



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de un Trivial para Android

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Espasa Font, Rubén

Tutor: Carsí Cubel, José Ángel

2018-2019

Resumen

El propósito de este proyecto consiste en adaptar el clásico juego de mesa Trivial Pursuit en un videojuego para dispositivos móviles Android, con el añadido de que los jugadores puedan disputar tanto partidas en línea como en local.

Asimismo, se exponen los diferentes procesos que han sido necesarios para el desarrollo de la aplicación desde la especificación de requisitos hasta las pruebas realizadas sobre el producto.

Finalmente se plantean una serie de conclusiones sobre si se han alcanzado los objetivos propuestos al inicio del proyecto.

Palabras clave: videojuego, Trivial Pursuit, Android, juego de mesa.

Abstract

The purpose of this project is to adapt the classic Trivial Pursuit board game into a videogame for Android mobile devices, considering that players can play both online and offline games.

Likewise, the different processes that have been necessary for the development of the application are exposed, from the specification of requirements to the tests carried out on the product.

Finally, a series of conclusions are raised as to whether the objectives proposed at the beginning of the project have been achieved or not.

Keywords: videogame, Trivial Pursuit, Android, board game.

Tabla de contenidos

1.	Introducción	7
1.1	Motivación	7
1.2	Objetivos del proyecto	7
1.3	Metodología	8
1.4	Estructura de la memoria	9
2.	Estado del arte	11
2.1	Introducción.....	11
2.2	Trivia 360	11
2.3	Cuestionados	11
2.4	Preguntados.....	12
2.5	Conclusiones.....	13
3.	Especificación de requisitos.....	15
3.1	Introducción.....	15
3.1.1	Propósito	15
3.1.2	Ámbito del sistema.....	15
3.1.3	Definiciones, acrónimos y abreviaturas.....	15
3.1.4	Referencias.....	16
3.2	Descripción general.....	16
3.2.1	Perspectiva del producto.....	16
3.2.2	Funcionalidad del producto	17
3.2.3	Características de los usuarios	17
3.2.4	Restricciones	17
3.2.5	Suposiciones y dependencias.....	18
3.3	Requisitos específicos.....	18
3.3.1	Requisitos funcionales	18
3.3.2	Requisitos no funcionales	25
4.	Análisis	27
4.1	Diagrama de clases.....	27
5.	Diseño	31
5.1	Capa de presentación	31
5.2	Capa de negocio	39



5.3 Capa de persistencia.....	39
6. Implementación	41
6.1 Lenguajes empleados	41
6.2 Tecnologías utilizadas	42
6.2.1 Android Studio	42
6.2.1 Google Play Game Services	42
6.3 Desarrollo del juego	42
6.3.1 Capa de presentación.....	42
6.3.2 Capa lógica.....	45
6.3.3 Capa de persistencia	50
7. Pruebas	51
8. Conclusiones y futuras mejoras.....	53
9. Bibliografía.....	55
10. Apéndice	57
10.1 Monetización del juego.....	57
10.2 Gamificación	58
10.3 Manual de usuario.....	59
10.3.1 Menú principal	59
10.3.2 Modo partidas en línea.....	60
10.3.3 Modo práctica	64
10.3.4 Modo partidas en local	65
10.3.5 Jugando la partida	67

1. Introducción

En este apartado se va a explicar los motivos de la elección de este proyecto, los objetivos que se pretende conseguir, la metodología empleada para el desarrollo del proyecto y por último la estructura de esta memoria.

1.1 Motivación

En los últimos años la popularidad de los juegos en los dispositivos móviles ha aumentado significativamente debido a la mejora significativa del rendimiento en aspectos tanto del *hardware* como el *software*. Tanto es así que, en 2016, un 72.3% de los usuarios de dispositivos móviles en Estados Unidos jugaban a juegos. Se estima que esta cifra alcance a un 77% en 2020 (1). Teniendo en cuenta que cada vez los videojuegos en dispositivos móviles son más populares se decidió desarrollar un videojuego para estos terminales.

La elección de desarrollar un Trivial para Android viene dada por que se trata de un juego de mesa muy popular, conocido y con un gran número de usuarios potenciales, ya que se trata de un juego de relativa facilidad y apto para todas las edades. Es cierto que actualmente existen una gran cantidad de videojuegos parecidos al Trivial publicados en la Google Play Store, pero muy pocos o casi ninguno son una fiel representación del juego de mesa original. Por ello, otro de los motivos por el cual se ha decidido desarrollar este videojuego es realizar una fiel representación del juego de mesa para dispositivos móviles.

1.2 Objetivos del proyecto

Como se ha comentado en el apartado anterior, el principal objetivo de este proyecto es realizar una fiel adaptación del juego de mesa en un videojuego para dispositivos móviles.

Para ello las partidas se deberán desarrollar sobre un tablero idéntico al del juego original, con un mínimo de dos jugadores y un máximo de seis. Por otro lado, existirán dos tipos de partidas: las en línea y las que se juegan en local. El primer tipo permitirá a los usuarios jugar cuando les sea posible y desde su propio terminal sin necesidad de estar junto a los demás jugadores. En cambio, las partidas en local

permitirán a los usuarios jugar a la vez en mismo dispositivo móvil y sin requerir de una conexión a internet.

Otro objetivo que debe cumplir la aplicación es permitir guardar y cargar partidas para que se pueda retomar cuando y donde desee.

Por último, para fomentar la competitividad entre los jugadores que juegan partidas en línea, se deberá implementar un *ranking* y un sistema de logros.

1.3 Metodología

La metodología de desarrollo que se va a utilizar en este proyecto va a ser iterativa e incremental. Se ha decidido utilizar esta metodología ya que se trata de una metodología ágil y nos va a permitir disponer de una aplicación funcional desde la primera iteración.

Se va a dividir el desarrollo del juego en cinco iteraciones o *sprints*. En la primera iteración se va a desarrollar la gestión y la funcionalidad de las partidas en local. Como en esta iteración se desarrolla la parte más importante del juego será la que más durará.

En el segundo *sprint* se desarrollarán la gestión y funcionalidades de las partidas online. Este no será tan largo como el primero ya que como la funcionalidad principal de la partida ya estará implementada y solo será necesario adaptarla con las funcionalidades en línea.

En el tercer *sprint* se desarrollará el sistema de *ranking* y logros de las partidas en línea. En cuanto a la cuarta iteración, se implementará la funcionalidad de jugar contra jugadores controlados por la aplicación en las partidas en local.

Para finalizar, en el último *sprint* se llevará a cabo el modo de entrenamiento o práctica, en el cual se permitirá al usuario contestar preguntas durante un tiempo determinado.

1.4 Estructura de la memoria

Esta memoria está estructurada en diferentes capítulos, en cada cual se profundizará sobre el tema que traten.

En este capítulo se trata el porqué de la elección del proyecto, los objetivos que se desean alcanzar y la metodología que se ha utilizado para llevar a cabo el proyecto.

En el segundo se hablará sobre el estado del arte del juego a desarrollar.

En el tercer apartado se realizará la especificación de los requisitos del juego siguiendo el estándar IEEE 830.

En el cuarto, se llevará a cabo un análisis para obtener y representar en un diagrama de clases las diferentes entidades que formarán parte de la aplicación.

En el capítulo cinco se explicará el diseño que ha seguido el juego, así como la funcionalidad de cada uno de los componentes que conforman este proyecto.

En el apartado seis se expondrán tanto los lenguajes como las tecnologías utilizadas para el desarrollo del proyecto. A continuación, se explicarán los aspectos más importantes implementados del juego.

En el séptimo, se hará un breve resumen de las pruebas realizadas durante el desarrollo y los resultados obtenidos de estas.

En el octavo, se expondrán las conclusiones a las que se ha llegado al terminar el proyecto, así como las posibles mejoras que se harán sobre el juego en un futuro.

En el capítulo nueve se presentará la bibliografía del trabajo y para finalizar, se comentarán tres apéndices en los cuales se explicarán las posibilidades para monetizar el juego, la gamificación y un manual de usuario.



2. Estado del arte

2.1 Introducción

En este capítulo se va a realizar un estudio estratégico sobre el videojuego que se va a desarrollar. Debido a que el Trivial es un juego muy popular existen muchas aplicaciones en el mercado relacionadas con él mismo, por tanto, lo que se va a realizar es un análisis sobre las características que pueda compartir o no el proyecto en desarrollo con distintas aplicaciones existentes actualmente.

2.2 Trivia 360

En esta aplicación (2) el usuario debe contestar unos cuestionarios de diez preguntas, durante un minuto con la posibilidad de fallar hasta tres preguntas. Si se consigue superar, el jugador ganará una serie de puntos que le permitirán ir aumentando de nivel, para así ir subiendo en la clasificación.

Cabe destacar que el usuario solo se limita a contestar los cuestionarios, es decir, se puede afirmar que no se trata de una adaptación fiel del juego de mesa original ya que no requiere la interacción con otros usuarios, ni desplazarse en un tablero para contestar preguntas y ganar la partida. Otro aspecto que hay que recalcar, es que se necesita de acceso a internet para poder acceder a las funcionalidades de la aplicación.

2.3 Cuestionados

Este juego (3) dispone de tres modos de juego. El primero trata de recrear una partida del Trivial original, ya que sigue la misma dinámica de acertar las preguntas e ir desplazándose en un tablero, para así poder llegar a las casillas principales y conseguir los quesitos de cada categoría. Aunque la mecánica sea parecida, no es una fiel representación, ya que el tablero es circular y no dispone de las casillas centrales como el tradicional, ni permite jugar a más de dos jugadores en una misma partida.

El segundo modo de juego consiste en retar a otro usuario a un duelo de doce preguntas donde el jugador que más acierte ganará la pugna. Las cuestiones de los duelos podrán ser de cualquier categoría o bien de solo la seleccionada.

Por último, la modalidad restante tiene la misma estructura que la anterior solo que servirá de modo de práctica ya que el usuario es el único participante respondiendo a las preguntas. También hay que destacar que se requiere de una conexión a internet para poder acceder a las funcionalidades del juego.

2.4 Preguntados

En esta aplicación (4) existen diversos modos de juego, pero el análisis se va a centrar en solo el principal, ya que es el que más se asemeja a las funcionalidades que se van a implementar en este proyecto.

Esta modalidad consiste en una partida en línea de uno contra uno en que cada usuario tratará de conseguir todos y cada uno de los quesitos o premios de cada categoría, o bien tener más que el otro jugador al finalizar los 25 turnos de la partida. Sin embargo, la mecánica para conseguir estos premios consiste en girar una ruleta donde hay un total de siete casillas posibles, donde seis corresponden a cada uno de los tipos de preguntas y la restante es una especial. Si el usuario cae en una estas seis casillas deberá contestar la pregunta correspondiente con el tema de la casilla. Cuando este alcance un total de tres aciertos o bien se sitúe en la casilla especial, el jugador podrá o bien seleccionar una categoría de la cual aún no tenga el premio y contestar la pregunta pertinente, o bien retar al otro jugador a un duelo de seis preguntas para quitarle un quesito que el contrincante tenga y él no.

Como se puede observar en el punto anterior, esta aplicación solo se asemeja al juego original en la dinámica de conseguir los seis quesitos, ya que ni requiere de un tablero para el desarrollo de la partida, ni permite más dos jugadores en una misma partida. Otro aspecto común con las dos aplicaciones analizadas es que también necesita una conexión a internet para poder jugar.

2.5 Conclusiones

Mediante el análisis realizado se ha conseguido recopilar una serie de características que el proyecto a desarrollar comparte y no con las aplicaciones examinadas.

