



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Grado en Ingeniería Informática

Informatikako Ingeniaritzako gradua

Proyecto fin de grado

Gradu amaierako proiektua



Deusto

Facultad de Ingeniería
Ingeniaritza Fakultatea

Grado en Ingeniería Informática

Informatikako Ingeniaritzako gradua

Proyecto fin de grado

Gradu amaierako proiektua

Resumen

El objetivo de este proyecto es diseñar y programar una aplicación multiplataforma web y móvil para asistir las partidas de un juego de mesa educativo sobre *blockchain*. La aplicación estará pensada para ser utilizada en dispositivos móviles y tablets que tengan sistemas operativos Android ya que una vez la aplicación esté terminada se subirá a Google Play donde se podrá descargar de manera gratuita. La aplicación web se desplegará en un servidor web de la Universidad de Deusto al cual se podrá acceder independientemente del dispositivo que de disponga.

La finalidad del proyecto consiste en enseñar las bases de una tecnología desarrollada en la última década la cual ha cobrado una gran importancia en el ámbito de las criptomonedas, esta tecnología es *blockchain*. Las bases de dicha tecnología son las que se desean enseñar mediante un juego educativo. El juego en cuestión es un juego de cartas físico llamado Block Miners creado por Pablo Garaizar Sagarminaga, profesor de la Facultad de Ingeniería de la Universidad de Deusto.

La aplicación se desarrollará mediante el entorno de desarrollo Android Studio para la plataforma Android y hará uso de la librería LibGDX, que es capaz de compilar código Java para múltiples plataformas.

Descriptores

- LibGDX
- *Blockchain*
- Android Studio
- Juego educativo
- Java

Índice

1.	Introducción.....	1
1.1	Presentación del documento.....	1
1.2	Presentación del problema.....	1
1.3	Estado del arte	3
1.3.1	Diferenciación de nuestro proyecto.....	7
2.	Objetivos del proyecto	9
2.1	Visión general.....	9
2.2	Definición del proyecto	9
2.2.1	Objetivos y alcance del proyecto	9
2.2.2	Producto final	10
3.	Desarrollo.....	11
3.1	Descripción de la realización.....	11
3.1.1	Metodología de desarrollo.....	11
3.2	Especificación de requisitos.....	12
3.2.1	Stakeholders.....	12
3.2.2	Restricciones obligatorias	12
3.2.3	Requisitos funcionales	13
3.2.4	Requisitos no funcionales	16
3.2.5	Análisis de casos de uso.....	17
3.3	Herramientas	21
3.3.1	Herramientas analizadas.....	21
3.3.2	Herramientas seleccionadas	28
3.4	Especificación del diseño	30

3.4.1	Arquitectura del software	31
3.4.2	Estructura de archivos del proyecto	32
3.4.3	Compilación del código en sus diferentes módulos	36
3.5	Interfaces	36
3.6	Plan de pruebas	50
3.6.1	Pruebas de Unidad	50
3.6.2	Pruebas de Integración	54
3.6.3	Pruebas de Validación.....	54
3.7	Manual de usuario	55
4.	Planificación	65
4.1	Equipo del proyecto	65
4.2	Estructura cronológica del proyecto	66
5.	Presupuesto	71
6.	Conclusiones.....	75
6.1	Resultado obtenidos.....	75
6.2	Valoración académica	75
6.3	Trabajo futuro	76
7.	Bibliografía	77

Figuras

Figura 1: Pantalla de blockchain demo - Anders Brownworth	3
Figura 2: Pantalla de blockchain demo – A visual demo of blockchain technology	4
Figura 3: Robot turtles.....	4
Figura 4: Code Master.....	5
Figura 5:Code Brink	5
Figura 6: Rover Control	6
Figura 7: Robot Repair	6
Figura 8: Figura MOON.....	7
Figura 9: Logo BlockMiners	10
Figura 10: Figura Cascada.....	11
Figura 11: Casos de uso	17
Figura 12: Diagrama de actividad	21
Figura 13: IntelliJ	22
Figura 14: Eclipse.....	22
Figura 15: NetBeans	23
Figura 16: Visual Studio Code	23
Figura 17: SlickEdit	24
Figura 18: Android Studio	24
Figura 19: Portable WebServer.....	25
Figura 20: Apache	25
Figura 21: WampServer	26
Figura 22: XAMPP.....	26

Figura 23: Git	26
Figura 24: GitHub	27
Figura 25: GitHub Desktop	27
Figura 26: Visual Paradigm	27
Figura 27: Team Gantt	28
Figura 28: Entorno de desarrollo seleccionado	28
Figura 29: Servidor local seleccionado	29
Figura 30: Control de versiones seleccionado	29
Figura 31: Almacenamiento de código seleccionado	29
Figura 32: Herramienta de visualización de cambios seleccionada	29
Figura 33: Herramienta para la elaboración de casos de uso seleccionada	30
Figura 34: Herramienta para la elaboración de diagramas de Gantt seleccionada	30
Figura 35: Arquitectura monolítica con envío de datos a servidor	31
Figura 36: Estructura de archivos base	32
Figura 37: Estructura de archivos completa	34
Figura 38: Compilación Android	36
Figura 39: Figura compilación web	36
Figura 40: Interfaz transición	37
Figura 41: Interfaz menú principal	38
Figura 42: Interfaz ajustes	39
Figura 43: Interfaz ayuda	40
Figura 44: Interfaz selección jugadores	41
Figura 45: Código erróneo	42
Figura 46: Bloque 1	42
Figura 47: Bloque 2	42

Figura 48: Bloque 3	42
Figura 49: Minado 1	43
Figura 50: Minado 2	44
Figura 51: Error minado	44
Figura 52: Tablero juego flecha superior activada	45
Figura 53: Tablero juego flecha inferior activada	46
Figura 54: Doble clic dos bloques	47
Figura 55: Resultado doble clic con dos bloques	47
Figura 56: Doble clic tres bloques	47
Figura 57: Resultado doble clic con tres bloques	47
Figura 58: Interfaz robar	48
Figura 59: Fin de juego	49
Figura 60: Confirmación salir	49
Figura 61: Error 0	50
Figura 62: Error 00	51
Figura 63: Error 80	51
Figura 64: Minado sin código	52
Figura 65: Error minado 0	52
Figura 66: Error minado 00	53
Figura 67: Error minado 90	53
Figura 68: Manual icono aplicación	55
Figura 69: Manual interfaz transición	55
Figura 70: Manual interfaz menú principal	56
Figura 71: Manual interfaz ajustes	57
Figura 72: Manual interfaz ayuda	57

Figura 73: Manual interfaz seleccionar cantidad de jugadores	58
Figura 74: Manual interfaz juego	58
Figura 75: Manual interfaz juego al ingresar un bloque	59
Figura 76: Manual interfaz juego al ingresar dos bloques	59
Figura 77: Manual interfaz juego al realizar un minado	60
Figura 78: Manual interfaz error minado	60
Figura 79: Manual interfaz juego al ingresar un código de carta no válido.....	61
Figura 80: Manual interfaz robo de gamecoins	61
Figura 81: Manual interfaz juego flecha superior	62
Figura 82: Manual interfaz juego flecha inferior	62
Figura 83: Manual interfaz juego doble clic en el bloque inferior	62
Figura 84: Manual interfaz juego resultado de realizar doble clic en el bloque inferior	62
Figura 85: Manual interfaz finalización partida	63
Figura 86: Organigrama	66
Figura 87: Diagrama Gantt	69

Tablas

Tabla 1: Reparto de tareas	68
Tabla 2: Coste /h	71
Tabla 3: Coste laboral	72
Tabla 4: Coste software	73
Tabla 5: Coste hardware	73
Tabla 6: Coste total	74

1. INTRODUCCIÓN

1.1 PRESENTACIÓN DEL DOCUMENTO

El presente documento tiene como objetivo describir el proceso de diseño e implementación seguido para la realización del proyecto de fin de carrera cuyo título es: *“Diseño y desarrollo de una aplicación móvil y web para el soporte de un juego de mesa educativo sobre blockchain”*.

El proyecto ha sido propuesto por el director Pablo Garaizar Sagarminaga y posteriormente realizado por Mikel Gómez Astobieta, alumno de 4º de Ingeniería Informática.

El presente documento se divide en los siguientes apartados:

- **Introducción:** En el presente capítulo se realiza una introducción a la memoria del proyecto exponiendo la presentación del problema y el arte actual.
- **Objetivos del proyecto:** Capítulo en el que se detalla una descripción concreta de los objetivos del proyecto y las condiciones de ejecución necesarias.
- **Desarrollo:** En este capítulo se realiza un informe detallado de las funcionalidades que el proyecto debe ofrecer al usuario. En dichas funcionalidades se distinguen los requisitos funcionales y requisitos no funcionales, las tecnologías y herramientas usadas, la arquitectura empleada y el impacto en el resultado final del proyecto.
- **Planificación:** Capítulo en el que se refleja el equipo del proyecto junto con la estructura cronológica de este.
- **Presupuesto:** En este capítulo se detalla el coste total del proyecto.
- **Conclusiones:** Capítulo en el que se detallan las conclusiones a las que se han llegado tras el proceso de diseño e implementación de este proyecto.

1.2 PRESENTACIÓN DEL PROBLEMA

En la última década se ha producido un inmenso desarrollo tecnológico que ha conllevado a una revolución tecnológica presente en todos los ámbitos tanto laborales como cotidianos. Consecuencia de dichas mejoras tecnológicas se creó *blockchain*. *Blockchain* es una base de datos que almacena información agrupada en bloques que forman una cadena, la cual posee múltiples mecanismos de seguridad que impiden modificar la información de los bloques añadidos previamente, logrando así que la información no sea alterada. *Blockchain* es comparado generalmente con un libro de contabilidad, pero con la particularidad de que gracias los mecanismos de seguridad y protección contra manipulaciones y falsificaciones consigue que modificar cualquier registro previamente realizado sea prácticamente imposible.

Para entender correctamente la idea del desarrollo e implementación realizado en este proyecto, se debe entender el funcionamiento de *blockchain*. Continuando con la comparación con el libro de contabilidad mencionado previamente, a cada bloque de datos *blockchain* se le asigna un número de bloque, que equivale al número de página correspondiente a dicho libro; los datos se

deben almacenar en el libro, conocidos como *nonce*, junto con el resumen de los datos del bloque previo. Esto último es fundamental ya que es el parámetro que relaciona los bloques y asegura la integridad. El *nonce* es un dato muy valioso ya que *blockchain* posee una gran variedad de medidas de seguridad entre las que se incluye que para que valide un bloque, previamente dicho bloque debe ser minado.

Minar un bloque consiste en averiguar un *nonce* que, al ser sumado con el resto de los datos, el resultado de dicha suma se obtenga una cantidad de ceros contiguos. Si el minado del bloque es correcto, el siguiente bloque incluirá el siguiente número de bloque con sus datos, el resumen del bloque recién minado y un nuevo *nonce* que se volverá a averiguar.

Estos mecanismos necesarios para minar un bloque nuevo certifican que no únicamente se garantiza la integridad de la información, sino también los datos del bloque anterior. Al aplicar las medidas previamente mencionadas, sería fácilmente detectable cualquier modificación de la información almacenada en una cadena de bloques.

Las cadenas de bloques nos proporcionan la fiabilidad al saber que la información que almacenan no se ha visto comprometida y en un supuesto caso de que se viera comprometida, seríamos capaces de notarlo rápidamente y actuar al respecto. Esta tecnología se está empezando a usar en ámbitos diversos tales como atestados policiales al almacenar la información de dichos incidentes.

Con el avance de la tecnología, cualquier persona puede poseer un “monedero digital” en el cual poder recibir y enviar criptomonedas a otras cuentas. Todo esto al realizarse de una manera descentralizada dificulta saber la identidad de la persona que realmente se oculta tras ese “monedero digital”.

Es por ello que, mediante el juego, para el cual se ha diseñado y desarrollado este proyecto, llamado “Block Miners” se quiere enseñar el funcionamiento de *blockchain* con el uso de criptomonedas. Por ello, se ha inventado una moneda ficticia a la cual se le ha llamado “gamecoin”.

1.3 ESTADO DEL ARTE

Esta sección contiene el análisis de algunas de las aplicaciones web cuya finalidad u objetivo se asemeja bastante al objetivo de este proyecto.

- **Blockchain Demo – Anders Brownworth:** Es un simulador web cuyo objetivo es enseñar el funcionamiento de diversas tecnologías tales como: “Hash”, “Bloque”, “Blockchain”, “Distribuido”, “Tokens” y “Coinbase”. El simulador de *blockchain* muestra cinco bloques los cuales cada bloque dispone de un número de bloque, un *nonce*, los datos a almacenar en dicho bloque junto con el *Hash* del bloque anterior y el *Hash* correspondiente al propio bloque. Al modificar el parámetro del número del bloque o los datos correspondientes al bloque, el *Hash* del bloque se convierte en otro diferente y no concuerda con el que debería ser realmente.[1]

Blockchain

The image shows two side-by-side forms for a blockchain demo. Each form represents a block and contains the following fields:

- Bloque:** A dropdown menu with a '#' symbol and a value (1 for the first block, 2 for the second).
- Nonce:** A text input field containing a numeric value (11316 for block 1, 35230 for block 2).
- Datos:** A large empty text area for entering data.
- Anterior:** A text input field showing the previous block's hash (a long string of zeros for block 1, and a specific hash for block 2).
- Hash:** A text input field showing the current block's hash (calculated based on the previous hash and the block's data).
- Minar:** A blue button at the bottom of each form.

Figura 1: Pantalla de blockchain demo - Anders Brownworth

- **Blockchain Demo – A visual demo of blockchain technology:** Esta web también es un simulador para tratar de entender el funcionamiento de la tecnología *blockchain*. No obstante, la forma de representarlo varía ligeramente del caso previo además del uso de terminología diferente, la cual se detallará a continuación. Primeramente, se representan los datos que se van a almacenar dentro del bloque, seguidamente se muestra el valor correspondiente al *Hash* anterior, posteriormente se visualiza el *Hash* asignado al bloque en cuestión, debajo del *Hash* asignado al bloque en cuestión se muestra el cardinal asignado a dicho bloque, finalmente a la derecha está representado en un rectángulo la cantidad numérica que se le ha asignado al *nonce* del bloque. En el primer bloque se ha utilizado el término “GENESIS BLOCK”, dicho término corresponde al primer bloque de cada *blockchain*. [2]

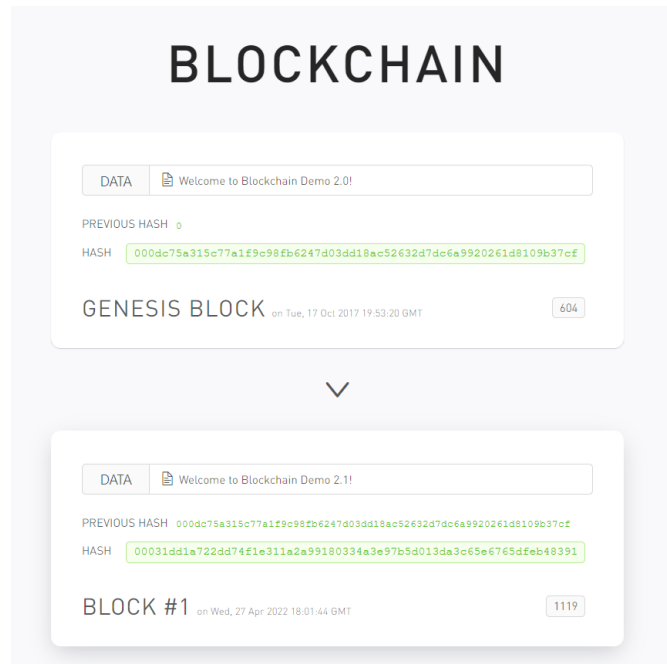


Figura 2: Pantalla de blockchain demo – A visual demo of blockchain technology

Otros juegos educativos cuya materia de aprendizaje no coincide con *blockchain* pero se han tenido en cuenta para la posterior idea son los siguientes:

- **Robot turtles** - Es un juego de mesa desarrollado por un informático llamado Dan Saphiro, con el propósito de enseñar los rudimentos más básicos de la programación informática de una manera más divertida a sus hijos. El juego consiste en programar los movimientos de unas tortugas para que recojan las gemas repartidas en el tablero y evitando los obstáculos surgidos.[3]



Figura 3: Robot turtles

- **Code Master** - Es un juego de mesa en el que se plantean retos de complejidad creciente en el cual el jugador debe ayudar al aventurero con el rescate de tesoros ubicados en las diferentes islas mediante caminos de diferentes colores y evitando los peligros que puedan surgir. [4]



Figura 4: Code Master

- **Code: On the Brink** - Es el primero de una serie de tres juegos de mesa educativos. Este juego está relacionado con la programación de las acciones que debe realizar el robot para lograr su objetivo. La dificultad aumenta a medida que se van completando los retos previos.[5]



Figura 5: Code Brink

- **Code: Rover Control** - Es el segundo de la serie de juegos de mesa educativos mencionado previamente. Este juego está relacionado con la lógica binaria de las

acciones que debe realizar el robot para cumplir su objetivo. Al igual que el primer juego de la serie, este también sigue con el aumento de la dificultad a medida que se completan los retos previos.[6]



Figura 6: Rover Control

- **Code: Robot Repair** - Es el tercero y último de la serie de juegos de mesa educativos. Este juego se relaciona con la interpretación de código. Al igual que en los dos juegos de la serie, este también cuenta con la característica del aumento de la dificultad a medida que se resuelven los retos previos.[7]



Figura 7: Robot Repair

- **MOON** - Es un juego de mesa educativo sobre informática diseñado por Pablo Garaizar Sagarminaga, director de este proyecto. Consiste en evitar que los últimos minutos del alunizaje de la nave Eagle en la luna no se conviertan en una catástrofe. Por ello que el jugador deberá tomar el control del sistema de la nave y realizar los cálculos pertinentes para que el aterrizaje sea correcto. El juego permite al jugador aprender a realizar cálculos binarios, operaciones lógicas y matemáticas entre muchas otras.[8]



Figura 8: Figura MOON

1.3.1 Diferenciación de nuestro proyecto

La mayoría de las aplicaciones existentes de *blockchain* disponibles en Google Play y en la web tienen un carácter económico, es decir, consisten en obtener una cantidad de puntos los cuales posteriormente se ingresarán en un “monedero digital” especificado por el usuario. La aplicación diseñada y desarrollada en este proyecto tiene como objetivo enseñar las características de una tecnología la cual puede ser muy importante en el futuro; es por ello, que en la aplicación se pueden observar cómo se realizan las transacciones con “gamecoins” que es la moneda ficticia utilizada en el juego.

