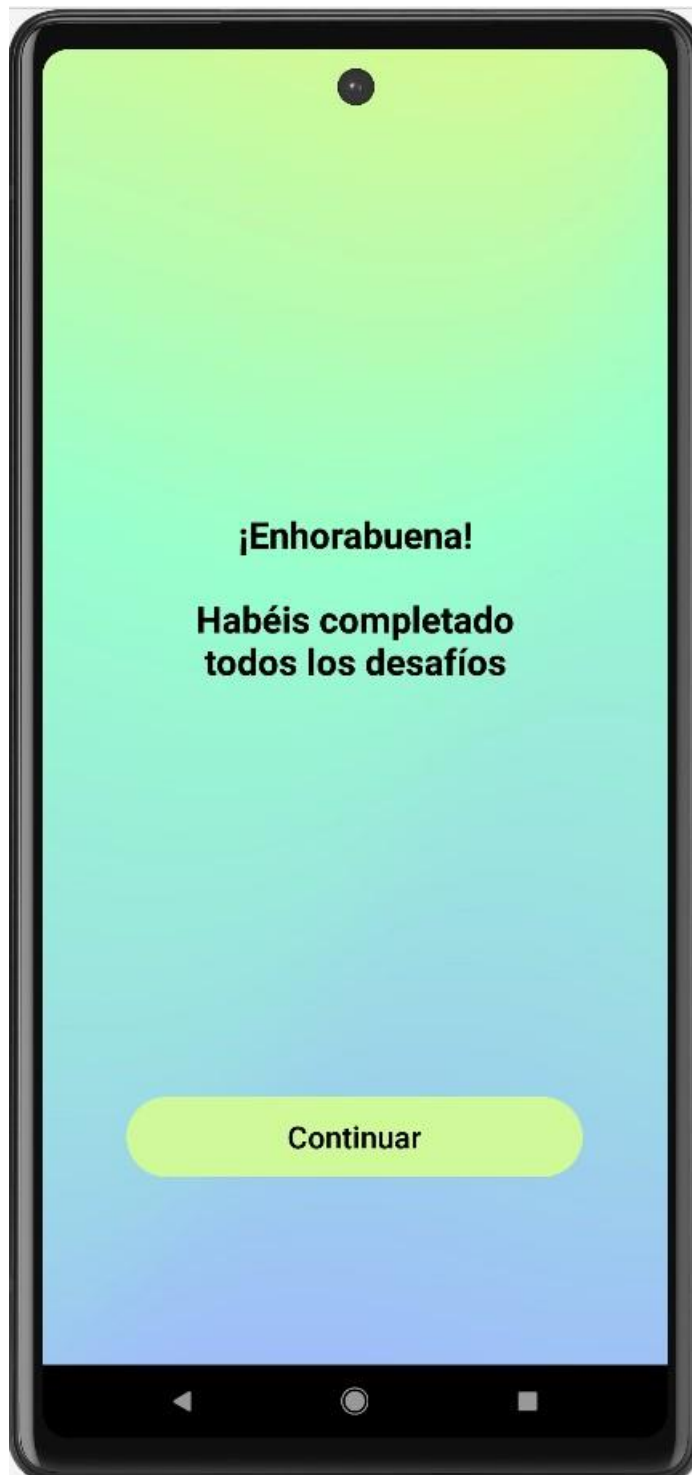


Figura 9: Interfaz de final de partida

- Ancho de 300dp y altura de 60dp.
- Margen inferior de 120dp.
- Color de fondo configurado a #D0F999.
- Texto "Continuar".
- Tamaño de texto de 20sp.
- Alineado al centro horizontalmente y anclado al fondo de ConstraintLayout.

3. **TextView:**

- Identificado por @+id/textView.
- Ancho de 237dp y altura de 145dp.
- Margen superior de 220dp.
- Texto predeterminado.
- Tamaño de texto de 24sp.
- Estilo de texto en negrita.
- Alineado al centro horizontalmente y anclado a la parte superior de ConstraintLayout, justo debajo del TextView.

Ambos elementos están anclados a los bordes del ConstraintLayout y están bien distribuidos verticalmente en la pantalla. El TextView tiene un tamaño relativamente grande y está ubicado en la parte superior de la pantalla, mientras que el Button está centrado verticalmente y ubicado en la parte inferior de la pantalla.

10. **Interfaz de foto final**

La interfaz de usuario para la actividad de la foto final es la siguiente:



Figura 10: Interfaz de foto final

Esta es la descripción de los elementos y su disposición:

1. **ConstraintLayout:**

- Este es el contenedor principal que abarca toda la pantalla.

2. **Button:**

- Identificado por @+id/buttonSaveAndFinish.

- Ancho de 300dp y altura de 60dp.
- Margen inferior de 120dp.
- Color de fondo configurado a #D0F999.
- Texto "Finalizar Juego".
- Tamaño de texto de 20sp.
- Alineado al centro horizontalmente y anclado al fondo de ConstraintLayout.

3. **ImageButton:**

- Identificado por @+id/imageButtonEndCamera.
- Ancho y alto de 275dp.
- Escala ajustada al centro.
- Alineado debajo del TextView y arriba del Button.
- Se usa como un botón para tomar una foto.

4. **TextView:**

- Identificado por @+id/textViewEndPhotoMessage.
- Ancho de 274dp y altura de 83dp.
- Margen superior de 60dp.
- Texto predeterminado.
- Tamaño de texto de 20sp.
- Estilo de texto en negrita.
- Alineado al centro horizontalmente y anclado a la parte superior de ConstraintLayout.

Ambos elementos están anclados a los bordes del ConstraintLayout y están bien distribuidos verticalmente en la pantalla. El TextView tiene un tamaño relativamente grande y está ubicado en la parte superior de la pantalla, mientras que el Button está centrado verticalmente y ubicado en la parte inferior de la pantalla. El ImageButton está en el centro de la pantalla, debajo del TextView y arriba del Button.