



Universidad de Valladolid

ESCUELA DE INGENIERÍA INFORMÁTICA

**GRADO EN INGENIERÍA INFORMÁTICA
MENCIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN**

**RefreshMe: Herramienta para el desarrollo cognitivo
en personas mayores**

Alumno: Dimitrov Dilkov, Nikodim
Tutor: Sánchez Mayoral, Pablo

Dedico este trabajo a mis padres que tanto apoyo me han dado durante estos años para seguir adelante en todo momento y que tanto han confiado en mí y en mis posibilidades de lograrlo.

AGRADECIMIENTOS

A mi familia, por todo el apoyo que me han brindado durante todo este tiempo y así poder lograr mis objetivos.

A mis amigos, que siempre han estado ahí aportando su granito de arena en cualquier momento y que han hecho que mi estancia en la universidad haya sido una experiencia única.

También quiero agradecerse a mis tutores Pablo Sánchez Mayoral, Rodrigo Gómez Malmierca y José Carlos Delás Ruiz, ya que con sus constantes revisiones y correcciones tanto de la aplicación como de la memoria han ayudado a que este trabajo vaya por buen camino.

RESUMEN

Con el transcurso del tiempo nos vamos haciendo mayores y queramos o no esto supone una pérdida cognitiva, en ocasiones no muy pronunciada pero otras sí, probablemente debido a factores patológicos. Este proceso puede ser retrasado mediante la ejercitación diaria de las capacidades mentales superiores (memoria, razonamiento, lenguaje, etc..) para así estimularlas y conseguir mantener las conexiones neuronales existentes. Esto puede conseguirse de una manera entretenida y divertida mediante el uso de juegos especializados.

Los videojuegos desde sus inicios han tenido un carácter lúdico, es decir, distraer y divertir a los jugadores. Sin embargo, con la evolución de estos, este carácter lúdico también se vería afectado por las inmensas posibilidades que nos pueden brindar los videojuegos. En nuestro caso pretendemos dotar a estos *serious games* de un carácter “sanador”. Por ende, los emplearemos para estimular la actividad cognitiva del usuario intentando que estas capacidades no se mermen.

Esto se pretende llevar a cabo mediante el desarrollo de tres juegos que pueden ser desplegados tanto en Android, iOS, “desktop” o plataformas web si así se desea, ya que estas son algunas de las posibilidades que nos brinda el framework LibGDX.

Palabras Clave: LibGDX, pérdida cognitiva, estimular, serious games

ABSTRACT

With the passage of time, we get older and whether we like it or not, this implies a cognitive loss, sometimes not very pronounced but others it is, probably due to pathological factors. This process can be delayed through the daily exercise of higher mental capacities (memory, reasoning, language, etc.) in order to stimulate them and manage to maintain the existing neural connections. This can be achieved in a fun and entertaining way through the use of specialized games.

Since their inception, video games have had a playful character which distract and amuse players. However, with the evolution of these, this playful character would also be affected by the immense possibilities that video games can offer us. In our case we intend to provide these video games with a “healing” character. Therefore, we will use them to stimulate the user's cognitive activity, trying not to diminish these capacities.

This is intended to be carried out through the development of three games that can be deployed on Android, iOS, "desktop" or web platforms if desired, since these are some of the possibilities offered by the LibGDX framework.

Keywords: LibGDX, cognitive loss, stimulate, serious games

Índice

1.	Introducción	15
1.1.	Antecedentes y motivaciones	15
1.2.	Objetivos del Trabajo de Fin de Grado.....	16
1.3.	Estructura de la memoria.....	16
2.	Marco del trabajo de fin de grado	19
2.1.	Declive cognitivo natural y demencia	19
2.2.	Serious Games.....	21
2.3.	Serious Games como terapia	24
3.	Planificación	27
3.1.	Recursos humanos y técnicos	27
3.2.	Tecnología software empleada	28
3.2.1.	Android Studio.....	28
3.2.2.	LibGDX.....	29
3.2.3.	Gimp.....	29
3.2.4.	Audacity.....	30
3.2.5.	GitLab	30
3.3.	Estimación temporal	30
3.4.	Identificación de riesgos.....	33
4.	LibGDX.....	37
4.1.	Que es LibGDX.....	37
4.2.	Herramientas auxiliares	38
4.2.1.	Project Setup	38
4.2.2.	Hiero.....	39
4.2.3.	GDX Texture Packer.....	39
4.3.	Estructura LibGDX	41
5.	Desarrollo de la aplicación	45
5.1.	Adquisición de requisitos	45
5.1.1.	Requisitos funcionales.....	46
5.1.2.	Requisitos no funcionales	46
5.2.	Análisis.....	47
5.3.	Diseño de la aplicación.....	48
5.4.	Implementación	54

5.4.1.	Elementos visuales	56
5.4.2.	Versiones	58
5.5.	Pruebas.....	58
6.	Ejecución del proyecto	63
6.1.	Evolución temporal del proyecto	63
6.2.	Costes del proyecto.....	65
6.2.1.	Horas efectivas empleadas y tasas horarias del personal.....	66
6.2.2.	Amortizaciones del equipo informático	69
6.2.3.	Costes del material consumible	70
6.2.4.	Costes indirectos	70
6.2.5.	Costes totales del proyecto.....	71
7.	Conclusiones.....	73
	Referencias.....	75
	Apéndices	77
	Apéndice A: Manual de Instalación.....	77
	Apéndice B: Manual de Usuario	83
	Apéndice C: Generación de la dificultad de resultados en Maths	87
	Apéndice D: Contenido del CD	91

Índice de Tablas

Tabla 3.1: Número de personas por rol	27
Tabla 3.2: Fases del proyecto y tareas correspondientes	31
Tabla 3.3: Evaluación y clasificación de riesgos	34
Tabla 3.4: Desglose de las probabilidades de ocurrencia de un riesgo	35
Tabla 3.5: Desglose del nivel de impacto	35
Tabla 3.6: Clasificación de los riesgos según su probabilidad e impacto	35
Tabla 5.1: Descripción del caso de uso: seleccionar juego	47
Tabla 5.2: Descripción del caso de uso: Cerrar aplicación	47
Tabla 6.1: Calculo días/horas de cada fase del proyecto.....	64
Tabla 6.2: Desglose de los días y horas de trabajo anuales.....	66
Tabla 6.3: Semanas anuales de trabajo	67
Tabla 6.4: Sueldos medios de cada rol.....	67
Tabla 6.5: Coste mensual y por hora de cada rol	67
Tabla 6.6: Desglose de las horas por etapa y rol.....	68
Tabla 6.7: Costes totales de la mano de obra	69
Tabla 6.8: Coste del material informático.....	70
Tabla 6.9: Coste del material consumible	70
Tabla 6.10: Costes indirectos	71
Tabla 6.11: Costes totales del proyecto.....	71
Tabla C.1: Tamaño de los operandos A y B generados para cada operación	87
Tabla C.2: Generación de resultados para la operación suma	88
Tabla C.3: Generación de resultados para la operación la operación resta.....	88
Tabla C.4: Generación de resultados para la operación multiplicación.....	89

Índice de Figuras

Figura 2.1: Número de personas afectadas por Alzheimer	21
Figura 2.2: Gráfico de características de un "serious game"	22
Figura 2.3: Ejemplo de juego edutainment	23
Figura 2.4: Capturas de pantalla de Sea Hero Quest.....	25
Figura 3.1: Calendarización inicial de las tareas.....	32
Figura 3.2: Descomposición de las tareas en un diagrama de Gantt.....	32
Figura 4.1: Captura de pantalla del generador de proyectos de LibGDX.....	38
Figura 4.2: Captura de pantalla de la herramienta Hero	39
Figura 4.3: Captura de pantalla de la herramienta GDX Texture Packer	40
Figura 4.4: Spritesheet generado con GDX Texture Packer	40
Figura 4.5: Diagrama de módulos de LibGDX.....	41
Figura 4.6: Diagrama del ciclo de vida de una aplicación en LibGDX	43
Figura 5.1: Diagrama de casos de uso	48
Figura 5.2: Diagrama de actividades Android	48
Figura 5.3: Diagrama de secuencia de selección de juego.....	49
Figura 5.4: Diagrama de secuencia de MatchPairs	50
Figura 5.5: Diagrama de secuencia de Colors.....	51
Figura 5.6: Diagrama de secuencia de Maths	52
Figura 5.7: Diagrama de clases de la aplicación.....	53
Figura 5.8: Diagrama de clases que representa la relación con el AssetManager	54
Figura 5.9: Captura de pantalla de MatchPairs	54
Figura 5.10: Captura de pantalla de Colors.....	55
Figura 5.11: Captura de pantalla de Maths	56
Figura 5.12: Primer fondo de pantalla	56
Figura 5.13: Figuras de los juegos MatchPairs y Colors	57
Figura 5.14: Botones del menú de Game Over.....	57
Figura 5.15: Textura donde se colocan la puntuación	57
Figura 6.1: Calendarización real de las tareas.....	64
Figura 6.2: Descomposición de las tareas en un diagrama de Gantt.....	65
Figura 6.3: Descomposición de las tareas en función de las horas trabajadas por cada rol	68
Figura A.1: Sistema de ficheros de Android	77
Figura A.2: Fichero APK de instalación	77
Figura A.3: Opción del menú de ajustes	78
Figura A.4: Opción para aplicaciones desconocidas	78
Figura A.5: Fuentes desde donde se permiten instalar aplicaciones desconocidas.....	79
Figura A.6: Permiso de instalación de aplicaciones desconocidas	79
Figura A.7: Icono de localización de ficheros APK disponibles	80
Figura A.8: Archivo APK de instalación.....	80
Figura A.9: Pestaña de confirmación instalación.....	80
Figura A.10: Pestaña de confirmación de instalación de aplicación desconocida	81
Figura A.11: Pestaña de confirmación de instalación.....	81
Figura A.12: Pestaña de confirmación de análisis de seguridad	81
Figura B.1: Icono estándar de aplicación LibGDX.....	83

Figura B.2: Captura del menú de acceso.....	83
Figura B.3: Figuras no parejas	84
Figura B.4: Figuras parejas	84
Figura B.5: Secuencia de figuras inicial en Colors	85
Figura B.6: Ejemplo de operación aritmética en Maths.....	85

Capítulo I

INTRODUCCIÓN

En este capítulo se realizará una breve introducción al contenido de este proyecto en la cual se hablará de las motivaciones y los objetivos que se desean alcanzar con la realización de este trabajo. Con esto se pretende que el lector consiga tener una visión global del proyecto y de la estructura de la memoria.

1.1. Antecedentes y motivaciones

Los antecedentes en este campo suelen deberse a trabajos realizados por personas que se dedican a la medicina y a la informática. En ambos casos el objetivo es el mismo, implementar una serie de juegos para así estimular las funciones mentales de las personas mayores y de esta manera conseguir mantener o mejorar su estado cognitivo.

Cabe mencionar que tales objetivos son mucho más complicados o casi imposibles de conseguir si se da el caso de que la persona se vea afectada por alguna enfermedad neurodegenerativa como algún tipo de demencia. En estos casos, la pérdida de las capacidades mentales se ven afectadas gravemente y van empeorando con el avance de la enfermedad siendo casi imposible obtener resultados notorios.

Este trabajo de fin de grado se enmarca en la colaboración entre el observatorio HP y la Escuela de Ingeniería Informática de Valladolid. El programa HP SCDS se lleva desarrollando durante varios años con las universidades españolas y este es uno de los proyectos propuestos en cooperación con la Cruz Roja de León.

También hay que mencionar el trasfondo social que posee, ya que la intención de esta aplicación es intentar ayudar y entretener a las personas mayores mediante el uso de videojuegos.

1.2. Objetivos del Trabajo de Fin de Grado

El principal objetivo que se persigue con la realización de este proyecto es implementar uno o varios juegos que permitan la estimulación cognitiva del público objetivo.

Lo que se quiere lograr con estos juegos es estimular las capacidades mentales de los usuarios, para ello se crearán juegos para ejercitar la memoria. Se trabajará este aspecto mediante la memorización espacial de diferentes figuras y también mediante la memorización de una secuencia de figuras acompañadas de sonidos diferentes para cada una de ellas. Lo que se pretende conseguir con los sonidos es estimular más aún la capacidad de memorizar. Y finalmente, se trabajarán aspectos como el razonamiento y el cálculo mediante la realización de operaciones aritméticas.

Los objetivos formativos son aprender a usar la librería LibGDX, comprender la lógica de programación de videojuegos, entender las estructuras de Android y emplear los conocimientos adquiridos a lo largo del grado para realizar el proyecto.

1.3. Estructura de la memoria

En el segundo capítulo situaremos este trabajo en contexto, es decir, se realizará una explicación de cómo y porqué se produce las pérdidas cognitivas tanto normales como las debidas a enfermedades. También se realizará una breve introducción a los serious games y de su papel dentro de este ámbito.

En el tercer capítulo se explicará la planificación que se ha llevado a cabo, una breve descripción de las tecnologías que se piensan emplear y, una estimación temporal junto con una evaluación de los posibles riesgos que puedan aparecer en el transcurso del trabajo.

En el cuarto capítulo se hace una presentación de lo que es la librería LibGDX la cual ha sido empleada para la realización de los videojuegos. Primero se hará una introducción a algunas herramientas importantes y para posteriormente acabar hablando de su modelo de dominio y de las clases más importantes que intervienen en el ciclo de vida.

En el quinto capítulo de este trabajo, explicaremos los pasos llevados a cabo para crear estos juegos, los requisitos necesarios planteados durante la etapa inicial, el diseño, la implementación y una justificación de porque se han empleado las tecnologías citadas en el apartado anterior.

En el sexto capítulo se hablará de la ejecución real que ha supuesto el proyecto y las desviaciones que ha habido respecto al plan inicial. También se podrá ver una estimación de los costes que supondría realizar un proyecto como este en una empresa real.

Y finalmente, en el séptimo capítulo se plasman las conclusiones obtenidas durante la realización del trabajo de fin de grado.

Capítulo II

MARCO DEL TRABAJO DE FIN DE GRADO

En este segundo capítulo del trabajo de fin de grado, se situará a este dentro de contexto que nos ayudarán a comprender mejor los fundamentos de este proyecto.

2.1. Declive cognitivo natural y demencia

El envejecimiento es un proceso normal y natural de cualquier ser vivo, este proceso se ve acompañado de una serie de cambios físicos, mentales y emocionales. Estos cambios suelen manifestarse en forma de un deterioro visual y/o auditivo, ligeras pérdidas de memoria, faltas de atención y torpeza a la hora de expresarse .

Estos síntomas pueden hacer sentir a la persona incómoda, torpe o incluso inútil, pudiéndose así generar sentimientos depresivos ya que estas cosas no les ocurrían cuando eran jóvenes. [1]

Con el párrafo anterior no se quiere dar a entender que por envejecer se pierden totalmente las capacidades cognitivas ni mucho menos, ya que hay personas que a pesar de su avanzada edad siguen gozando de lucidez unas buenas capacidades mentales.

Según estudios estadísticos [2] también es cierto que una vez pasados los 65 años aumenta la probabilidad de poder sufrir algún tipo de enfermedad neurodegenerativa como una demencia, siendo la más común de estas la enfermedad del Alzheimer.

En ausencia de este tipo de enfermedades, las capacidades cognitivas se verán afectadas, pero en mucho menor grado. Esta afirmación se debe a que el cerebro es el órgano más afectado por el transcurso del tiempo y esto se ve acompañado de una pérdida celular lo que se traduce como una pérdida neuronal y, por ende, una reducción de las conexiones existentes entre las mismas. Estos cambios naturales (no patológicos) no suponen una limitación o discapacidad de la persona para la realización de las tareas cotidianas. [3]

Una de las causas de este deterioro cognitivo (no patológico) es el aislamiento de los propios ancianos ya que al no tener contacto con otras personas provoca en ellos una dejadez y una falta de interés generalizada, lo cual repercute en sus capacidades[4].

El estrés y la depresión aumentan el desgaste neuronal afectando directamente a sus facultades mentales. Estos estados depresivos dan lugar a que las personas dejen de hacer o incluso de intentar hacer cosas que antes sí que podían[4].

Según el CDC (Centro para el Control y Prevención de Enfermedades)[5], demencia es un término que se emplea para denominar las alteraciones de las capacidades cognitivas de las personas. Estas alteraciones, son la dificultad o imposibilidad de recordar, pensar o tomar decisiones, las cuales suponen un inconveniente en la realización de las actividades cotidianas. La demencia afecta principalmente a los adultos mayores, pero no forma parte del proceso de envejecimiento normal.

Existen diferentes tipos de demencia, las cuales podemos agrupar en tres grandes tipos: degenerativas primarias, vasculares y secundarias.[6]

- Las demencias degenerativas primarias son de origen desconocido y a día de hoy son irreversibles. Dentro de este grupo encontramos el Alzheimer, los cuerpos de Lewy o la enfermedad de Parkinson.
- Las demencias vasculares se deben a accidentes cerebrovasculares y a bloqueos sanguíneos. En estas, si se consigue eliminar la causa, la degeneración de las funciones neuronales se retrasa o bien se evita.
- Las demencias secundarias se producen debido a trastornos neurológicos no degenerativos. Suelen tener baja incidencia y son reversibles si se tratan a tiempo.