Mediante este proyecto se hace uso del pensamiento computacional cuya definición es la siguiente: *“se define como el proceso por el cual un individuo, a través de habilidades propias de la computación y del pensamiento crítico, del pensamiento lateral y otros más, logra hacerle frente a problemas de distinta índole”*. [9]

El juego de cartas físico para el cual se ha diseñado y desarrollado este proyecto, recalcar una vez más que la aplicación es un complemento y su uso no es obligatorio, utiliza el concepto de pensamiento computacional desenchufado (*computational thinking unplugged*) también llamado Informática desenchufada. Este concepto se define de la siguiente manera: *“hace referencia a un conjunto de actividades que se elaboran para fomentar en los niños habilidades que pueden*

ser evocadas después, para favorecer el pensamiento computacional” [10]. Como bien indica su nombre, este tipo de actividades se realizan sin el uso de dispositivos móviles, sin ordenadores, etc. Este tipo de actividades se suelen realizar mediante el uso de fichas, cartulinas, juegos de mesa, juguetes mecánicos, etc. La finalidad de este pensamiento, al igual que el juego de cartas “Block Miners”, está pensado y diseñado para ser incluidas en las primeras fases del desarrollo cognitivo de una persona; no obstante, al ser un juego educativo puede ser jugado por cualquier persona independientemente de su edad ya que el conocimiento se puede adquirir en cualquier etapa de la vida.

Otra gran diferencia que representa un incentivo es que, tanto en el juego físico como en sus versiones para móvil y web, para jugar una partida debe realizarse con dos a cuatro jugadores fomentando así la relación entre personas. Esto es un punto muy destacable en un momento en que los avances tecnológicos han conllevado a que las nuevas generaciones se relacionen más mediante dispositivos electrónicos que físicamente.

2. OBJETIVOS DEL PROYECTO

2.1 VISIÓN GENERAL

A continuación, se ofrece una visión general de los diferentes apartados que se van a desarrollar en el presente capítulo.

- **Definición del proyecto:** Se especifican los objetivos del proyecto, donde se plasman los aspectos funcionales y el alcance de este.
- **Producto final:** Se describe el producto entregado como consecuencia del desarrollo del proyecto.

2.2 DEFINICIÓN DEL PROYECTO

2.2.1 Objetivos y alcance del proyecto

El principal objetivo de este proyecto es asistir en el aprendizaje y en la formación de una tecnología como es *blockchain*.

Debido al desarrollo de las nuevas tecnologías y al auge de las cadenas de bloques, conocido como *blockchain*, ha conllevado a la creación de criptodivisas. Muchas personas se han dedicado a realizar inversiones en las diferentes criptomonedas existentes; no obstante, puede darse la situación que no se hayan percatado del funcionamiento interno en la creación de dicha moneda virtual.

Por ello y debido a la inmensa velocidad con la que suceden los avances y mejoras tecnológicas, en el futuro *blockchain* podría ser utilizado en una cantidad mayor de ámbitos en los cuales podrían verse avances importantes para la sociedad. Mediante el diseño y desarrollo de esta aplicación se desea plasmar el énfasis en la formación de las personas de una manera más práctica e interactiva para el usuario.

La finalidad de este proyecto es ser un complemento al juego de cartas físico cuyo nombre es "Block Miners", la idea consiste en que los usuarios que deseen puedan hacer uso de la aplicación móvil, la cual en un futuro estará disponible en Google Play de manera gratuita, para realizar los cálculos de las transacciones de una manera más rápida y sencilla. La aplicación diseñada y desarrollada mediante este proyecto no solo consiste en lo mencionado previamente, también tiene como objetivo ayudar al usuario en la contabilidad del juego y detectar posibles errores en el transcurso de la partida. La aplicación no estará disponible únicamente para dispositivos con sistema operativo Android, también estará disponible como aplicación web la cual se desplegará en un servidor web de la Universidad de Deusto.



Figura 9: Logo BlockMiners

Por lo tanto, el proyecto debe contemplar el cumplimiento de los requisitos acordados entre el cliente y el equipo de desarrollo, la selección de las tecnologías necesarias para el desarrollo del proyecto, junto con la entrega correspondiente del manual de usuario y la entrega del código fuente de este.

No obstante, quedará fuera del alcance de este proyecto el mantenimiento del servidor en el que se alojará la versión web de la aplicación y el pago de las licencias necesarias.

2.2.2 Producto final

Tras el diseño e implementación del proyecto, se obtendrá una aplicación que se subirá a la tienda de juegos Google Play y su variante para servicio web que se desplegará en uno de los servidores de la Universidad de Deusto.

3. DESARROLLO

3.1 DESCRIPCIÓN DE LA REALIZACIÓN

En el presente capítulo, se recogen los aspectos técnicos que forman parte del proyecto. Por lo tanto, se ha realizado una especificación de todos los requisitos tanto del sistema como los requisitos del diseño. Añadido a lo previamente mencionado, se describen las tecnologías de las que se han hecho uso junto con su respectiva descripción de la solución y plan de pruebas pertinentes.

3.1.1 Metodología de desarrollo

El diseño y desarrollo de este proyecto se ha realizado mediante el uso de la metodología modelo en cascada. La metodología en cascada se ejecuta secuencialmente, por lo tanto, para el comienzo de una nueva etapa debe haberse finalizado la anterior previamente. Ya que las fases definidas en el desarrollo del proyecto se ejecutan secuencialmente, se ha decidido hacer uso del modelo en cascada debido a la similitud con este proyecto. Dichas fases se detallan en [la estructura cronograma del proyecto](#) .[11]

El modelo en cascada es muy riguroso respecto a los problemas que puedan aparecer durante el transcurso del proyecto, incluso provocando retrasos en este. Es por ello que, se ha creado una fase transversal de “Seguimiento” para identificar y solucionar los problemas con la mayor antelación posible.

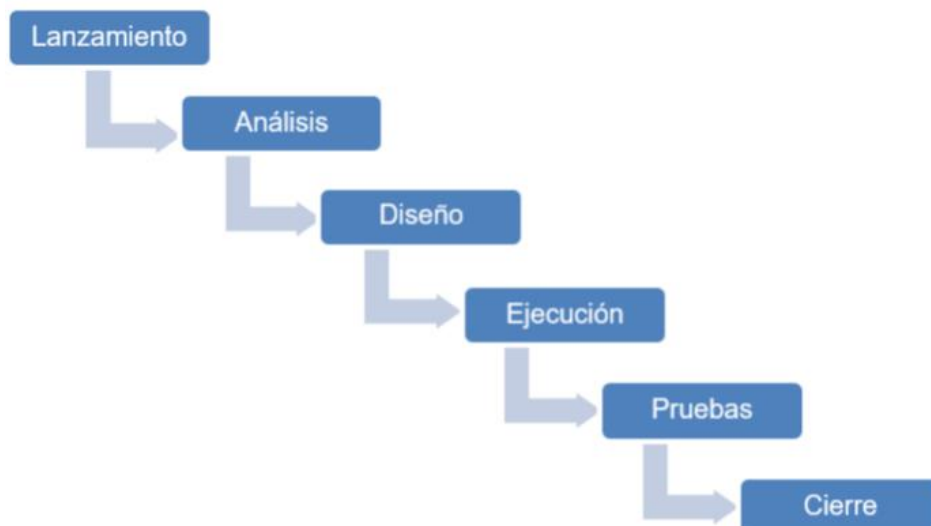


Figura 10: Figura Cascada

3.2 ESPECIFICACIÓN DE REQUISITOS

En este apartado se realiza la especificación de los requisitos que se deben implementar durante el transcurso del proyecto, los cuales deben estar presentes en el producto final.

En este apartado se hace uso del término *stakeholder* junto con las restricciones obligatorias que debe poseer la aplicación.

Del mismo modo, se describen los requisitos del sistema. Dichos requisitos se agrupan en dos:

- Requisitos funcionales: Este tipo de requisitos describen lo que el sistema debe cumplir junto con los servicios que debe prestar el sistema en cuestión.
- Requisitos no funcionales: En estos se describen las restricciones respecto las funciones que son ofrecidas por el sistema.

3.2.1 Stakeholders

Los stakeholders son aquellas personas, organizaciones o empresas las cuales poseen alguna forma de participación interna o externa y se ven afectados el sistema; por lo tanto, obtienen algún beneficio del sistema.

Para el desarrollo e implementación de este proyecto, únicamente se ha dispuesto de un único *stakeholder*.

- Cliente: En este caso el cliente y el director del proyecto es la misma persona, Pablo Garaizar Sagarminaga. Como se ha comentado previamente, el diseño y desarrollo de una aplicación móvil y web es un complemento para el juego de cartas físico creado por Pablo Garaizar Sagarminaga. Por lo tanto, para la realización de este proyecto se cuenta con la base de que tanto la idea como el juego físico ya han sido pensadas y llevadas a cabo previamente. Es por ello por lo que la única persona de la que se van a obtener los requisitos es el director de este proyecto.

3.2.2 Restricciones obligatorias

En este apartado se plasman las restricciones a cumplir durante el transcurso del proyecto.

La restricción más importante está relacionada con el uso de la librería LibGDX [12], la cual posee la peculiaridad de compilar un mismo código de Java para distintas plataformas.

Otra restricción importante está relacionada con la versión mínima de Android necesaria para poder ejecutar la aplicación correctamente. Actualmente los dispositivos móviles con sistema

operativo Android vienen con versiones superiores a once, por lo tanto, la restricción es que la versión mínima soportable sea la séptima.

La última restricción que debe cumplir la aplicación es que debe mandar las puntuaciones a un servidor de la Universidad de Deusto junto con el almacenamiento de todos los datos en el propio dispositivo.

Respecto a las tecnologías de las que se han hecho para la ejecución del proyecto, no se ha recibido ninguna restricción por parte del director del proyecto, que a su vez es el cliente. Consecuentemente, las herramientas han sido seleccionadas por la persona encargada de realizar el diseño y desarrollo del proyecto.

Concluyendo con las restricciones, la persona encargada de supervisar el proyecto al no especificar fecha alguna de entrega, la fecha estará definida por los plazos de entrega establecidos para el Proyecto de Fin de Grado por la Universidad de Deusto.

3.2.3 Requisitos funcionales

A continuación, se describen los requisitos funcionales del proyecto:

- **RF1** - Se debe crear la estructura del proyecto mediante el uso de la librería LibGDX.
- **RF2** - Se deben crear los módulos de Android y Html.
- **RF3** - La pantalla de transición debe mostrar el nombre de la aplicación junto con los nombres del creador del juego y el diseñador gráfico. La pantalla de transición enlazará con la pantalla del menú del juego.
- **RF4** - La pantalla del menú debe contener cuatro botones. El primero debe ser el botón de jugar, el segundo el botón de ajustes, el tercero el botón de ayuda y el cuarto debe ser el botón de salir.

Requisitos funcionales después de jugar

- **RF5** - Al seleccionar el botón de jugar, se abrirá una pantalla la cual solicitará la cantidad de jugadores. En la pantalla en cuestión, habrá tres botones indicando el número dos, tres y cuatro respectivamente; junto con un botón para regresar a la pantalla anterior y otro botón para continuar. La cantidad de jugadores predeterminada será de dos, de lo contrario, se escogerá haciendo clic en la cantidad deseada y posteriormente dándole a continuar. Esto último enlazará con el comienzo del juego.
- **RF6** - La pantalla del juego debe mostrar el marcador de los jugadores. Al ingresar por primera vez en esta pantalla, la puntuación de todos los jugadores debe ser la misma, comenzando todos los jugadores con una cantidad de cincuenta.

- **RF7** - La pantalla del juego debe tener un área en la que mostrar un tablero, dicho tablero estará inicialmente oculto. El tablero mostrará simultáneamente como mucho dos bloques de transacciones.
- **RF8** - La pantalla del juego debe tener dos botones, uno superior y otro inferior para navegar por el tablero. Inicialmente no tendrán funcionalidad alguna. Deben comenzar a tener funcionalidad cuando se dispongan dos bloques de transacciones en el tablero y se ingrese un tercer bloque. Es decir, al introducir tres o más bloques es cuando comenzarán a tener funcionalidad. Si el tablero dispone de dos bloques y se introduce un tercero, se habilitará la flecha superior la cual al hacer clic deberá mostrar el bloque superior en la posición inferior y el bloque anterior a este en la posición superior. La flecha inferior se habilitará si previamente se ha hecho clic en la flecha superior. Una vez pulsada la flecha inferior, el bloque inferior pasará a la posición superior y en la posición inferior debe mostrarse el bloque ocultado mediante la pulsación en la flecha superior previamente. Ambas flechas deben comprobar la posición de los bloques del tablero para deshabilitar la flecha al llegar al límite de bloques tanto superior como inferior.
- **RF9** - Al hacer una doble pulsación en el bloque inferior del tablero, el bloque debe eliminarse. En caso de que existan dos bloques, el inferior se eliminará; no obstante, en caso de existir tres bloques o más y eliminarse el inferior, el superior deberá visualizarse en el bloque inferior y el bloque previo a este último deberá situarse en la posición superior.
- **RF10** - La pantalla del juego debe mostrar la cantidad del *nonce previo* y el *total*. Inicialmente ambas cantidades comenzarán con un valor de cero. El valor representado por el *nonce previo* se modificará tras realizarse un *minado* de carta correctamente. El valor representado por el *total* indica la cantidad del *nonce previo* sumado a la cantidad de los valores de las transacciones realizadas en el tablero.
- **RF11** - La pantalla del juego debe tener un área inferior, suficientemente amplia para contener un teclado numérico. También debe contener tres botones, cuya funcionalidad corresponda a *intro*, *minar* y *robar* respectivamente. Sumado a todo lo anterior, en esta área inferior también debe mostrarse la cantidad numérica introducida mediante el teclado.
- **RF12** - El botón *intro* de la pantalla del juego debe realizar las comprobaciones correspondientes a la cantidad numérica introducida en el teclado. Dependiendo de las comprobaciones realizadas, se introducirá la carta asociada a la cantidad comprobada en el teclado. En caso de que la cantidad introducida genere cualquier tipo de error, el botón debe ser capaz de manejar las excepciones producidas.
- **RF13** - El botón *minar* de la pantalla del juego debe estar deshabilitado hasta que la cantidad de bloques disponibles en el tablero coincida con la cantidad numérica de jugadores escogida en la pantalla previa. Para realizar un *minado*, el *nonce previo*

sumado con las transferencias del tablero más el *nonce* correspondiente de la carta con la que se desea, debe ser equivalente a un múltiplo de diez. El botón de *minado* debe realizar las comprobaciones correspondientes, en caso de que sea correcto se minará el bloque modificándose el *nonce* previo con el *nonce* de la carta con la que se ha procedido a realizar el minado. Al realizar un minado, deben eliminarse del tablero los bloques introducidos previamente.