Característica	Trivia 360	Cuestionados	Preguntados	Proyecto
Partidas en local	x	x	x	✓
Partidas en línea	✓	✓	✓	✓
Tablero	x	✓±	x	✓
Partidas de un jugador	✓	x	x	✓
Partidas de dos jugadores	x	✓	✓	✓
Partidas de más de dos jugadores	x	x	x	✓
Duelos entre jugadores	x	✓	✓	x
<i>Ranking</i>	✓	✓	✓	✓

Como se observa en la tabla anterior, el juego que más se asemeja a lo que se desea implementar es el de Cuestionados. Aun así, aspectos como las partidas en local o las partidas de más de dos jugadores están ausentes. Una característica que comparten es la mecánica del desarrollo de la partida mediante un tablero, pero tampoco se asemejan del todo ya que el tablero utiliza Cuestionados no es idéntico al original. Por tanto, se puede afirmar que no existe actualmente en el mercado de videojuegos para dispositivos Android, un juego que adapte fielmente el Trivial Pursuit.

3. Especificación de requisitos

Es este capítulo se va a llevar a cabo el análisis de requisitos del juego desarrollado siguiendo el estándar IEEE 830.

3.1 Introducción

3.1.1 Propósito

El objetivo que de este capítulo es especificar los requisitos tanto funcionales como no funcionales que debe cumplir el juego que se va a desarrollar.

3.1.2 Ámbito del sistema

La aplicación que se va a desarrollar trata de adaptar fielmente el juego de mesa Trivial a los dispositivos móviles que utilizan el sistema Android. Por tanto, el juego deberá permitir al usuario disputar y gestionar tanto las partidas en línea como en el local, así como el modo de práctica donde se podrá mejorar la contestación de las preguntas y ver los resultados mediante estadísticas al finalizar.

3.1.3 Definiciones, acrónimos y abreviaturas

Término	Descripción
Partida en línea	Partida jugada mediante una conexión a internet la cual permitirá al jugador jugar su turno cuando le sea posible. Una vez terminado su turno le otorgará el turno al siguiente jugador.
Partida en local	Partida jugada en el propio terminal sin requerir de conexión a internet.
Tablero	Recreación del tablero del juego de mesa original, el cual dispondrá de una casilla principal, seis casillas de quesito y diferentes casillas base que podrán pertenecer a una de las distintas categorías o bien a volver a tirar.
Casilla principal	Casilla central del tablero donde las fichas se situarán al principio de la partida y posteriormente los jugadores tendrán que alcanzar una vez conseguido todos los quesitos, para así poder contestar la pregunta de la categoría que deseen y ganar la partida en caso de acierto.

Casilla base	Casilla normal del tablero la cual pertenecerá a una de las seis categorías o bien a volver a tirar. Cada vez que el jugador se sitúe sobre esta casilla y pertenezca a una categoría se le formulará una pregunta. De lo contrario otorgará otra tirada sin necesidad de contestar a ninguna pregunta.
Casilla quesito	Casilla especial perteneciente a una de las seis categorías que otorgará un quesito al jugador en caso de que acierte la pregunta y no tuviera anteriormente el quesito correspondiente. Si el usuario ya poseyera el premio, ésta se comportaría como una casilla base normal.
Quesito	Premio que recibe el jugador al contestar correctamente la pregunta formulada en cada una de las casillas quesito. Para poder ganar la partida el usuario debe disponer de los seis quesitos.
Ficha	Representación gráfica de la posición del jugador en el tablero.
Categoría	Representa el ámbito de conocimiento por el cual formulará una pregunta. Hay seis tipos: geografía (azul), entretenimiento (violeta), deportes (naranja), ciencias (verde), arte y literatura (roja) e historia (amarilla).

3.1.4 Referencias

- Especificación de Requisitos según el estándar IEEE 830 (5).

3.2 Descripción general

3.2.1 Perspectiva del producto

Esta aplicación trata de adaptar de la manera más fiel posible el juego de mesa original del Trivial, como consecuencia la dinámica de las partidas será totalmente idéntica de la mecánica de este. También cabe destacar que al ser un juego muy popular existen muchas aplicaciones en el mercado, pero como se ha expuesto en el apartado del estado del arte, no existe ninguna adaptación realmente fiel en él.

3.2.2 Funcionalidad del producto

Las funcionalidades del juego se pueden dividir en distintos módulos como los siguientes:

Gestión de las partidas en local: el juego debe permitir al usuario el poder crear, guardar, cargar y borrar las partidas jugadas en el propio terminal. También debe dejar al usuario añadir o borrar jugadores antes de empezar la partida.

Gestión de las partidas en línea: para poder acceder a esta funcionalidad el sistema debe permitir al usuario iniciar sesión o cerrar sesión de su cuenta de Google Play Games. Además, el usuario debe poder crear partidas nuevas, aceptar o rechazar invitaciones, reanudar su turno, ver la clasificación y ver los logros conseguidos.

La funcionalidad de la partida es igual independientemente si es en línea o en local. Por ello en una partida, el jugador debe poder tirar los dados, mover la ficha a uno de los destinos posibles, contestar la pregunta formulada y seguir jugando o pasar el turno en caso de fallo.

Por último, en el modo de práctica el jugador podrá elegir el tiempo que quiere que dure la práctica y la categoría de las preguntas formuladas. Al finalizar el tiempo de la práctica o el usuario la abandone, se le mostrará una serie de estadísticas con el resultado de las contestaciones realizadas.

3.2.3 Características de los usuarios

El juego está enfocado para cualquier usuario frecuente de dispositivos móviles, por tanto, no requerirá de amplios conocimientos en el uso de terminales ya que el juego va a ser intuitivo y fácil de utilizar. El único conocimiento mínimo que debería tener el usuario sería conocer mínimamente la dinámica de una partida de Trivial.

3.2.4 Restricciones

Como el proyecto está enfocado al desarrollo de un juego en Android, solo los terminales que dispongan de este sistema operativo tendrán acceso a él. Además, los terminales que no tengan una versión de Android 5.0 (*Lollipop*) tampoco podrán



ejecutar la aplicación debido a que la API mínima seleccionada para este juego es la 21, que solo permite ejecutarse en versiones igual o superiores a la 5.0.

Por otro lado, todos aquellos usuarios que no dispongan de una conexión a internet no podrán acceder a las funcionalidades en línea, limitando así el acceso a todas las funcionalidades del juego.

3.2.5 Suposiciones y dependencias

Por lo que respecta a este apartado, la funcionalidad de las partidas en línea depende íntegramente de la API de Google Play Game Services (6), ya que este gestiona toda la parte *backend* de las partidas en línea. Como consecuencia, en caso de que fallara o no estuviera disponible durante un periodo de tiempo, la aplicación no permitiría a los usuarios jugar este tipo de partidas.

3.3 Requisitos específicos

3.3.1 Requisitos funcionales

Los requisitos de la gestión de las partidas se pueden dividir en dos ya que la aplicación tiene dos tipos de partida.

Gestión de partidas en línea

Identificación	RF01
Nombre	Autenticación del usuario.
Características	El sistema comprobará que los datos sean correctos y concederá acceso en caso afirmativo.
Descripción	El usuario deberá identificarse para poder acceder a las funcionalidades de las partidas en línea.
Prioridad	Alta

Tabla RF 1

Identificación	RF02
Nombre	Cerrar sesión.
Características	El sistema cerrará la sesión del usuario y lo devolverá al menú de inicio de sesión.
Descripción	Permite al usuario cerrar sesión de su cuenta de Google Games.
Prioridad	Media

Tabla RF 2

Identificación	RF03
Nombre	Comprobar partidas en línea.
Características	El sistema ofrecerá al jugador información general sobre el estado de sus partidas.
Descripción	Muestra información del estado de las partidas del usuario, ya estén pendientes de aceptar o rechazar una invitación, sea su turno o bien el turno de los otros jugadores.
Prioridad	Alta

Tabla RF 3

Identificación	RF04
Nombre	Crear partida en línea
Características	El sistema creará la partida con los jugadores seleccionados y permitirá que el creador juegue su turno. Una vez finalice su turno enviará una invitación al usuario que le toque jugar el próximo turno.
Descripción	El usuario podrá crear una partida nueva. También deberá proporcionar a los jugadores que desea que participen en ella.
Prioridad	Alta

Tabla RF 4

Identificación	RF05
Nombre	Aceptar/Rechazar invitación
Características	El sistema ofrecerá la posibilidad de aceptar o rechazarla. En caso positivo, el jugador jugará su turno, de lo contrario se cancelará la partida.
Descripción	Muestra al usuario la información de la partida propuesta y la opción de rechazar o aceptar participar en ella.
Prioridad	Alta

Tabla RF 5

Identificación	RF06
Nombre	Abandonar la partida.
Características	El sistema borrará al jugador de la partida y le otorgará el turno al próximo jugador. En caso de que solo quedara un jugador activo, la partida se cancelaría.
Descripción	El usuario abandonará la partida.
Prioridad	Alta

Tabla RF 6

Identificación	RF07
Nombre	Ver clasificación.
Características	El sistema mostrará el <i>ranking</i> de los jugadores con más partidas ganadas y en la que se encuentra el usuario.
Descripción	Muestra la información de todos los jugadores que han ganado partidas ordenada por quien haya ganado más.
Prioridad	Media

Tabla RF 7

Identificación	RF08
Nombre	Ver logros.
Características	El sistema mostrará la información de los logros conseguidos o no del usuario.
Descripción	Muestra el listado de logros que puede conseguir el jugador, si lo ha logrado o no y su progreso actual.
Prioridad	Media

Tabla RF 8

Gestión partidas en local

Identificación	RF09
Nombre	Crear partida en local.
Características	El sistema crea una partida con su identificador correspondiente y sin jugadores.
Descripción	Permite al usuario crear una partida nueva.
Prioridad	Alta

Tabla RF 9

Identificación	RF10
Nombre	Añadir jugador a la partida.
Características	El sistema añadirá el jugador a la partida.
Descripción	Permite al usuario añadir un jugador a la partida. Se debe proporcionar el nombre del jugador, el color, y si se trata de un jugador controlado por el sistema o no. En caso de que lo esté, también se deberá proporcionar su dificultad.
Prioridad	Alta

Tabla RF 10

Identificación	RF11
Nombre	Editar jugador
Características	El sistema modificará la información del jugador en cuestión
Descripción	Permite al usuario modificar las características.
Prioridad	Media

Tabla RF 11

Identificación	RF12
Nombre	Borrar jugador de la partida
Características	El sistema borrará al jugador.
Descripción	Permite al usuario borrar al jugador que desee.
Prioridad	Alta

Tabla RF 12

Identificación	RF13
Nombre	Empezar partida.
Características	El sistema empezará la partida, situando a todos los jugadores en la casilla principal y otorgándole el turno al primer jugador
Descripción	Permite al usuario empezar la partida.
Prioridad	Alta

Tabla RF 13

Identificación	RF14
Nombre	Cargar partida.
Características	El sistema reanudará la partida, situando a los jugadores en la casilla que se encontraban en el tablero y otorgándole el turno al jugador que lo tenía.
Descripción	Permite al usuario reanudar la partida ya empezada que desee.
Prioridad	Alta

Tabla RF 14

Identificación	RF15
Nombre	Borrar la partida.
Características	El sistema borrará la partida de la base de datos del dispositivo móvil.
Descripción	Permite al usuario borrar la partida que se desee.
Prioridad	Alta

Tabla RF 15

Identificación	RF16
Nombre	Guardar partida.
Características	El sistema guardará en la base de datos del terminal, el estado en que se encuentren los jugadores y la partida.
Descripción	Permite al usuario guardar una partida.
Prioridad	Alta

Tabla RF 16

Una vez se han especificado los requisitos de los dos tipos de partida, se va a detallar los requerimientos del desarrollo de las partidas.