Estas enfermedades neurodegenerativas se suelen diagnosticar mediante la realización de una serie de pruebas cognitivas y neurológicas, pero que en algunos casos la sintomatología presente puede dar lugar a confusión con otras patologías produciéndose así un diagnóstico erróneo.

En ocasiones, el deterioro cognitivo no es muy pronunciado y se diagnostica como un deterioro cognitivo leve. Este se presenta con pérdida de memoria, pérdida de concentración y ciertos problemas en el habla, pero que en general estos no suponen una

imposibilitación para la realización de las tareas cotidianas. Aunque este sea el diagnóstico es aconsejable mantener un seguimiento para ver si los síntomas empeoran ya que entre el 15 y el 41%[1] de estos casos acaban formando parte de la etapa temprana de una enfermedad neurodegenerativa.

Según datos del SEN (Sociedad Española de Neurología)[7], en España en 2018 los afectados por Alzheimer, que no por demencia, alcanzan el número de 800.000 personas. El coste que generan los tratamientos supone el 1,5% del producto interior bruto nacional. Para ser más concretos, un gasto de 28.200 € anuales por persona en sus etapas iniciales, mientras que cuando su estado empeora el gasto aumenta hasta los 41.700 € anuales.

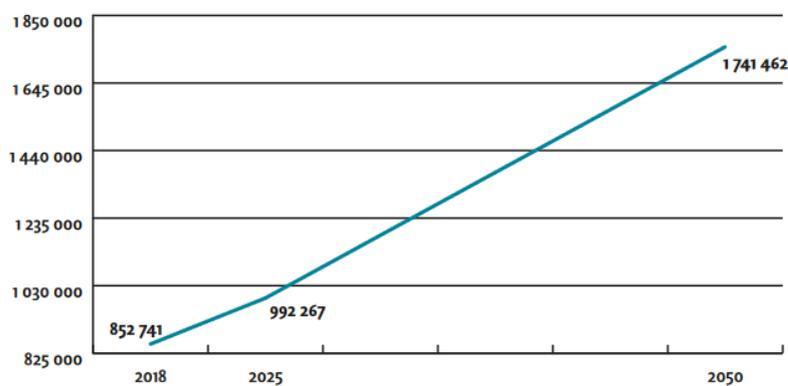


Figura 2.1: Número de personas afectadas por Alzheimer

En la figura anterior, 2.1 [8], podemos observar las cifras estimadas del crecimiento de las personas afectadas por la enfermedad del Alzheimer desde 2018 hasta el 2050. Como se ve, se estima un aumento en más del doble respecto a 2018, lo cual son unas cifras muy preocupantes tanto socialmente como económicamente hablando.

2.2. Serious Games

Los *serious games* no son como los videojuegos normales en lo que a objetivo se refiere, ya que no tienen el aspecto lúdico como objetivo principal sino más bien un carácter formativo donde el usuario debe aprender cosas útiles sobre cierto tema.

Citando a Squire y Jenkins (Corti,2006), los *serious games* se emplean para tener un “impacto” en sus usuarios y ayudar a que estos desarrollen ciertas habilidades o conocimientos poniéndolos en situaciones que en la vida real sería imposible debido a razones de seguridad, de coste o de tiempo.



Figura 2.2: Gráfico de características de un "serious game"

Estos tipos de juegos comienzan a ser cada vez más comunes y su abanico de categorías es muy amplio, se pueden aplicar tanto al mundo de los negocios, a lo militar, a la educación o a la sanidad por citar algunos ámbitos.

Desde los puntos de vista de varios autores hay discrepancias en lo que se refiere a la definición de un serious game, para algunos son juegos donde lo primordial es conseguir un objetivo, como aprender algo, y la diversión es opcional, mientras que otros dicen que no, que la diversión ni siquiera debe estar presente ya que distrae al usuario de su objetivo principal. En esta última corriente de pensamiento incluso dicen que en el desarrollo de los serious games deben intervenir las tecnologías 3D de los videojuegos, pero no la teoría de diseño de estos.[9]

Pero, aunque discrepen en cierta medida, están de acuerdo en la idea de que los serious games no tienen como objetivo principal la diversión sino otros como son la educación, el entrenamiento o la salud, por ejemplo.

Los edutainment fueron famosos en los años 90 con el auge del contenido multimedia, lo que pretendían estos juegos era entretener y enseñar cierto conocimiento. El grupo objetivo eran los niños de preescolar, haciéndose hincapié en la lectura, las matemáticas o la ciencia. El mayor problema que tenían es que eran descritos por los niños como juegos muy aburridos. Se caracterizaban porque su manera de aprendizaje era muy repetitiva y monótona (van Eck, 2006) [9]. Finalmente, estos juegos dejaron de hacerse porque no resultaban rentables.

Los juegos gubernamentales, son juegos donde se intentan simular políticas interiores o exteriores, o bien juegos donde te hacen una serie de preguntas para comprobar tu conocimiento acerca de las políticas de un país, por ejemplo. En otros juegos se intenta simular la gestión de presupuestos, la gestión de ciudades o la gestión de políticas de salud, etc.



Figura 2.3: Ejemplo de juego edutainment

En la categoría de los serious games del ámbito de la simulación y estrategia militar, podemos hablar de uno de los proyectos más interesantes llevados a cabo por el ejército estadounidense, *America 's Army: Proving Grounds*". Se trata de un FPS (first-person-shooter) que les permitió ahorrarse una gran cantidad de dinero en campañas de reclutamiento. Este proyecto lo describen como una plataforma para la formación de comunicación estratégica y de reclutamiento.

Esta plataforma se cerró el 5 de mayo de 2022 ya que, para el gobierno, este juego "de momento" ha cumplido con su misión, y decimos que es de momento, porque en su página oficial anuncian futuras entregas [9, 10]. Existen otros juegos de simulación militar mucho más realistas, en estos se pueden controlar vehículos aéreos, terrestres o marinos, además la acción se desarrolla en mapas mucho más extensos de manera que las opciones estratégicas y tácticas son tantas como podamos imaginar. La única diferencia respecto al título anterior es que estos no están financiados directamente por el gobierno de un país como es el caso del título anterior.

Otra categoría perteneciente al género de los serious games son los juegos corporativos, son juegos creados por empresas y estos les sirven para entrenar a sus empleados en determinadas tareas sin necesidad de tener que disponer a otros empleados para que los formen en tales tareas. Además, resulta ser la manera fácil de llegar a un público cada vez más metido en el mundo de los videojuegos. Estos juegos corporativos se consideran como una herramienta útil cuando la materia a enseñar es aburrida o demasiado técnica.

Se ha demostrado que los videojuegos desarrollan habilidades en los jugadores como son el pensamiento estratégico, la planificación, la comunicación, la colaboración, la toma de decisiones y la negociación.

En Estados Unidos, por ejemplo, se están realizando inversiones de capital por diferentes entidades para el desarrollo y estudio de los serious games. Algunos de los proyectos que actualmente se están construyendo para estas empresas consisten en: juegos de simulaciones de contención de fugas de gases tóxicos, juegos de rescate de rehenes detrás de las líneas enemigas o juegos de instrucción de bomberos en el apagado de efectivo de incendios.

En general, casi todos los países occidentales más desarrollados están invirtiendo en diferentes programas de desarrollo de serious games.

2.3. Serious Games como terapia

Sea Hero Quest es un “serious game” con el cual los investigadores pretenden recoger información para ver de qué manera se ven afectados los sentidos de la memoria espacial y de la orientación tanto en personas sanas como en personas afectadas por Alzheimer. Esto es importante ya que estos dos sentidos suelen ser unos de los primeros damnificados por la enfermedad, por lo tanto, si se consigue una rápida detección de este factor es probable que se detecte el Alzheimer en fases mucho más tempranas.

Se trata de un videojuego sencillo donde debemos orientarnos por el mapa para surcar los mares de inicio a fin. Al cabo de varios niveles llegaremos a una última etapa donde deberemos fotografiar al monstruo que habita esa zona. Para poder orientarnos por los diferentes mares, al inicio de cada ronda se muestra el mapa con las posibles rutas que se pueden seguir y deberemos memorizarlas durante el breve periodo durante el que se muestran. Estos mapas pueden resultar fáciles de recordar para una persona sana, mientras que para un enfermo de Alzheimer puede suponer un reto.

Actualmente, está en la Play Store y se puede descargar de forma gratuita, ya que los desarrolladores están buscando que lo prueben el mayor número de personas posible, independientemente de si padecen Alzheimer o no, para así poder obtener una muestra de datos con la que poder trabajar.

Lo que pretenden los juegos destinados a la estimulación cognitiva es estimular estas mismas de manera que la pérdida neuronal sea más lenta para que así el deterioro cognitivo no sea tan notorio. Esta estimulación se consigue intentando que los usuarios se piensen el siguiente paso a dar en vez de reaccionar de manera espontánea. Con lo cual juegos como puzles, palabras encadenadas, “scrabble”, cálculos matemáticos o recordar secuencias de figuras/colores y/o sonidos son muy adecuados para tal tarea. Cada uno de estos juegos estimulan diferentes capacidades cognitivas como la memoria espacial, la memoria secuencial o la capacidad de razonamiento.

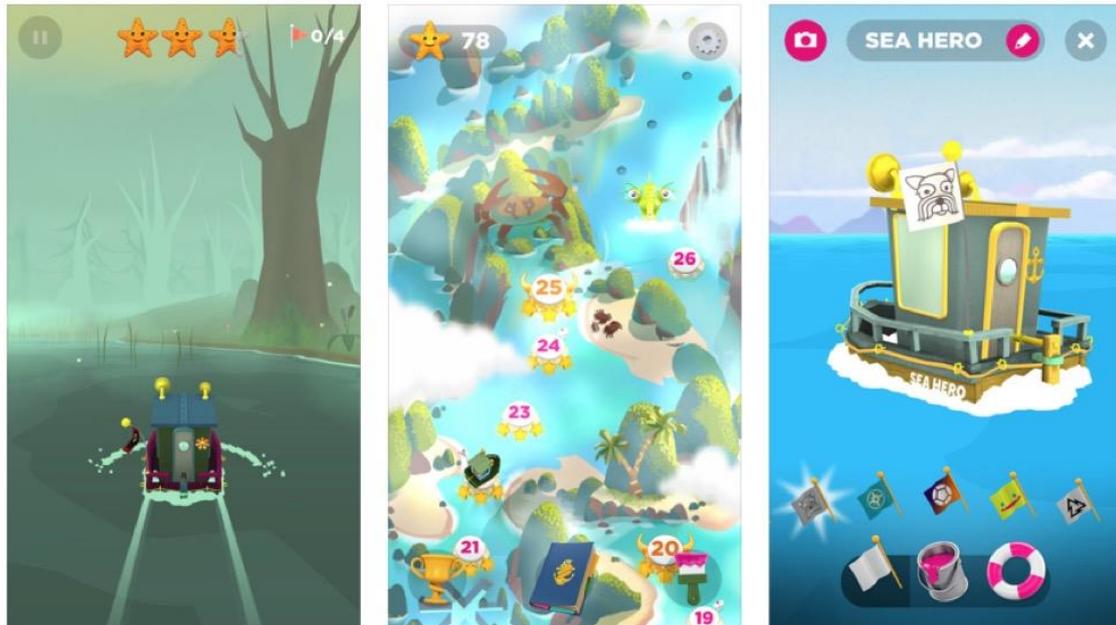


Figura 2.4: Capturas de pantalla de Sea Hero Quest

Una manera incluso más beneficiosa de ejercitar la mente es hacerlo en grupo, socializar es una actividad fundamental para las personas mayores, ya que esto ayuda a hacer frente a las depresiones y ansiedades que pueden padecer los ancianos. Estos factores negativos también repercuten en las capacidades cognitivas de los que los sufren.

Las actividades que suponen un mayor reto a los ancianos suelen ser tareas que requieren de atención dividida o tareas que requieran de recomponer una secuencia de números o letras en orden inverso. Por ello, muchos juegos de memoria intentan trabajar estos dos aspectos con diferentes juegos.

Los juegos de estimulación cognitiva no sólo se emplean en personas mayores sino también en personas que hayan sufrido daños cerebrales. Con estos juegos se pretende mejorar sus habilidades intelectuales y para ello se adaptan a las necesidades específicas de cada persona.

Uno de los juegos más “recientes” y más conocidos en cuanto a estimulación cognitiva es *Brain Training* de Nintendo. Este título se emplea con bastante frecuencia en el ámbito médico en rehabilitaciones y como tratamiento adicional en enfermedades como puede ser la esclerosis múltiple, donde los pacientes presentan buenos resultados tras las sesiones de juego. También se han realizado estudios[11] con personas mayores en los que se han presentado mejoras en los tiempos de respuesta y en las funciones ejecutivas.

Capítulo III

PLANIFICACIÓN

Este capítulo se dedica a explicar la estimación inicial realizada del proyecto, así como las tecnologías seleccionadas y los posibles riesgos que pueden surgir durante la realización del proyecto.

3.1. Recursos humanos y técnicos

A continuación, presentamos los posibles roles que intervendrían en un proyecto real llevado a cabo por una pequeña empresa que consta de 5 personas, en realidad cada uno de estos roles será llevado a cabo por el alumno.

Rol	Número de personas
Jefe de proyecto / Programador Senior	1
Programador Intermediate	1
Programador Junior	1
Administrativo	1

Tabla 3.1: Número de personas por rol

- **Jefe del Proyecto:** Máximo responsable del proyecto, encargado de proporcionar ideas, coordinar, aconsejar y guiar al resto del equipo durante el desarrollo. También es el encargado de estar en contacto con el cliente y conseguir que se cumplan todas sus expectativas. Para este puesto se necesita

una persona con gran experiencia, con unos extensos conocimientos técnicos y con una buena capacidad de liderazgo.

- **Programador Intermediario:** Ingeniero con bastante experiencia en el sector, profesional cualificado que entiende los requisitos del proyecto. Éste, aparte de programar también se encargará de proporcionar apoyo y ayuda a los programadores menos experimentados. También realizará ciertas tareas de documentación referentes al código escrito por él.
- **Programador Junior:** Ingeniero poco experimentado, encargado de realizar las tareas de programación menos complejas del proyecto. También se encargará de hacer la documentación realizando una búsqueda de información, estructuración y redacción de esta.

Respecto a los recursos técnicos necesarios para elaborar la tarea se necesitan ordenadores, en este caso portátiles, con capacidades computacionales suficientes para ejecutar los programas necesarios, específicamente Android Studio ya que es el que más demanda de recursos tiene. En este caso, serán necesarios un portátil por empleado acompañado de un ratón externo para mayor comodidad de trabajo.

Se considera oportuno disponer de una impresora para que el administrativo pueda hacer la documentación necesaria.

También es necesario disponer de una infraestructura con conexión a internet que permita conectar con las tecnologías de control de versiones y con los servicios de búsqueda de información.

3.2. Tecnología software empleada

En este apartado se hará una breve explicación de las diferentes tecnologías que se han empleado para la realización de este proyecto de fin de carrera. Aunque se hayan empleado estas herramientas no quiere decir que no haya otras disponibles en el mercado, pero estas son de uso gratuito y no se han tenido que pagar licencias para su uso, lo cual supone una gran ventaja y una razón de mucho peso para seleccionarlas.

3.2.1. Android Studio

Este es el entorno de desarrollo (IDE) empleado para programar la aplicación. Esta plataforma es desarrollada por IntelliJ IDEA. Entorno muy completo ya que dispone de una gran cantidad de herramientas. Está disponible para Windows, macOS y Linux.

Permite escribir código Android tanto en Kotlin como en Java y dispone de herramientas de autocompletado para ambos.

La última versión estable disponible actualmente es la Chipmunk (v.2021.2.1), la cual pesa 929 MB [12]. Como bien se ve es un IDE bastante pesado y que además tiene un consumo de recursos considerable cuando lanzamos alguna simulación de un terminal móvil.

La elección de esta herramienta en vez de otros editores de código Java se debe a que como bien se dice en el apartado anterior, nos permite simular terminales móviles en los que poder probar el funcionamiento de nuestra aplicación. Se puede simular casi cualquier funcionalidad de un terminal real, llamadas, mensajes de texto, especificación de ubicación de dispositivo, Play Store, etc.

Aparte de los terminales móviles también se pueden emular tabletas Android, Wear OS y dispositivos Android TV.

3.2.2. LibGDX

Framework basado en Java que hace uso de la tecnología OpenGL [13] para el desarrollo de videojuegos en Android, este nos proporciona las librerías necesarias para una codificación más sencilla. También nos proporciona las herramientas necesarias para la gestión de la mayoría de assets, pero no para su creación. No se procede a realizar una mayor explicación de esta tecnología ya que se le dedica el capítulo IV.

3.2.3. Gimp

Herramienta similar a Photoshop, pero con la ventaja de que es un programa libre y gratuito. Esta ha sido utilizada para el tratamiento de las distintas imágenes que se han empleado durante el desarrollo del proyecto. Este programa fue inicialmente creado para ser usado exclusivamente en Linux, pero posteriormente paso a ser multiplataforma.

Se trata de un programa muy flexible que nos brinda gran cantidad de posibilidades a la hora de realizar operaciones sobre cualquier formato de imagen. Al igual que en Photoshop podemos dividir una imagen en diferentes capas y en cada una de estas realizar las tareas que deseemos.