- **RF14** - En caso de que las comprobaciones realizadas en el botón *minar* de la pantalla del juego sean incorrectas, debe mostrarse una pantalla en la cual se indica que se ha producido un error de minado junto con un botón de continuar. El botón debe poder situarnos en la pantalla del juego con el estado previo. En caso de que el código no sea válido, debe mostrarse un mensaje indicando que se ha producido un error.
- **RF15** - Al hacer clic en el botón *robar* de la pantalla del juego debe mostrarse una pantalla en la que se solicita a qué jugador robar una cantidad equivalente a cinco *gamecoins*. Se deben mostrar seis botones. Cuatro para los jugadores, uno para cada jugador; uno para volver atrás y otro para continuar. Para llevar a cabo el robo, se debe escoger el jugador haciendo clic en el botón y posteriormente haciendo clic en el botón de continuar. De forma predeterminada, "*Argonaut*" deberá ser la opción para robar. Tanto el botón para volver atrás como el de continuar llevan a la pantalla del juego, que una vez se regresa a la pantalla del juego debe preservar el estado en el que estaba previamente. El botón para volver atrás deberá cancelar el robo, mientras que el botón de continuar confirmará esto último.
- **RF16** - La pantalla del juego debe tener un botón con el cual poder salir de la aplicación. Haciendo clic en el botón debe mostrarse una pantalla en la que se pregunte si realmente se desea terminar la partida junto con los jugadores y la puntuación que poseen respectivamente. También debe haber dos botones, uno para volver a la pantalla del juego y otro para continuar y, por ende, terminar la partida. El botón de volver atrás debe situarnos en la pantalla del juego en el estado en el que se encontraba previamente la partida; por otra parte, el botón de continuar nos sitúa en la pantalla del menú principal. Además de situarnos en el menú principal, el botón de continuar debe aumentar la cantidad de partidas jugadas, la cantidad de *gamecoins* minados y realizar un borrado del estado de juego actual junto con el borrado de las puntuaciones actuales.

Requisitos funcionales de ajustes

- **RF17** - Al seleccionar el botón de ajustes, debe abrirse una pantalla de ajustes en la que debe mostrarse el id del usuario, la cantidad de juegos que ha realizado junto con la cantidad de *gamecoins* minados. La creación del id del usuario debe realizarse aleatoriamente mediante caracteres alfanuméricos, cuya longitud sea de siete caracteres. La pantalla de ajustes debe contener un botón cuya funcionalidad consiste en situar al usuario en el menú principal.

Requisitos funcionales de ayuda

- **RF18** - Al seleccionar el botón de ayuda, se abrirá una pantalla en la que debe mostrarse el propósito de la aplicación, un botón para continuar y un enlace el cual al hacer clic se nos abra en un navegador la página web en la cual encontrar información correspondiente al juego. El botón del continuar debe situarnos en el menú principal de la aplicación.

Requisitos funcionales de salir

- **RF19** - Al seleccionar el botón de salir, debe mostrarse una pantalla en la cual se solicite una confirmación para salir de la aplicación. Para responder a dicha pregunta deben existir dos botones, uno para volver atrás y otro para continuar. Al hacer clic en el botón de volver atrás, nos situará en el menú principal. En cambio, al hacer clic en el botón de continuar, nos cerrará la aplicación.

3.2.4 Requisitos no funcionales

A continuación, se procederá a realizar una descripción de los requisitos no funcionales del proyecto.

- **RNF1** - Las interfaces de la aplicación, tanto en su versión móvil como web, deben ser amigables para los usuarios; esto es, las interfaces deben ser intuitivas y fáciles de utilizar.
- **RNF2** - La aplicación, tanto en su versión móvil como web, deben ser responsivos; es decir, deben poder adaptarse y ejecutarse correctamente independientemente de las características del dispositivo.
- **RNF3** - Los errores relacionados a los números insertados en el teclado numérico, deberán mostrarse de manera que se visualice correctamente.
- **RNF4** - La aplicación, tanto en su versión móvil como web, deben estar disponibles para poder ser usadas en todo momento.
- **RNF5** - La aplicación, tanto en su versión móvil y web, deben ofrecer un tiempo de respuesta óptimo para contribuir positiva y satisfactoriamente, transmitiendo así una experiencia de calidad al usuario.
- **RNF6** - La aplicación, tanto en su versión móvil y web, no deberá recopilar ningún tipo de información personal relacionada con el usuario.

- **RNF7** - La aplicación, tanto en su versión móvil y web, debe asegurar que la información recopilada de las partidas sea eliminada en el dado caso que el usuario decida desinstalar la aplicación de su dispositivo.

3.2.5 Análisis de casos de uso

En este apartado, se realiza un análisis de los casos de uso de la aplicación, asimismo se proporcionan situaciones en las que se definen las interacciones entre el sistema y el usuario. Para la elaboración de los diagramas de casos de uso se ha utilizado diagramas UML

Casos de uso

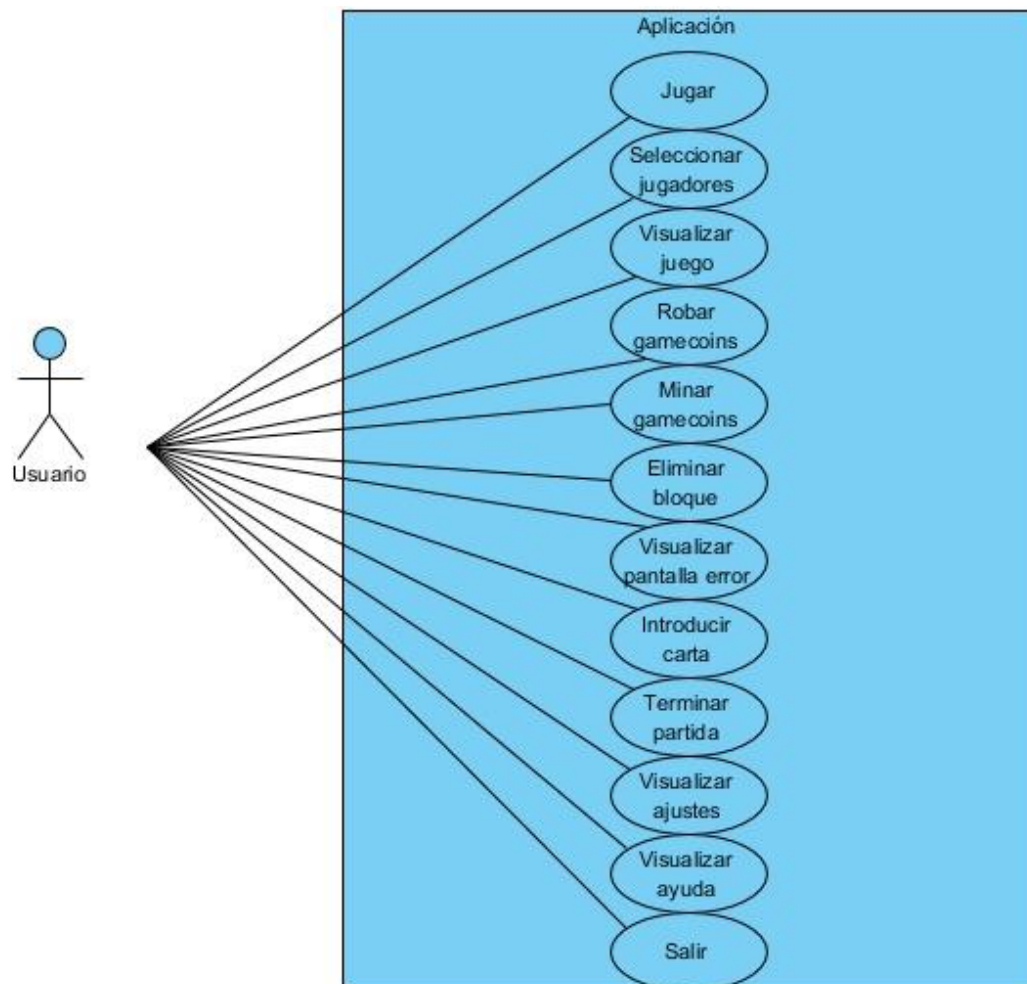


Figura 11: Casos de uso

A continuación, se procederá a detallar los casos de uso:

- **CU1 - Jugar:**

- **Propósito** - Acceder a la pantalla de jugar.
- **Participante** - Usuario.
- **Precondición** - Situarse en el menú principal.
- **Escenario principal y variantes:**
 - Accede a la pantalla para escoger la cantidad de jugadores.
 - Accede a ajustes.
 - Accede a ayuda.
 - Accede a salir.
 - Se acaba CU.
- **CU2 - Seleccionar jugadores:**
 - **Propósito** - Acceder a la pantalla de jugar.
 - **Participante** - Usuario.
 - **Precondición** - Opción *Jugar* del menú principal seleccionada.
 - **Escenario principal y variantes:**
 - Seleccionar la cantidad de jugadores y continuar. Avanza a la pantalla de juego.
 - Regresa al menú principal.
 - Se acaba CU.
- **CU3 - Visualizar juego:**
 - **Propósito** - Jugar.
 - **Participante** - Usuario.
 - **Precondición** - Cantidad de jugadores seleccionados.
 - **Escenario principal y variantes:**
 - Ingresar carta y pulsar enter para añadir un bloque.
 - Ingresar carta y pulsar minar para minar un bloque.
 - Robar para robar gamecoins a otros jugadores.
 - Visualizar las cartas de tablero.
 - Pulsar el botón para salir.
 - Se acaba CU.
- **CU4 - Robar gamecoins:**
 - **Propósito** - Robar moneda ficticia a otro jugador.
 - **Participante** - Usuario.
 - **Precondición** - Haber seleccionado la opción de robar en la pantalla de juego.
 - **Escenario principal y variantes:**
 - Seleccionar el jugador al que robar y continuar. Vuelve a la pantalla de juego.
 - Cancela el robo y vuelve a la pantalla de juego.
 - Se acaba CU.
- **CU5 – Minar gamecoins:**
 - **Propósito** - Minar un bloque para obtener gamecoins y realizar las operaciones de los bloques.
 - **Participante** - Usuario.

- **Precondición** - Haber ingresado la carta por teclado y seleccionado la opción de minar en la pantalla de juego.
- **Escenario principal y variantes:**
 - Se realiza un minado de carta, se eliminan los bloques del tablero y el *nonce previo* pasa a ser el *nonce* la carta con la que se ha realizado el minado.
 - Carta incorrecta introducida, error visible. CU6
 - Carta introducida pero no comprobación errónea de minado. Pantalla error minado.
 - Se acaba CU.
- **CU6 - Eliminar bloque:**
 - **Propósito** - Eliminar el último bloque introducido.
 - **Participante** - Usuario.
 - **Precondición** - Haber ingresado una cantidad de dos o más bloques.
 - **Escenario principal y variantes:**
 - Hacer doble clic en el bloque inferior para eliminarlo.
 - Hacer un clic en el bloque inferior y posteriormente hacer otro clic en cualquier otra parte de la pantalla, cancela la eliminación del bloque inferior.
 - Se acaba CU.
- **CU7 - Visualizar pantalla error:**
 - **Propósito** - Notificar al usuario que la carta con la que ha procedido el minado ha fallado.
 - **Participante** - Usuario.
 - **Precondición** - Haber ingresado la carta por teclado y seleccionado la opción de minar en la pantalla de juego.
 - **Escenario principal y variantes:**
 - Hacer clic en el botón para volver a la pantalla de juego
 - Se acaba CU.
- **CU8 - Introducir carta:**
 - **Propósito** - Añadir un bloque nuevo al tablero.
 - **Participante** - Usuario.
 - **Precondición** - Haber ingresado la carta por teclado y seleccionado la opción de intro en la pantalla de juego.
 - **Escenario principal y variantes:**
 - Carta correcta, se añade al tablero.
 - Carta incorrecta, se visualiza error en pantalla
 - Se acaba CU.
- **CU9 - Terminar partida:**
 - **Propósito** - Terminar la partida.
 - **Participante** - Usuario.
 - **Precondición** - Haber pulsado el botón superior derecha de la pantalla de juego.
 - **Escenario principal y variantes:**

- Botón atrás, cancela el fin de partida y continua con la partida en la pantalla de juego.
 - Botón de continuar, termina la partida aumentando las puntuaciones correspondientes en ajustes e ingresa en el menú principal.
 - Se acaba CU.
- **CU10 - Visualizar ajustes:**
 - **Propósito** - Visualizar las estadísticas de las partidas previas jugadas.
 - **Participante** - Usuario.
 - **Precondición** - Haber pulsado ajustes en el menú principal.
 - **Escenario principal y variantes:**
 - Visualizar las estadísticas y pulsar en continuar para regresar al menú principal.
 - Se acaba CU.
- **CU11 - Visualizar ayuda:**
 - **Propósito** - Notificar al usuario la ayuda necesaria respecto a la aplicación.
 - **Participante** - Usuario.
 - **Precondición** - Haber seleccionado la opción de ayuda en el menú principal
 - **Escenario principal y variantes:**
 - Visualizar la ayuda y pulsar continuar para regresar al menú principal.
 - Se acaba CU.
- **CU12 - Salir:**
 - **Propósito** - Cerrar la aplicación.
 - **Participante** - Usuario.
 - **Precondición** - Haber seleccionado la opción de salir en el menú principal.
 - **Escenario principal y variantes:**
 - Pulsar continuar para confirmar. Se cierra la aplicación.
 - Pulsar cancelar para volver al menú principal.
 - Se acaba CU.

Diagrama de actividad

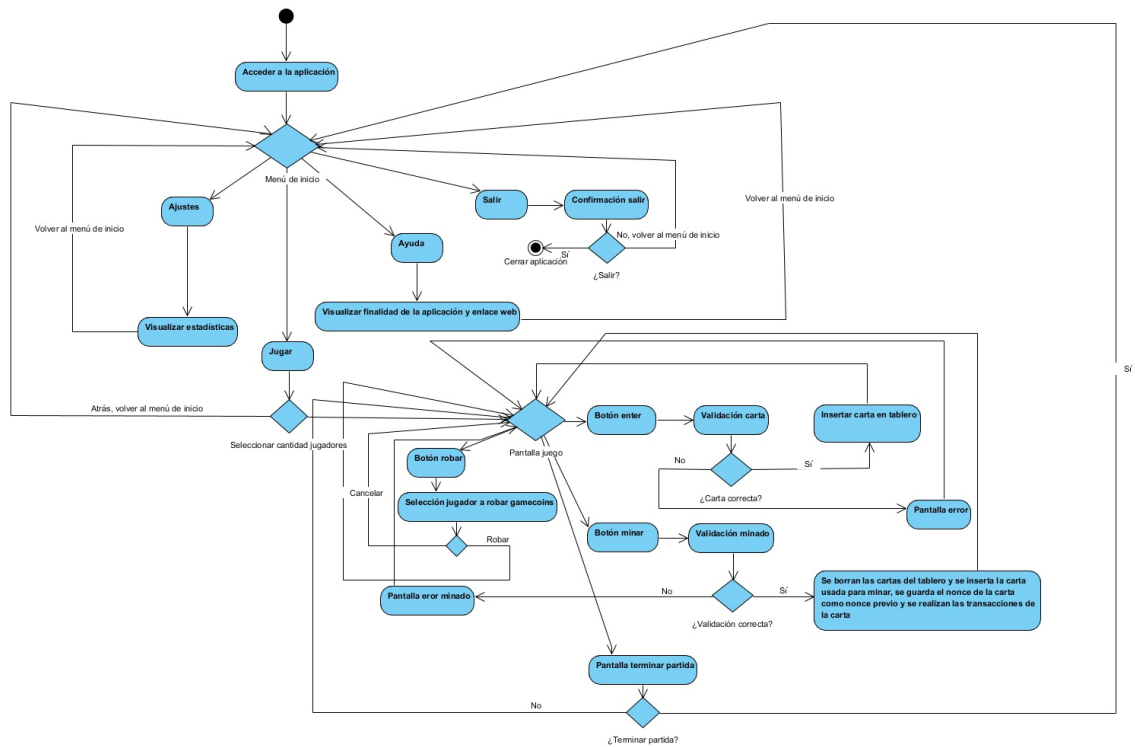


Figura 12: Diagrama de actividad

3.3 HERRAMIENTAS

3.3.1 Herramientas analizadas

En este apartado se van a describir las tecnologías y herramientas analizadas para la posterior elección de las que más se adaptan al diseño y desarrollo de este proyecto.