Dinámica de las partidas

Identificación	RF17
Nombre	Tirar dado.
Características	El sistema proporcionará un resultado entre 1 y 6 aleatoriamente y lo mostrará gráficamente con la imagen de un dado con el número generado.
Descripción	Permite al usuario lanzar el dado para poder moverse de casilla.
Prioridad	Alta

Tabla RF 17

Identificación	RF18
Nombre	Calcular destinos posibles.
Características	El sistema calculará las posibles casillas destino mediante el resultado de la tirada del dado. Una vez se hayan calculado, mostrara gráficamente en el tablero los posibles destinatarios.
Descripción	Permite al usuario visualizar los posibles destinos a los que podrá mover su ficha.
Prioridad	Alta

Tabla RF 18

Identificación	RF19
Nombre	Mover ficha.
Características	El sistema moverá la ficha del jugador a la casilla que éste haya seleccionado. En caso de que el jugador esté controlado por la aplicación se calculará la casilla óptima donde moverse.
Descripción	Permite al usuario mover la ficha a la casilla elegida.
Prioridad	Alta

Tabla RF 19

Identificación	RF20
Nombre	Generar pregunta.
Características	El sistema formulará la pregunta pertinente al usuario dependiendo de la casilla en la que se encuentre.
Descripción	Formula la pregunta al jugador con el turno.
Prioridad	Alta

Tabla RF 20

Identificación	RF21
Nombre	Procesar respuesta de la pregunta
Características	El sistema procesará la respuesta de la pregunta y le revelará si ha acertado o no al jugador. En caso de error también se le mostrará la opción que era correcta.
Descripción	Procesa la respuesta proporcionada por el usuario.
Prioridad	Alta

Tabla RF 21

Identificación	RF22
Nombre	Generar quesito.
Características	El sistema proporcionará al jugador, el quesito correspondiente en caso de que el jugador acierte la pregunta formulada.
Descripción	Permite al usuario obtener el premio correspondiente a la categoría de la pregunta.
Prioridad	Alta

Tabla RF 22



Identificación	RF23
Nombre	Jugar siguiente turno
Características	El sistema otorgará el turno al siguiente participante en caso de que el jugador con el turno haya fallado. En caso de que este acierte la pregunta o se haya situado en una casilla de volver a tirar, el turno se le volverá a otorgar a él.
Descripción	Continúa con la dinámica de la partida.
Prioridad	Alta

Tabla RF 23

Identificación	RF24
Nombre	Finalizar partida
Características	El sistema finaliza la partida y muestra al jugador ganador cuando este haya conseguido los seis quesitos y acertado la pregunta de la casilla principal.
Descripción	Finaliza la partida y revela al ganador de esta.
Prioridad	Alta

Tabla RF 24

Por último, se van a especificar los requisitos del modo de práctica.

Identificación	RF25
Nombre	Empezar práctica.
Características	El sistema empezará la práctica
Descripción	Permite al usuario empezar la práctica. Este deberá proporcionar el tiempo que quiere que dure y la categoría de las preguntas que quiera responder.
Prioridad	Media

Tabla RF 25

Identificación	RF26
Nombre	Finalizar práctica.
Características	El sistema empezará finalizará la práctica una vez haya terminado el tiempo o bien el usuario la haya cancelado.
Descripción	Termina la sesión de práctica.
Prioridad	Alta

Tabla RF 26

Identificación	RF27
Nombre	Mostrar los resultados de la práctica.
Características	El sistema representará gráficamente los resultados obtenidos por el jugador.
Descripción	Permite al usuario ver los resultados de la práctica realizada.
Prioridad	Media

Tabla RF 27

3.3.2 Requisitos no funcionales

Identificación	RNF01
Nombre	Interfaz del sistema sencilla e intuitiva
Características	El sistema presentará una interfaz sencilla e intuitiva para que al usuario le resulte fácil el manejo del sistema.
Descripción	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad	Alta

Tabla RNF 1

Identificación	RNF02
Nombre	Interfaz adaptativa a los dispositivos móviles
Características	El sistema presentará una interfaz que se adaptará a las distintas resoluciones de los terminales.
Descripción	El sistema debe adaptarse a cualquier tipo de resolución de pantalla.
Prioridad	Alta

Tabla RNF 2

4. Análisis

4.1 Diagrama de clases

En siguiente diagrama de clases se recogen el conjunto de clases que formarán de la aplicación. Este diagrama se ha obtenido analizando las funcionalidades que debe proporcionar el juego durante una partida.

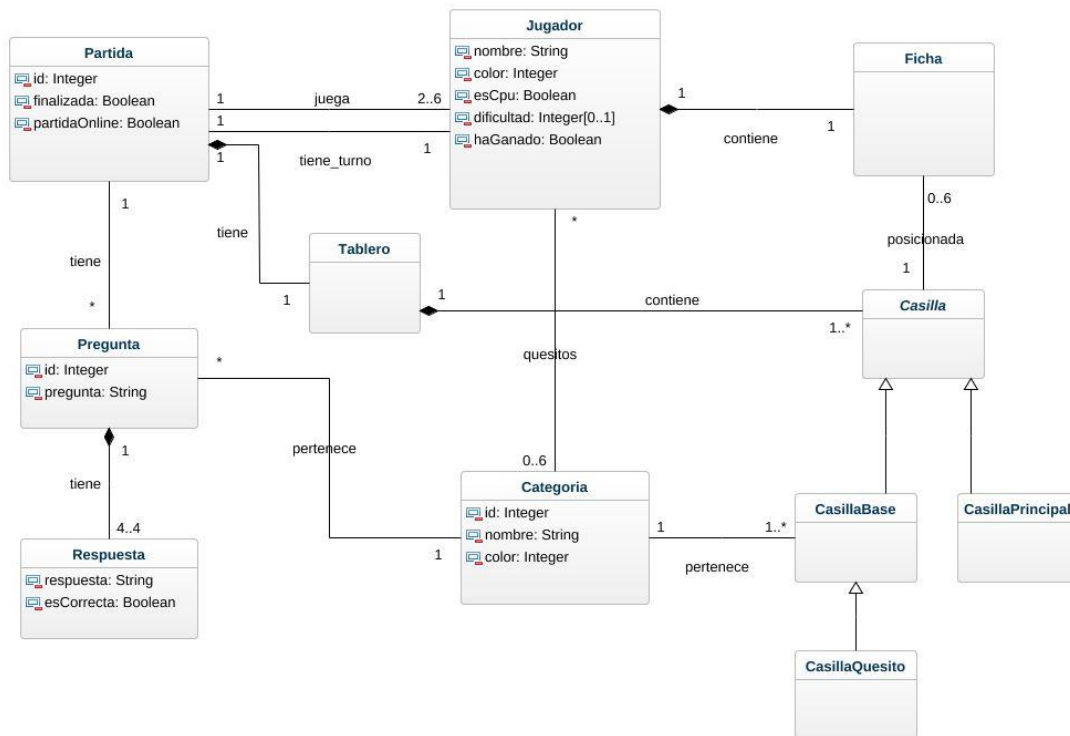


Ilustración 1. Diagrama de clases

Este diagrama de clases encapsula toda la información necesaria para el correcto funcionamiento de las partidas.

La clase partida es una de las clases principales del juego, ya que contiene los jugadores que formarán parte de ella, el tablero donde se desarrollan y las preguntas y respuestas que se formularán. Esta clase también contendrá al jugador con el turno actual, así como información de si la partida ha terminado o si se trata de una partida en línea o no.

Jugador también será una clase importante ya que especifica lo que podrá hacer durante su turno, como por ejemplo mover su ficha por el tablero, indicar si ha ganado la partida o mostrar los quesitos que ha ganado hasta el momento. También indicará si el jugador estará controlado por la aplicación, así como su dificultad en caso de lo esté.

Ficha es una entidad que contiene la posición del jugador en el tablero y también lo representa gráficamente en él.

La clase Tablero contiene las casillas que lo constituyen.

Casilla representa a todas y cada una de las casillas que componen el tablero. Existen tres tipos:

- Casilla principal: solo habrá una por tablero y será la casilla inicial de la partida. Cuando un jugador con todos los quesitos obtenidos se sitúe sobre ella, se formulará una pregunta que en caso de acierto finalizará la partida y proclamara a este como ganador de ella.
- Casilla base: podrá representar a una de las seis categorías principales del juego, así como a la casilla de volver a tirar. En caso de pertenecer a una de estas seis, se le formulará una pregunta al jugador. Si se acierta o la categoría era de volver a tirar, conservará el turno y podrá volver a jugar, en caso contrario cederá el turno al próximo jugador.
- Casilla quesito: solo habrá seis de ellas en el tablero y cada una de estas corresponderá a una de las seis categorías que hay de preguntas. En caso de que el jugador acierte la pregunta formulada y no tuviera anteriormente el quesito correspondiente a esta, se le otorgará.

Pregunta es la entidad que contendrá la pregunta a formular y sus cuatro posibles respuestas. Por otra parte, cada objeto respuesta incluye la posible respuesta de la pregunta y si es la correcta o no.

Por lo que respecta a la clase Categoría, simboliza tanto las siete posibles categorías de que puede ser una casilla base, como una de las seis que puede ser

una casilla queso. Por otra parte, también representarán a cada uno de los seis quesitos que puede ganar un jugador a lo largo de la partida.



5. Diseño

En este capítulo se va a explicar la arquitectura que se ha seguido durante el desarrollo del proyecto, así como el funcionamiento de esta.

Para la arquitectura del juego se ha optado por una formada por tres capas, ya que esto permite desarrollar cada una de estas independientemente la una de la otras, es decir, si se da la necesidad de modificar o añadir algún componente en alguna de estos módulos, no se requerirá la modificación de las otras capas. Las tres capas en cuestión son la de presentación, la de negocio o lógica y la de persistencia.

5.1 Capa de presentación

Esta capa se encarga de interactuar con el usuario y obtener la información que este proporciona. También se encarga de transmitir estos datos a la capa de negocio. Debido a todo esto, este nivel es el que contiene la interfaz gráfica con lo que a continuación se va a explicar los prototipos de las interfaces y las diferentes interacciones que tienen entre ellas.



Menú principal



Ilustración 2. Menú principal

Esta será la primera interfaz que se mostrará al iniciarse la aplicación. Debido a ello, nos servirá de enlace a cada uno de los diferentes modos de juego. Cuando el usuario presione el botón de partida online se le redirigirá al menú de inicio de sesión y en caso de que hubiera iniciado sesión anteriormente se le mostrará directamente el menú partidas en línea. Por lo contrario, si se presiona el botón de modo de práctica se le enseñará el menú de la práctica. Finalmente, si el usuario presionara en multijugador se le aparecería la interfaz de menú multijugador.

Ahora se va a explicar las diferentes interfaces que componen el modo en línea y sus interacciones entre sí.