3.2.4. Audacity

Programa multiplataforma también libre y gratuito, empleado para la grabación y edición de audio. Éste proporciona ciertas posibilidades como pueden ser: grabar audio de otras aplicaciones, programar grabaciones de audio del entorno, reproducir cualquier pista con la posibilidad de poder cambiar la velocidad de reproducción y cortar, copiar y pegar trozos de pistas[14]. También sirve para mezclar diferentes pistas de audio, añadir efectos de sonido y exportar archivos a diferentes formatos.

En este proyecto, Audacity se ha empleado para recortar los efectos de sonidos para así cuadrarlos de la mejor manera posibles con las animaciones.

3.2.5. GitLab

Plataforma de control de versiones donde podemos subir código y compartirlo con compañeros de trabajo o clientes. Este tipo de servicio nos permite tener la seguridad de que no vamos a perder el código escrito siempre y cuando el usuario se acuerde de subirlo. Permite tener un control sobre las respectivas versiones y actualizaciones realizadas sobre el proyecto por cada uno de los participantes. Esta plataforma se puede emplear indistintamente del lenguaje de programación empleado.

GitLab también proporciona funcionalidad para “mezclar” código de diferentes usuarios del proyecto. Permite mantener control de versiones, podemos volver a una versión anterior si así lo desea el usuario, esto se realiza a través del ver registro de actualizaciones. Esta herramienta también permite importar un proyecto ya existente y trabajar con ramas distintas que pueden corresponder a módulos diferentes del proyecto.

Esta ha sido la plataforma elegida para mantener el control de versiones del proyecto. No ha sido necesaria la utilización de software externo para poder usar la plataforma ya que el propio Android Studio nos brinda esta posibilidad tras una pequeña configuración de parámetros.

3.3. Estimación temporal

Se plantea iniciar el proyecto para el 26 de abril de 2021 con lo cual en esa fecha se abordará la primera fase del proyecto.

El proyecto se dividirá en cuatro fases: puesta en marcha, recopilación de información, desarrollo del proyecto y finalmente la elaboración de la memoria. A continuación, se explican las tareas que se realizarán en cada una de las etapas en sus correspondientes tablas:

Fase	Duración	Descripción
Puesta en Marcha	6 semanas y media	Selección de las tecnologías a emplear en el desarrollo, reparto de tareas y planificación del desarrollo del trabajo.
Recopilación de información	8 semanas	Búsqueda en todo tipo de fuentes de la información necesaria para poder proceder con el trabajo de manera adecuada.
Desarrollo del proyecto	19 semanas	Análisis, diseño, codificación del proyecto y aprendizaje de las tecnologías necesarias para llevar a cabo la tarea. También se realiza una batería de pruebas para comprobar el correcto funcionamiento de los juegos.
Elaboración de la memoria	5 semanas	Redacción completa de la documentación del proyecto y de la memoria donde se incluirá toda la información.

Tabla 3.2: Fases del proyecto y tareas correspondientes

La situación personal actual del programador imposibilita el desarrollo diario y la jornada de trabajo no se puede precisar con exactitud y se dedicará el máximo tiempo posible. Se realiza una aproximación en función de una jornada de unas 2 horas.

En función de la fecha de inicio y de las horas que se van a dedicar, se ha elaborado la siguiente planificación temporal del desarrollo del trabajo de fin de grado. Como podemos ver, la calendarización está compuesta por las cuatro fases planificadas y cada una de ellas se compone de una serie de tareas que deben ser completadas.

A partir de esta división del trabajo, se ha podido realizar el diagrama de Gantt de la figura 3.2 donde se puede observar la distribución temporal de cada tarea de manera más clara.

Etapas		Duración Estimada en días	Fecha de Inicio	Fecha de Finalización
Puesta en Marcha				
	Propuesta	1	26/04/2021	26/04/2021
	Elaboración de requisitos	20	27/04/2021	26/05/2021
	Selección de tecnologías	1	27/05/2021	27/05/2021
	Configuración de tecnologías	1	28/05/2021	28/05/2021
	Planificación inicial	10	31/05/2021	11/06/2021
Recopilación de información				
	LibGDX	15	14/06/2021	02/07/2021
	Android	15	05/07/2021	23/07/2021
	Android Studio	2	26/07/2021	27/07/2021
	Selección de assets	3	28/07/2021	30/07/2021
	GDX Texture Packer	1	01/09/2021	01/09/2021
	Patrones empleados en videojuegos	5	02/09/2021	08/09/2021
Desarrollo del proyecto				
	Diseño de las apps	3	09/09/2021	13/09/2021
	Programación juego n°1	30	14/09/2021	26/10/2021
	Documentación del código	1	27/10/2021	27/10/2021
	Programación juego n°2	30	28/10/2021	13/12/2021
	Documentación del código	1	14/12/2021	14/12/2021
	Programación juego n°3	30	10/01/2022	18/02/2022
	Documentación del código	1	21/02/2022	21/02/2022
	Pruebas	3	22/02/2022	24/02/2022
Elaboración de lamemoria				
	Capítulo I	1	25/02/2022	25/02/2022
	Capítulo II	5	28/02/2022	04/03/2022
	Capítulo III	5	07/03/2022	11/03/2022
	Capítulo IV	5	14/03/2022	18/03/2022
	Capítulo V	5	21/03/2022	25/03/2022
	Conclusiones	1	28/03/2022	28/03/2022
	Apéndices	2	29/03/2022	30/03/2022
	Total días	197	Horas de trabajo diario	2
				Total horas

Figura 3.1: Calendarización inicial de las tareas

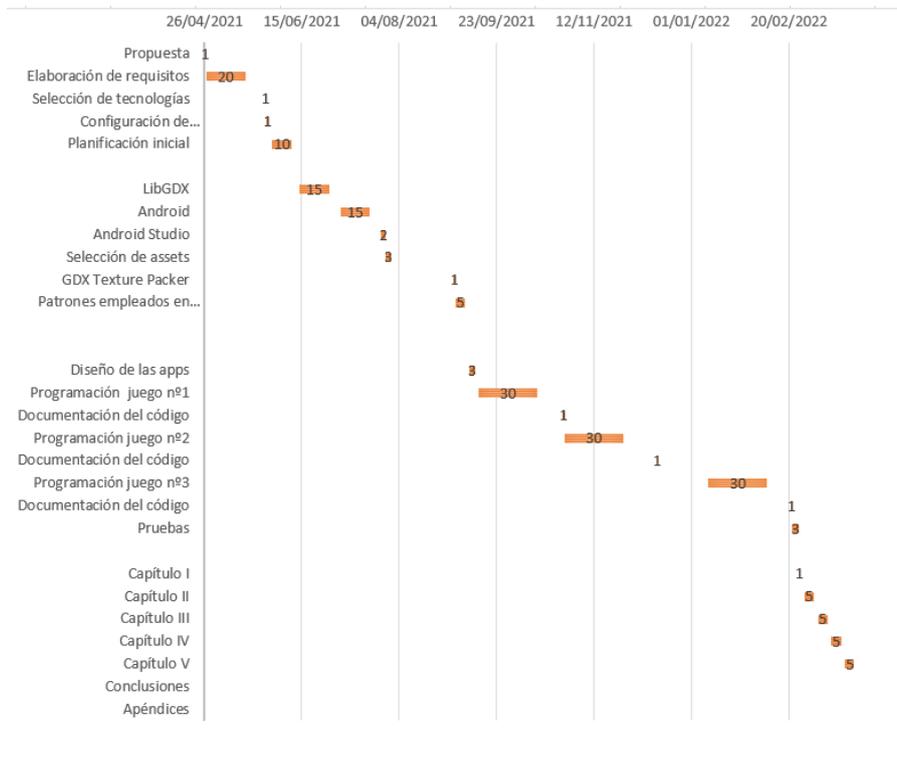


Figura 3.2: Descomposición de las tareas en un diagrama de Gantt

3.4. Identificación de riesgos

En las siguientes listas podemos observar cada uno de los riesgos identificados que se pueden sufrir a lo largo del desarrollo del trabajo de fin de grado junto con la probabilidad que hay que de que ocurran.

Identificador R-01	Título: Cambios de requisitos
Descripción	Modificación de algunos de los requisitos iniciales proporcionados por el cliente.
Plan de contingencia	Evaluar los cambios propuestos y ver si son viables o no de implementar en función del tiempo que conlleve hacerlos.

Identificador R-02	Título: Baja médica del desarrollador
Descripción	El único desarrollador podría darse de baja debido a alguna enfermedad, lo cual generaría un retraso en los plazos estimados.
Plan de contingencia	Hacer una evaluación del tiempo perdido y aumentar las horas dedicadas para recuperar el ritmo.

Identificador R-03	Título: Pérdida de datos locales
Descripción	Pérdida de los datos locales del proyecto debido a problemas con las unidades de almacenamiento o por la pérdida de datos debido a cortes de luz.
Plan de contingencia	Adquirir nueva unidad de almacenamiento y/o emplear herramientas para intentar recuperar los datos perdidos.

Identificador R-04	Título: Pérdida de los datos almacenados en la nube
Descripción	Pérdida de los datos del proyecto almacenados en la nube.
Plan de contingencia	Hacer una reclamación a los servicios de almacenamiento para que recuperen los datos perdidos.

Identificador R-05	Título: Daños materiales en el equipo empleado en el desarrollo
Descripción	Daños en componentes hardware o del equipo empleado para desarrollar.
Plan de contingencia	Reponer los componentes dañados, repararlos o bien adquirir nuevo equipo si es necesario.

Identificador R-06	Título: Falta de experiencia con LibGDX
Descripción	Conocimientos insuficientes acerca del funcionamiento de la librería empleada para la creación de los juegos.
Plan de contingencia	Emplear una mayor cantidad de tiempo para adquirir los conocimientos necesarios para poder proceder con la codificación.

Identificador R-07	Título: Falta de experiencia con Java
Descripción	Conocimientos insuficientes en Java.
Plan de contingencia	Aumentar el tiempo necesario dedicado a formación en Java.

Identificador R-08	Título: Compra de licencias
Descripción	Algunos de los elementos o tecnologías empleados pueden requerir de la compra de licencias para hacer uso de estos.
Plan de contingencia	Emplear contenido/software sin licencia.

Una vez identificados todos los posibles riesgos se procede a una evaluación y clasificación de estos en una tabla que podremos utilizar luego para calcular su impacto.

Las consecuencias que podrían generar cada uno de estos riesgos son retrasos temporales respecto a la fecha de finalización. Los únicos que aparte de ser un riesgo temporal son un riesgo monetario son el daño al equipo informático empleado y la necesidad de adquirir de licencias. Este último, tiene fácil solución como lo es buscar otro software o assets que sea gratuito, con lo cual este también se resume a una demora temporal en lo que se encuentra un sustituto.

Id	Riesgo	Probabilidad	Retraso estimado (días)	Exposición al riesgo (días)
R-06	Falta de experiencia con LibGDX	0.8	10	8
R-07	Falta de experiencia con Java	0.03	7	0.21
R-03	Pérdida de datos locales	0.02	2	0.04
R-02	Baja médica	0.2	14	2.8
R-01	Cambios de requisitos	0.1	13	1.3
R-05	Daños equipo informático	0.01	5	0.05
R-08	Compra de licencias	0.01	1	0.01
R-04	Pérdida de datos en la nube	0.005	10	0.05

Tabla 3.3: Evaluación y clasificación de riesgos

Muchos de los riesgos observados tienen muy poca probabilidad de que ocurran, pero aun así deberemos tenerlos en cuenta. Como se puede ver, el que mayor exposición tiene es la falta de experiencia/conocimientos respecto al empleo de LibGDX y el riesgo de que el programador sufra una baja médica.

Probabilidad de ocurrencia				
Muy baja	Baja	Moderada	Alta	Muy alta
[0.0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.0]

Tabla 3.4: Desglose de las probabilidades de ocurrencia de un riesgo

Nivel de impacto				
Despreciable	Marginal	Medio	Crítico	Catastrófico
[0, 2)	[2, 4)	[4, 6)	[6, 8)	[8, 10]

Tabla 3.5: Desglose del nivel de impacto

En la tabla 3.6 recogeremos y clasificaremos los riesgos con un código de colores en función de la probabilidad de ocurrencia y del nivel de impacto que pueden tener cada uno de estos. De esta forma podremos ser más conscientes de las tareas que necesitan ser atendidas con mayor presteza y en caso de ser necesario desplegar el plan de contingencia para evitar que ocurran.

Muy Alta					R-06
Alta					
Moderada					
Baja		R-02			
Muy Baja	R-07, R-03, R-01, R-05, R-08, R-04				
	Despreciable	Marginal	Medio	Crítico	Catastrófico

Tabla 3.6: Clasificación de los riesgos según su probabilidad e impacto

Como vemos en la tabla 3.6, hay un riesgo que debe ser tratado lo antes posible, el riesgo referente a la falta de conocimientos de LibGDX, con lo cual se deberá activar el plan de contingencia correspondiente (dedicar más tiempo a la adquisición de conocimientos) desde un inicio para que no tenga mayor efecto sobre el tiempo necesario para la realización de las tareas.

El siguiente riesgo con mayor probabilidad de ocurrencia es la baja médica, pero ésta al igual que el resto de los riesgos se encuentran por debajo de la franja de tolerancia, con lo cual, deberemos monitorearlos periódicamente para evitar su desplazamiento hacia las zonas rojas, pero no debemos activar los planes de contingencia respectivos.

Capítulo IV

LIBRERÍA LIBGDX

En este capítulo, se explica de manera más detallada algunas de las características de la librería empleada. También se hará una breve explicación de algunas de las herramientas que nos facilita y que hemos empleado para la realización de este trabajo.

4.1. Que es LibGDX

El framework LibGDX fue creado en 2009 y está específicamente diseñado para el desarrollo de videojuegos. La API de este framework permite crear un único proyecto que luego se puede exportar para una gran variedad de plataformas, Windows, IOS, Android. Además, no se trata de una API nueva, está ya está bastante testeada y es confiable.

LibGDX no nos obliga a interactuar con toda la API al crear una aplicación, nos permite utilizar lo que deseamos en cada momento de manera flexible, dando así gran libertad al desarrollador. Este framework no solamente usa Java, sino que, para las tareas donde el rendimiento sea crítico emplea C.

Está basado en OpenGL, que es una especificación estándar que define una API para aplicaciones con gráficos 2D y 3D.

Este framework dispone de abundante documentación online, podemos obtener información acerca de su utilización no sólo de su página web, sino también desde la comunidad de Discord o libros dedicados al tema.

4.2. Herramientas auxiliares

LibGDX también dispone de una serie de herramientas indispensables a la hora de hacer un juego. Estas utilidades facilitan ciertos aspectos del ciclo de programación evitando configuraciones tediosas.

4.2.1. Project Setup

Para la creación de proyectos se nos facilita la herramienta, LibGDX Project Generator. Este generador nos permite seleccionar las plataformas para las que queremos desarrollar el proyecto, Desktop, Android, iOS o navegador, además de darnos la opción de añadir una serie de posibles extensiones tanto propias de la librería como de terceros.

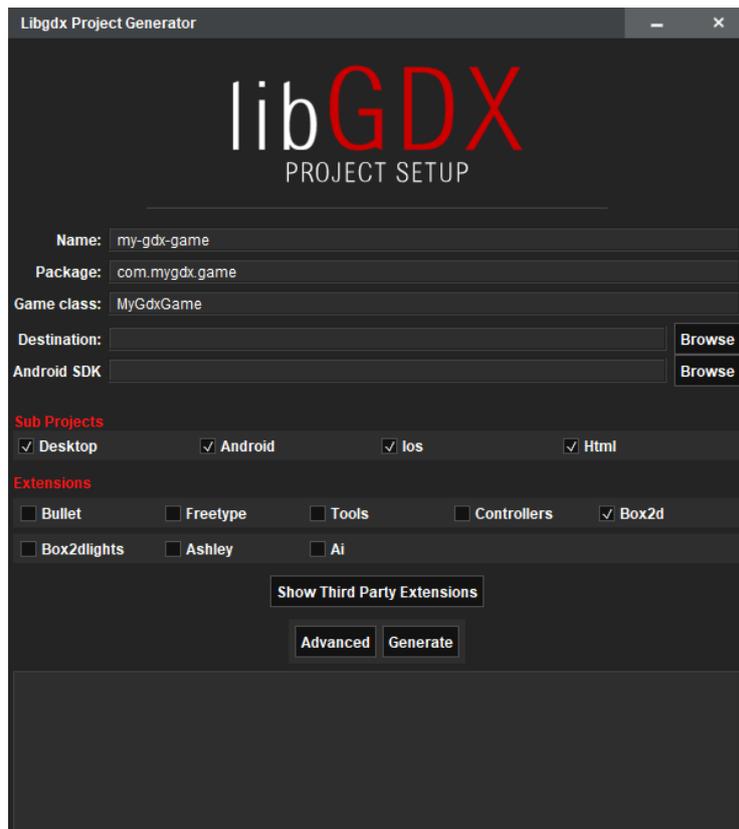


Figura 4.1: Captura de pantalla del generador de proyectos de LibGDX

4.2.2. Hiero

Hiero es una herramienta propia de la librería la cual nos sirve para crear fuentes con el formato “bitmap font” para poder usarlas dentro del juego. Podremos seleccionar de entre todas las fuentes disponibles en nuestro sistema ya que las detecta automática y posteriormente aplicarlas una serie de propiedades antes de exportarlas al formato deseado. Al exportar se generan dos tipos de archivos que debemos incluir en nuestro proyecto, un file.font y file.png, los cuales se emplean para el renderizado de los caracteres dentro del juego.