- IntelliJ IDEA** - Es un entorno de desarrollo integrado, también llamado IDE en sus siglas en inglés, para el desarrollo de programas informáticos. Este entorno es desarrollado por JetBrains. La descarga de este entorno de desarrollo no trae incluido el JDK, siglas de Java Development Kit, que es un software que provee las herramientas necesarias para el desarrollo de programas en Java. Al igual que el JDK, no trae preinstalado el SDK, siglas de Software Development Kit, de Android.[13]

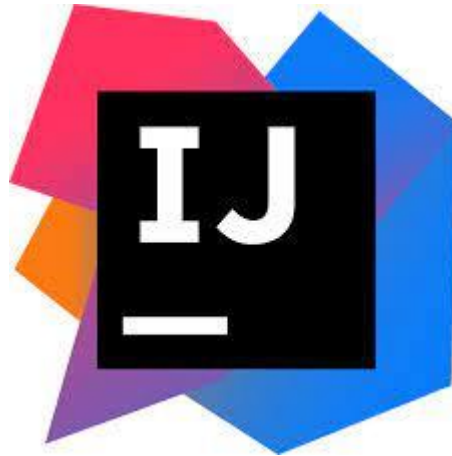


Figura 13: IntelliJ

- **Eclipse** - Es una plataforma de software compuesta por un conjunto de herramientas de programación que proporcionan la posibilidad de crear código multiplataforma y es una herramienta de código abierto. Eclipse, ha sido mayormente usada para desarrollar entornos de desarrollo integrados junto con el JDT, siglas en inglés de Java Development Toolkit. Esta plataforma de desarrollo no trae preinstalado tanto el Java Development Kit y de forma oficial no está soportado el Software Development Kit de Android; no obstante, se puede instalar unos ADT, siglas de Android Development Tools, de versiones antiguas.[13]



Figura 14: Eclipse

- **NetBeans** - Es un entorno de desarrollo integrado de código abierto, el proyecto de desarrollo de este IDE pensado principalmente para el lenguaje de programación Java. Este entorno de desarrollo no trae preinstalado el Java Development Kit y oficialmente no soporta el Software Development Kit de Android. [13]



Figura 15: NetBeans

- **Visual Studio Code** - Es un editor de código fuente desarrollado por Microsoft, el cual incluye soporte para depuración de código, control integrado de Git, resalta los diferentes tipos de sintaxis. De forma predeterminada Visual Studio Code no trae el soporte necesario para Java; no obstante, se pueden realizar instalaciones de una gran variedad de extensiones en la propia aplicación. Lo mismo ocurre con el Software Development Kit de Android, de forma predeterminada no viene instalado, se puede instalar mediante extensiones proporcionas en el propio IDE.[13]



Figura 16: Visual Studio Code

- **SlickEdit** - Anteriormente conocido como Visual SlickEdit, es un entorno de desarrollo integrado que realiza las funciones de editor de código fuente y editor de texto. Este IDE soporta el lenguaje de programación Java.[14]



Figura 17: SlickEdit

- **Android Studio** - Es un entorno de desarrollo integrado oficial para plataformas Android, reemplazando a Eclipse como IDE oficial para el desarrollo de aplicaciones Android. Este entorno de desarrollo está basado en IntelliJ IDEA. Android Studio proporciona de forma predeterminada el Java Development Kit junto, con obviamente, el Software Development Kit de Android ya que es el entorno de desarrollo integrado oficial para plataformas Android. [13]



Figura 18: Android Studio

- **Portable WebServer** - También conocido como PWS, es un servidor web ligero y portable para sistemas operativos Windows. Este servidor destaca por su interfaz minimalista, en la cual haciendo unos clics se puede navegar por la interfaz y sus configuraciones. [15]



Figura 19: Portable WebServer

- **Servidor HTTP Apache** - Es un servidor web HTTP de código abierto que está disponible gratuitamente para diferentes sistemas operativos. Asimismo, es uno de los servidores web más usados.[16]



Figura 20: Apache

- **WAMPServer** - Es una pila de soluciones dirigidas a Windows. Una pila de soluciones consiste en un conjunto de subsistemas necesarios para la creación de una plataforma completa. Se incluye un servidor Apache, un servidor MySQL y un servidor PHP junto con PHPMyAdmin para la gestión de las bases de datos.[15]



Figura 21: WampServer

- **XAMPP** - Es un paquete de software libre que incorpora un servidor Apache, un sistema de gestión de base de datos MySQL y lenguajes como PHP y Perl, junto con la posibilidad de gestión de cuentas FTP y acceso a las bases de datos mediante PHPMyAdmin. Sumando a todo lo previamente mencionado, incluye también un servidor de correo Mercury para realizar envíos de correos, un servidor Tomcat para servlets JSP junto con un servidor FTP FileZilla.[15]



Figura 22: XAMPP

- **Git** - Es un software de control de versiones y tiene como propósito llevar a cabo un registro de los cambios en los archivos de la computadora junto con la coordinación del trabajo realizado por diferentes personas, las cuales pueden haber realizado cambios en el mismo archivo compartido en un repositorio de código.[17]



Figura 23: Git

- **GitHub** - Es una plataforma de desarrollo colaborativo en la cual se alojan proyectos utilizando el sistema de control de versiones Git.[18]



Figura 24: GitHub

- **GitHub Desktop** - Es una aplicación que proporciona la interacción con GitHub utilizando interfaces de usuario en lugar de la línea de comandos o buscador web.[19]



Figura 25: GitHub Desktop

- **Visual Paradigm** - Es una herramienta UML CASE la cual facilita la transformación de los requisitos a diseños precisos. Soporta la representación en diagramas del ciclo de vida completo del proceso de desarrollo de software.[20]



Figura 26: Visual Paradigm

- **Team Gantt** - Es una herramienta online gratuita para la creación de diagramas Gantt.[21]



Figura 27: Team Gantt

3.3.2 Herramientas seleccionadas

En este apartado se va a realizar una descripción del motivo por el cual se han seleccionado dichas herramientas.

Para ello, se va a dividir en tres categorías:

- **Entorno de desarrollo** - A continuación, se va a describir el entorno de desarrollo integrado seleccionado indicando sus motivos. El IDE escogido ha sido Android Studio, ya que al ser el entorno de desarrollo integrado oficial para plataformas Android, proporciona garantías de soporte para las diferentes versiones de Android. La instalación es simple y en la propia instalación se incorpora el Software Development Kit, mientras que en los otros IDEs la instalación no era soportada o había que realizarla de manera independiente. Otro motivo por el cual ha sido seleccionado Android Studio es gracias a la posibilidad de crear dispositivos móviles virtuales, esto proporciona la capacidad de probar la aplicación desarrollada en los dispositivos creados mediante la propia aplicación.



Figura 28: Entorno de desarrollo seleccionado

- **Servidor local** - Esto es importante en la realización de este proyecto ya que el desarrollo al consistir en una aplicación móvil y web, deben realizarse pruebas periódicas para saber si el código desarrollado e implementado produce algún tipo de error o incompatibilidad y solucionarlo antes de la finalización del proyecto. Es por ello que se necesita un servidor en el cual poder alojar el código compilado en su versión html. Se ha decidido optar por XAMPP, el cual posee una interfaz de usuario simple y minimalista. Además, XAMPP, proporciona una gran cantidad de servicios aparte del servidor Apache del cual se va a hacer uso en el transcurso de este proyecto.



Figura 29: Servidor local seleccionado

- **Control de versiones** - El control de versiones es una herramienta imprescindible en la tarea de cualquier persona encargada de realizar labores de programación, ya que permite llevar un registro del código, las modificaciones realizadas y proporciona un método de respaldo en caso de que ocurriera cualquier tipo de catástrofe natural, robo de información, etc. Como herramienta de control de versiones se ha escogido Git, que es el sistema de control de versiones más popular y, por lo tanto, se posee la certeza de que el comportamiento y la fiabilidad que se va a obtener es la esperada. Para almacenar los repositorios de código realizado a lo largo del proyecto se ha decantado por GitHub, que junto a la herramienta de control de versiones Git hacen una combinación idónea. Finalmente se ha decidió hacer uso de la herramienta GitHub Desktop, cuya funcionalidad es bastante similar a la de Git; no obstante, al disponer de una interfaz gráfica GitHub Desktop nos posibilita observar los cambios realizados antes de subirlos al lugar de alojamiento.



Figura 30: Control de versiones seleccionado



Figura 32: Herramienta de visualización de cambios seleccionada



Figura 31: Almacenamiento de código seleccionado

- **Herramienta UML CASE** - El uso de este tipo de herramientas es vital para el desarrollo de un proyecto debido a que la realización de una buena planificación junto con el análisis idóneo conlleva a un resultado exitoso. Es por ello que se ha decidido hacer uso de Visual Paradigm como herramienta para la elaboración de los casos de uso y el diagrama de actividad. La herramienta permite su uso gratuito durante treinta días, no obstante, al disponer de una licencia de estudiante proporcionada por la Universidad de Deusto aumenta el tiempo de uso sin coste añadido.



Figura 33: Herramienta para la elaboración de casos de uso seleccionada

- **Herramienta para el diseño de diagramas de Gantt** - La herramienta escogida para la elaboración del diagrama de Gantt ha sido Team Gantt. Es una herramienta online cuya versión gratuita es suficiente para crear el diagrama necesario de este proyecto.



Figura 34: Herramienta para la elaboración de diagramas de Gantt seleccionada

3.4 ESPECIFICACIÓN DEL DISEÑO

En el presente apartado se recoge las especificaciones del diseño, describiendo detalladamente su arquitectura del software, así como el almacenamiento de los datos.

3.4.1 Arquitectura del software

La arquitectura del software utilizada en la realización de este proyecto está basada en una *arquitectura monolítica* [22] excepto en un proceso en el que se hace uso de un servidor alojado en la Universidad de Deusto.

Una *arquitectura de software monolítica* consiste en agrupar la capa de interfaz de usuario, la capa de lógica de negocio y la capa que proporciona el acceso a los datos en una misma aplicación y sobre la misma plataforma en la que se está ejecutando la aplicación. Al realizar una aplicación mediante esta arquitectura de software, otorga a la aplicación autonomía e independencia respecto a otras aplicaciones que hacen uso de otro tipo de arquitectura de software. Al estar agrupadas la capa de interfaz de usuario, la capa de lógica de negocio y la capa de datos, la aplicación dispone de todos los datos necesarios para su correcto funcionamiento desde el momento de su instalación; por lo tanto, la propia aplicación es responsable no solamente de realizar únicamente una tarea, sino que es capaz de realizar todos los pasos necesarios hasta completar su objetivo.

Monolithic Architecture

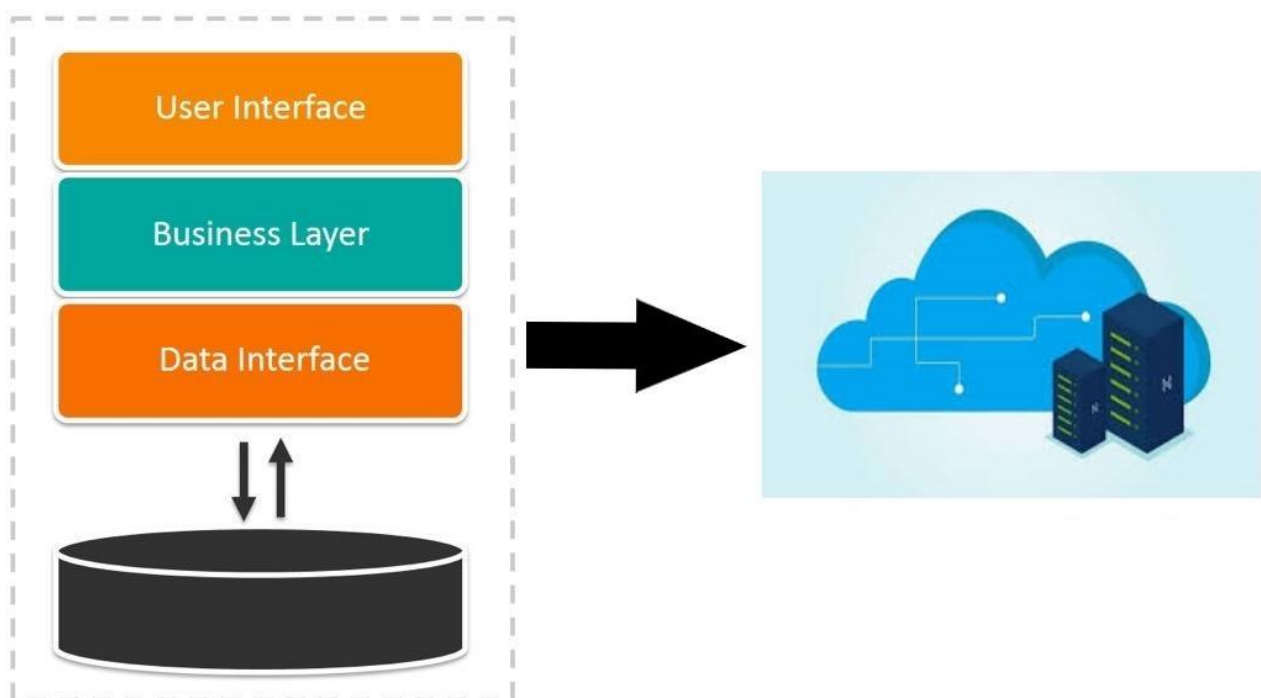


Figura 35: Arquitectura monolítica con envío de datos a servidor

No obstante, la elección de esta arquitectura de software en un primer momento puede ser sorprendente, en una situación tecnológica en la que prácticamente los avances tecnológicos junto con las nuevas arquitecturas software han conseguido conectar a los usuarios con incontables aplicaciones. Al pensar la forma en la que se estructura la arquitectura en cuestión, se observa como únicamente en la capa de datos existe una salida de datos a un servidor de la Universidad de Deusto. La finalidad de dicho envío de datos es para que el director del proyecto

observe y analice los datos en cuestión para posibles futuras modificaciones y que el objetivo del proyecto se cumpla.

3.4.2 Estructura de archivos del proyecto

3.4.2.1 Estructura de archivos base del proyecto

La siguiente figura muestra el esqueleto general que crea la librería LibGDX al generar el proyecto. Previamente se ha indicado que el desarrollo estaba orientado para sistemas operativos Android y para plataformas web; es por ello, que se procederá a continuación a realizar una aclaración de la finalidad relacionada con cada módulo de la imagen.

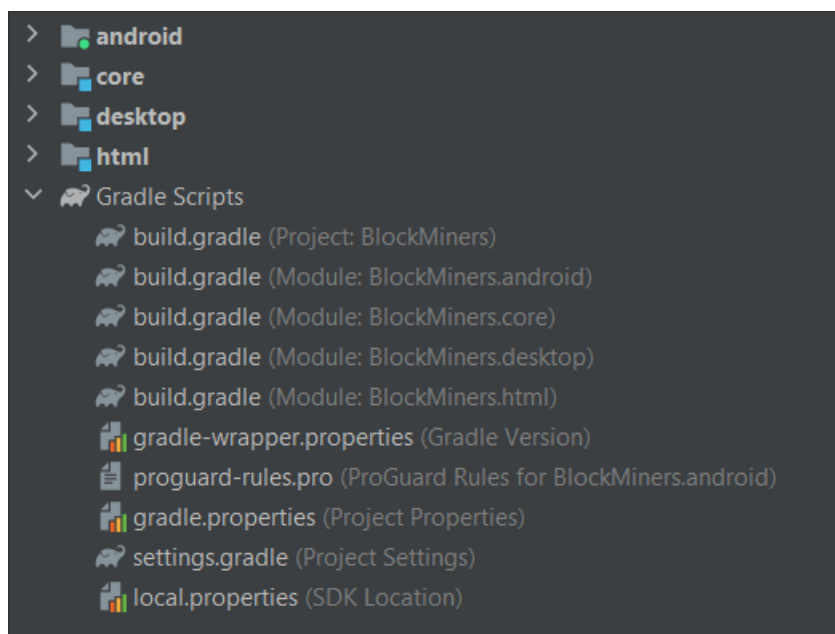


Figura 36: Estructura de archivos base

La aclaración de la existencia de dichos módulos es la siguiente:

- **Módulo core** - Es un módulo creado por defecto al hacer uso de la librería LibGDX en el cual se va a desarrollar todo el código de la aplicación. Dicho código es el que posteriormente va a ser compilado para las diferentes plataformas para las que se haya creado una estructura previa.