Menú inicio de sesión

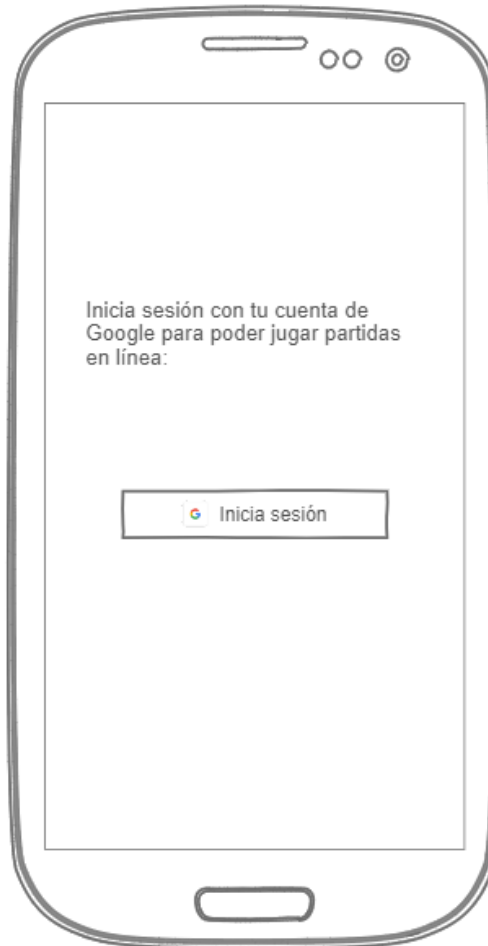


Ilustración 3. Menú de inicio de sesión

Esta será la interfaz que se mostrará al acceder al modo de partidas en línea y el usuario no haya aún iniciado sesión, no tuviera conexión a internet o haya cerrado sesión anteriormente. Una vez el jugador iniciada la sesión, se le redirigirá al menú partidas en línea.

Menú partidas online



Ilustración 4. Menú partidas en línea



Ilustración 5. Ver estado de la partida

Una vez iniciada sesión se mostrará este menú. Ahora bien, los servicios de Google Play Game Services, proporcionan por defecto las interfaces gráficas que muestran las partidas actuales y su estado, así como las posibles invitaciones que haya recibido el jugador. Por ello, cuando el usuario presione el botón de comprobar partidas se le aparecerá la interfaz en cuestión. En caso de que el jugador juegue su turno se le aparecerá la pantalla de disputar la partida. Por el contrario, cuando no sea su turno y presione en ver partida, se le aparecerá la ventana emergente que se muestra en la ilustración 5, en la cual se podrá visualizar el estado de la partida y jugador con el turno actual.

Tanto en el caso que se presione en los botones de nueva partida, ver clasificación o ver logros, se mostrará la interfaz proporcionada por Google Play Game Services correspondiente a cada acción. Por último, si el usuario decide cerrar sesión se le redirigirá al menú de inicio de sesión.

Ahora se va a exponer las diferentes interfaces que conforman el modo partidas en local y sus diferentes interacciones entre sí.

Menú partidas en el dispositivo

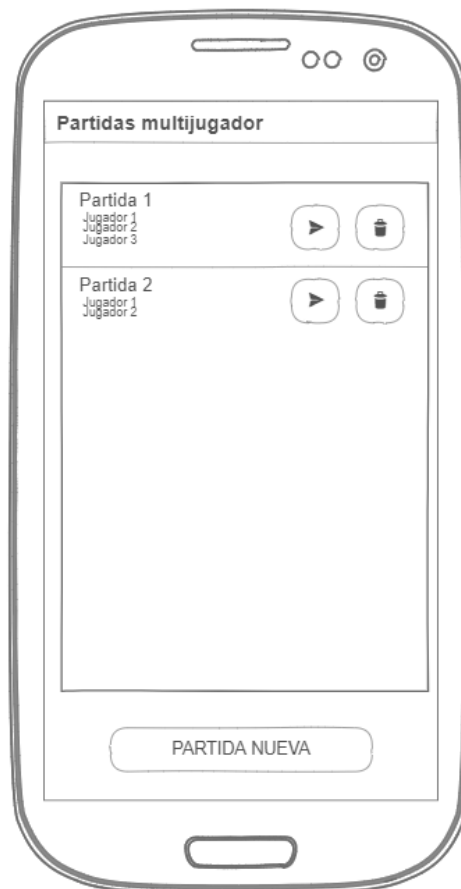


Ilustración 6. Menú multijugador

Este será la interfaz que se le mostrará al usuario cuando acceda el menú de partidas en local. Como se observa, este menú permite visualizar las partidas guardadas, en caso de que las haya. También ofrecerá la posibilidad de borrarlas o reanudarlas. Por último, si se desea crear una nueva partida nos llevará al menú partida nueva.

Menú partida nueva



Ilustración 7. Menú partida nueva

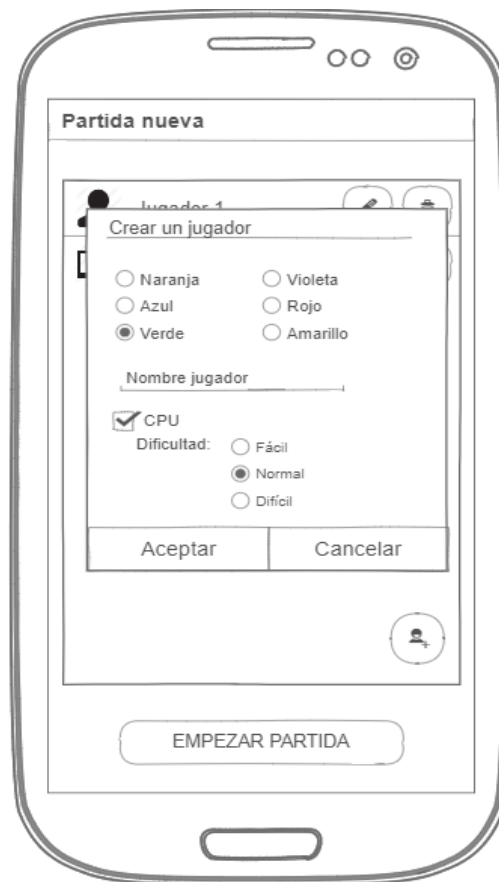


Ilustración 8. Ventana emergente añadir jugador

Esta será la interfaz que aparezca cuando el usuario quiera crear una nueva partida. Se podrá crear, editar, borrar o visualizar a los jugadores que disputarán la partida. Cuando se quiera crear un jugador aparecerá la ventana emergente de la ilustración 8 y en caso de que se edite un jugador también se mostrará esta pantalla, pero con la diferencia que contendrá los datos de este. Por último, cuando se pulse el botón de empezar la partida aparecerá la interfaz de la partida.

Ahora se va a presentar las interfaces que formarán parte del modo de práctica.

Menú de la práctica y resultados de la práctica

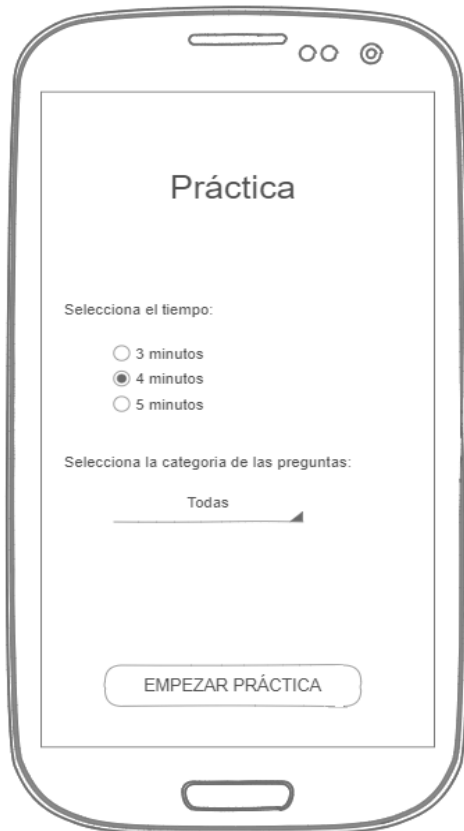


Ilustración 9. Menú de la práctica



Ilustración 10. Resultado de la práctica

Estas serán las dos interfaces, junto con la ventana Contestar pregunta que formarán parte del modo de práctica. En primer lugar, cuándo se acceda a esta modalidad se mostrará el menú de la ilustración 9. Una vez se haya seleccionado los parámetros deseados se mostrará la ventana de contestar preguntas. Por último, cuando se termine de contestar a las preguntas se visualizará el resultado de la práctica como se muestra en la ilustración 10.

Por terminar, se va a explicar las interfaces gráficas que forman parte de la dinámica de las partidas.

Interfaces gráficas del desarrollo de la partida



Ilustración 11. Partida



Ilustración 12. Contestar pregunta



Ilustración 13. Ver puntuación

Cuando se esté disputando la partida se mostrará la ventana mostrada en la imagen 11. En esta se muestra la posición de los jugadores en el tablero, la información del jugador con el turno y los quesitos que tiene. Cuando el jugador tenga que contestar a la pregunta formulada se le aparecerá la interfaz de la ilustración 12. Para terminar, como las partidas pueden albergar hasta seis jugadores, no hay suficiente espacio para poder representar toda esta información. Debido a esto, se ha decidido mostrar esta información en otra ventana diferente como la representada en la ilustración 13.

5.2 Capa de negocio

En esta capa contiene la funcionalidad del juego, así como las clases que definen los modelos de datos necesarios para poder desarrollarla. Para representar las diferentes clases del juego y sus relaciones se ha utilizado el diagrama de clases especificado en el capítulo de análisis.

Cabe destacar que no se ha requerido modificar estructuralmente el diagrama de clases, solo se ha necesitado añadir algunos atributos a diferentes clases.

Para representar la dinámica de las partidas y los procesos que requerirá se ha utilizado un diagrama de transición.

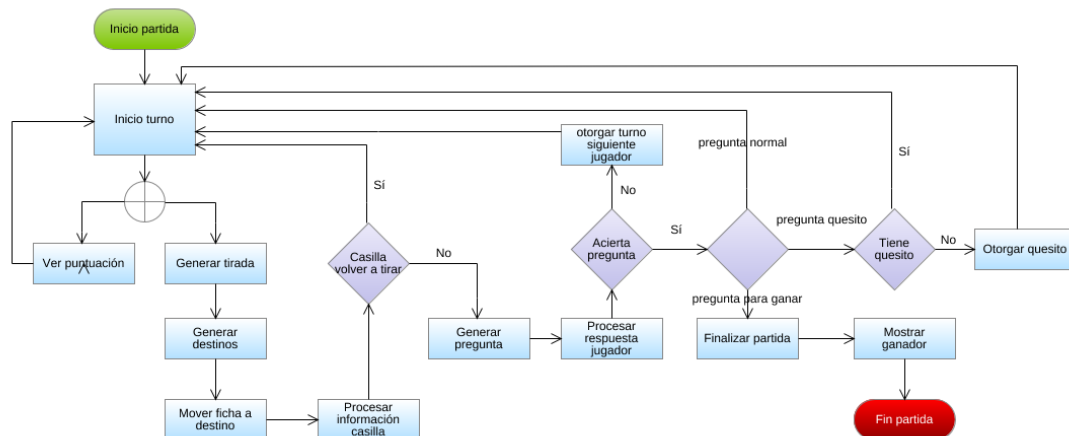


Ilustración 14. Diagrama transición partida

5.3 Capa de persistencia

Esta capa es la responsable de almacenar y proporcionar toda la información que el juego requiere. Para ello se ha decidido utilizar el sistema de base de datos SQLite (7) ya que nos permite implementar una base de datos SQL en el dispositivo móvil sin necesidad de requerir de una conexión a internet.

El juego debe poder almacenar las partidas, los jugadores que la componen, las categorías de las preguntas, las preguntas y las respuestas que se formulan en el juego. Por ello, se ha estructurado las tablas de la base de datos como se muestra en la ilustración 15.