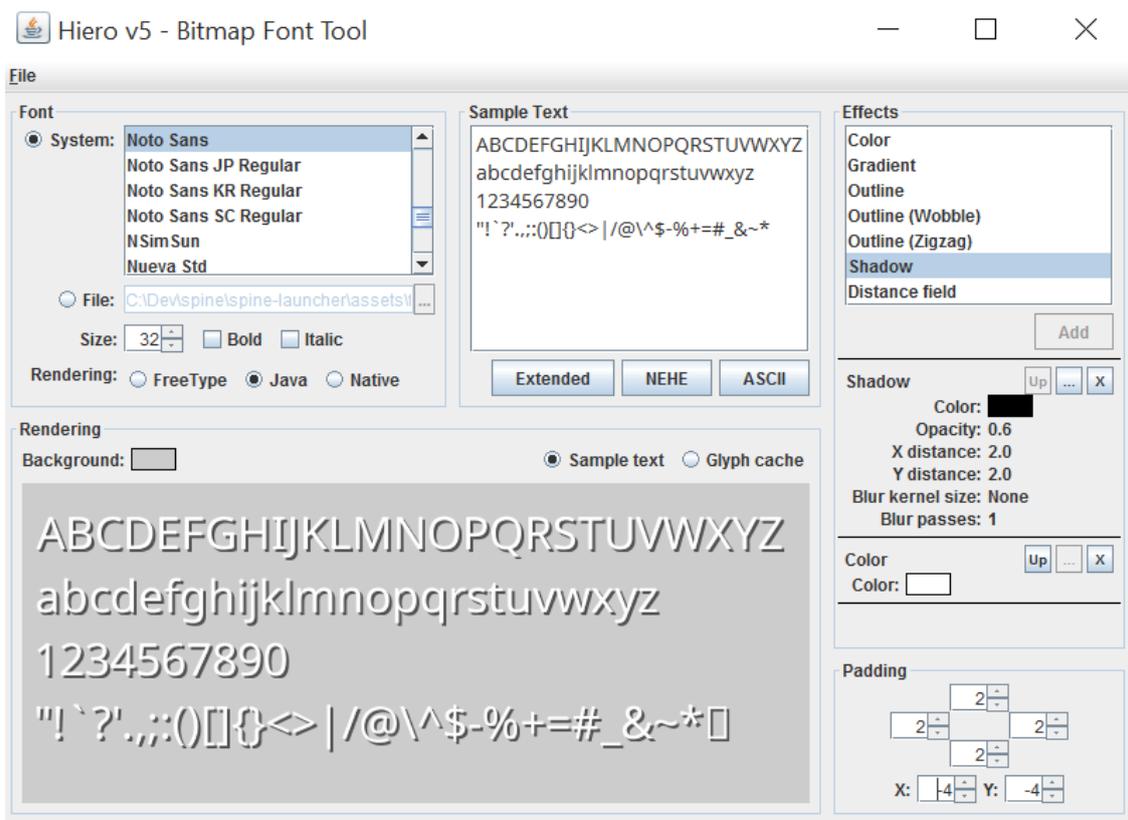


Figura 4.2: Captura de pantalla de la herramienta Hero

4.2.3. GDX Texture Packer

GDX Texture Packer, es una utilidad que se emplea para agrupar los diferentes assets en un mismo sitio, dentro en un *spritesheet*, de esta manera se intentan usar las texturas de la manera óptima, intentando reducir al máximo la cantidad de memoria necesaria para almacenarlas.

Este programa nos permite crear *spritesheets*, esto son imágenes grandes compuestas por otras varias imágenes o *sprites* más pequeños que emplearemos como imágenes estáticas o bien como animaciones dentro de un juego.

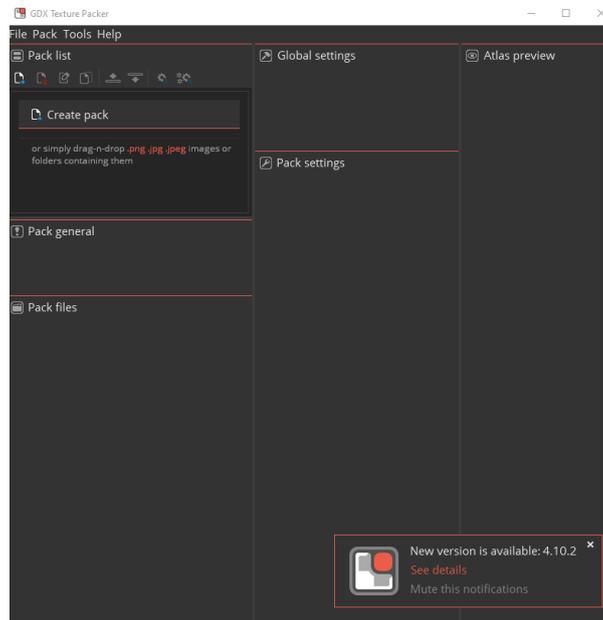


Figura 4.3: Captura de pantalla de la herramienta GDX Texture Packer

Para crear estos *spritesheets* debemos cargar todas las imágenes que deseamos poner en un mismo lienzo y el programa automáticamente se encargará de delimitar el área de cada una de ellas y de establecerles unos índices. Al finalizar la tarea se exportan dos tipos de ficheros: un fichero imagen.png y otro fichero imagen.atlas. Este último, es un fichero de texto en el que aparecen una serie de datos, como son el nombre, la rotación, la posición y tamaño entre otros, de cada una de las imágenes que se han introducido en el lienzo.

En este proyecto se han empleado para generar las animaciones del juego Colors, donde cada una de las figuras tiene que realizar una animación de pulso, es decir, primero aumentar su tamaño hasta cierto punto y luego volver a su estado inicial.



Figura 4.4: Spritesheet generado con GDX Texture Packer

4.3. Estructura LibGDX

LibGDX no nos proporciona una interfaz gráfica con la que el usuario pueda trabajar, sino que nos proporciona funcionalidad, código.

El código general del proyecto, la lógica del juego irá dentro de la carpeta *core* independientemente de la plataforma para la que se desarrolle. En caso de que sea necesario código específico para alguna de las plataformas con las que opera LibGDX, deberemos colocar el código en el directorio del backend correspondiente.

El colocar la lógica en el directorio *core* no es casualidad, esto se debe a que desde esta ubicación podremos emplear las interfaces genéricas de LibGDX, lo que nos dará acceso a la pantalla, sonidos, archivos, red.

Para proporcionarnos todos los servicios necesarios LibGDX consta de una serie de módulos que podremos ver en el siguiente diagrama:

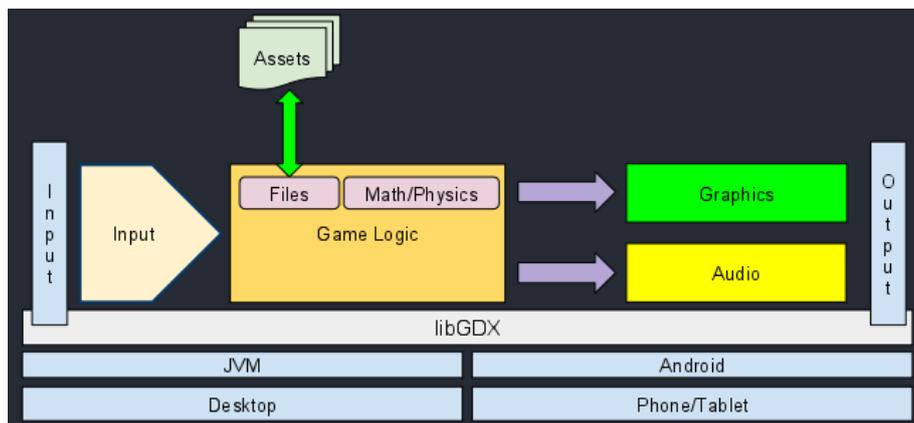


Figura 4.5: Diagrama de módulos de LibGDX

El módulo *Input*, nos proporciona un manejador de eventos de entrada para cada una de las plataformas en las que se soporta la librería. Este soporta la entrada de datos mediante teclado, pantalla táctil, acelerómetro o ratón.

El siguiente módulo, *Graphics*, se comunica directamente con la GPU y nos proporciona métodos para obtener instancias de OpenGL al igual que métodos para la creación de *Pixmap*s (mapa de píxeles que almacena y muestra una imagen como un array de valores de color de píxeles) y Texturas.

Files es el directorio donde almacenaremos todos los archivos de escritura/lectura que pueda necesitar nuestro proyecto. Aquí es donde irán almacenados todos nuestros ficheros multimedia necesarios, imágenes, sonidos y fuentes, los que se suelen denominar como *assets*.

Entrando en más detalle con esta librería vamos a explicar los aspectos más relevantes de su modelo de dominio:

- *ApplicationListener* es la interfaz más importante, alberga la funcionalidad básica de la aplicación. Se encarga de procesar y escuchar todos los cambios de estado que se puedan producir en la aplicación, es decir, cambio de tamaño de la pantalla, pausado y reanudación de la partida. A esto es lo que se denomina el “ciclo de vida” de una aplicación LibGDX. Cuando implementamos esta interfaz también debemos implementar los métodos *create*, *dispose*, *pause*, *render*, *resize* y *resume*.
- El método *create* es llamado sólo una vez cuando se inicia la aplicación, es dentro de este método donde deberemos inicializar todas nuestras variables y estructuras de datos que hemos creado y que deseamos utilizar. Una cosa a tener en cuenta es que cuando implementamos la interfaz *Screen* la cual contiene la mayoría de los métodos de *ApplicationListener* se añaden dos nuevos, *show* y *hide*. De hecho, el método *create* se ve sustituido por *show* y es en éste donde colocamos el contenido de *create*.
- *Render* es un método que se ejecuta en bucle y se ejecuta en cada tick del juego, este tick determinará el número de fotogramas que se muestran por segundo, lo que vienen siendo los fps (frames per second). Es en este método donde se realizará la actualización del estado del juego, con lo cual, la lógica del juego o bien la colocamos dentro de éste o bien en otros que posteriormente se les llamará desde el mismo.
- El método *resize* se emplea para redimensionar la pantalla del juego sin tener que estar en el estado de pausa, este método se ejecuta por primera vez justo después de que termine la ejecución de *create*.
- Al método *pause* se le invoca cuando pulsamos el botón *home* de Android, cuando recibimos una llamada o bien justo antes de que se cierre el juego o aplicación. Y *resume* será invocado al volver al contexto de nuestro juego.
- *Dispose* será el método encargado de reciclar y liberar todos los recursos que empleamos en nuestra aplicación. Se le invoca justo antes de terminar la aplicación, pero debemos tener cuidado ya que si no incluimos en el cuerpo de este los recursos que se deben eliminar no hará nada, simplemente se ejecutará sin ningún efecto.

Al terminar nuestra aplicación debemos asegurarnos de que se liberan todos los recursos correctamente y para cerciorarnos de ello, aparte de *dispose*, debemos emplear `Gdx.app.exit()`. Esta llamada nos asegura que todos los recursos serán liberados evitando así las fugas de memoria, lo cual puede suponer un riesgo de seguridad para nuestro dispositivo.

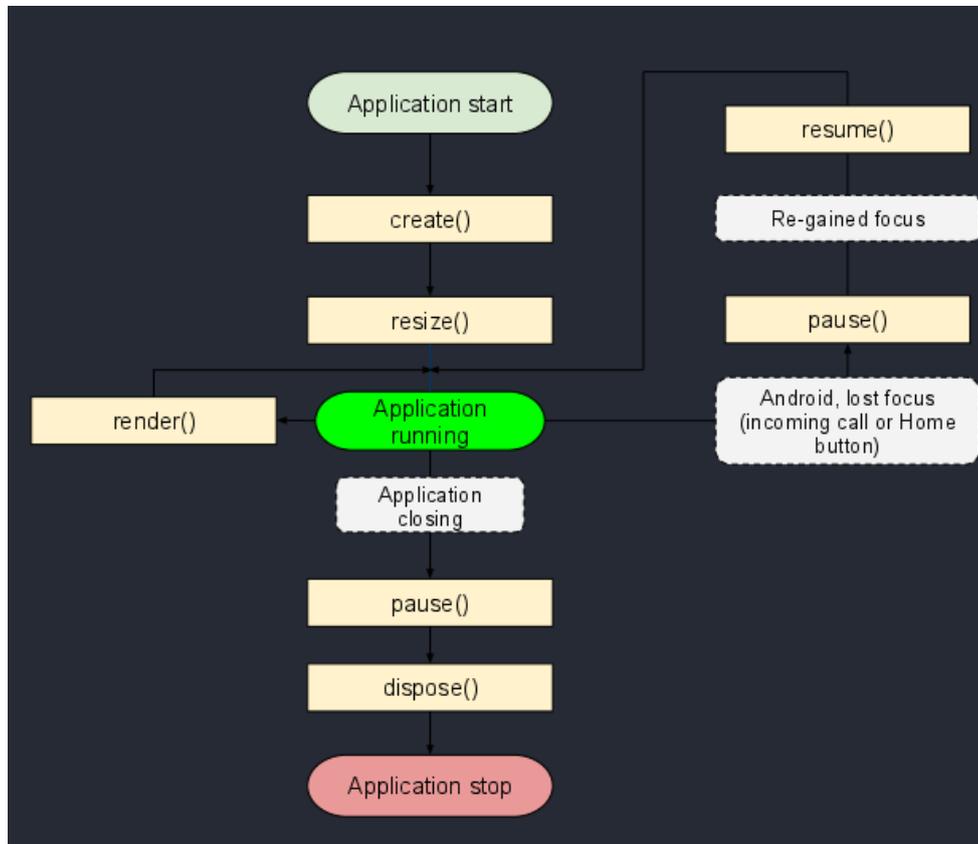


Figura 4.6: Diagrama del ciclo de vida de una aplicación en LibGDX

En la figura 4.6 se representa el ciclo de vida de una aplicación estándar de LibGDX, en esta podemos apreciar de manera gráfica y más sencilla lo que se explica en los párrafos anteriores a esta figura.

Capítulo V

DESARROLLO DE LA APLICACIÓN

En este apartado se mostrarán los requisitos que se consiguieron reunir de la reunión con los clientes y del proceso de selección de éstos, también se realizará una descripción del diseño de la arquitectura de la aplicación.

5.1. Adquisición de requisitos

Aquí se realizará el análisis y especificación de los requisitos del sistema que se han podido descubrir de la reunión realizada con los clientes (tutores de la empresa HP). De esta reunión se presentó la idea de que se desarrollarán el mayor número de juegos posibles para personas mayores y que posiblemente fueran destinados a personas con cierto deterioro cognitivo pudiéndose incluir personas que presentaran algún tipo de demencia o enfermedad neurodegenerativa.

Con lo cual, el objetivo es desarrollar juegos con independencia de menú de entrada o menú de *game over* ya que estos juegos van a ser incluidos en el game suite de la empresa.

Un factor importante que no se estableció fue la dificultad que se debería dar en estos juegos ya que la empresa decidirá la dificultad para cada usuario en función de un test previo que ellos les realizarán para así conocer el estado en el que se encuentran los usuarios.

Como idea para los mismos se ha propuesto fijarse en juegos desarrollados para ejercer la memoria como puede ser Train your Brain o Stimulus por citar algunos de ellos.

Como los requisitos son algo vagos, vamos a suponer una serie de requisitos como necesarios para poder comenzar.

Los juegos no deberán tener un nivel de dificultad muy elevado ya que lo que se pretende es estimular el funcionamiento cognitivo del usuario e intentar que se divierta.

Para evitar la frustración, se evitará mostrar puntuaciones negativas. Esto puede ser un arma de doble filo, ya que, si no se puede perder, la satisfacción al ganar es menor. Por ello, uno de los tres juegos es de dificultad incremental, es decir, la dificultad irá aumentando hasta donde seamos capaces de memorizar cierta secuencia.

5.1.1. Requisitos funcionales

Los requisitos funcionales se emplean para establecer el comportamiento del software, es decir, se emplean para describir las funciones que debe proporcionar el sistema. Para este proyecto se han identificado los siguientes:

- **RF-01: Aplicación para dispositivos móviles :**La aplicación se ejecutará en dispositivos móviles.
- **RF-02: Juegos desarrollados en LibGDX:** Los juegos desarrollados para la aplicación deberán estar programados utilizando la librería LibGDX.

Al no disponer de unas mayores especificaciones de lo que debe ser la aplicación, como se dice con anterioridad en esta sección se ha optado por añadir un menú desde el que poder seleccionar cada uno de los juegos y una pantalla de game over.

5.1.2. Requisitos no funcionales

Los requisitos funcionales son los que especifican comportamientos generales y restricciones del sistema. Estos suelen presentar cierta dificultad debido a la libre interpretación que se puede dar a la hora de definirlos.

- **RNF-01: La aplicación no debe mostrar anuncios:** No se deben mostrar anuncios que consigan equivocar a los usuarios de la aplicación.

- **RNF-02: Juegos intuitivos:** Los juegos se deben entender sin necesidad de una explicación previa de lo que se debe hacer.
- **RNF-03: Manual de instalación:** La aplicación deberá disponer de un manual de cómo se puede instalar.

5.2. Análisis

En esta sección del capítulo 5, se detallarán los casos de uso identificados para esta aplicación.