- **Módulo Android** - Es el módulo encargado de realizar los procesos pertinentes para compilar el código del módulo core y crear una aplicación móvil ejecutable en sistemas operativos Android.
- **Módulo desktop** - Este módulo es el responsable de realizar la compilación del código del módulo core y poder ejecutarlo posteriormente en el escritorio del ordenador. Este módulo se crea por defecto al hacer uso de la librería LibGDX, ya que el desarrollo base se realiza para la versión de escritorio.
- **Módulo html** - Este módulo es el encargado de realizar la compilación del código del módulo core y posteriormente ser utilizado en navegadores web. El código resultante a la compilación de este módulo es el que se alojará posteriormente en un servidor de la Universidad de Deusto.
- **Gradle Scripts** - Este apartado contiene las configuraciones respectivas a los módulos mencionados previamente. Entre las que se destacarían las características a mencionar a continuación.
 - **build.gradle (Module: BlockMiners.android)** - Dicho archivo de configuración corresponde al módulo de Android y es en este donde se especifica la versión de compilación del Software Development Kit utilizado junto con el mínimo SDK soportado. Además de todos los temas relacionados con la configuración de las características de la aplicación, en este archivo se describe de donde debe obtener la aplicación las imágenes, archivos de audio, fuentes de letras, etc. Al final del archivo se configuran los comandos que realiza el propio IDE cuando la persona encargada de realizar el desarrollo comienza la compilación del código. todo viene preconfigurado, por lo tanto, generalmente el único parámetro a modificar es el relacionado con la versión del SDK.
 - **build.gradle (Module: BlockMiners.html)** - Dicho archivo de configuración corresponde al módulo de html y es donde se especifica la versión de Google Web Toolkit, siglas de GWT, encargado de compilar el código de Java convirtiendo dicho código en Html y JavaScript. También contiene las instrucciones a seguir para crear directorios en los que se almacena el código una vez terminado la compilación.

3.4.2.2 Estructura de archivos avanzada del proyecto

En esta fase se describe más detalladamente la estructura subyacente a los módulos descritos previamente.

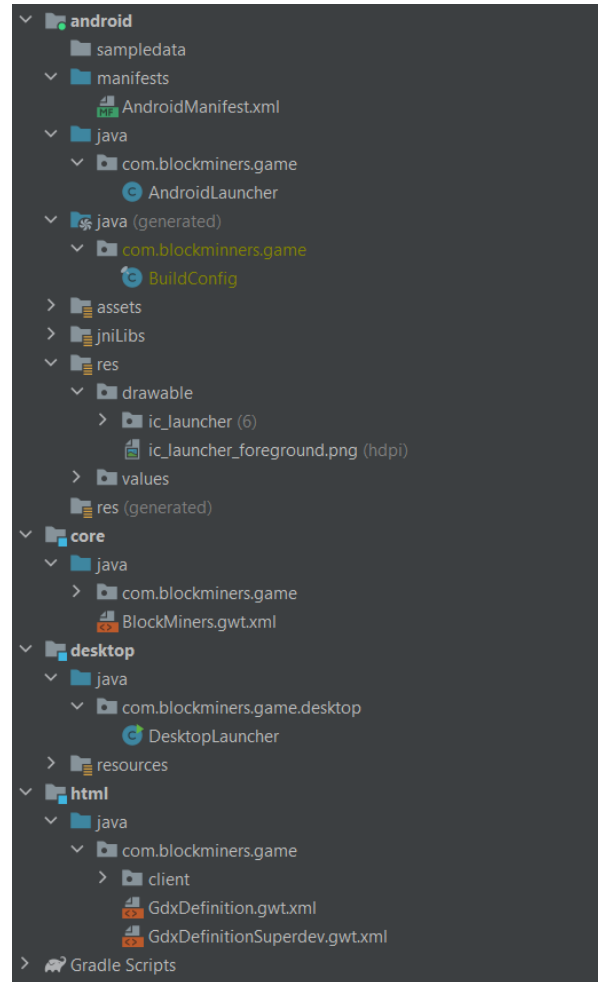


Figura 37: Estructura de archivos completa

- **android** - Este módulo contiene la siguiente estructura de archivos:
 - **manifests** - Contiene un archivo llamado *AndroidManifest.xml* situado en la raíz de la fuente del proyecto, su nombre debe ser idéntico y no debe modificarse. Dicho archivo describe la información esencial de la aplicación para las herramientas de creación de Android y el sistema operativo Android. En el manifiesto, se especifica el archivo que se debe ejecutar junto la ubicación de su paquete correspondiente, se declara el icono que se le va a asignar a la aplicación, el nombre que estará visible junto a la aplicación una vez se haya instalado en el dispositivo móvil y los permisos de los que la aplicación deberá hacer uso, entre tantos parámetros.
 - **java** - Contiene el paquete llamado *com.blockminers.game* que a su vez contiene la clase a ejecutar en el módulo de Android cuyo nombre es *AndroidLauncher*. Esta clase es en la que se encarga de iniciar la ejecución de la aplicación.

- **java (generated)** - Contiene características relacionadas con la última compilación realizada. Almacena el id de la aplicación, el número de la versión, entre otros.
- **assets** - Contiene los archivos multimedia que serán usados en el proyecto. En esta carpeta también se almacenan los archivos relacionados con el cambio de fuentes de texto.
- **jniLibs** - En esta ubicación, se encuentran las librerías y paquetes necesarios para la correcta ejecución de la aplicación.
- **res** - Contiene una carpeta con nombre *drawable* en la que se ha configurado la imagen asignada al icono de la aplicación, dicha imagen está representada como la imagen *ic_launcher_foreground.png*.
- **core** - Este módulo contiene la siguiente estructura de archivos:
 - **java** - Contiene el paquete llamado *com.blockminers.game*. Este paquete contiene todas las clases Java creadas para la realización del proyecto. El archivo con nombre *BlockMiners.gwt.xml*, contiene la ruta que enlaza con los archivos del módulo HTML.
- **desktop** - El módulo en cuestión contiene la siguiente estructura de archivos:
 - **java** - Contiene un paquete llamado *com.blockminers.game.desktop*. Dicho paquete contiene la clase principal a ejecutar en ordenadores. Esta clase es la encargada de iniciar la ejecución de la aplicación.
 - **resources** - Carpeta en la que se almacenan los archivos multimedia que posteriormente serán usados en el proyecto, junto con el almacenamiento de los archivos relacionados con la modificación de las fuentes de texto.
- **html** - Este módulo contiene la siguiente estructura de archivos:
 - **java** - Contiene el un paquete llamado *com.blockminers.game*. El paquete en cuestión alberga a su vez, una carpeta llamada *client* en la cual se encuentra la clase encargada de iniciar la ejecución de la aplicación. Los archivos llamados *GdxDefinition.gwt.xml* y *GdxDefinitionSuperdev.gwt.xml* corresponden a archivos de configuración del módulo en cuestión.

3.4.3 Compilación del código en sus diferentes módulos

En este apartado se va a realizar la descripción de las diferentes formas de compilación del código para las diferentes plataformas.

- **Compilación Android** - En esta situación se puede realizar de dos maneras:
 - **Instalación directa** - Se hace clic en el botón de ejecución con el modo de Android seleccionado. La aplicación se instala directamente en el dispositivo en cuestión, tanto en un dispositivo físico como en un dispositivo virtual creado para instalar la aplicación.

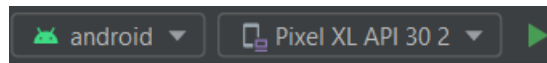


Figura 38: Compilación Android

- **Creación de un archivo con extensión .apk firmado** - Esto permite crear una aplicación móvil con extensión .apk, permitiendo la instalación de la aplicación en otros dispositivos móviles con sistema operativo Android.[23]
- **Compilación web** - Para la compilación del código y posteriormente ser usado en navegadores web, se realiza de una manera diferente a la compilación de Android. En este caso se debe abrir la *Terminal*, proporcionado por el propio Android Studio, para ejecutar comandos mediante línea. Una vez abierta dicha *Terminal*, se debe ejecutar el siguiente comando: “*gradlew html:dist*” [24] tal y como se puede observar en la siguiente imagen.

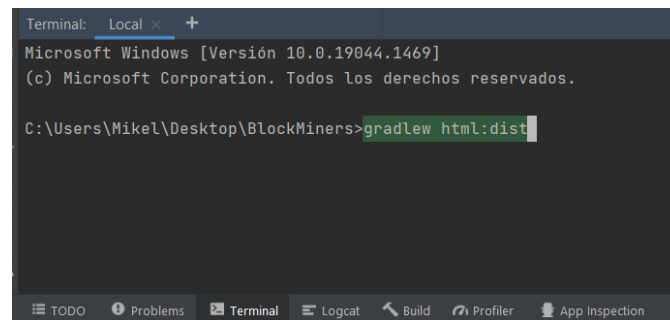


Figura 39: Figura compilación web

El resultado de la ejecución del comando anterior almacenará el código compilado en la carpeta del módulo html, más concretamente en: “*html → build → dist*”. Los archivos ubicados en la ruta anterior son los que posteriormente se deben situar en el servidor en el que se alojará la aplicación en su versión web.

3.5 INTERFACES

En este apartado se va a realizar un análisis de las distintas interfaces de la aplicación.

Interfaz de transición

La interfaz de transición, también conocida en inglés como *Splash Screen*, es la primera interacción que realiza la aplicación con el usuario. En la *Splash Screen*, se muestra el nombre de la aplicación, junto con el nombre de su creador y el diseñador de esta. Al transcurrir unos segundos la interfaz desaparece y muestra la interfaz del menú principal.

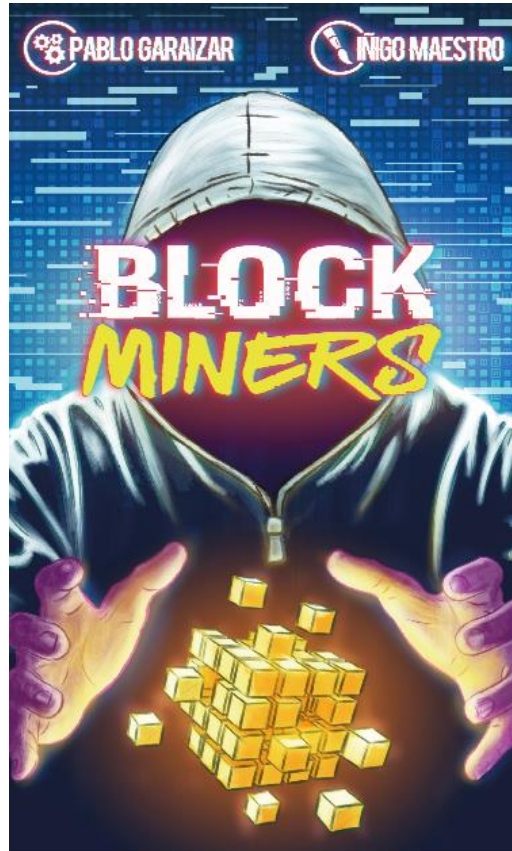


Figura 40: Interfaz transición

Interfaz de menú

Esta interfaz es la que se muestra una vez terminada la interfaz de transición. En esta interfaz se visualizan las opciones a escoger que son las siguientes: “Jugar”, “Ajustes”, “Ayuda” y “Salir”. Dependiendo de la opción escogida, se le redirigirá al usuario a una interfaz distinta.



Figura 41: Interfaz menú principal

Interfaz de ajustes

La interfaz de ajustes es una interfaz a la cual únicamente se puede acceder haciendo clic en la opción “Ajustes” del menú principal.

En esta interfaz se muestra el id asociado al usuario, dicho id es generado automáticamente. También en esta interfaz se observa la cantidad de juegos que ha realizado el usuario junto con la cantidad de *gamecoins* que ha minado durante el transcurso de las partidas realizadas. Además, también existe el botón de continuar cuyo funcionamiento es regresar a la interfaz del menú principal.



Figura 42: Interfaz ajustes

Interfaz de ayuda

La interfaz de ayuda, al igual que la interfaz de ajustes, únicamente se puede mediante el menú principal y para ello debe hacerse clic en “Ayuda”. La interfaz de ayuda contiene información relativa a la aplicación junto con la finalidad de esta. También incluye un enlace en el cual, al hacer clic, redirige a la página en cuestión. Si se hace clic en icono situado en la parte inferior derecha, se redirigirá al menú principal. La interfaz de ayuda se puede observar en la siguiente imagen.

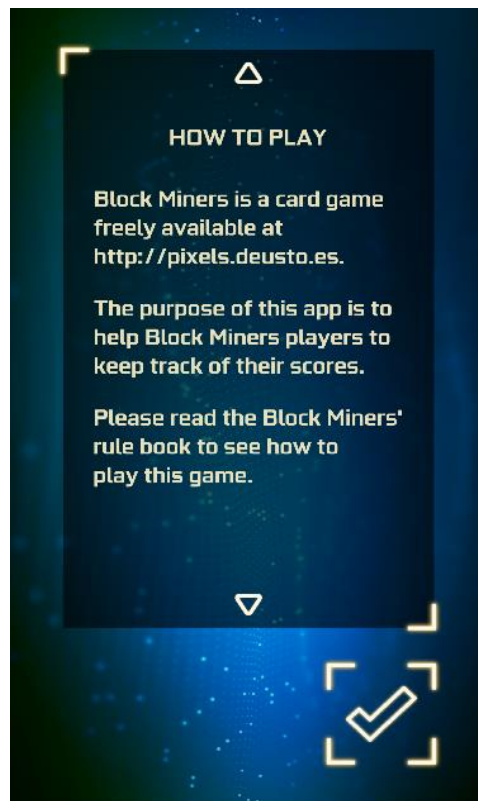


Figura 43: Interfaz ayuda

Interfaz de seleccionar jugadores

A la interfaz de seleccionar la cantidad de jugadores se accede desde el menú haciendo clic en “Jugar”. Esta interfaz contiene tres cantidades numéricas para escoger la cantidad de jugadores que vayan a jugar, siendo la cantidad de dos jugadores la opción marcada de forma predeterminada. En caso de ser una cantidad diferente de jugadores, se selecciona la cantidad deseada y dicha cantidad cambiara de color para resaltar e indicar visualmente la cantidad escogida. Posteriormente se hace clic en la esquina inferior derecha para continuar a la siguiente interfaz; no obstante, si se hace clic en la esquina inferior izquierda se regresará al menú principal.



Figura 44: Interfaz selección jugadores

Interfaz de juego

Esta interfaz contiene el juego en sí. Se pueden diferenciar tres áreas destacables en esta interfaz que son las siguientes:

- **El área de los marcadores** - Es la parte situada cerca del margen lateral derecho de la interfaz, en la cual están visibles las puntuaciones respectivas a los distintos jugadores de la partida.
- **El área del teclado** - Es la parte situada en la en el lado inferior de la interfaz. En dicha área, se destacan los dígitos del cero al nueve junto con el icono de borrar la cantidad indicada. Por otra parte, se observan los iconos de “Intro”, “Minar” y “Robar” y sus funciones son las siguientes:
 - **Intro** - Al hacer clic en dicho icono se desencadenan una serie de validaciones. Se valida que la cantidad numérica introducida no produce ningún error en el sistema, en caso de producirse algún error, se muestra por pantalla un indicativo de error del código numérico introducido. Por otra parte, otra de las validaciones que realiza es comparar la cantidad de bloques de transacciones que se han

realizado, inicialmente el tablero está oculto y al introducir un número válido se visualiza el primer bloque; al introducir un segundo número, se comprueba la cantidad de bloques existentes y se pondría debajo del primer bloque. En caso de querer introducir un tercer bloque de transacciones, se introduce y el primer bloque del tablero desaparece, el segundo bloque toma la posición del primero y el tercer bloque introducido toma la posición del segundo. Se debe señalar que el bloque desaparece visualmente pero no es eliminado. En las siguientes imágenes se muestran situaciones de error, al insertar un código, dos códigos y un tercer código.