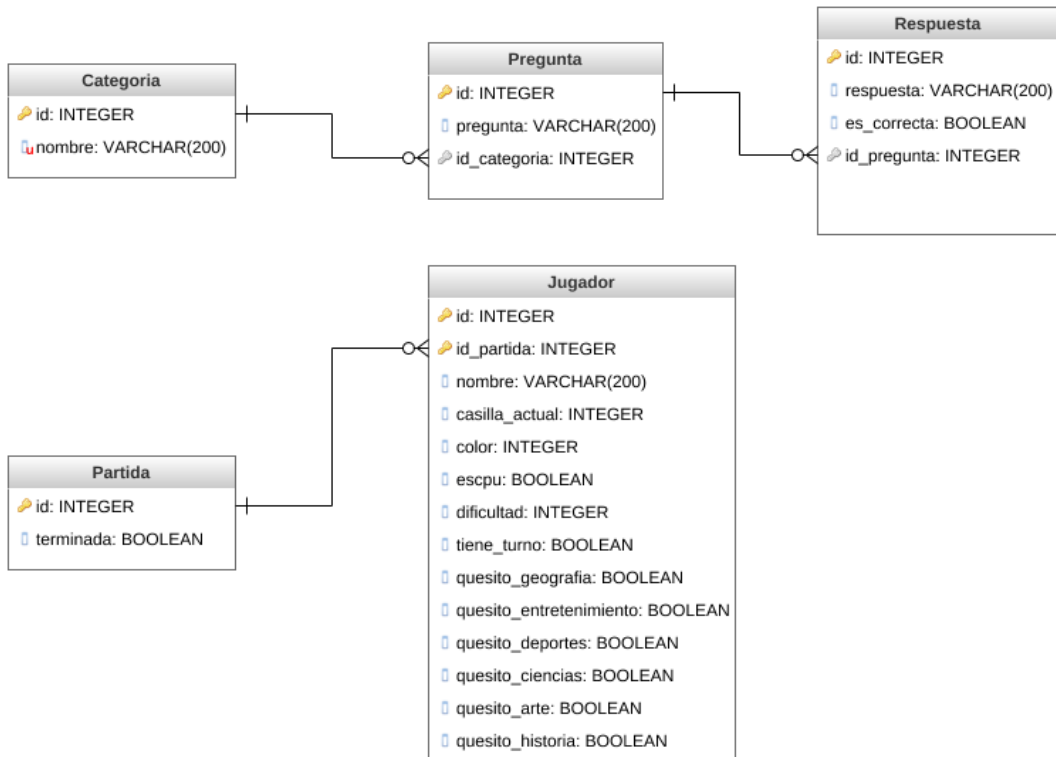


Ilustración 15. Diagrama de la base de datos

Por otro lado, cuando se instale la aplicación se deberá de disponer de una colección de preguntas y respuestas. Para llevar esto a cabo, cuando se cree las tablas Pregunta y Respuesta de la base de datos, se realizará una carga masiva de estas mediante la lectura de un archivo CSV.

Otro aspecto que destacar es que no será necesario almacenar en la base de datos los datos de las partidas en línea ya que esta información se guarda en los servidores de Google Play Game Services.

6. Implementación

En este capítulo se va a explicar la etapa de implementación del código de la aplicación siguiendo tanto el diseño y los requisitos especificados en los capítulos anteriores. También se expondrán los lenguajes y tecnologías empleadas para realizar este desarrollo.

6.1 Lenguajes empleados

Para el desarrollo de este juego se ha empleado un total de tres lenguajes.

El primero y el más utilizado en esta implementación ha sido Java el cual es un lenguaje de programación multiplataforma, de propósito general, concurrente y orientado a objetos. También es una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems que más tarde fue adquirida por Oracle (8). También es un lenguaje. Para el desarrollo de aplicaciones Android se pueden utilizar tanto Java como Kotlin, pero debido a que se tiene más conocimiento del uso de Java se ha optado por utilizar este.

El segundo es Extensible Markup Language, conocido comúnmente como XML (9). Este es un lenguaje de marcado de propósito general que fue desarrollado por el World Wide Web Consortium (W3C). En este proyecto se utiliza para el diseño de las interfaces gráficas ya que es el lenguaje que utiliza Android para este propósito.

Por último, SQL (10) (Structured Query Language) es un lenguaje estándar e interactivo de acceso a base de datos relacionales que permite realizar consultas y operaciones sobre este tipo de base de datos.

6.2 Tecnologías utilizadas

6.2.1 Android Studio

Android Studio (11) ha sido el entorno de desarrollo utilizado para la implementación del proyecto ya permite la gestión de todos los aspectos del proyecto, desde las interfaces gráficas de la aplicación, las clases o los recursos de este. También dispone de un emulador de dispositivos Android el cual ha permitido la posibilidad de probar el juego en todo el proceso de desarrollo.

6.2.1 Google Play Game Services

Es una plataforma de videojuegos en dispositivos Android y Web que permite almacenar y sincronizar datos del juego en la nube, implementar sistemas de logros y clasificaciones y gestionar partidas multijugador en línea tanto en tiempo real, como por turnos (6). Para poder acceder a este servicio se requiere de una cuenta de desarrollador de Google, la cual tiene un precio único de 25 dólares.

6.3 Desarrollo del juego

6.3.1 Capa de presentación

En este apartado se va a comentar los aspectos más importantes de la implementación de la capa de presentación.

Gestión de la resolución de los terminales

Como se ha expuesto en la especificación de requisitos, concretamente con el requisito no funcional número dos, las interfaces gráficas del juego deben adaptarse a cualquier resolución que pueda tener el terminal que utilice el juego.

Para cumplir este requisito se ha utilizado el elemento `ConstraintLayout` (12) que permite crear una interfaz receptiva, es decir, que se adapte al tamaño de la pantalla tanto él, como los elementos que contiene.

Representación gráfica de las fichas

La representación del tablero en el juego se hace mediante una imagen que se encuentra contenida dentro de un `ImageView`. Este a su vez pertenece a otro contenedor de tipo `RelativeLayout` el cual se adapta al tamaño de la imagen. Por tanto, para representar la posición de la Ficha en el tablero su clase se extiende de la clase `View`, permitiendo así crear una instancia de una `View` personalizada que se añadirá al contenedor padre `RelativeLayout` para poder representar esta ficha. Cuando se cree el objeto de tipo `Ficha`, se le deberá pasar al método constructor la `Casilla` en la que se encuentra la ficha, el color que representa al jugador, así como la amplitud y la altura del tablero, para que de esta manera esta vista tenga el mismo tamaño este.

Ahora solo falta dibujar el círculo que representara la ficha en el tablero y para ello se sobrescribe el método `onDraw` de la clase `View` como en la ilustración 16 ya que este ejecutará cuando se cree el objeto `Ficha` y dibujará el círculo deseado que representará la ficha.

```
@Override
protected void onDraw(Canvas canvas)
{
    super.onDraw(canvas);

    float x = casilla.getCentro().x;
    float y = casilla.getCentro().y;
    paint.setStrokeWidth((widthView/100)*1.3f);
    paint.setStyle(Paint.Style.STROKE);
    paint.setColor(Color.BLACK);
    canvas.drawCircle(x, y, (widthView/100)*3, paint);
    paint.setStyle(Paint.Style.FILL);
    paint.setColor(getResources().getColor(color));
    canvas.drawCircle(x, y, (widthView/100)*3, paint);
}
```

Ilustración 16. Método `onDraw` del objeto `Ficha`.

Como se puede observar en la imagen anterior, se obtiene las coordenadas `x` e `y` de la posición de la `Casilla` en la cual la ficha se encuentra en el tablero y mediante el método `drawCircle` se dibuja un círculo del color del jugador y de un tamaño proporcional al de la amplitud del tablero en estas coordenadas, para que así este siempre tenga un tamaño acorde con el tablero independientemente de cuál sea su dimensión.

Representación gráfica de los destinos posibles y mover la ficha al seleccionado

Para representar los posibles destinos se utiliza la misma mecánica explicada en el apartado anterior, solo que en este caso el círculo será casi transparente como los de la ilustración 17 para que el jugador pueda observar la casilla en la cual se podrá posicionar.



Ilustración 17. Representación gráfica de los destinos posibles

Una vez dibujados los destinos posibles en el tablero, el usuario seleccionará la casilla deseada. Para comprobar que esta acción se haya hecho en el destino correcto, se hará uso del método `setOnTouchListener` de la `ImageView`. En este `listener` se sobrescribirá el método `onTouch` para que mediante el `MotionEvent` generado se pueda obtener las coordenadas del punto donde el usuario ha tocado el tablero.

Haciendo uso de estas coordenadas, se comprobará si se encuentran dentro de uno de los posibles destinos y en caso afirmativo se devolverá la Casilla en cuestión. Una vez obtenida esta, se procederá a borrar mediante el método `removeView` la representación gráfica de los destinos posibles y se dibujará la ficha del jugador en la casilla seleccionada.

6.3.2 Capa lógica

En este apartado se va a comentar los aspectos más importantes implementados en la capa de negocio.

Cálculo de los posibles destinos

Para calcular los destinos a los que se pueda mover la ficha el jugador después de la tirada de los dados se ha diseñado el método `getDestinos`. Este contiene dos parámetros, el valor del resultado de la tirada y la casilla en la que se encuentra la ficha. Para el correcto funcionamiento del algoritmo se ha requerido de una clase auxiliar llamada `Camino`, la cual está compuesta por la casilla inicial y la casilla destino.

Para explicar las operaciones que realiza este método se va a hacer uso de un ejemplo.

Se supone que la tirada obtenida por el jugador es de valor 2 y la casilla en la que se encuentra es la A. Por tanto, el algoritmo empezará creando un objeto `Camino` inicial con la casilla origen en valor `null` y como destino la A. Seguidamente, se entra en un bucle que realizará tantas iteraciones como sea el valor de la tirada, en este caso 2.

En cada iteración se obtendrán los adyacentes de la casilla destino y se generará nuevos caminos con estos. Por ejemplo, en la primera iteración tenemos el camino inicial (`null`, A) y se obtiene que los adyacentes de A son B y C, por tanto, los caminos generados serán (A, B) y (A, C).

En la segunda pasada del bucle se realizará el mismo proceso que el anterior con los caminos generados en la iteración previa. Así pues, primero se selecciona el camino (A, B) y se obtiene que los adyacentes de B son las casillas A y D adquiriendo así los caminos (B, A) y (B, D). Pero como podemos observar (B, A) vuelve a la casilla A, la cual es el origen del camino predecesor, con lo que este se descarta.

Para el camino (A, C), la casilla destino tiene como adyacentes A y E de manera que siguiendo el proceso anterior obtendremos un nuevo camino (C, E).



Como consecuencia, en esta iteración se han creado los nuevos caminos (B, D) y (C, E) y como el valor de la tirada era de 2 el bucle ha finalizado, obteniendo así que los posibles destinos de la tirada serán las casillas D y E.

Gestión del turno del jugador controlado por la aplicación

Para automatizar la tirada del jugador controlado por la aplicación se ha seguido el siguiente procedimiento.

Cuando este comience su turno se obtendrá automáticamente el valor de la tirada y haciendo uso del método getDestinos se obtendrán los posibles destinos a los que se podrá mover. Ahora bien, para calcular que destino será mejor se realiza las siguientes comprobaciones.

En primer lugar, se comprobará que el jugador tenga todos los quesitos y en caso positivo se calculará si alguno de los posibles destinos sea la casilla central. Si esto se cumple el jugador se moverá a esta casilla, en caso contrario se estimará que destino está más cerca de la central y se moverá la ficha ahí.