Caso de Uso	Seleccionar Juego	Identificador: CU-01
Actores	Usuario	
Precondición	Ninguna	
Postcondición	Ninguna	
Descripción	El usuario ha abierto la aplicación y está en el menú principal de ésta donde debe seleccionar uno de los tres posibles juegos	
Secuencia	<ol style="list-style-type: none"> 1. El usuario selecciona un juego. 2. El sistema carga el juego y cambia a la pantalla del juego 	
Excepción	2.a. Se produce un error de ejecución y la aplicación se cierra.	

Tabla 5.1: Descripción del caso de uso: seleccionar juego

Caso de Uso	Cerrar aplicación	Identificador: CU-02
Actores	Usuario	
Precondición	Ninguna	
Postcondición	Ninguna	
Descripción	El usuario pulsa el botón virtual de Android de retroceder y la aplicación se cierra	
Secuencia	<ol style="list-style-type: none"> 1. El usuario pulsa el botón virtual de Android de retroceder. 2. El sistema cierra la aplicación y vuelve al contexto previo. 	
Excepción		

Tabla 5.2: Descripción del caso de uso: Cerrar aplicación

En la figura 5.1 podemos observar el diagrama de casos de uso que se empleará para facilitar el desarrollo de la aplicación y que define el comportamiento de nuestra aplicación.

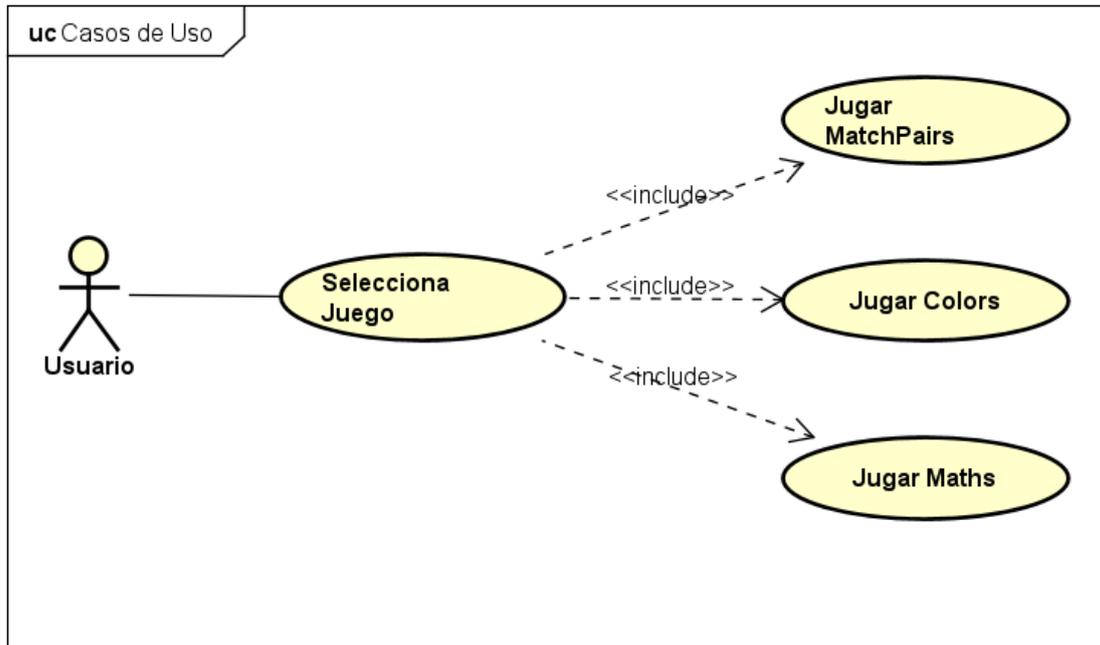


Figura 5.1: Diagrama de casos de uso

5.3. Diseño de la aplicación

Aquí se lleva a cabo una presentación de la estructura de clases que presenta nuestra aplicación. En esta sección se mostrarán los diagramas más relevantes para nuestra aplicación, de manera que se entienda de la mejor manera posible.

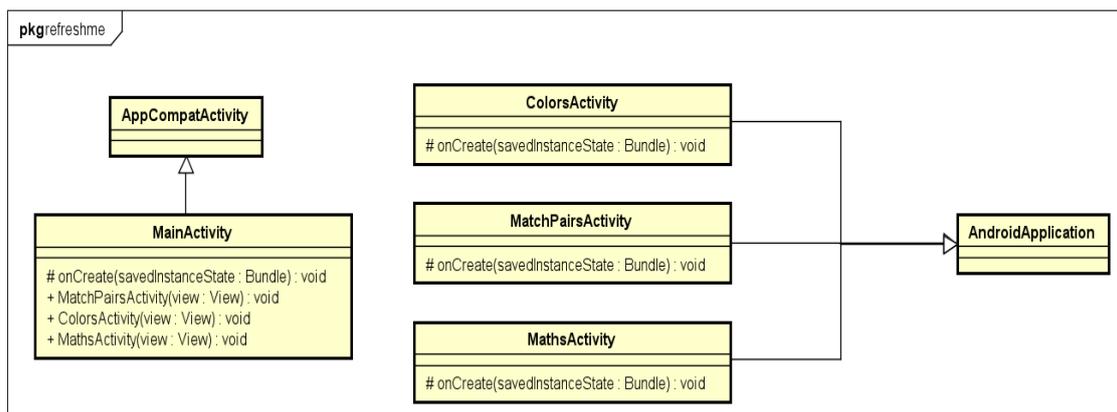


Figura 5.2: Diagrama de actividades Android

En el diagrama anterior se muestra cómo se crean las actividades correspondientes a cada uno de los diferentes juegos. Esto es así debido a que para el diseño del menú de la aplicación no se ha realizado con la LibGDX sino con las clases

de Android. De manera que cuando pulsamos en uno de los botones del menú lo que hacemos es lanzar una nueva activity, la cual se corresponde a alguno de los tres juegos.

En la figura 5.3 podemos ver como se lanzan cada uno de los juegos mediante sus actividades Android correspondientes.

A continuación se añaden los diagramas de secuencia de cada uno de los juegos, estos estan simplificados para una comprensión más fácil.