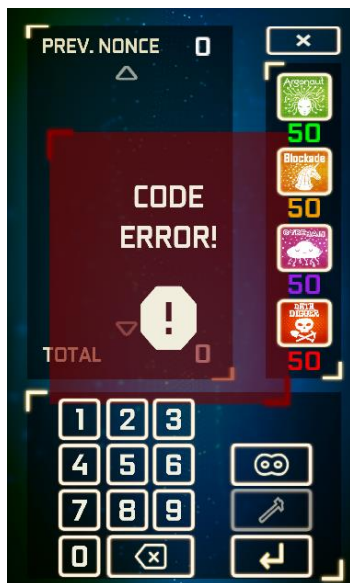


Figura 45: Código erróneo



Figura 46: Bloque 1



Figura 47: Bloque 2



Figura 48: Bloque 3

- **Minar** - Al hacer clic en el icono, se empieza a comprobar si las condiciones de para minar un bloque se cumplen. Para que se mine un minado debe ocurrir lo siguiente: la cantidad de bloques introducidos debe ser igual o mayor a la cantidad de jugadores seleccionados. También, la suma del *nonce anterior* + la suma de todas las transacciones de los bloques del tablero + el *nonce* correspondiente al número de la carta introducida debe ser múltiplo de diez. Para mostrarlo de una manera más simple, el *total* contiene la cantidad de las transacciones de los bloques + el *nonce previo*. Si se cumple esa validación, se vacía el tablero y el *nonce* de la carta con la que se ha realizado el minado se convierte en el *nonce previo*. En las siguientes imágenes se muestra un ejemplo de cómo sería. La partida tendría la siguiente situación.



Figura 49: Minado 1

Como se puede observar en la imagen anterior, la cantidad del parámetro *total* es siete, por lo tanto, se debe ingresar una carta cuyo *nonce* sea igual a tres para que la suma sea diez y diez es múltiplo de diez. La carta cinco tiene un *nonce* de tres, se ingresa la carta cinco y se le hace clic en *minar*. Si la validación del minado es correcta, ocurre lo que se puede ver en la siguiente imagen.

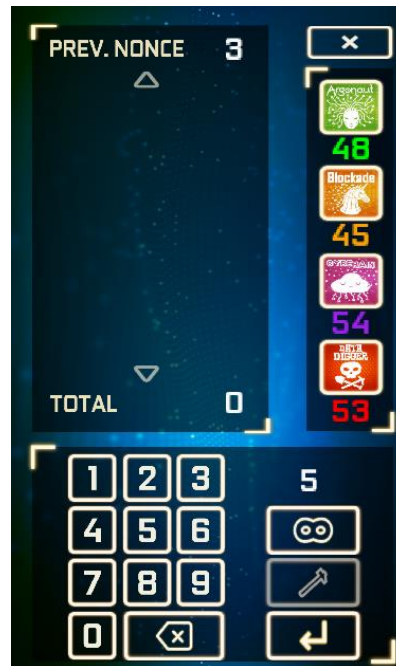


Figura 50: Minado 2

En caso de que el valor numérico introducido junto a la suma de todos los parámetros anteriormente comentados no fuera múltiplo de diez, nos dirigirá a una interfaz de error de minado en la cual se muestra un texto indicando que el minado ha producido un error. Para salir de dicha interfaz, se debe hacer clic en el icono inferior derecha, el cual nos sitúa en la interfaz de juego.

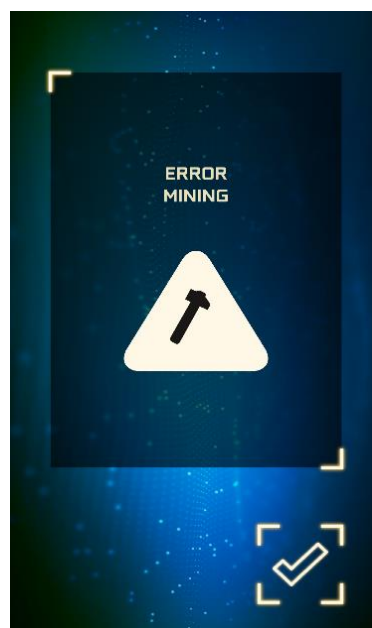


Figura 51: Error minado

En caso de que se ingrese un número de carta que no exista en el listado de las cartas, se mostrará el mismo error que la figura 38.

- **Robar** - La funcionalidad de dicha interfaz se va a describir cuando se detalle las características de la interfaz en cuestión.
- **El área del tablero** - Es la última área de esta interfaz y, por lo tanto, es el área restante. En esta área se visualizan los bloques de transacciones que se realizan durante el transcurso de la partida, como se ha podido ver en los casos anteriores.



Figura 52: Tablero juego flecha superior activada

- **Flecha superior** - Como se puede observar en la imagen anterior, al ingresar un tercer bloque se activa la flecha superior. El funcionamiento de esta es mostrar al usuario la carta anterior al bloque superior. En la siguiente imagen se muestra como al pulsar la flecha superior, la carta superior bajará al bloque inferior y la carta previa a la superior, se visualizará en el bloque superior.



Figura 53: Tablero juego flecha inferior activada

- **Flecha inferior** - Siguiendo con la imagen anterior, se nos activa la flecha inferior que indica la existencia de un bloque inferior. Al pulsar en la flecha inferior volveremos a la situación en la se nos desactiva la flecha inferior y se nos activará la superior, tal y como ocurre en la figura 45.
- **Doble clic** - Al hacer un doble clic en el bloque inferior se borrará el bloque en cuestión, en caso de que no exista un bloque inferior, no ocurrirá nada. En esta interacción pueden darse dos situaciones dependiendo de la cantidad de bloques existentes en el tablero:
 - **Dos bloques en el tablero** - En el caso de que existan dos bloques en el tablero y el usuario realice una doble pulsación en el bloque inferior, este desaparecerá decrementando la puntuación total.



Figura 54: Doble clic dos bloques



Figura 55: Resultado doble clic con dos bloques

- **Tres o más bloques en el tablero** - En el caso de que existan tres o más bloques en el tablero y el usuario realice una doble pulsación en el bloque inferior, este desaparecerá dejando la posición inferior a la carta superior y la posición del bloque superior a la carta previa.



Figura 56: Doble clic tres bloques



Figura 57: Resultado doble clic con tres bloques

Interfaz de robar

Esta interfaz contiene los iconos de los cuatro jugadores y se debe escoger un jugador al que robar cinco *gamecoins*. Se selecciona el jugador al que se le quiere robar y se le da a continuar, esto hará que se vuelva a interfaz de juego, en caso de que se le hubiera dado a la opción para “Atrás”, se hubiera llegado al mismo punto; es decir, el usuario hubiera llegado a la interfaz de juego.



Figura 58: Interfaz robar

Interfaz fin de partida

Para llegar a esta interfaz se debe hacer clic en el icono superior derecha ubicado en la interfaz de juego. En esta interfaz se muestran las puntuaciones de los jugadores, juntos con dos iconos inferiores. El izquierdo nos vuelve a situar en la interfaz de juego, mientras que el derecho nos sitúa en la interfaz del menú principal.



Figura 59: Fin de juego

Interfaz de confirmación de salida del juego

Esta interfaz es accesible mediante el icono de “Salir” situado en el menú principal. Esta interfaz contiene una confirmación para salir del juego, si hacemos clic en el icono inferior izquierda regresamos al menú principal; no obstante, si hacemos clic en el icono inferior derecha, se nos cerrará la aplicación. Destacar que esta opción no funciona en la versión web ya que no se puede salir debido a que la aplicación siempre va a estar disponible en todo momento.



Figura 60: Confirmación salir

3.6 PLAN DE PRUEBAS

Durante el desarrollo del presente proyecto, se han llevado a cabo diferentes tipos de pruebas con la intención de garantizar con el cumplimiento de la calidad y funcionalidad.

3.6.1 Pruebas de Unidad

Una prueba de unidad consiste en probar unidades individuales de software con el propósito de validar correctamente el funcionamiento.

Las pruebas de la aplicación móvil y web se han realizado mediante la validación de los datos ingresados en los campos numéricos en la interfaz de juego ya que es en el único lugar de la aplicación en el que el usuario puede ingresar datos; por lo tanto, se debe comprobar dicho aspecto. Para la versión web se ha hecho uso de XAMPP como servidor local para alojar el código html y JavaScript. Respecto a la versión de móvil, se ha hecho uso de un dispositivo móvil personal junto con diversos dispositivos móviles virtuales de distintas dimensiones y versiones de Android, siempre versiones iguales o superiores a siete, creados mediante Android Studio.

Se han hecho cuatro pruebas unitarias para los casos posibles de conflictos relacionados inserción de bloques y cinco relacionados con el minado de bloques, ambos ubicados en la interfaz de juego:

- **Pruebas inserción de bloques:**
 - **No se ha introducido ningún número** - Se intenta ingresar una carta sin haber puesto ningún número previamente por teclado. El resultado esperado es el siguiente:
 - figura 38
 - **El código introducido es 0** - Se intenta realizar una búsqueda del código en cuestión en el listado de cartas guardado, pero no existe ninguna carta con dicho código. El resultado esperado es el siguiente:

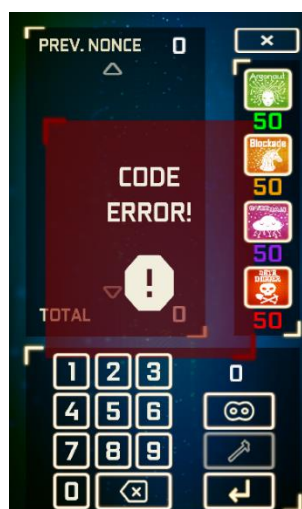


Figura 61: Error 0

- **El código introducido es 00** - Se intenta realizar una búsqueda del código 00 en el listado de cartas guardado, pero no existe ninguna carta con dicho código. El resultado esperado es el siguiente:

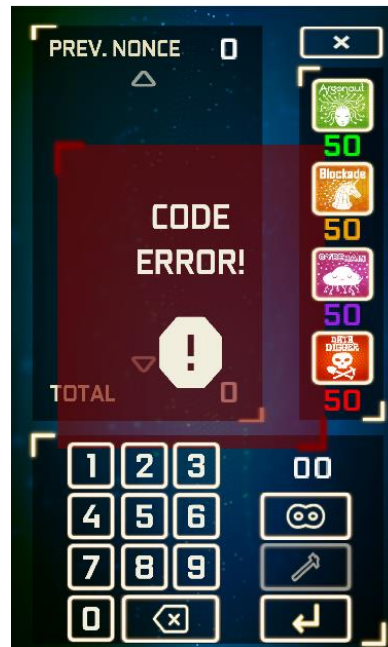


Figura 62: Error 00

- **El código introducido es mayor a 73 o mayor** - El listado contiene cartas del uno al setenta y dos; por lo tanto, al ingresar una cantidad mayor a setenta y dos el resultado esperado es el siguiente:

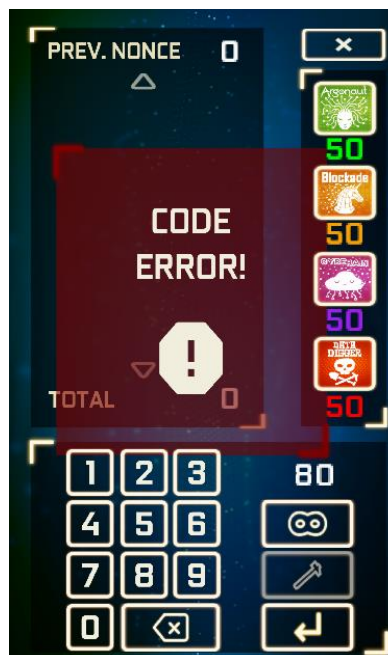


Figura 63: Error 80

- **Pruebas minado de bloques:**

- **No se ha introducido ningún número** - Se intenta realizar un minado de bloques sin haber puesto ningún número previamente por teclado. El resultado esperado es el siguiente:

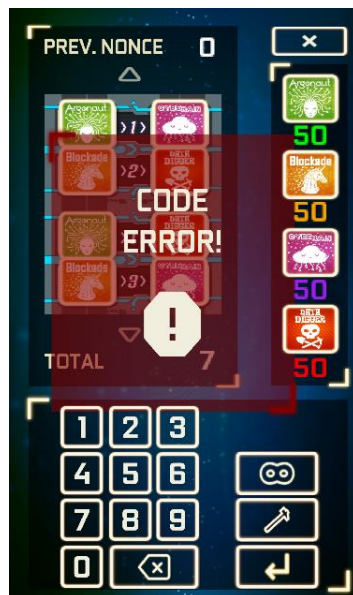


Figura 64: Minado sin código

- **El código introducido es 0** - Se intenta realizar un minado con el *nonce* de la carta introducida, pero no existe ninguna carta con dicho código. El resultado esperado es el siguiente:

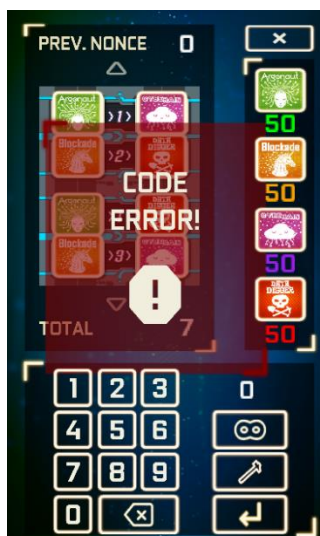


Figura 65: Error minado 0

- **El código introducido es 00** - Se intenta realizar un minado con el *nonce* del código 00 en el listado de cartas guardado, pero no existe ninguna carta con dicho código. El resultado esperado es el siguiente:

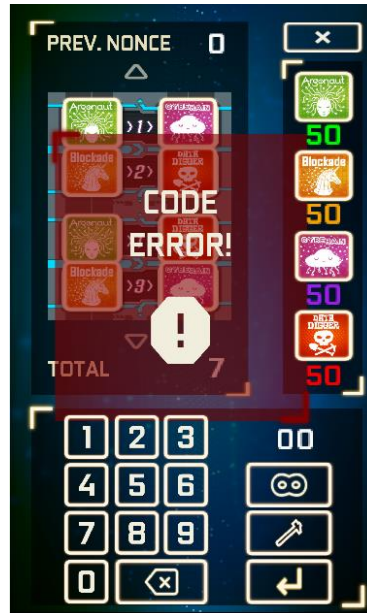


Figura 66: Error minado 00

- **El código introducido es mayor a 73 o mayor** - El listado contiene cartas del uno al setenta y dos; por lo tanto, al ingresar una cantidad mayor a setenta y dos e intentar realizar un minado, el resultado esperado es el siguiente:

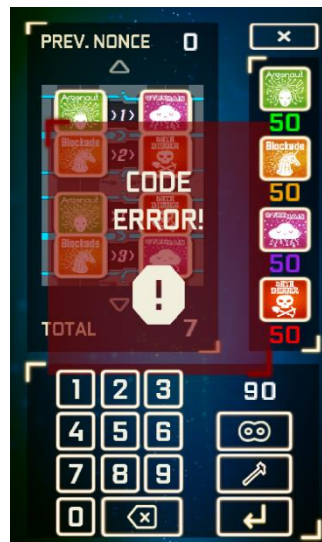


Figura 67: Error minado 90

- **El código introducido no realiza las validaciones pertinentes** - Para que se mine un bloque se tienen que cumplir una serie de condiciones. No obstante, en

caso de que exista una carta con la cantidad ingresada por teclado y no cumpla las condiciones, se espera el siguiente resultado:

figura 44

3.6.2 Pruebas de Integración

Las pruebas de integración comprueban que las unidades en la que se han realizado las pruebas funcionan correctamente en todos los casos, tanto en su versión móvil como en su versión web. La finalidad de este tipo de pruebas es exponer los fallos en las interacciones. Las pruebas de integración se han realizado en las diferentes fases del proyecto, para saber si había algún tipo de conflicto al realizar compilación del código a las diferentes plataformas a las que está dirigido este proyecto.

Las pruebas de integración han sido las siguientes para ambas versiones y en ambas se ha seguido el mismo orden a la hora de la realización:

- **PI1** - Comprobar si se lograba acceder a la aplicación correctamente, este paso es el más importante ya que si fallase esto, no se podrían realizar los demás procesos de la aplicación.
- **PI2** - Comprobar el enlace con las siguientes interfaces asegurándose que la navegación es correcta y no se da el caso en el que se acceda a una interfaz y no haya manera de salir.
- **PI3** - Comprobación del correcto funcionamiento al hacer clic en el enlace situado en la interfaz de *Ayuda*

3.6.3 Pruebas de Validación

Las pruebas de validación tienen como objetivo la evaluación del cumplimiento de los requisitos, independientemente si los requisitos son funcionales como no funcionales.

Durante el transcurso de la realización de este proyecto, se han realizado reuniones con el director del proyecto en las cuales se ha validado el progreso realizado. Las pruebas de validación se han realizado compilando el código en su versión web y alojándolo en el servidor Apache proporcionado por XAMPP, esto ha facilitado realizar las validaciones debido a la situación de pandemia actual junto con la problemática de horarios poco compatibles.

3.7 MANUAL DE USUARIO

El presente apartado contiene el manual de usuario entregado al director del proyecto. El objetivo de este manual es ofrecer al usuario las posibilidades y directrices a seguir a la hora de jugar partidas.