En caso de que no tenga todos los quesitos, se comprobará si alguno de los destinos se corresponde con una CasillaQuesito que no haya conseguido aún. Si se cumple esta condición se moverá la ficha a la casilla en cuestión, de lo contrario se calculará que CasillaQuesito que le falte por conseguir esté más cerca de los posibles destinos y se procederá a mover la ficha al destino más cercano a una de queso. Cabe destacar, que siempre que uno de los posibles destinos sea una CasillaBase de vuelve a tirar se moverá a esta.

Una vez la ficha se haya movido a la casilla seleccionada y no se trate de un vuelve a tirar, la aplicación contestará automáticamente la pregunta generada como se describe en el siguiente punto.

Algoritmo de selección de respuesta del jugador controlado por el juego

Para obtener una contestación automática de este jugador se ha diseñado un método decidirá si contestar correctamente o no dependiendo de la dificultad que tenga asignada el jugador.

La elección de esta respuesta está condicionada por el valor que se generará aleatoriamente entre los números 1 y 100.

En primer lugar, si el jugador tiene asignada una dificultad fácil, solo contestará correctamente si el valor generado es mayor de 65, de lo contrario seleccionará una opción errónea.

En segundo lugar, si el jugador tiene una dificultad normal, seleccionará la opción correcta si el valor es mayor que 40.

Por último, si la dificultad es difícil se contestará correctamente si el valor generado es mayor de 30.

Implementación de las partidas en línea mediante Google Play Game Services

En este apartado se va a exponer los puntos más importantes sobre cómo se gestionan las partidas en línea.

Google Play Game Services dispone de dos tipos de partidas en línea, unas en tiempo real y las otras por turnos. Las que se juegan en tiempo real todos los jugadores deben estar conectados al mismo tiempo para poder jugar, en cambio que las que se juegan por turnos, cada jugador podrá disputar su turno cuando le sea posible. Por tanto, se ha optado por utilizar las partidas por turnos (13) ya que uno de los objetivos del proyecto es que las partidas en línea se puedan jugar cuando a los usuarios les sea posible.

El funcionamiento de estas partidas se basa en que cuando un jugador juega su turno y lo finaliza debe pasar la información de la partida a los demás jugadores. Para ello Google Play Game Services proporciona un array de bytes en la cual se podrá guardar el estado de la partida y actualizar los datos en los servidores.



Cuando un jugador seleccione una partida, se comprobará el estado de la partida y del turno. Si la partida se ha cancelado o ha expirado se mostrará un mensaje en la pantalla informando al usuario de ello. En caso de que algún jugador haya ganado y la partida esté finalizada se mostrará una ventana emergente con el resultado de la partida.

Ahora bien, cuando el jugador tenga el turno y seleccione la partida se le aparecerá la pantalla de la partida y lo jugará. Cada vez que conteste a una pregunta independientemente de que falle o acierte se actualizará el estado de la partida para que los otros jugadores puedan ver el progreso. Cuando falle será cuando se le asigne el turno a próximo jugador.

Esta gestión se hace mediante el método `takeTurn` del objeto `TurnBasedMultiplayerClient`. Las funciones que realiza pueden ser dos. En caso de que se quiera actualizar la información de la partida pero que el usuario conserve su turno se le pasará por parámetro el id de la partida, el array de bytes con la información y el id del usuario con el turno.

```
//Guardar tirada turno
TurnBasedMultiplayerClient multiplayerClient = Games.getTurnBasedMultiplayerClient(this, GoogleSignIn.getLastSignedInAccount(this));
multiplayerClient.takeTurn(idPartidaOnline, partida.persist(), idJugadorEnPartidaOnline);
```

Ilustración 18. Actualizar información partida en línea

Por el contrario, si se desea otorgar el turno al siguiente jugador, el único parámetro que deberá cambiar es del id del jugador que en este caso debería ser el del usuario al que se le quiere conceder el turno.

En el caso de que un jugador quiera abandonar la partida durante su turno también deberá actualizar la información antes de abandonarla como se muestra en la siguiente ilustración.

```
mTurnBasedMultiplayerClient.takeTurn(mMatch.getMatchId(), mTurnData.persist(), myParticipantId).addOnSuccessListener(new OnSuccessListener<TurnBasedMatch>() {
    @Override
    public void onSuccess(TurnBasedMatch turnBasedMatch) {
        String nextParticipantId = getNextParticipantId();
        mTurnBasedMultiplayerClient.leaveMatchDuringTurn(mMatch.getMatchId(), nextParticipantId);
    }
});
```

Ilustración 19. Código para abandonar la partida durante el turno

Por último, cuando un jugador gane la partida, también deberá actualizar la información antes de terminarla ya que en caso contrario los demás jugadores no obtendrían esta información.

```
mTurnBasedMultiplayerClient.takeTurn(mMatch.getMatchId(), mTurnData.persist(), myParticipantId).addOnSuccessListener(new OnSuccessListener<TurnBasedMatch>() {  
    @Override  
    public void onSuccess(TurnBasedMatch turnBasedMatch) {  
        mTurnBasedMultiplayerClient.finishMatch(mMatch.getMatchId())  
    }  
});
```

Ilustración 20. Código actualizar partida y finalizarla

Para concluir este apartado se va a explicar cómo se actualiza la clasificación global del total de partidas ganadas por un jugador y el progreso de los logros.

Para actualizar el número de partidas que el usuario ha ganado en total primero se deberá llamar al método `loadCurrentPlayerLeaderboardScore` de la clase `LeaderboardsClient` (14) para obtener la puntuación actual del jugador. Una vez conseguido este valor, se le sumará uno y se actualizará mediante la función `submitScoreImmediate`. En caso de que el jugador nunca hubiera ganado ninguna partida también se haría uso de este método para insertar el nuevo valor.

Por lo que respecta a los logros, se actualizará su progreso mediante la función `increment` de la clase `AchievementsClient` (15). Para ello se le deberá pasar el id del logro a actualizar y el valor que incrementará el progreso. Este método se ejecutará cada vez que se cumpla la condición especificada. Incluso cuando se haya conseguido desbloquear el logro se seguirá llamando esta función si se da la circunstancia, pero el servidor simplemente ignorará esta petición.

6.3.3 Capa de persistencia

En este apartado se va a comentar los aspectos más importantes implementados en la capa de persistencia.

Para la implementación de esta capa se ha hecho uso del patrón *Data Access Object* (DAO) que permite abstraer el acceso a los datos independientemente de la base de datos que se esté utilizando. La ventaja que proporciona este diseño es que en caso de que en un futuro se quiera migrar la base datos a otra diferente solo se requeriría hacer una implementación del DAO sin necesidad de modificar la existente.

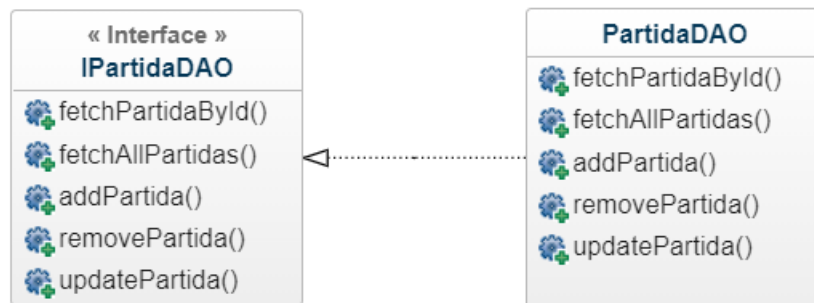


Ilustración 21. Ejemplo patrón DAO

En la imagen anterior se puede observar un ejemplo de lo que sería la implementación de este patrón. Primero se crea una interfaz con los métodos necesarios para acceder o modificar los datos. En segundo lugar, se implementará esta interfaz según la base de datos que se esté utilizando. Por ejemplo, si se estuviera utilizando dos bases de datos distintas, se tendría una implementación de esta interfaz por cada una de ellas.

7. Pruebas

En este apartado se va a explicar las pruebas que se han realizado y los resultados obtenidos de ellas.

Para la realización de estas pruebas se ha dejado probar el juego a un grupo de diez personas de entre veinte y sesenta años. Cabe destacar que todas estas personas son usuarios habituales de dispositivos móviles, por lo que no se trata de gente inexperta en el uso de estos.

Como a lo largo del desarrollo se ha seguido una metodología ágil por iteraciones, las pruebas se han realizado a final de cada una de estas.

En el primer *sprint* se implementó las partidas en local. Aquí algunos usuarios se encontraron con el error de que las fichas no se representaban correctamente en el tablero. Esto se debía que para las coordenadas de las casillas se utilizaba el objeto Point el cual las coordenadas se almacenan en variables de tipo Integer. Por tanto, cuando se realizaban las operaciones para calcular estas coordenadas, como algunos resultados contenían decimales se hacía una aproximación errónea al guardar estos valores en el objeto Point. Se solucionó cambiando el objeto Point por PointF, ya que este utiliza variables de tipo float para almacenar las coordenadas.

En la segunda iteración se realizó el desarrollo de la funcionalidad de las partidas en línea. En este proceso los usuarios que tenían partidas en curso y durante el transcurso de estas se habían cambiado su apodo de la cuenta de Google Games se encontraron con que el juego fallaba. Esto se debía que cuando se pasaba el turno al próximo jugador, este se buscaba en la lista de jugadores mediante su apodo y no mediante su id en la partida. Por tanto, si un jugador se lo había cambiado durante el transcurso de la partida, al llegarle su turno su antiguo apodo no coincidía con ningún jugador y la aplicación sufría un error.

En el tercer *sprint* no se encontró ningún fallo de gran importancia.

Por lo que respecta a la cuarta fase, se programó la funcionalidad de los jugadores controlados por la aplicación. Aquí los usuarios encontraron que cuando era el turno de estos jugadores podían elegir el destino donde mover la ficha o contestar la pregunta. Se solucionó haciendo que cuando el turno era de un jugador controlado por la máquina no se pudiera hacer clic ni el tablero ni en las opciones de las preguntas.

Por último, cuando se implementó el modo de práctica no se encontró ningún error significativo.

8. Conclusiones y futuras mejoras

Con la conclusión de este proyecto se puede afirmar que se ha logrado cumplir todos los objetivos propuestos al inicio de este. Se ha conseguido adaptar exactamente el juego de mesa original a un videojuego para dispositivos Android ya que tanto las partidas en línea como las que se juegan en local, siguen la misma dinámica que las del juego original.

Por otro lado, también se ha logrado el propósito personal de desarrollar un juego para dispositivos móviles. Otro aspecto importante que destacar es que se han puesto en práctica los conocimientos aprendidos en las asignaturas de la carrera, como por ejemplo el uso de una metodología ágil para el desarrollo de un proyecto o el procedimiento de como especificar los requisitos de una aplicación.

Por lo que respecta a las futuras mejoras, se hubiera querido adaptar el juego con las redes sociales para poder fomentar más la competitividad entre los jugadores, ya que los resultados de las partidas y las clasificaciones se podrían publicar en estas.

Otra posible mejora para el futuro es la implementación de otra base de datos mediante Firebase ya que se trata de una base de datos alojada en la nube que permitiría que cada jugador pudiera añadir sus propias preguntas y que otros usuarios pudieran moderarlas. Esto posibilitaría aumentar y actualizar el número de preguntas y respuestas del juego.

Por último, otra mejora que se contempla para un futuro es introducir un número de ventajas limitadas que tendrá cada jugador en las partidas en línea. Por ejemplo, si se falla la pregunta poder contestarla otra vez o bien mostrar solo dos posibles contestaciones de una cuestión. Independientemente de si el jugador hubiera consumido todas las ventajas o no, este lograría más a medida que vaya desbloqueando logros.