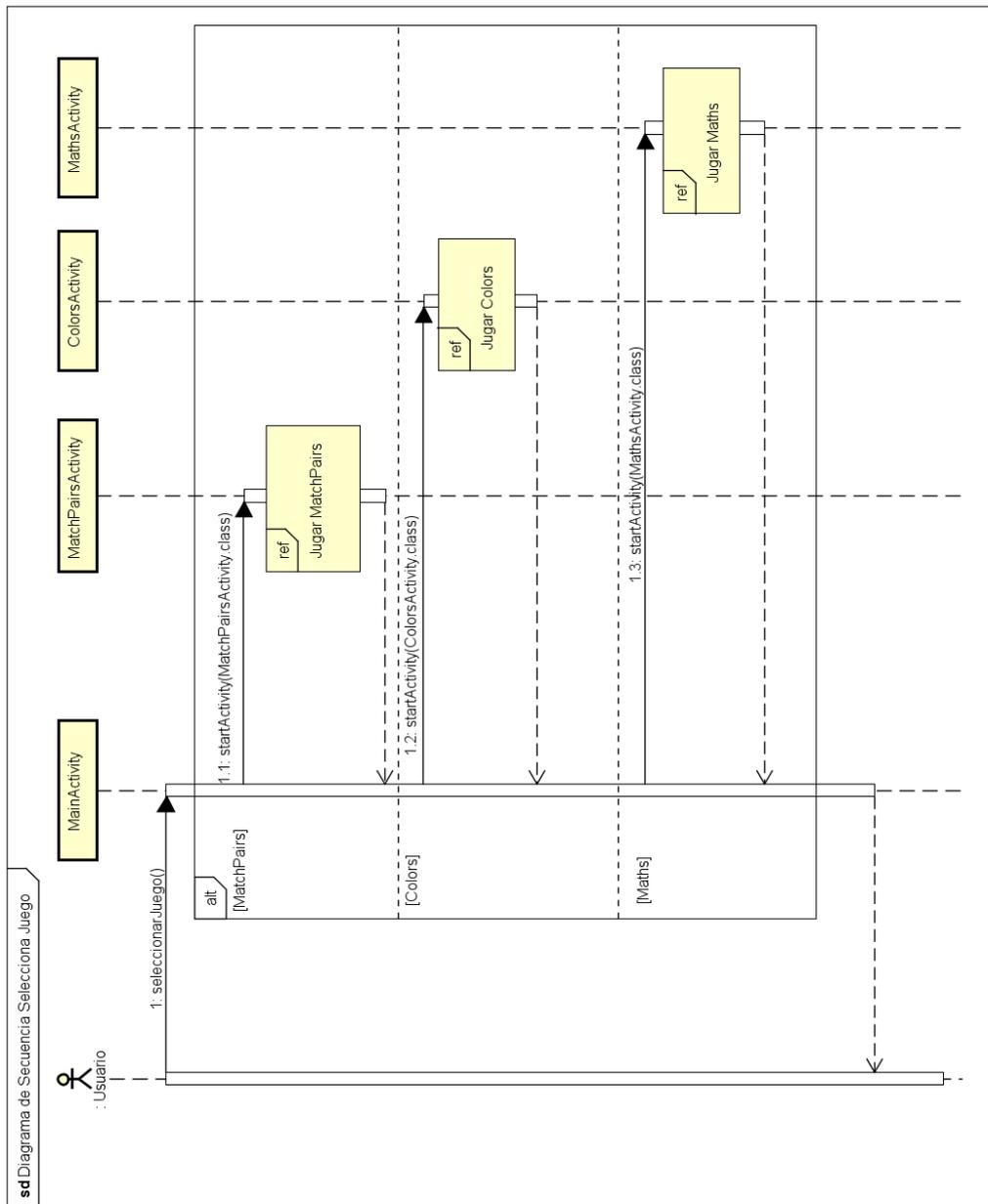


Figura 5.3: Diagrama de secuencia de selección de juego

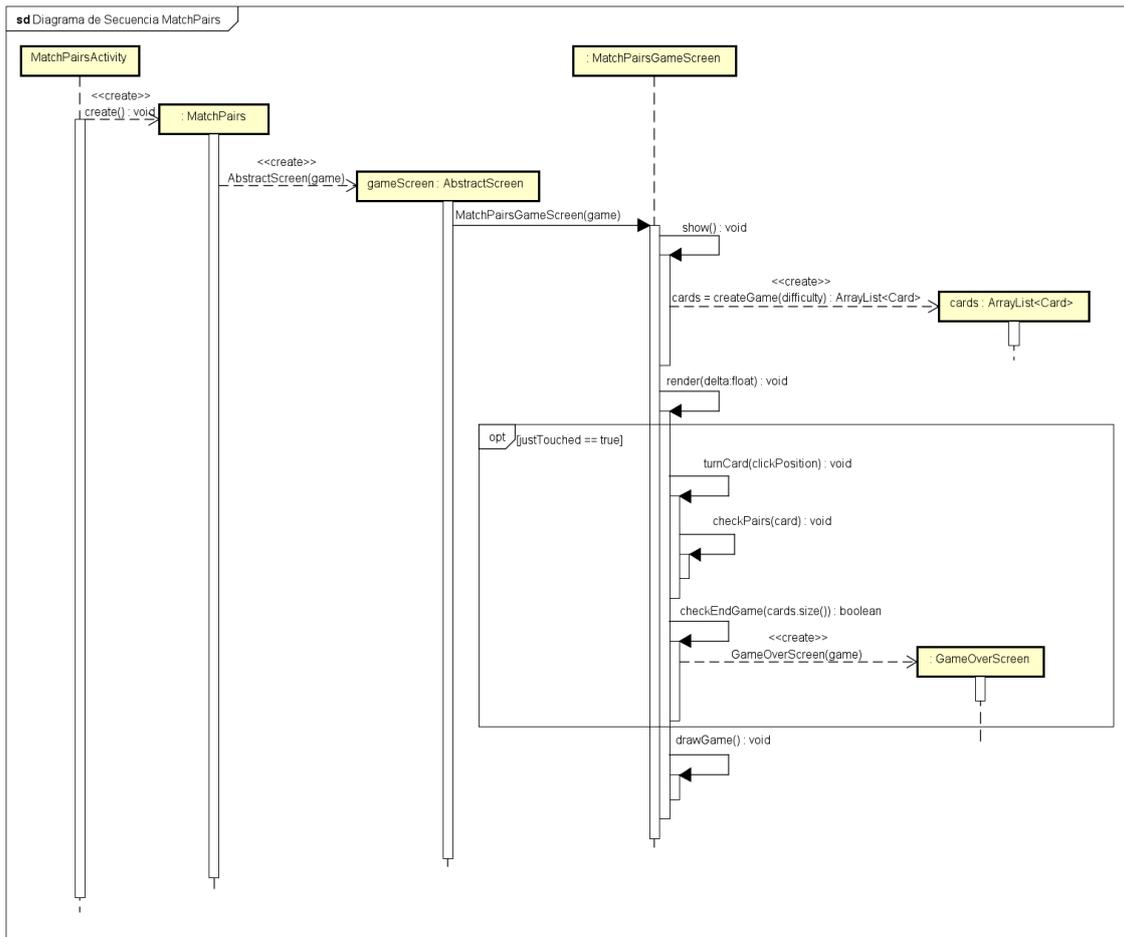


Figura 5.4: Diagrama de secuencia de MatchPairs

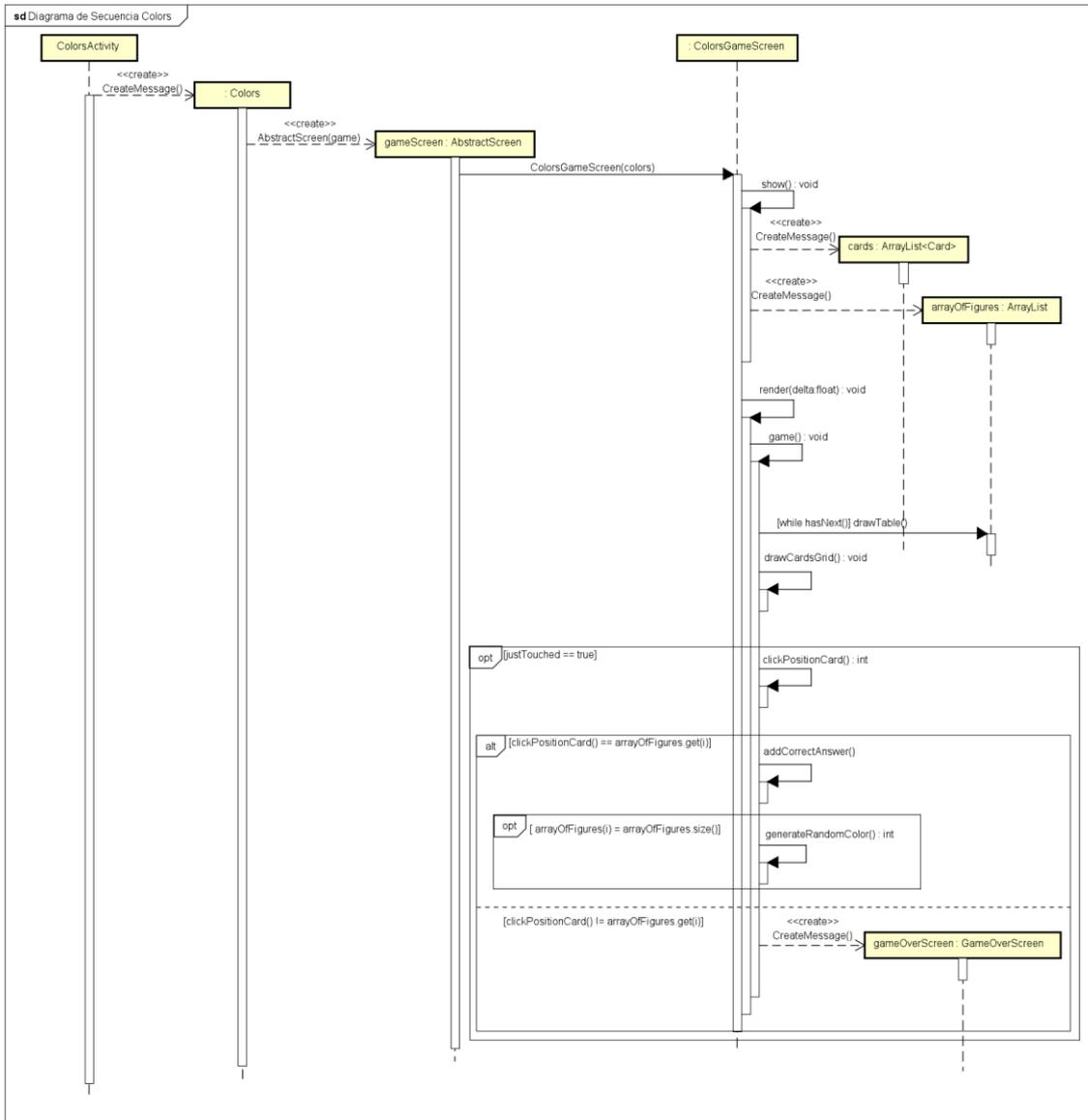


Figura 5.5: Diagrama de secuencia de Colors

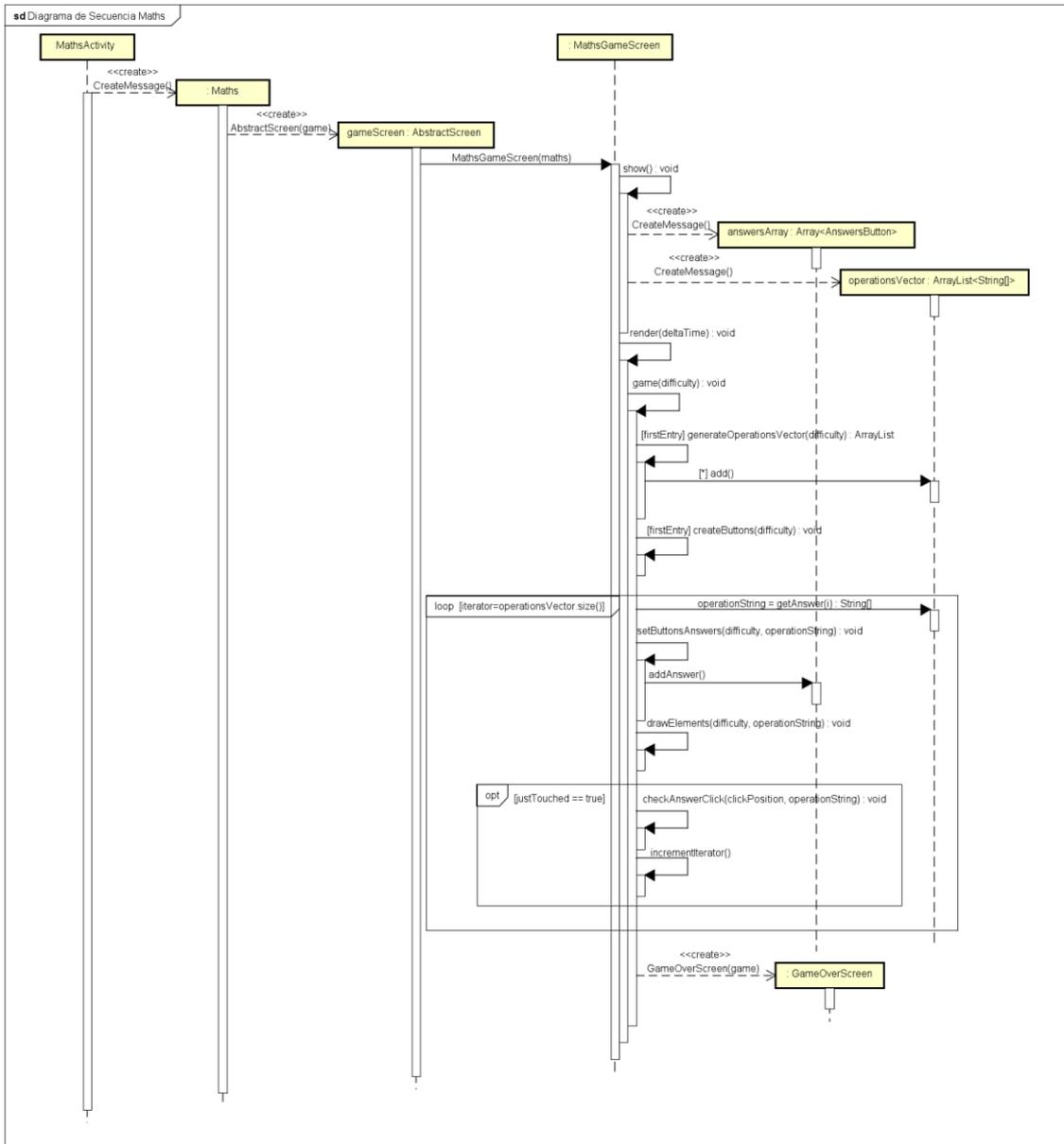


Figura 5.6: Diagrama de secuencia de Maths

En el diagrama de la figura 5.7 se aprecia el diagrama de clases que describiría la estructura de clases de los tres juegos y como interaccionan las clases entre sí. En este aparecen los diagramas de clases de los tres juegos en uno, ya que, comparten algunas clases y la separación de cada juego con su propio diagrama no nos proporcionaría la visión global que nos proporciona el actual.

Se muestran todos en conjunto ya que hay partes que emplean en común y evitar así repetir contenido en los diagramas.

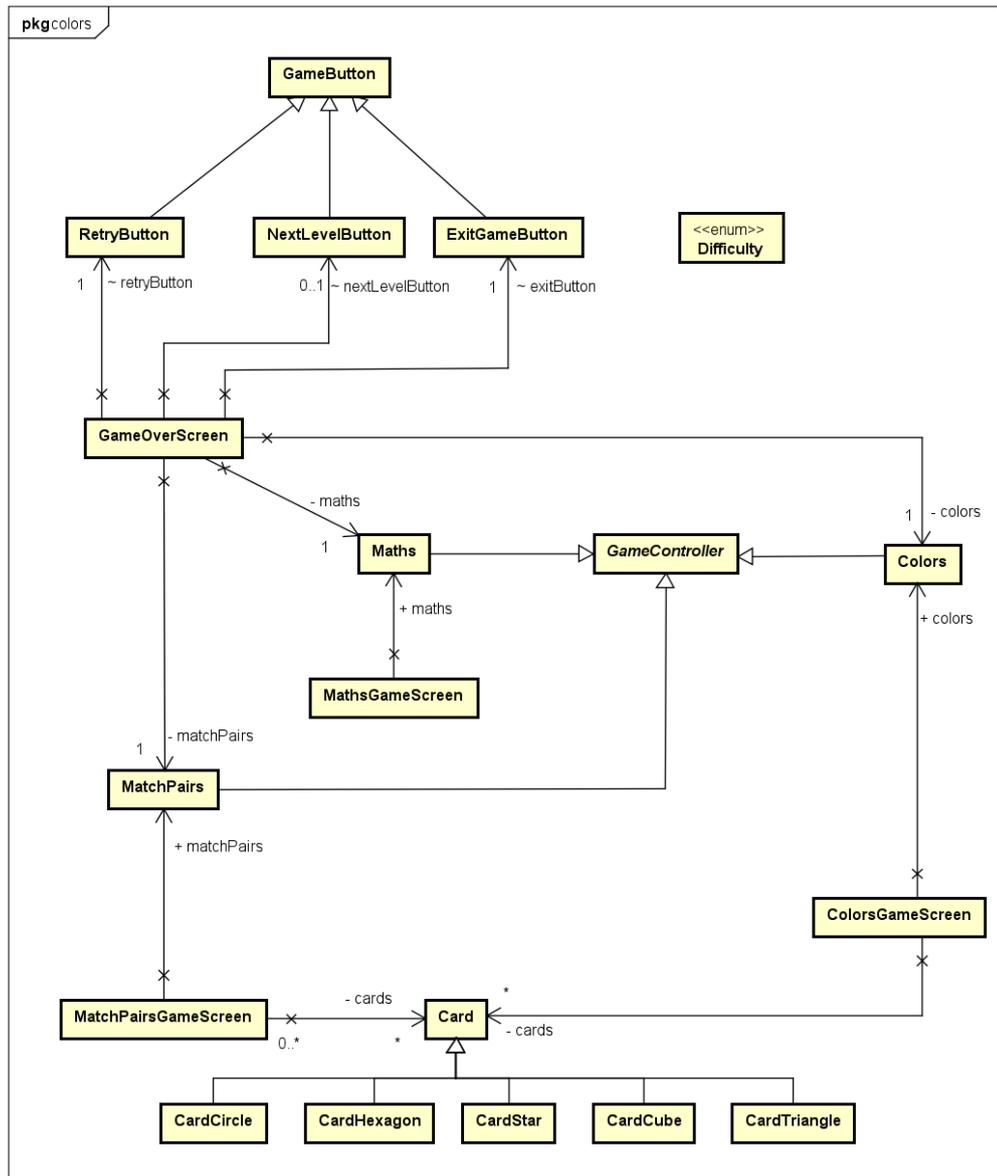


Figura 5.7: Diagrama de clases de la aplicación

Por mayor legibilidad del diagrama se ha creado otro diagrama adicional donde se muestra la relación de las clases de los tres juegos con lo que es el mánager de recursos, encargado de cargar las texturas y sonidos correspondientes.

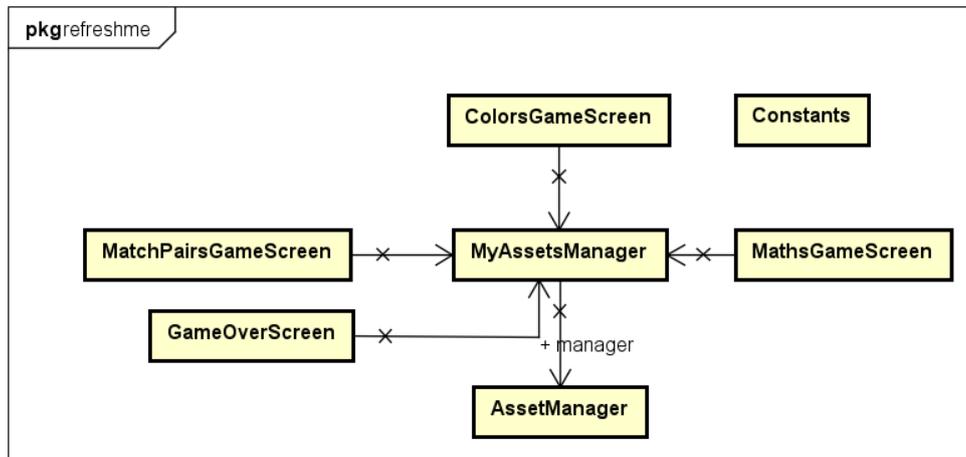


Figura 5.8: Diagrama de clases que representa la relación con el AssetManager

5.4. Implementación

En esta sección haremos una breve descripción de cada uno de los juegos que han sido desarrollados, estos son: MatchPairs, Colors y Maths.

En el primer juego, MatchPairs, los jugadores deberán ir desvelando las cartas para encontrar su correspondiente pareja, si la carta desvelada no es la correcta ambas cartas se volverán a tapar. De manera que el jugador debe memorizar las posiciones donde se encuentran cada pareja.

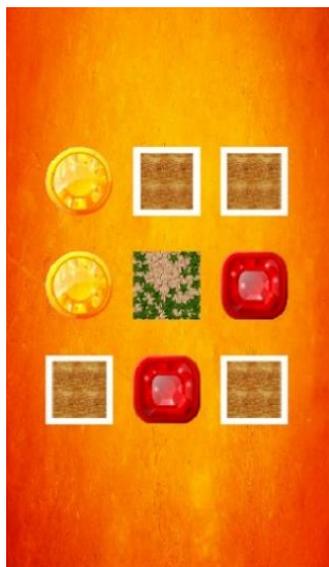


Figura 5.9: Captura de pantalla de MatchPairs

El juego consta de cuatro niveles de dificultad. En el primer nivel de dificultad, easy, los usuarios deberán emparejar dos pares de cartas. En el nivel medio se complica algo más teniendo que emparejar dos tríos de cartas. En high, se vuelve a subir un poco más la dificultad, aquí tendrán que memorizar la posición correcta de tres parejas de cartas. Y finalmente, en el nivel hard, se incrementa en uno el número de parejas respecto al anterior nivel.

En todos los niveles de dificultad tendremos un tablero de 3x3, pero en cada nivel se interactúa con un número de figuras.

En MatchPairs y en Colors se han empleado los mismos sprites para las figuras.

Colors es un juego de memorizar secuencias, el típico Simon Dice. Este tiene una dificultad incremental de manera que la dificultad depende de nuestras propias posibilidades. Inicialmente se comienza con una secuencia de 3 figuras que posteriormente se irán incrementando infinitamente siempre y cuando pulsemos la secuencia correcta en cada iteración.



Figura 5.10: Captura de pantalla de Colors

El último juego se denomina Maths y como su propio nombre indica se trata de un juego de cálculos matemáticos, en el cual debemos seleccionar el resultado correcto respecto a la operación presentada.

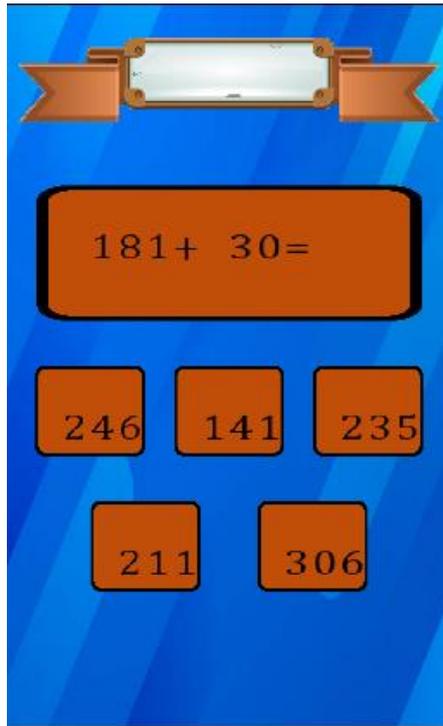


Figura 5.11: Captura de pantalla de Maths

5.4.1. Elementos visuales

En este apartado se mostrarán los elementos visuales empleados en cada uno de los juegos. Estas imágenes han sido diseñadas o bien editadas mediante el uso de Gimp. Se hará una breve explicación de cada una de ellas.

En una versión inicial este fue el fondo de pantalla diseñado para todos los juegos, pero en versiones posteriores de la aplicación se ha sustituido por un fondo concreto para cada juego.

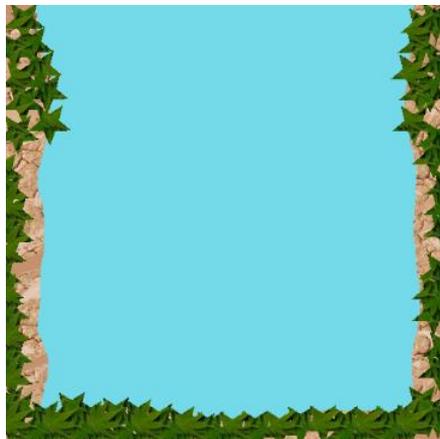


Figura 5.12: Primer fondo de pantalla

Los fondos empleados han sido seleccionados en función de su significado psicológico. Por ello, los colores seleccionados han sido el naranja se asocia con el entusiasmo y la acción, el color azul con la tranquilidad y la inteligencia, y finalmente el verde que representa la esperanza. Esto se debe a que lo que se pretende con esta aplicación es intentar hacer que los usuarios estén tranquilos y se diviertan durante un momento mientras ejercitan sus capacidades cognitivas.

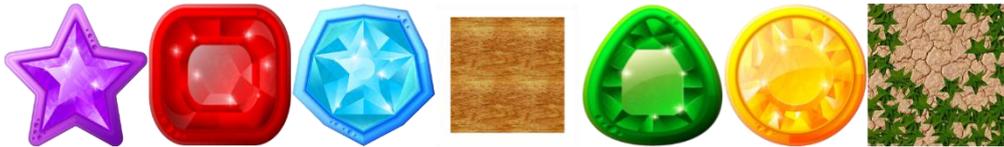


Figura 5.13: Figuras¹ de los juegos MatchPairs y Colors

En la figura anterior se ven representadas todas imágenes empleadas en los dos primeros juegos. La cuarta se corresponde a la tapa que se emplea en MatchPairs para ocultar las cartas, y la última para rellenar huecos libres para mantener la simetría del tablero.



Figura 5.14: Botones² del menú de Game Over

En esta imagen se ven los 3 botones usados en el menú de la pantalla de *Game Over*, botón de salir, siguiente nivel y volver a jugar respectivamente.

En esta última figura se ve la imagen usada para poner como fondo para la puntuación de los tres juegos.



Figura 5.15: Textura donde se colocan la puntuación

¹ Diseñadas por “macrovector” de la página de “Freepik.com”.

² Diseñados por “Johnstocker” compartidos en la página de “Pngtree.com”.

5.4.2. Versiones

El versionado de la aplicación ha comenzado con la versión 0.1 donde se consiguió implementar el primer juego, MatchPairs. Con lo cual las versiones posteriores se ha seguido el mismo criterio, es decir, se ha pasado a las siguientes versiones con la adición de juegos nuevos.

Se incrementó a la versión 0.2 con la entrega de Colors y a la 0.3 con la de Maths. En las siguientes se comienzan a ir arreglando ciertos bugs y a aplicar ciertas correcciones que se consideraron oportunas.

Uno de estos cambios consistió en cambiar el fondo de pantalla ya que todos los juegos tenían el mismo fondo de pantalla y se optó por poner un fondo de pantalla diferente para cada uno de ellos para así evitar ocasionar en el usuario la falsa sensación de que seguía en el mismo juego. En esta corrección también se eliminó un *placeholder* del juego de Colors que se empleaba para mantener la simetría del tablero que también podía dar lugar a confusiones.

Para mantener el control de versiones de la aplicación se ha empleado la plataforma GitLab. Permitiendo así mantener de forma segura el código y llevar un registro de los cambios realizados en cada una de las actualizaciones.