Lo primero es abrir la aplicación, para ello se deberá ingresar en la página web de la Universidad de Deusto donde estará alojada la aplicación en su versión web. En caso de estar en dispositivos móviles con sistema operativo Android, se deberá abrir la aplicación que contenga el siguiente icono:



Figura 68: Manual icono aplicación

Una vez se haya abierto la aplicación se mostrará la siguiente pantalla que enlazará posteriormente con el menú principal.

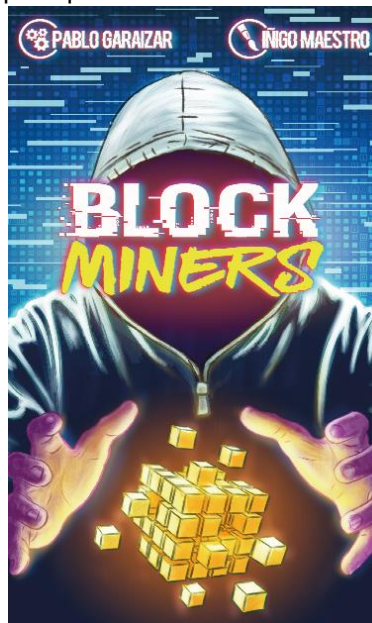


Figura 69: Manual interfaz transición

Una vez la animación de la pantalla anterior termine, como se ha comentado previamente, se visualizará el menú principal.



Figura 70: Manual interfaz menú principal

El menú dispone de cuatro opciones a escoger: Jugar, Opciones, Ayuda y Salir.

La opción de Ajustes contiene el id correspondiente al usuario de la aplicación junto con las métricas de las partidas jugadas y los *gamecoins* generados a lo largo de las partidas. Una vez se desinstala la aplicación móvil o se borran los datos de la aplicación, los datos relacionados con los ajustes se eliminarán. Lo mismo ocurre en su versión web en caso de desinstalación del navegador o borrado de los datos. La pantalla de ajustes contiene un botón en la esquina inferior derecha para volver al menú principal.

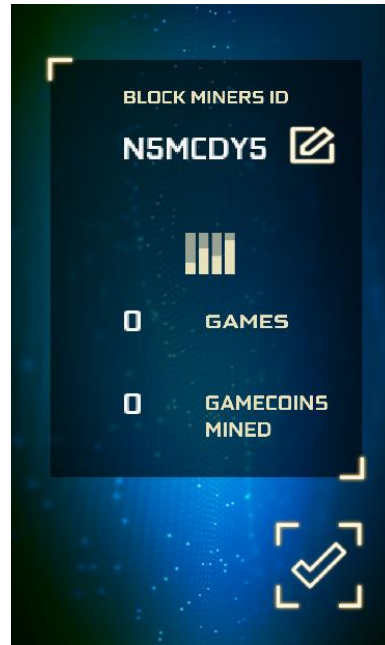


Figura 71: Manual interfaz ajustes

La opción de Ayuda contiene información relacionada sobre el propósito de esta aplicación junto con un enlace a <http://pixels.deusto.es> en donde se puede encontrar más información al respecto. También la pantalla en cuestión contiene un botón en la esquina inferior derecha para volver al menú principal.

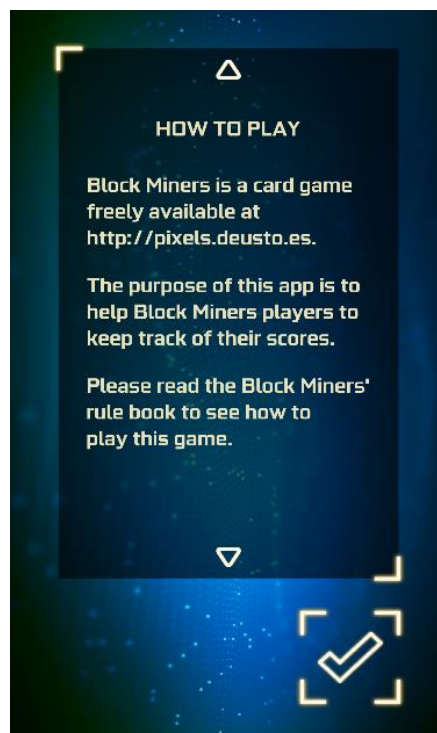


Figura 72: Manual interfaz ayuda

Al pulsar en la opción de Jugar, se abrirá una pantalla indicando que se seleccione la cantidad de jugadores. La aplicación mostrará la opción de dos jugadores escogida, si el usuario está de acuerdo pulsará el botón de continuar; en caso de que desee otra cantidad de jugadores, deberá seleccionar la cantidad y posteriormente pulsar en continuar, mostrando así la pantalla de juego. Por el contrario, si el usuario desea volver al menú principal, deberá pulsar en el botón de cancelar situado en la esquina inferior izquierda.



Figura 73: Manual interfaz seleccionar cantidad de jugadores

La pantalla de juego es donde se gestiona toda la parte relacionada con la partida. Al ingresar a esta pantalla se observará una pantalla como la siguiente:

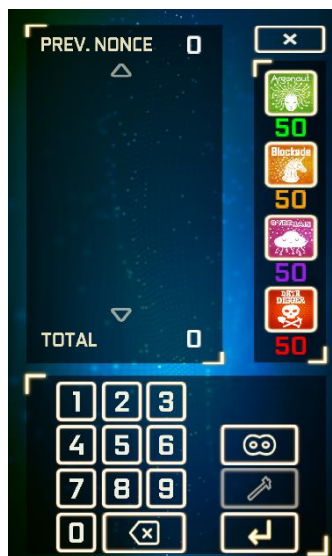


Figura 74: Manual interfaz juego

Para ingresar un bloque en el tablero, se debe ingresar el número de la carta, las cartas tienen un número asignado entre uno y setenta y dos. Por lo tanto, se selecciona una carta en dicho rango numérico y se pulsa en intro. Por ejemplo, se va a ingresar la carta uno. Las cartas cuyo rango sea entre uno y nueve se pueden representar de tanto con un 0 delante o no. Es decir, al ingresar la carta uno daría igual si ingresamos 1 o 01.

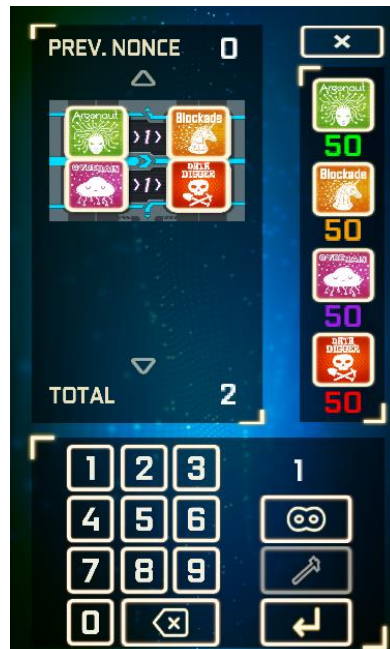


Figura 75: Manual interfaz juego al ingresar un bloque

Todos los bloques se insertan de la misma manera que el caso anterior.

Para realizar un minado de bloques, debe haber tantos bloques en el tablero como jugadores seleccionados. Si se cumple ese requisito, se activará el icono de minar situado encima de intro.



Figura 76: Manual interfaz juego al ingresar dos bloques

En la imagen anterior observamos como están en el tablero dos bloques y la cantidad de jugadores es dos, por lo tanto, el icono de minar se ha activado. Para minar un bloque además de la condición previamente comentada se debe cumplir otra validación, la suma del *nonce* previo más la cantidad de transacciones de los bloques más el *nonce* de la carta introducida debe ser múltiplo de diez. Si se cumple se realizarán las transacciones de las cartas y el *nonce* la carta con la que se ha procedido a minar pasará a ser el *nonce* previo tal y como se puede ver en la siguiente imagen.

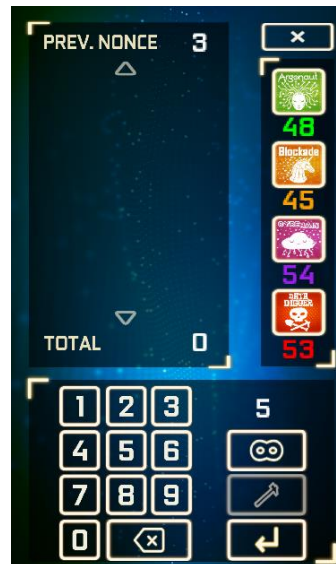


Figura 77: Manual interfaz juego al realizar un minado

En caso de introducir un número cuyo *nonce* no valide la condición previa, se mostrará la siguiente pantalla, en dicha pantalla se muestra un texto indicando que el minado ha producido un error y con el botón de continuar situado en la esquina inferior derecha, volverá a la pantalla de jugar.



Figura 78: Manual interfaz error minado

En caso de ingresar un número de carta que no esté en la lista de cartas o hacer clic en intro o minar sin haber ingresado un número previo, se mostrará el siguiente error por pantalla:

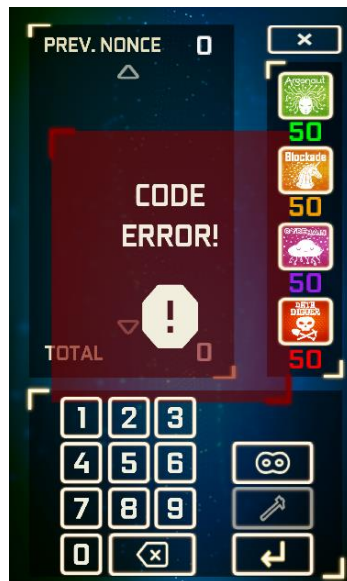


Figura 79: Manual interfaz juego al ingresar un código de carta no válido

Los jugadores pueden realizar robos de *gamecoins* desde el inicio de la partida. Para ello, el usuario debe pulsar el botón de minar situado encima del botón de minar. Una vez pulsado el botón, se visualizará una pantalla preguntando al jugador al que se le desea robar cinco *gamecoins*. De forma predeterminada, estará escogida la opción de *Argonaut* pero se puede escoger cualquier jugador haciendo clic en el icono correspondiente y posteriormente darle a continuar, esto volverá a situar al usuario en la pantalla de juego pero se le habrá restado los cinco *gamecoins* al jugador seleccionado. Por otra parte, el botón de cancelar consiste en cancelar y no permitir llevar a cabo el robo, volviendo a situar al jugador en la pantalla de juego.



Figura 80: Manual interfaz robo de gamecoins

Tras insertar una cantidad de bloques superior a tres, el usuario podrá navegar con las flechas superiores e inferiores del tablero. Para que estas funcionen correctamente, una vez que se pulsa la flecha superior se debe ascender hasta el primer bloque; lo mismo ocurre con la flecha inferior, es recomendable descender hasta el último bloque, todo ello para que la aplicación no genere ningún error.



Figura 81: Manual interfaz juego flecha superior



Figura 82: Manual interfaz juego flecha inferior

El usuario puede hacer doble clic en el bloque inferior del tablero, esto eliminará el bloque en cuestión dando la posibilidad de insertar nuevos bloques.



Figura 83: Manual interfaz juego doble clic en el bloque inferior

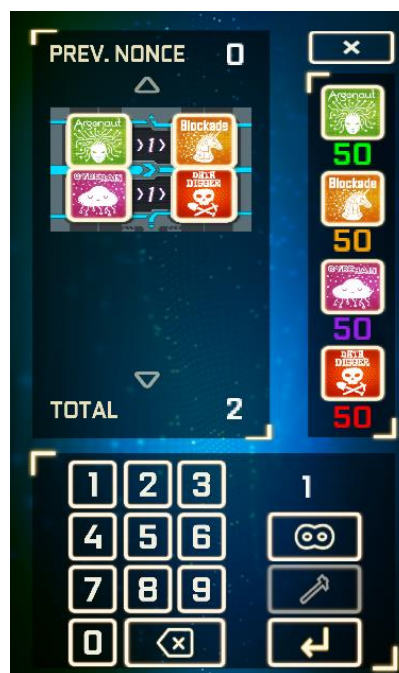


Figura 84: Manual interfaz juego resultado de realizar doble clic en el bloque inferior

Para terminar la partida el usuario debe pulsar el botón de salir situado en la esquina superior derecha, tras lo cual se mostrará una pantalla con las puntuaciones de los jugadores y la conformación para terminar la partida. En caso de querer terminar la partida se pulsa en continuar y mostrará el menú principal, en cambio, si se pulsa atrás se regresará a la pantalla de juego.



Figura 85: Manual interfaz finalización partida

Finalmente, para salir del juego se debe pulsar en Salir. Esto situará al usuario en una pantalla con la confirmación para salir del juego. Si se selecciona cancelar, se volverá al menú principal, en cambio, si se selecciona continuar, se cerrará el juego.

Se debe destacar que en la versión web de la aplicación, esta función de salir no estará operativa.

4. PLANIFICIÓN

En el presente capítulo, se describe detalladamente la planificación del proyecto. Para ello, se han definido y analizado los diferentes perfiles que componen el proyecto, así como las fases y tareas en las que se ha organizado.

4.1 EQUIPO DEL PROYECTO

En este apartado se realiza un análisis de los diferentes perfiles que conforman parte de la estructura interna del proyecto.

- **Analista** - Perfil encargado de realizar la especificación de los requisitos del proyecto. El analista también será el perfil encargado de realizar la planificación y el presupuesto del proyecto, junto con la posterior redacción de la memoria del proyecto y el manual de usuario pertinente.
- **Diseñador** - Este perfil es el encargado de realizar la especificación del diseño junto con su arquitectura. También se encargará de diseñar el plan de pruebas.
- **Desarrollador** - Perfil cuya tarea consiste en realizar la implementación de la funcionalidad de la aplicación en sus versiones móvil y web. El diseñador también va a ser el responsable de la gestión de los datos generados en la aplicación junto con la validación de pruebas unitarias a medida que se va avanzando en el proyecto; una vez terminado el proyecto, se encargará de realizar las pruebas de integración.
- **Director del proyecto** - Se encargará de realizar el seguimiento del proyecto, ayudando en las correcciones de la memoria final y el manual de usuario.

Destacar que los perfiles mencionados previamente, a excepción del director del proyecto, han sido realizados por el alumno encargado de realizar este proyecto.

En la siguiente imagen se puede observar un organigrama con la estructura previamente descrita.

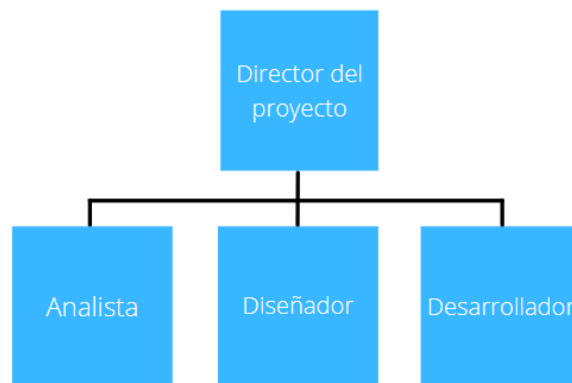


Figura 86: Organigrama

4.2 ESTRUCTURA CRONOLÓGICA DEL PROYECTO

En este apartado se indican las tareas que han sido necesarias para el diseño y desarrollo del proyecto final y los pasos intermedios. Las tareas se encuentran agrupadas en las diferentes fases que intervienen en el proceso de desarrollo que ha sido escogido.

- Fase I - Fase inicial:
 - **T01** - Selección del director del proyecto que será el responsable de la coordinación y asesoramiento del proyecto.
 - **T02** - Reunión entre el responsable del proyecto y el alumno encargado de realizar el proyecto. En dicha reunión, se va a realizar la presentación de la propuesta del proyecto junto con el alcance de este.
 - **T03** - Realizar el registro del proyecto en la plataforma ALUD.
- Fase II - Análisis:
 - **T04** - Reunión entre el director del proyecto y la persona encargada de realizar el proyecto. En dicha reunión se van a comentar los antecedentes, concretar los objetivos y alcance del proyecto.
 - **T05** - Reunión con el responsable del proyecto para detallar los requisitos del proyecto tanto los requisitos funcionales como los no funcionales, además de las restricciones que debe cumplir el proyecto.
 - **T06** - Detallar los plazos para el desarrollo del proyecto y el presupuesto de este, teniendo en cuenta el gasto de los equipos y el software.