9. Bibliografía

1. WePC.com, video game statistics - mobile gaming.
<https://www.wepc.com/news/video-game-statistics/#mobile-gaming>
2. Trivia 360.
<https://play.google.com/store/apps/details?id=smartowlapps.com.quiz360&hl=es>
3. Cuestionados.
<https://play.google.com/store/apps/details?id=com.diablins.android.leagueofquiz&hl=es>
4. Preguntados.
<https://play.google.com/store/apps/details?id=com.etermax.preguntados.lite&hl=es>
5. Especificación de requisitos según el estándar IEEE 830.
<https://www.fdi.ucm.es/profesor/gmendez/docs/is0809/ieee830.pdf>
6. Google Play Game Services. <https://developers.google.com/games/services/>
7. SQLite. <https://www.sqlite.org/index.html>
8. Java. https://java.com/es/download/faq/whatis_java.xml
9. XML. https://developer.mozilla.org/es/docs/Web/XML/Introducción_a_XML
10. SQL. <https://devcode.la/blog/que-es-sql/>
11. Android Studio. <https://developer.android.com/studio>
12. Google Developers, Build a Responsive UI with ConstraintLayout.
<https://developer.android.com/training/constraint-layout>
13. Google Developers, Turn-based Multiplayer Support in Android Games.
<https://developers.google.com/games/services/android/turnbasedMultiplayer>
14. Google Developers, Leaderboards in Android Game.
<https://developers.google.com/games/services/android/leaderboards>
15. Google Developers, Achievements in Android Games.
<https://developers.google.com/games/services/android/achievements>

10. Apéndice

10.1 Monetización del juego

La monetización del juego es un aspecto que se debe contemplar para cuando se lance la aplicación al mercado. En este caso se han contemplado dos opciones.

La primera opción consiste en tener dos versiones del juego, una gratuita con publicidad y otra de pago sin anuncios. La versión gratuita enseñaría anuncios durante la carga de las partidas o bien al final de los turnos de los jugadores, pero no siempre ya que no se pretende bombardear al jugador con publicidad ya que de esta manera se cansarían fácilmente del juego. Por otra parte, si el jugador quisiera eliminar los anuncios permanentemente debería realizar un pago único de aproximadamente de un euro.

La segunda consistiría en que una vez las ventajas mencionadas en las futuras mejoras estuvieran implantadas, el jugador pudiera comprar *packs* de estas para las partidas en línea. No obstante, se limitaría de alguna forma el uso de estas durante una partida para que no hubiera desigualdad entre los usuarios que optan por gastarse dinero y por los que no.



10.2 Gamificación

En este apartado se va a explicar cómo se pretende que la gente no haga uso de la aplicación una sola vez o durante un periodo corto de tiempo.

Una posibilidad sería enviar notificaciones cuando haya pasado un periodo de tiempo sin que el jugador haya jugado su turno en las partidas en línea. Esto evitaría que las partidas se estancaran debido a que los usuarios no recuerdan que tienen partidas activas y les toca jugar.

Otra opción sería fomentar aún más la competitividad entre los jugadores. Por ejemplo, si el jugador desciende bastantes posiciones en la clasificación del total de partidas ganadas, notificárselo para que sea consciente de ello y pueda aumentar otra vez su posición en el *ranking*.

También se podría notificar a cada jugador su posición en la clasificación al finalizar la semana, para así fomentar que quieran escalar posiciones semana tras semana.

Por último, otra posibilidad sería llevando a cabo la futura mejora de que los jugadores puedan añadir sus propias preguntas. Así estas se irían actualizando, evitando que se repitan constantemente y los usuarios caigan en la monotonía de contestar siempre las mismas preguntas.

10.3 Manual de usuario

10.3.1 Menú principal

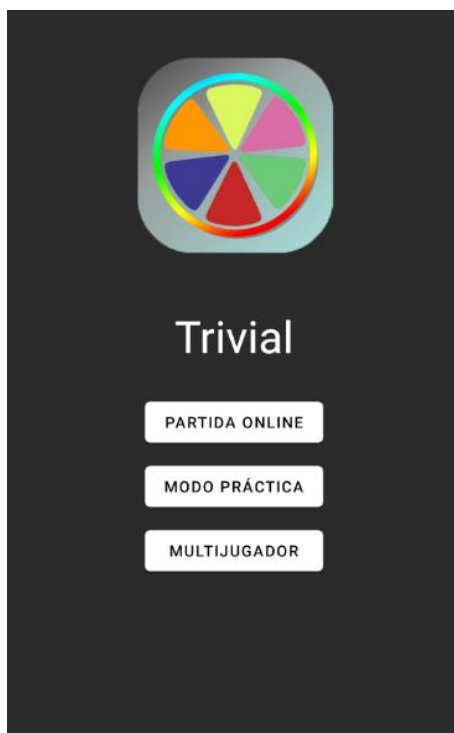


Ilustración 22. Menú principal

Este es el primer menú que aparece cuando los usuarios inician el juego. Desde aquí se podrá acceder a los tres modos de juego que dispone el juego.

10.3.2 Modo partidas en línea

La primera pantalla que se muestra cuando el usuario accede por primera vez a este modo de juego es la de inicio de sesión. También aparecerá cuando se cierre la sesión.

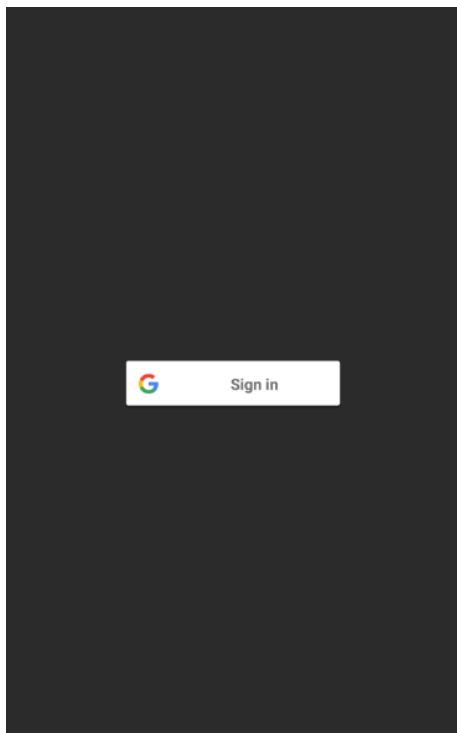


Ilustración 23. Menú inicio de sesión

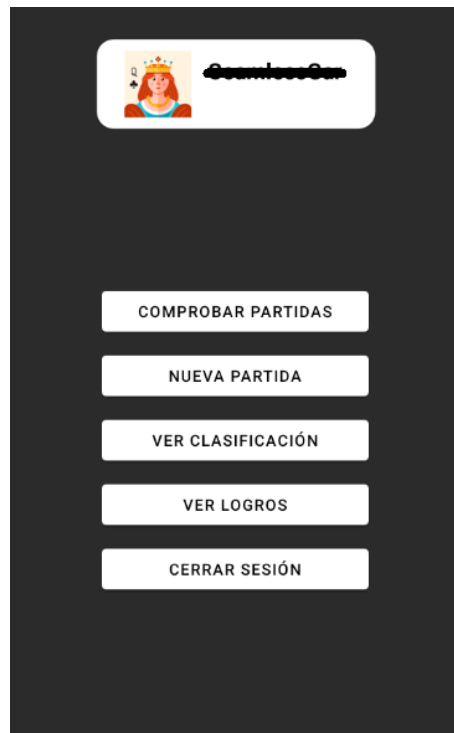


Ilustración 24. Menú modo partidas en línea

Cuando el usuario pulse en el botón para iniciar sesión, se comprobará que tenga instalada en su terminal la aplicación de Play Juegos ya que se requiere de esta para poder acceder a estas funcionalidades. En caso de no tenerla instalada, se le redirigirá a la Play Store donde se lo podrá descargar. En el caso de que sí, aparecerá una ventana emergente con las cuentas de Google disponibles para iniciar sesión.

Una vez se haya seleccionado una cuenta e iniciada la sesión sin problemas, aparecerá un menú como el de la ilustración 24, donde el usuario podrá acceder a todas las funcionalidades de este modo de juego.

Si se quiere ver las invitaciones recibidas, las partidas en curso del jugador o bien las ya completadas se deberá pulsar el botón Comprobar partidas. En este caso aparecerá un listado como el de la ilustración 25.

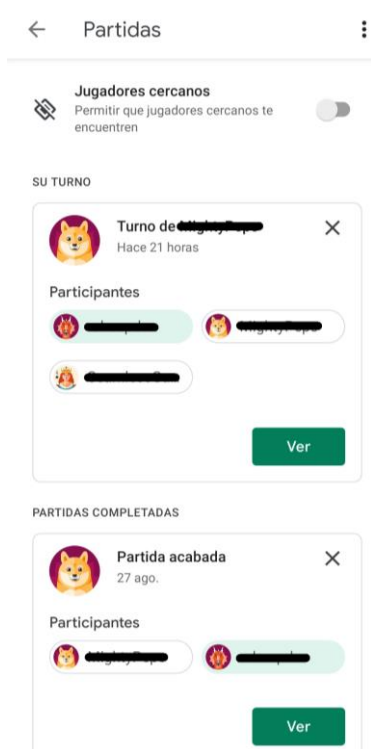


Ilustración 25. Partidas del jugador

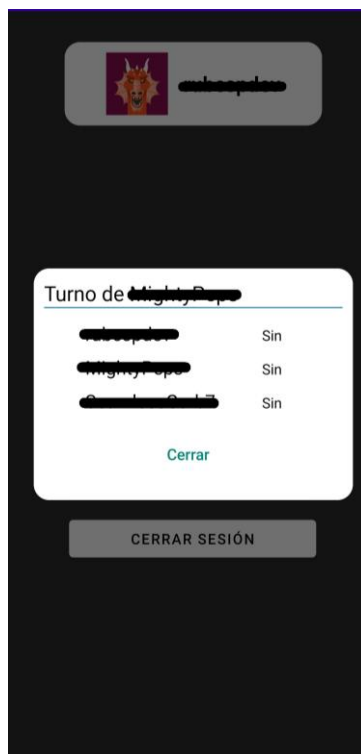


Ilustración 26. Estado partida en curso



Ilustración 27. Estado partida finalizada

En este listado se podrá aceptar o rechazar invitaciones a partidas, comprobar el estado de las partidas tanto en curso como las ya finalizadas o bien jugar en caso de que le toque el turno.

Tanto cuando sea el turno del jugador o se acepte la invitación se le redirigirá a la pantalla del juego principal donde podrá jugar su turno. En caso de no tener el turno se le aparecerá una ventana emergente como en las ilustraciones 26 y 27. En la imagen 26 se muestra la información que aparecerá en una partida que aún se esté jugando, por el contrario, la 27 es un ejemplo de la información mostrada de una partida finalizada.

Por otro lado, cuando el jugador pulse el botón partida nueva del menú principal aparecerá un menú como el de la siguiente imagen.