5.5. Pruebas

Al tratarse de un videojuego sería interesante desarrollar estas pruebas con un conjunto de unos 10 o 20 usuarios y que estos nos reporten los posibles fallos que se puedan dar durante la ejecución del juego para así poder encontrar y corregir los bugs y glitches que pueda haber. De esta manera, podríamos establecer unas pruebas versión beta del juego. Pero se optará por realizar una pequeña serie de pruebas para ver si todo funciona adecuadamente. A continuación, se presentan las pruebas realizadas:

Prueba	PR-01 Abrir juegos
Prerrequisitos	
Procedimientos	Acceder a cada uno de los juegos
Resultado esperado	Poder ver la pantalla correspondiente a MatchPairs, Colors y Maths.
Resultado obtenido	OK

Prueba	PR-02 MatchPairs, generar tableros de diferente dificultad
Prerrequisitos	
Procedimientos	Lanzar repetidas veces el juego en las diferentes dificultades disponibles.
Resultado esperado	Generar las diferentes distribuciones de posiciones de figuras correspondientes a cada una de las dificultades.
Resultado obtenido	OK

Prueba	PR-03 MatchPairs, generar parejas o tríos dependiendo de la dificultad
Prerrequisitos	
Procedimientos	Lanzar repetidas veces el juego en los diferentes niveles de dificultad.
Resultado esperado	Generar parejas o tríos dependiendo de la dificultad establecida.
Resultado obtenido	OK

Prueba	PR-04 MatchPairs, mostrar la carta seleccionada
Prerrequisitos	
Procedimientos	Hacer clic encima de una carta
Resultado esperado	La carta seleccionada se destapa y se muestra una figura
Resultado obtenido	ERROR
Solución	Establecer unas coordenadas universales independientes de la resolución de la pantalla de cada dispositivo.

Prueba	PR-05 MatchPairs, tapar cartas que no son pareja
Prerrequisitos	
Procedimientos	Seleccionar una carta Seleccionar otra carta diferente
Resultado esperado	Las figuras al no ser idénticas se vuelven a tapar
Resultado obtenido	OK

Prueba	PR-06 MatchPairs, finalizar el juego cuando se destapen todas las parejas de cartas
Prerrequisitos	
Procedimientos	Emparejar todas las cartas
Resultado esperado	Mostrar menú de Game Over
Resultado obtenido	OK

Prueba	PR-07 MatchPairs, al abrir el juego se comienza a reproducir la música
Prerrequisitos	
Procedimientos	Abrir MatchPairs
Resultado esperado	Escuchar música de fondo
Resultado obtenido	OK

Prueba	PR-08 Colors, crear secuencia de animaciones
Prerrequisitos	
Procedimientos	Abrir Colors y esperar a que se genere la secuencia de animaciones de figuras inicial.
Resultado esperado	Se deben realizar 3 animaciones de figuras, estas pueden ser de la misma figura.
Resultado obtenido	ERROR
Solución	Emplear la clase propia de LibGDX para hacer las animaciones

Prueba	PR-09 Colors, tiempo entre animaciones
Prerrequisitos	
Procedimientos	Abrir Colors y ver como se realizan las animaciones
Resultado esperado	Las animaciones se realizan de manera secuencial sin pisarse
Resultado obtenido	ERROR
Solución	Insertar un pequeño delay de tiempo entre animaciones

Prueba	PR-10 Colors, añadir una figura nueva a la secuencia siguiente
Prerrequisitos	
Procedimientos	Insertar correctamente la secuencia mostrada
Resultado esperado	Añadir una nueva figura al final y mostrar de nuevo la secuencia incluyendo la nueva figura
Resultado obtenido	ERROR
Solución	Llevar un contador, añadir la siguiente figura y reiniciar el vector de figuras cada vez que se completa correctamente para que las animaciones comiencen desde el principio.

Prueba	PR-11 Colors, reproducir sonido al animar cada figura
Prerrequisitos	
Procedimientos	Abrir Colors y escuchar si a cada animación la acompaña el sonido correspondiente a cada figura.
Resultado esperado	Cada figura tiene asignado un sonido diferente
Resultado obtenido	OK

Prueba	PR-12 Colors, reproducir sonido al hacer clic encima de una figura
Prerrequisitos	
Procedimientos	Abrir Colors e introducir la secuencia correcta
Resultado esperado	Cada vez que se pulse encima de una figura se debe reproducir su sonido asignado
Resultado obtenido	OK

Prueba	PR-13 Colors, terminar partida cuando no se introduce la secuencia correcta
Prerrequisitos	
Procedimientos	Abrir Colors, esperar que haga la secuencia inicial e introducir una figura que no corresponda al orden establecido
Resultado esperado	Salir al menú de Game Over
Resultado obtenido	OK

Prueba	PR-14 Maths, generar operaciones en función de la dificultad establecida
Prerrequisitos	
Procedimientos	Abrir Maths en cada una de las dificultades y ver que operaciones genera
Resultado esperado	Generar operaciones en función de la dificultad establecida
Resultado obtenido	OK

Prueba	PR-15 Maths, generar el número de botones de respuesta adecuado a cada dificultad
Prerrequisitos	
Procedimientos	Abrir Maths en cada una de las dificultades y ver cuántos botones de respuesta crea
Resultado esperado	Crear 3 o 5 botones de respuesta dependiendo de la dificultad establecida
Resultado obtenido	OK

Prueba	PR-16 Maths, generar resultados diferentes
Prerrequisitos	
Procedimientos	Abrir Maths y generar operaciones
Resultado esperado	Los diferentes posibles resultados para cada operación no se deben repetir
Resultado obtenido	OK

Capítulo VI

EJECUCIÓN DEL PROYECTO

Durante este capítulo se explicarán de forma detallada los plazos temporales reales del proyecto y se hará una estimación del coste real que supondría encargar este proyecto a una empresa.

6.1. Evolución temporal del proyecto

Esta sección explica lo que realmente ha supuesto la realización de este trabajo de fin de grado, ya que se presenta un considerable retraso respecto al comienzo. Este retraso supone la necesidad de dedicar una mayor cantidad de tiempo para desarrollar las tareas ya que el tiempo apremia. Estos retrasos respecto a la planificación se deben a factores personales y de trabajo del desarrollador. También cabe mencionar que se empleó una mayor cantidad de tiempo en documentarse correctamente sobre LibGDX.

En la figura 6.1 se puede ver el tiempo real que se ha tardado en realizar cada una de las tareas programadas en cada una de las etapas. Otra diferencia que se puede observar respecto a la planificación inicial es el aumento de tareas en la etapa de realización de la memoria para una mayor comprensión de esta.

Etapas		Duración Estimada en días	Fecha de Inicio	Fecha de Finalización
Puesta en Marcha				
	Propuesta	0,5	01/11/2021	01/11/2021
	Elaboración de requisitos	10	01/11/2021	12/11/2021
	Selección de tecnologías	0,5	12/11/2021	12/11/2021
	Configuración de tecnologías	0,5	15/11/2021	15/11/2021
	Planificación	5	15/11/2021	22/11/2021
Recopilación de información				
	LibGDX	15	23/11/2021	13/12/2021
	Android	5	13/12/2021	17/12/2021
	Android Studio	1	20/12/2021	20/12/2021
	Selección de assets	1,5	21/12/2021	22/12/2021
	GDX Texture Packer	0,5	22/12/2021	22/12/2021
	Patrones empleados en videojuegos	5	03/01/2022	07/01/2022
Desarrollo del proyecto				
	Diseño de las apps	1	10/01/2022	10/01/2022
	Programación juego nº1	18,5	11/01/2022	03/02/2022
	Documentación del código	0,5	03/02/2022	03/02/2022
	Programación juego nº2	17	04/02/2022	01/03/2022
	Documentación del código	0,5	01/03/2022	01/03/2022
	Programación juego nº3	13,5	02/03/2022	18/03/2022
	Documentación del código	0,5	18/03/2022	18/03/2022
	Pruebas	0,5	19/03/2022	19/03/2022
Elaboración de la memoria				
	Capítulo I	0,5	19/03/2022	19/03/2022
	Capítulo II	3	21/03/2022	23/03/2022
	Capítulo III	3	24/03/2022	28/03/2022
	Capítulo IV	3	28/03/2022	30/03/2022
	Capítulo V	3	30/03/2022	01/04/2022
	Capítulo VI	3	01/04/2022	05/04/2022
	Capítulo VII	1	05/04/2022	07/04/2022
	Capítulo VIII/Bibliografía	3	07/04/2022	09/04/2022
	Conclusiones	3	09/04/2022	11/04/2022
	Apendices	3	11/04/2022	13/04/2022
	Correcciones	10	13/04/2022	25/04/2022

Figura 6.1: Calendarización real de las tareas

De la figura 6.1 se obtienen los siguientes datos, teniendo en cuenta de que la jornada laboral será de 4 horas:

Etapas	Días totales	Horas totales
Puesta en marcha	16,5	66
Recopilación de información	28	112
Desarrollo del proyecto	52	208
Elaboración de la memoria	35,5	142
Total	132	528

Tabla 6.1: Calculo días/horas de cada fase del proyecto

En la siguiente figura 6.2, podremos ver la descomposición de las tareas en el diagrama de Gantt empleado para representar de manera más gráfica el tiempo utilizado para cada tarea.

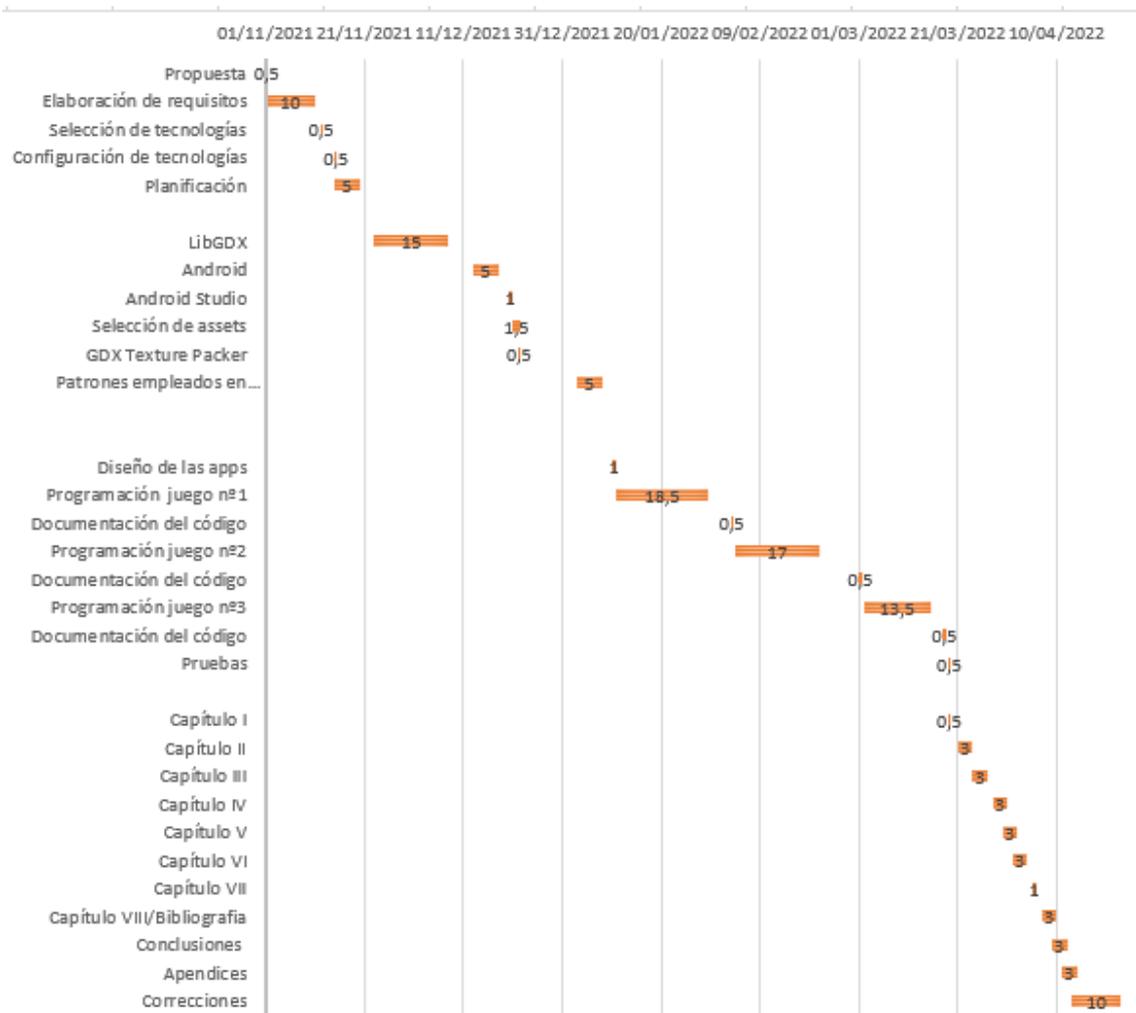


Figura 6.2: Descomposición de las tareas en un diagrama de Gantt

6.2. Costes del proyecto

Este estudio económico se realizará de una manera ficticia pero lo más cercana a la realidad. Supondremos que el equipo no está formado sólo por una persona y que el tiempo dedicado son las 528 horas totales de la figura 4.1.

Se van a desglosar las diferentes etapas del desarrollo, realizando un cálculo del coste de cada una de ellas y finalmente un coste total.

6.2.1. Horas efectivas empleadas y tasas horarias del personal

A continuación, se calcularán las horas anuales de trabajo y también las semanas trabajadas en un año. Para acabar haciendo así un cálculo del coste por hora de cada uno de los diferentes empleados de nuestra pequeña empresa.

Los días festivos durante el año 2021 fueron doce. A estos les debemos sumar al menos un día de fiestas locales.

Los sábados y domingos anuales (días no laborales) los calcularemos usando la siguiente fórmula:

$$365 / 7 = 52,14 \approx 52 \text{ (semanas tiene el año)}$$

$$52 * 2 = 104 \text{ (días del año se corresponden a sábados y domingos)}$$

En virtud del artículo 38 del Estatuto de los Trabajadores se establecen 30 días de vacaciones, si se han trabajado menos días, pues los días de vacaciones se calcularán por una regla de tres.

Por ley, se establece que al año se le deben garantizar a los trabajadores 20 horas de formación, lo que representa aproximadamente 2,5 días de trabajo.

Para conocer los días medios de baja, se ha empleado el estudio publicado en el periódico “La Razón”[15], donde se ha estimado que de media los trabajadores cogen de media 42 días de baja. Al tratarse del sector informático, vamos a suponer que los días de bajas se reducirán a la mitad ya que no se trata de una profesión de riesgo, pero vamos a tener en cuenta posibles bajas debido a COVID.

Por lo tanto, con estos datos elaboramos la siguiente tabla:

Concepto	Días/Horas
Año Medio	365
Sábados y domingos	104
Días efectivos de vacaciones	30
Días festivos reconocidos	13
Media de días perdidos por enfermedad	21
Formación y otros	2.5
Total estimado de días efectivos	194.5 \approx 195
Total de horas al año efectivas (8 horas / día)	1560

Tabla 6.2: Desglose de los días y horas de trabajo anuales

Como hemos dicho anteriormente tenemos 52 semanas anuales, de las cuales 19,1 (≈ 19) se corresponden a periodos vacacionales y festivos, y otras 3,35 (≈ 3) son para días de baja y de formación.

Concepto	Semanas
Año medio	52
Vacaciones y festivos	19
Enfermedad	3
Cursos de formación y otros	0.35
Total	29.55 \approx 30

Tabla 6.3: Semanas anuales de trabajo

En la tabla 6.4 se reflejan los salarios medios brutos de programadores de videojuegos en España. Estos pueden variar dependiendo de la comunidad autónoma en la que nos encontremos y de la propia empresa. [16, 17, 18]

Puesto	Experiencia	Salario Bruto Anual
Jefe de Proyecto / Programador Senior	De 10 a 20 años	45.200€
Ingeniero Intermedio	De 4 a 9 años	29.600€
Ingeniero Junior	De 0 a 3 años	21.500€
Auxiliar Administrativo		17.500€

Tabla 6.4: Sueldos medios de cada rol

Los siguientes datos se calculan en función de los sueldos brutos anuales. Supondremos que los días laborales son 5 y con un total de 8 horas al día.

Concepto	Jefe de Proyecto	Ing. Intermedio	Ing. Junior	Administrativo
Sueldo Bruto Mensual	3766 €	2466 €	1791,6 €	1458 €
Coste Hora	23,53 €	15,41 €	11,19 €	9,11 €

Tabla 6.5: Coste mensual y por hora de cada rol

El rol del administrativo no está incluido en la figura anterior ya que el trabajo de este es bastante puntual, como por ejemplo recibir llamadas, hacer facturación, nóminas, etc., con lo cual se realiza una aproximación temporal de trabajo de unas 20 horas.

Concepto	Jefe de proyecto	Ing. Intermedio	Ing. Junior	Administrativo
Coste por hora	23,53 €	15,41 €	11,19 €	9,11 €
Horas trabajadas	165	288	293	20
Total a pagar	3.882,45 €	4.438,08 €	3.278,67 €	182,2 €
TOTAL	11.781,4 €			

Tabla 6.7: Costes totales de la mano de obra

6.2.2. Amortizaciones del equipo informático

En la siguiente tabla se recogen los precios de los componentes hardware y software. Para la elección de los requisitos mínimos que han de cumplir los sistemas informáticos a emplear nos fijamos en los requisitos aconsejados por Android Studio, ya que es el programa que más fuerza de cómputo necesita.