- Fase III - Diseño:
 - **T07** - Búsqueda de tecnologías de desarrollo similares a las características y necesidades del proyecto. Realizar una búsqueda de aplicaciones móvil y web en el ámbito de *blockchain*.
 - **T08** - Especificación del diseño que debe poseer la aplicación, en la versión móvil y web.
- Fase IV - Ejecución:
 - **T09** - Implementación del código desarrollado en su versión móvil y web.
 - **T10** - Realización de pruebas individuales en ambas versiones de la aplicación durante el transcurso del proyecto, comprobando así, el correcto funcionamiento de este.
 - **T11** - Redacción de la memoria indicando todo lo ocurrido en el transcurso de la duración del proyecto.
 - **T12** - Redacción de un manual de usuario que permita al jugador entender el funcionamiento de la aplicación.
- Fase V - Pruebas:
 - **T13** - Realización de pruebas de integración, verificando así el correcto funcionamiento de la aplicación en su versión móvil y web.
 - **T14** - Realización de pruebas de validación, evaluando así el cumplimiento de los requisitos funcionales como no funcionales.
- Fase VI - Cierre del proyecto:
 - **T15** - Entrega de la memoria en los medios correspondientes
 - **T16** - Entrega del código desarrollado al director del proyecto para futuras modificaciones.
- Fase VII - Seguimiento:
 - **T17** - Revisar periódicamente con el director del proyecto las tareas desarrolladas durante la ejecución del proyecto junto con la resolución de las posibles dudas surgidas.

La siguiente tabla contiene el reparto y estimación realizado para cada tarea junto el responsable de esta.

4. PLANIFICACIÓN

Fases	Tareas	Encargado	Horas
Fase I	T01	Analista	2
	T02	Analista	4
		Director	2
	T03	Analista	1
Fase II	T04	Analista	1
		Director	1
	T05	Analista	6
		Director	3
	T06	Analista	1
		Director	1
Fase III	T07	Analista	8
	T08	Diseñador	12
Fase IV	T09	Desarrollador	100
	T10	Desarrollador	30
	T11	Analista	50
	T12	Analista	30
Fase V	T13	Desarrollador	30
	T14	Analista	15
Fase VI	T15	Analista	5
	T16	Analista	5
Fase VII	T17	Director	50

Tabla 1: Reparto de tareas

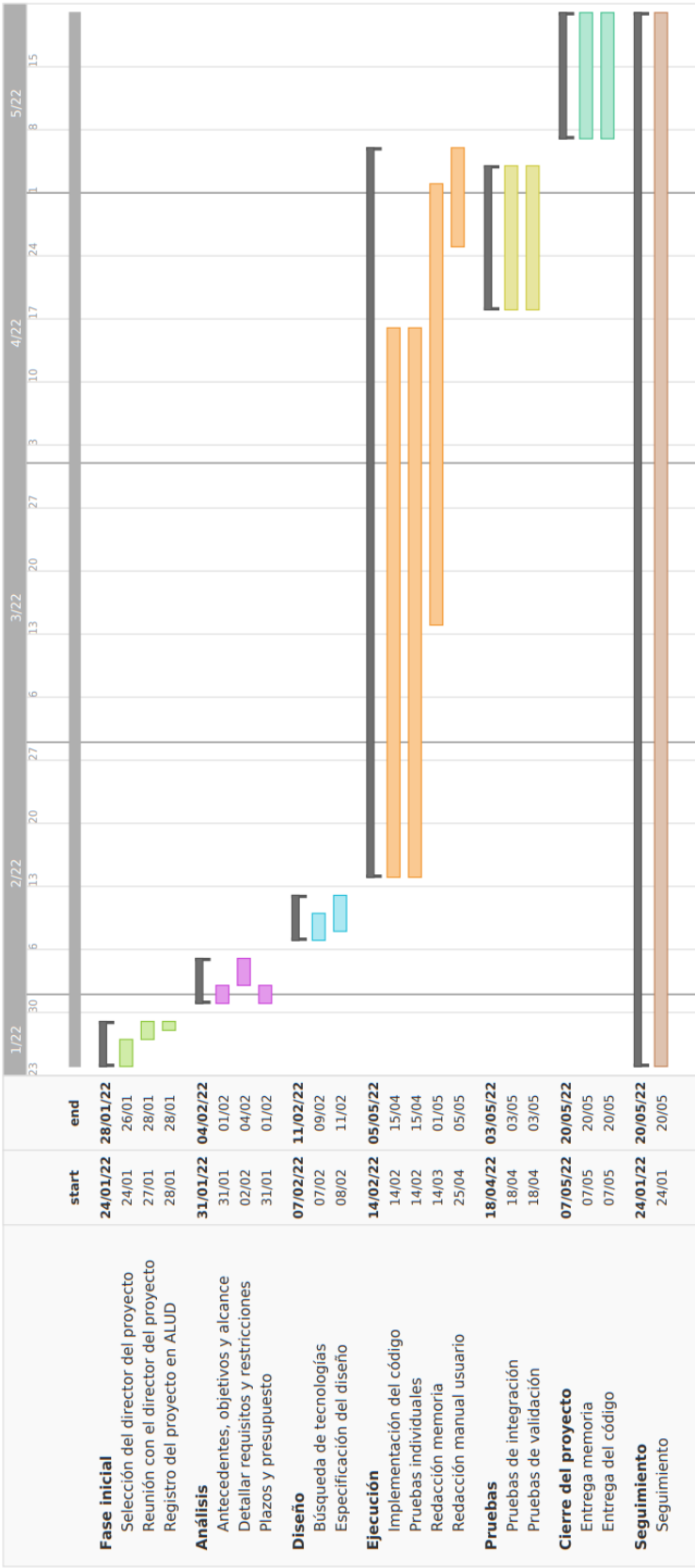


Figura 87: Diagrama Gantt

5. PRESUPUESTO

En este capítulo, se representa una estimación de los costes del proyecto. Para realizar dicha estimación se ha realizado una fragmentación entre los costes laborales y los costes materiales.

La siguiente tabla detalla los costes relacionados a cada perfil que trabajará en el proyecto.

Perfil	Coste
Director del proyecto	50 €/h
Analista	40 €/h
Diseñador	35 €/h
Desarrollador	40 €/h

Tabla 2: Coste /h

Tomando como referencia los precios asociados a cada perfil de la tabla anterior, la siguiente tabla muestra el coste laboral del proyecto agrupando estos en los perfiles correspondientes.

Director	57 horas	2.850 €
T02 - Reunión entre el director y alumno	2 horas	100 €
T04 - Reunión entre el director y el alumno para concretar los antecedentes, objetivos y alcance del proyecto.	1 horas	50 €
T05 - Reunión entre el director y el alumno para detallar los requisitos y restricciones del proyecto.	3 horas	150 €
T06 - Reunión para detallar los plazos de desarrollo y el presupuesto de este	1 horas	50 €
T17 – Seguimiento del proyecto periódicamente	50 horas	2.500 €
Analista	128 horas	5.120 €
T01 - Selección del director del proyecto	2 horas	80 €
T02 - Reunión entre el director y el alumno	4 horas	160 €

5. PRESUPUESTO

T03 - Realizar registro del proyecto en ALUD	1 horas	40 €
T04 - Reunión entre el director y el alumno para concretar los antecedentes, objetivos y alcance del proyecto.	1 horas	40 €
T05 - Reunión entre el director y el alumno para detallar los requisitos y restricciones del proyecto.	6 horas	240 €
T06 - Reunión para detallar los plazos de desarrollo y presupuesto de este	1 horas	40 €
T07 - Búsqueda de tecnologías	8 horas	320 €
T11 - Redacción de la memoria del proyecto	50 horas	2.000 €
T12 - Redacción del manual de usuario	30 horas	1.200 €
T14 – Realización de pruebas de validación	15 horas	600 €
T15 – Entrega de la memoria en medios correspondientes	5 horas	200 €
T16 – Entrega del código desarrollado al director del proyecto	5 horas	200 €
Diseñador	12 horas	420 €
T08 – Especificación de diseño	12 horas	420 €
Desarrollador	160 horas	6.400 €
T09 – Implantación del código desarrollado en su versión móvil y web	100 horas	4.000 €
T10 – Realización de pruebas individuales	30 horas	1.200 €
T13 – Realización de pruebas de integración	30 horas	1.200 €
TOTAL	357 horas	14.790 €

Tabla 3: Coste laboral

Los costes materiales a su vez se van a dividir en dos: *software* y *hardware*.

A continuación, se procederá a describir los costes respecto al *software*.

Concepto	Coste
Android Studio	Gratuito
LibGDX	Gratuito
XAMPP	Gratuito
Git	Gratuito
GitHub	Gratuito
GitHub Desktop	Gratuito
TeamGantt	Gratuito
VisualParadigm	Gratuito
TOTAL	Gratuito

Tabla 4: Coste software

Por otra parte, los costes *hardware* se reparten de la siguiente manera.

Material	Producto	Unidades	Precio (€/UD)	Importe
Equipo informático	MSI Modern 14 A10RAS-1049XES	1	1.050 €	131,25 €
Equipo informático	Huawei P30 Pro	1	1.099 €	183,16 €
TOTAL				314,41 €

Tabla 5: Coste hardware

El *hardware* experimenta una amortización constante a cuatro años en el caso del ordenador portátil y el dispositivo móvil presenta una amortización constante a tres años. Por lo tanto, en el caso del ordenador portátil se imputa una octava parte al coste del material y en el caso del dispositivo móvil se imputa una sexta parte al coste del proyecto.

En la siguiente tabla se resume el coste total relativo al proyecto.

VALORACIÓN	COSTE
Coste laboral	14.790 €
Coste software	Gratuito
Coste hardware	314,41 €
TOTAL	15.104,41 €

Tabla 6: Coste total

6. CONCLUSIONES

En el presente capítulo se representan los resultados y valoraciones obtenidas tras la finalización del proyecto.

6.1 RESULTADO OBTENIDOS

En relación con los aspectos obtenidos, se debe mencionar que tanto los objetivos y alcance se han completado con éxito.

Los requisitos tanto funcionales como no funcionales se han completado también con éxito, salvo por una restricción obligatoria. Dicha restricción consiste en enviar los datos a un servidor de la Universidad de Deusto. Dicha restricción se ha realizado, pero no se ha podido comprobar debido a que se escapa del alcance de este proyecto. La librería LibGDX proporciona opciones de comunicación internet; no obstante, al no poder comprobar si el envío de los datos se realizaba correctamente no se ha podido saber si la propia librería proporciona la comunicación o si por el contrario se deberían agregar permisos de internet en el *AndroidManifest*. En caso de que se necesitaran permisos de internet se debe agregar lo siguiente en el *AndroidManifest*: [25]

```
"<uses-permission android:name="android.permission.INTERNET" />"
```

```
"<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />"
```

6.2 VALORACIÓN ACADÉMICA

Este es un proyecto con necesidades y problemáticas reales, que supone además de una gran responsabilidad, un gran éxito.

El cliente que como se ha comentado previamente, es el director del proyecto, desde un primer momento tenía claras sus necesidades en relación con la librería que usar y las versiones que se debían realizar. Esto último a resultado en pocas, casi inexistentes, modificaciones en los requisitos del proyecto, debido a que al existir el juego de cartas previamente a la realización de este proyecto, las necesidades estaban bastante marcadas.

Las dificultades técnicas que han ido surgiendo no han sido un impedimento para el desarrollo del proyecto, que se han resuelto a base de dedicación y han supuesto un aprendizaje respecto a nuevas tecnologías.

A nivel académico la realización de este proyecto ha implicado la integración de conocimientos adquiridos en diferentes asignaturas del Grado en Ingeniería Informática durante el transcurso de los años.

Por un lado, las asignaturas de programación relacionadas con el lenguaje Java han sido fundamentales, ya que a pesar de usar la librería LibGDX ha sido Java el lenguaje de programación que se ha usado.

Por otro lado, la asignatura de Requisitos del Software ha sido fundamental para la obtención y análisis de los requisitos. En dicha asignatura se hizo uso el programa Visual Paradigm que posteriormente se ha usado en este proyecto.

Otra asignatura que ha sido de gran ayuda ha sido Proceso de Software y Calidad. En dicha asignatura se enseñó a usar metodologías para el desarrollo, a hacer diagramas de Gantt, se hizo uso de XAMPP, herramientas que posteriormente se han usado en este proyecto.

Destacar que en la gran mayoría de asignaturas relacionadas con la programación se hicieron uso de herramientas como Git y GitHub.

Finalmente se ha evitado recopilar cualquier tipo de información relacionada con los usuarios manteniendo la privacidad de los usuarios. Conceptos relacionados con la seguridad, están presentes en la asignatura de Seguridad de la Información.

6.3 TRABAJO FUTURO

Al finalizar el proyecto, como se ha mencionado previamente, se ha cumplido con todos los objetivos y requisitos establecidos inicialmente. No obstante, existen ciertas mejoras que podrían aplicarse en una segunda versión del proyecto.

Las cartas en su versión de mesa poseen un código QR asignado a cada carta; por lo tanto, una mejora para el futuro podría ser implementar la forma de leer mediante la aplicación los códigos QR de las cartas para añadirlas directamente, simplificando la labor del usuario.

Para finalizar, la aplicación al ser un soporte para un juego educativo, se podrían añadir tutoriales para usuarios con poca experiencia para que aprendan el funcionamiento de una manera más práctica.

7. BIBLIOGRAFÍA

- [1] “Blockchain Demo - Anders Brownworth”, <https://andersbrownworth.com/blockchain/blockchain>, (consultado 31/01/2022).
- [2] “Blockchain Demo - A visual demo of blockchain technology”, <https://blockchaindemo.io/>, (consultado 31/01/2022).
- [3] “Robot turtles”, <https://www.thinkfun.com/products/robot-turtles/>, (consultado 31/01/2022).
- [4] “Code Master”, <https://www.thinkfun.es/products/code-master/>, (consultado 31/01/2022).
- [5] “Code: On the Brink”, <https://www.thinkfun.com/products/on-the-brink/>, (consultado 31/01/2022).
- [6] “Code: Rover Control”, <https://www.thinkfun.com/products/rover-control/>, (consultado 31/01/2022).
- [7] “Code: Robot Repair”, <https://www.thinkfun.com/products/robot-repair/>, (consultado 31/01/2022).
- [8] “MOON”, <https://compus.deusto.es/es/moon-1110011/>, (consultado 31/01/2022).
- [9] “Pensamiento computacional”, https://es.wikipedia.org/wiki/Pensamiento_computacional#:~:text=El%20pensamiento%20computacional%20se%20define,a%20problemas%20de%20distinta%20%C3%ADndole., (consultado 31/01/2022).
- [10] Miguel Zapata-Ros, “Pensamiento computacional desenchufado”, *Education in the Knowledge Society*, 2019 (consultado 31/01/2022).
- [11] “Modelo en cascada”, https://es.wikipedia.org/wiki/Desarrollo_en_cascada, (consultado 31/01/2022).
- [12] “LibGDX”, <https://es.wikipedia.org/wiki/LibGDX>, (consultado 31/01/2022).
- [13] “IntelliJ IDEA, Eclipse, NetBeans, Visual Studio Code, Android Studio”, <https://libgdx.com/wiki/start/setup>, (consultado 31/01/2022).
- [14] “SlickEdit”, <https://en.wikipedia.org/wiki/SlickEdit>, (consultado 31/01/2022).
- [15] “Portable WebServer, WAMPServer, XAMPP”, <https://www.emezeta.com/articulos/15-aplicaciones-para-montar-servidores-web-en-local>, (consultado 31/01/2022).
- [16] “Apache”, https://es.wikipedia.org/wiki/Servidor_HTTP_Apache, (consultado 31/01/2022).
- [17] “Git”, <https://es.wikipedia.org/wiki/Git>, (consultado 31/01/2022).
- [18] “GitHub”, <https://es.wikipedia.org/wiki/GitHub>, (consultado 31/01/2022).
- [19] “GitHub Desktop”, <https://docs.github.com/en/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>, (consultado 31/01/2022).
- [20] “Visual Paradigm”, https://en.wikipedia.org/wiki/Visual_Paradigm, (consultado 31/01/2022).
- [21] “Team Gantt”, <https://www.appvizer.es/organizacion-planificacion/gestion-proyectos/teamgantt>, (consultado 31/01/2022).
- [22] “Arquitectura monolítica”, https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_monol%C3%ADtica, (consultado 10/02/2022).

- [23] "Firma de aplicación Android Studio", <https://developer.android.com/studio/publish/app-signing?hl=es-419>, (consultado 20/02/2022).
- [24] "Compilación web en LibGDX", <https://happycoding.io/tutorials/libgdx/setup#html-project>, (consultado 20/02/2022).
- [25] "Agregar permisos de internet a AndroidManifest", <https://developer.android.com/training/basics/network-ops/connecting?hl=es-419>, (consultado 20/02/2022).