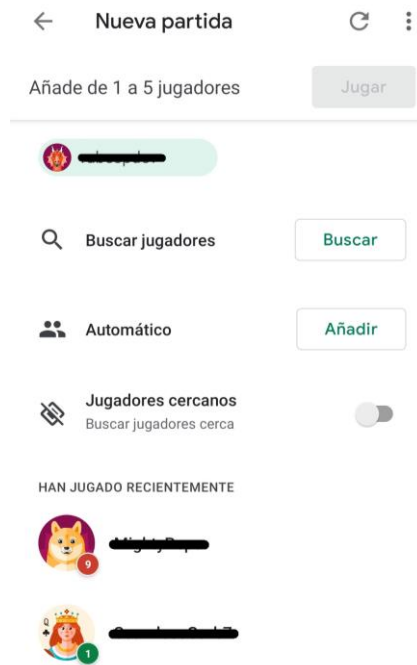


Ilustración 28. Menú partida nueva en línea

Aquí el jugador podrá seleccionar a un máximo de seis jugadores que participarán en la partida. Estos podrán ser o bien jugadores conocidos, o bien jugadores seleccionados de forma aleatoria. Una vez seleccionados, se deberá pulsar en el botón Jugar y el primer jugador empezará la partida.



Ilustración 29. Clasificación jugadores



Ilustración 30. Logros del jugador

Cuando el usuario pulse en ver clasificación podrá observar un listado como el de la ilustración 29, donde aparecen los jugadores que han ganado partidas, el número de partidas ganadas y su posición en el *ranking*. Por otro lado, cuando el usuario pulse en ver logros observará un listado como el de la imagen 30, donde podrá ver que consisten los logros disponibles, el progreso que lleva o bien si ya los ha conseguido.

10.3.3 Modo práctica

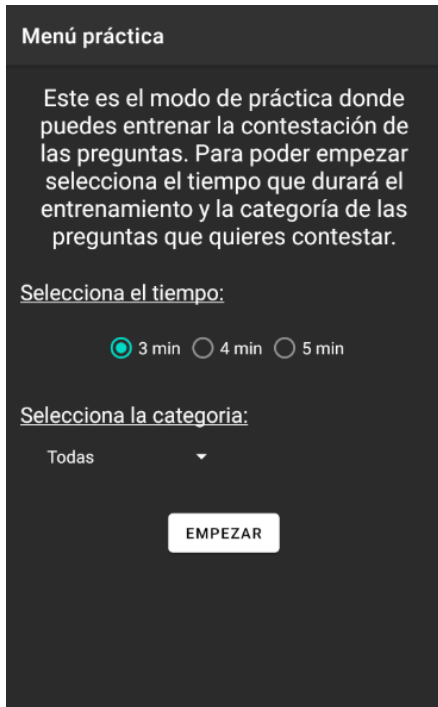


Ilustración 31. Menú configuración de la práctica

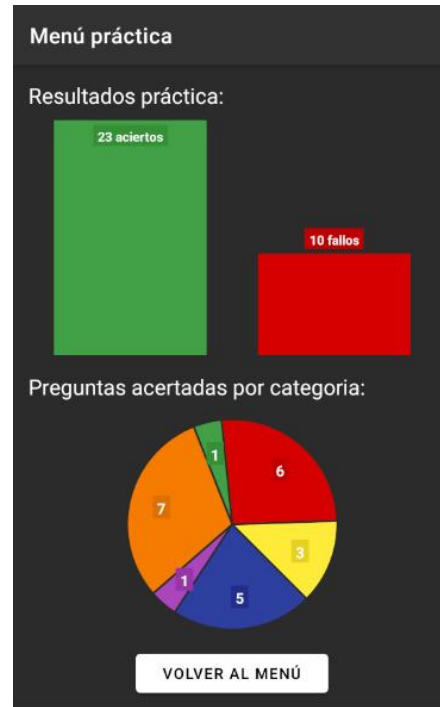


Ilustración 32. Resultado de la práctica

Cuando sea acceda a este modo de juego aparecerá el menú como el de la imagen 31 para configurar la práctica que se quiere realizar. Se podrán seleccionar tanto el tiempo que dure la práctica, como la categoría de las preguntas a las que se quiere responder.

Por otro lado, cuando se finalice la práctica se redirigirá al jugador a una pantalla como la de la ilustración 32, donde podrá ver las estadísticas del resultado del entrenamiento. En caso de que las preguntas a responder pertenezcan únicamente a una categoría, solo aparecerá el gráfico con en número de aciertos y fallos.

10.3.4 Modo partidas en local

Cuando el usuario acceda a este modo de juego se le aparecerá un listado con las partidas guardadas. En caso de que haya partidas guardadas el listado se mostrará como en la imagen 33, de lo contrario aparecerá como el de la ilustración 34.



Ilustración 33. Menú partidas guardadas



Ilustración 34. Menú partidas guardadas vacío

Si se quiere retomar una partida ya empezada, el jugador deberá pulsar en el botón con la flecha. Si la desea borrar, tendrá que seleccionar el botón la papelera.

Por otro lado, si el jugador quiere crear una partida nueva deberá pulsar en el botón partida nueva, el cual le redirigirá a la pantalla de configuración de la nueva partida.

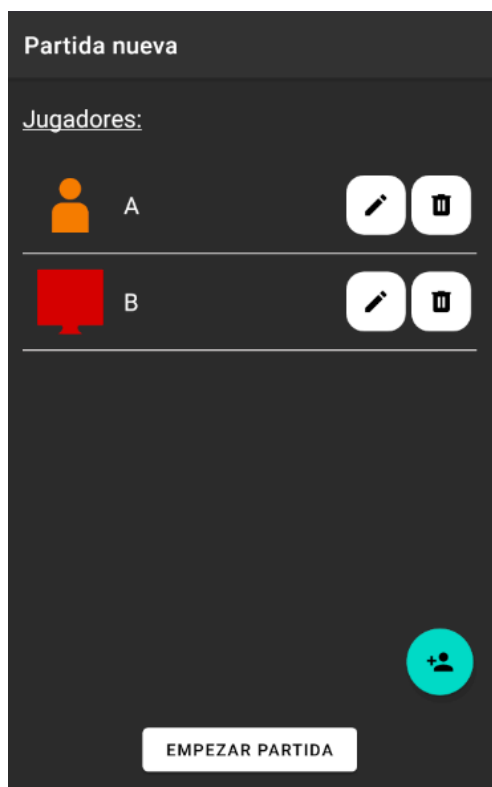


Ilustración 35. Menú configuración partida en local

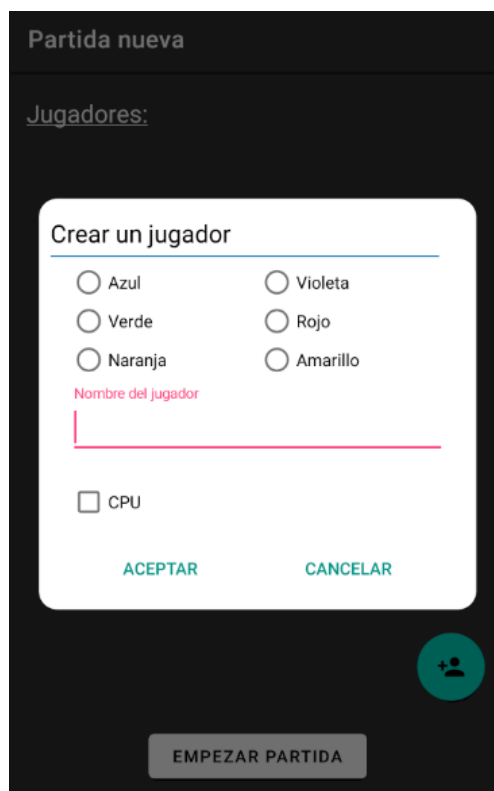


Ilustración 36. Menú añadir jugador

En el menú de configuración de la partida nueva se podrán añadir los nuevos jugadores y empezar la partida. Esta deberá contener un mínimo de dos jugadores para poder empezar, de lo contrario aparecerá un mensaje de error informando de ello.

Cuando se quiera añadir un jugador, aparecerá una ventana emergente como la de la imagen 36, donde se podrá seleccionar el color del jugador, su nombre y si se quiere que sea controlado por la aplicación o no. En caso de que si, también se tendrá que seleccionar su dificultad.

Por otro lado, también se ofrece la posibilidad tanto de modificar un jugador como de borrarlo.

10.3.5 Jugando la partida



Ilustración 37. Pantalla principal



Ilustración 38. Pantalla puntuación de los jugadores

Cuando los jugadores estén jugando la partida observarán una pantalla como la de la ilustración 37, donde se podrá ver la posición de cada jugador en el tablero, el jugador que le toque jugar el turno, los quesitos que tenga este y dos botones, uno para ver la puntuación actual de todos los jugadores y otro para realizar la tirada.

Si el jugador pulsa en el botón para ver la puntuación le aparecerá un listado como el de la imagen 38, donde podrá ver a todos los jugadores que conforman la partida y los quesitos que tienen. También se podrá distinguir si un jugador está controlado o no por la aplicación, ya que los que lo estén tendrán como imagen del avatar el icono de una pantalla.

Una vez el jugador realice la tirada, aparecerá la animación de un dado en movimiento que finalizará con el resultado obtenido de la tirada. A continuación, le aparecerán sobre el tablero los posibles destinos a los cuales podrá mover su ficha.



Ilustración 39. Animación del dado



Ilustración 40. Representación de los posibles destinos

Cuando el usuario se sitúe sobre una casilla que no sea de volver a tirar, se le formulará una pregunta de la categoría de la casilla en cuestión, con cuatro posibles respuestas. El jugador deberá seleccionar una de las posibles respuestas antes de treinta segundos, ya que pasado este tiempo la pregunta se le considerará como errónea.

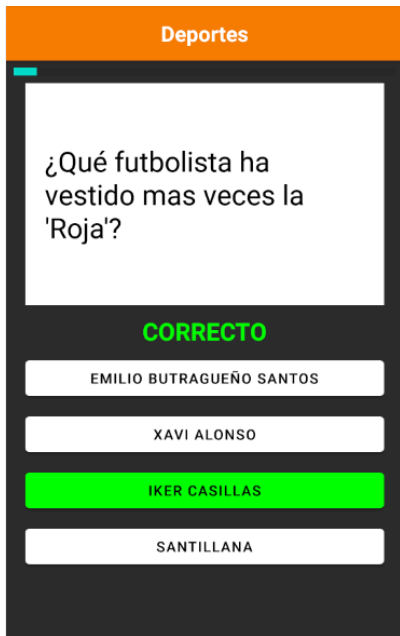


Ilustración 41. Pregunta acertada



Ilustración 42. Pregunta fallada

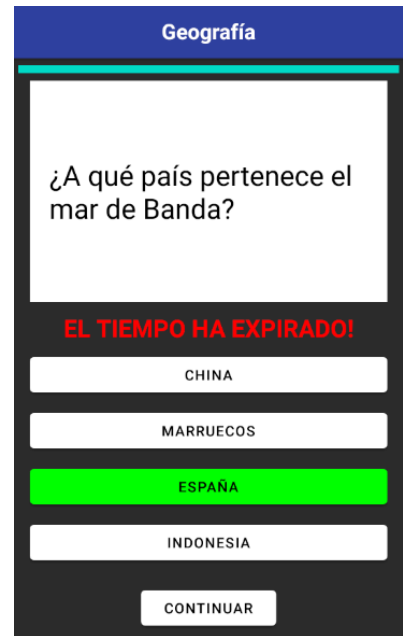


Ilustración 43. Pregunta errónea por expiración del tiempo

En caso de que el jugador hubiera llegado a una casilla quesito y acertado la pregunta, se le otorgaría el quesito de la categoría correspondiente. Por otro lado, si el jugador estuviera en la casilla central con todos los quesitos obtenidos y acertara la pregunta formulada ganaría la partida, de lo contrario la partida seguiría hasta que el u otro jugador lo consiguiera.