Requisitos Android Studio:

- Windows 7/8/10 (32 o 64 bits)
- 2 GB de RAM (8 GB de RAM recomendados)
- 2 GB de espacio libre mínimo (4 GB recomendados)
- Resolución mínima de 1280 x 800.
- Java 8.
- 64 bits y procesador Intel (emulador)

La amortización de este tipo de bienes materiales se calcula en 5 años, pero como nuestro proyecto se estima que dure tan sólo 3 meses pues procedemos a calcularlo en función de este periodo:

$$Amortización = \frac{\text{valor producto}}{\text{años} * n^{\circ}\text{meses año}} * 3$$

$$Amortización = \frac{\text{valor producto}}{5 * 12} * 3$$

Concepto	Coste por unidad	Unidades	Coste Total	Amortización a 3 meses
Portátil i5-1135G7/8GB/256GB SSD/14"	613,20 €	5	3066 €	153,30 €
Ratón Inalámbrico	12,59 €	5	62,95 €	3,15 €
HP LaserJet Pro	145,83 €	1	145,83 €	7,29 €
Monitor BenQ GW2280 21.5" LED FullHD	134,76 €	4	539,04 €	26,95 €
TOTAL			3730,94 €	190,69 €

Tabla 6.8: Coste del material informático

Como ya se ha indicado en la sección de tecnologías, todo el software empleado es de uso gratuito de manera que nos ahorraremos los gastos por software. Los equipos informáticos ya vienen con una instalación de Windows 10, por lo tanto, también nos quitamos de comprar licencias de sistema operativo.

6.2.3. Costes del material consumible

Estas estimaciones de gastos en materiales consumibles se realizan en función de la duración del proyecto.

Concepto	Coste por unidad	Unidades	Coste total
Papel Impresora 2500 folios	32 €	1	32 €
Pilas, pack 48	12,53 €	1	12,53 €
Tóner Impresora	59,90 €	1	59,90 €
TOTAL			104,43 €

Tabla 6.9: Coste del material consumible

6.2.4. Costes indirectos

Estos costes son los generados por la luz, internet, teléfono, etc. Por ejemplo, el alquiler de una oficina de 92 metros cuadrados en el centro de Valladolid costaría 961 € al mes. El alquiler de 2 líneas telefónicas y fibra por unos 60 € al mes y una factura de luz cada dos meses de 100 €. Si estimamos que nos toma 488 horas la elaboración completa del proyecto, a 8 horas laborales nos da 61 días laborales, lo que son aproximadamente 3 meses.

Concepto	Coste Mensual	Coste de 3 Meses
Luz	50 €	150 €
2 líneas móviles + internet	58,90 €	1776,70 €
Alquiler oficina	961 €	2883 €
TOTAL	1069,90 €	3209,70 €

Tabla 6.10: Costes indirectos

Los costes de esta tabla como son tomados como que en la empresa se realiza un único proyecto, y todos los costes van destinados al mismo. Pero en caso de que no fuera así, estos costes se deberían distribuir entre todos los proyectos activos durante el periodo en el que se realiza este.

6.2.5. Costes totales del proyecto

Los costes totales del proyecto se estimarán en función de las 488 horas empleadas en el desarrollo. Para hacer los cálculos procedentes emplearemos los datos que aparecen en la figura 6.3, para así obtener la cantidad de dinero a abonar a cada empleado por el trabajo realizado.

En esta última tabla se recogen lo que pueden ser los gastos totales para la creación de un proyecto de estas magnitudes suponiendo que se cumplan todas las previsiones anteriores.

Concepto	Costes Totales
Costes mano de obra	11.781,4 €
Costes de amortización	190,69 €
Costes Materiales Consumibles	104,43 €
Costes Indirectos	3209,70 €
Total	15.286,22 €

Tabla 6.11: Costes totales del proyecto

Capítulo VII

CONCLUSIONES

En lo referente a la consecución de los objetivos, se puede decir que se han conseguido ya que la idea principal era desarrollar varios juegos y se han conseguido hacer tres, los cuales están listos para ser incorporados a la plataforma de HP.

Una dificultad que se presentó desde el primer momento fue la imprecisa definición de requisitos del proyecto por parte del cliente, lo cual ha supuesto un mayor esfuerzo para intentar averiguar y establecer estos requisitos en función de mis criterios y en un punto de vista no profesional acerca del tema. Los juegos han sido seleccionados en base al criterio de que sean juegos que necesiten utilizar la memoria para poder avanzar o bien realizar cálculos matemáticos, ya que ambos estimulan las capacidades cognitivas.

Las dificultades principales que se han encontrado durante el desarrollo del trabajo han sido principalmente las relacionadas con el funcionamiento de las animaciones y con la lógica de programación, ya que en un principio me resultó costoso asimilar cómo se deberían emplear correctamente.

Estas dificultades han supuesto una mejora de aprendizaje, al conseguir buscar información adecuada, comprenderla y posteriormente adaptarla a mis necesidades.

La idea de este trabajo de fin de grado es muy interesante, ya que tiene mucho potencial y un trasfondo social importante al tratarse de una colaboración entre HP y Cruz Roja.

Respecto a las desviaciones presentes con relación a la planificación inicial, se puede decir que siempre es necesario realizar una mejor estimación temporal y valorar varias veces los riesgos, ya que este ha sido el principal problema de la gran demora.

El encontrar trabajo, los requisitos poco definidos por parte del cliente y por otros motivos personales que no proceden a ser explicados, se han producido los retrasos temporales no esperados. Pero esto me va a servir en un futuro para ser más previsor y a dejar cierto margen de tiempo para cada tarea. De esta manera, si surgen imprevistos temporales se podrán suplir con los márgenes establecidos.

Y finalmente, en cuanto a posibles mejoras estaría bien poder hacer una división de juegos en función del deterioro cognitivo del usuario, al igual que los niveles de dificultad. Con esto quiero decir, que para gente en mejor estado se les muestren juegos algo más complicados y que puedan seleccionar ellos mismos tener mayor dificultad, mientras que si su estado no es tan bueno disfrutar de juegos más sencillos pero que aun así supongan un pequeño reto a superar.

También se podrían recoger todos los datos de los resultados obtenidos y las demoras de tiempo para generar estadísticas de cada uno de los usuarios y ser comunicados o compartidos en una aplicación que puedan ver sus correspondientes médicos de cabecera para así estar al tanto de si se producen pérdidas de capacidades importantes.

.

Referencias

1. Arriola E., Canero C., Freire A., López R., López J.A., Manzano S., Olazarán J., *Deterioro Cognitivo Leve en el adulto mayor*. <https://www.segg.es/media/descargas/Consenso%20deteriorocognitivoleve.pdf>
2. Fillat Y., (diciembre, 2020) *Censo de personas con Alzheimer y otras demencias en España*. <https://www.ceafa.es/files/2020/12/censo-alz.pdf> [En línea]
3. Carolina, S., Osorno, D.A., 2013, Envejecimiento Cerebral y Cognoscitivo en el Adulto Mayor, *Revista Asociación Colombiana de Gerontología y Geriátrica*, Volumen 27(n°1), 1764-1774. <http://acgg.org.co//pdf/REVISTA%201-2013.pdf#page=34> [En línea]
4. ¿Qué causa el deterioro cognitivo en las personas mayores?. *Blog ISES* <https://www.isesinstituto.com/noticia/que-causa-el-deterioro-cognitivo-en-las-personas-mayores> [En línea]
5. CDC, *Alzheimers Disease and Healthy Aging*. (19 de diciembre 2019). <https://www.cdc.gov/aging/spanish/features/dementia.html> [En línea]
6. Sanitas. *Tipos de demencia*. <https://www.sanitas.es/sanitasresidencial/residencias-mayores/expertos-demencia/tipos-demencia>
7. Sociedad Española de Neurología (21 de septiembre 2021). <https://www.sen.es/saladeprensa/pdf/Link280.pdf> [En línea]
8. *Alzheimer Europe, Dementia in Europe Yearbook 2019*. https://www.alzheimer-europe.org/sites/default/files/alzheimer_europe_dementia_in_europe_yearbook_2019.pdf [En línea]
9. Susi T., Johannesson, M. y Backlund P., *Serious Games – An Overview*, (2 de mayo de 2007). <https://www.diva-portal.org/smash/get/diva2:2416/FULLTEXT01.pdf> [En línea]
10. America's Army. <https://www.americasarmy.com/> [En línea]
11. Nouchi R, Taki Y, Takeuchi H, Hashizume H, Akitsuki Y, Shigemune Y, et al. (2012) Brain Training Game Improves Executive Functions and Processing Speed in the Elderly: A Randomized Controlled Trial. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0029676>
12. Android Studio. <https://androidstudiofaqs.com/conceptos/android-studio-requisitos-minimos> [En línea]
13. LibGDX, <https://libgdx.com/> [En línea]
14. Audacity, <https://audacity.es/como-usar-audacia/> [En línea]
15. Cada trabajador está 42 días de baja al año. *La Razón*(3 de septiembre 2017).. <https://www.larazon.es/economia/cada-trabajador-esta-42-dias-de-baja-al-ano-HM15915992/> [En línea]
16. ¿Cuánto se gana en España, como Android developer?. Indeed. (19 de mayo de 2022). <https://es.indeed.com/career/android-developer/salaries> [En línea]

17. *Sueldo del Programador de Videojuegos en España.* Jobted.
<https://www.jobted.es/salario/programador-videojuegos> [En línea]
18. *Sueldo del Auxiliar Administrativo en España.* Jobted.
<https://www.jobted.es/salario/auxiliar-administrativo> [En línea]

Apéndices

Apéndice A: Manual de Instalación

Una vez dispongamos de nuestro archivo APK lo pasamos del ordenador a nuestro teléfono Android. Para ello debemos conectar el teléfono móvil al ordenador y permitir la transferencia de datos de uno a otro.

Cuando hayamos permitida la transferencia de datos, abrimos el explorador de archivos de nuestro sistema operativo y abrimos la ubicación de nuestro dispositivo.

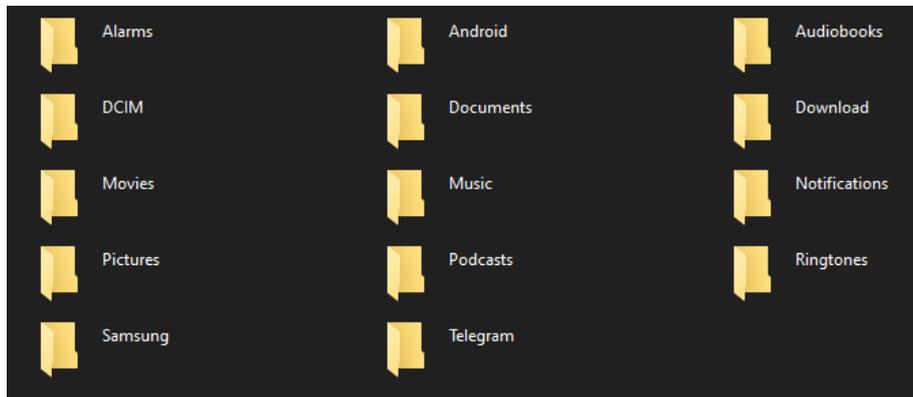


Figura A.1: Sistema de ficheros de Android

Aquí, podemos seleccionar donde queremos depositar nuestro archivo APK, por comodidad se puede meter dentro de *Download*.

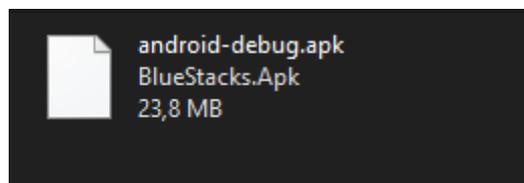


Figura A.2: Fichero APK de instalación

Completada esta fase, podemos desenchufar el dispositivo móvil del ordenador para proseguir con la instalación.

Posiblemente, tengamos que activar una opción de nuestro móvil para que nos permita instalar aplicaciones de terceros. Para ello, en un dispositivo Samsung, debemos seguir los siguientes pasos:

Primero debemos ir a ajustes, una vez dentro buscamos la siguiente opción, datos biométricos y seguridad.



Figura A.3: Opción del menú de ajustes

Una vez dentro, bajamos hasta encontrar la pestaña de Instalar apps desconocidas.



Figura A.4: Opción para aplicaciones desconocidas

Al seleccionar esta opción se nos mostrará otra pantalla donde se ven todas las posibles fuentes desde las que podemos instalar aplicaciones de terceros. Como lo hemos colocado el programa dentro del sistema de ficheros del dispositivo móvil hay que seleccionar como fuente “Mis Archivos”.

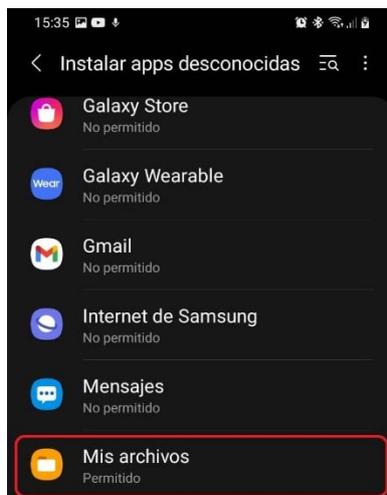


Figura A.5: Fuentes desde donde se permiten instalar aplicaciones desconocidas

Para finalizar con la fase de permisos, se abrirá una última pestaña donde deberemos habilitar el tick para poder conceder permisos.

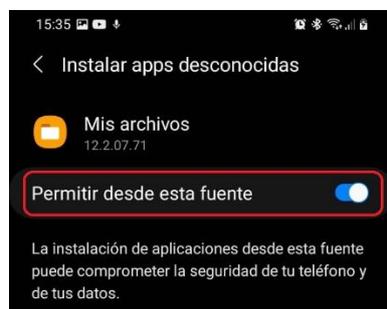


Figura A.6: Permiso de instalación de aplicaciones desconocidas

Dependiendo del dispositivo y de la versión Android del mismo, estos pasos pueden variar ligeramente. Una vez concedidos los permisos, ya podremos proceder con la instalación de la aplicación.

El primer paso consiste en abrir la aplicación mis archivos o el programa gestor de archivos del que disponga nuestro dispositivo. Una vez dentro, seleccionamos la opción marcada con el círculo rojo.

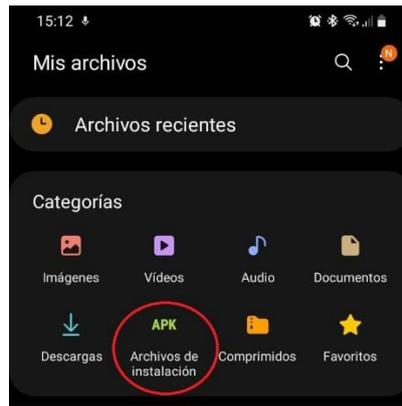


Figura A.7: Icono de localización de ficheros APK disponibles

Dentro de la ruta seleccionada se nos mostrarán todos los archivos APK que haya almacenados en nuestro dispositivo. Procedemos a seleccionar el archivo correcto, en nuestro caso no hemos modificado el nombre por defecto que genera Android Studio.

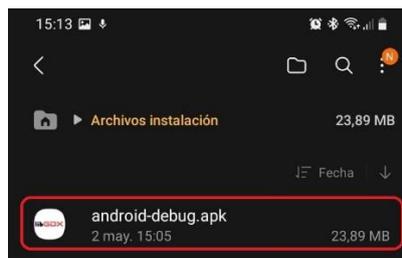


Figura A.8: Archivo APK de instalación

Una vez seleccionamos la aplicación que deseamos instalar, se nos muestra una ventana emergente que nos muestra las opciones Cancelar e Instalar. Seleccionamos instalar y dejamos que comience el proceso.

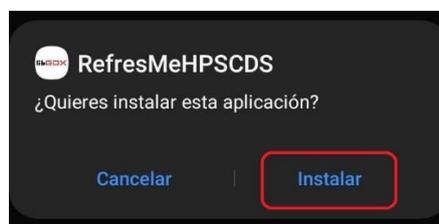


Figura A.9: Pestaña de confirmación instalación

Como se trata de una aplicación de terceros, volverá a solicitar que se apruebe la instalación. Debemos seleccionar la opción de Instalar de todas formas.



Figura A.10: Pestaña de confirmación de instalación de aplicación desconocida

Después de confirmar, se procede a la instalación de la aplicación. Al finalizar, se nos mostrará otra ventana emergente preguntando si deseamos abrir la aplicación o bien terminar el proceso de instalación sin abrir. Debemos seleccionar listo.



Figura A.11: Pestaña de confirmación de instalación

Antes de abrir la aplicación es probable que se nos muestre esta ventana donde se recomienda que se envíe la aplicación para su análisis en busca de malware a los servicios Google.

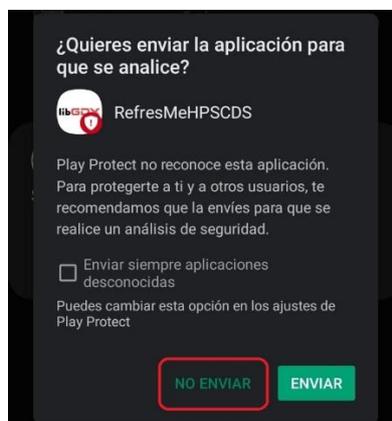


Figura A.12: Pestaña de confirmación de análisis de seguridad

Apéndice B: Manual de Usuario

Para lanzar los juegos lo único que debemos hacer es buscar el icono de la aplicación y pulsar sobre él.



Figura B.13: Icono estándar de aplicación LibGDX

Posteriormente, tras un periodo breve de carga de la aplicación se lanzará la aplicación mostrando el siguiente menú de la figura 2 apéndice II , desde el que deberemos seleccionar el juego al que deseamos jugar.

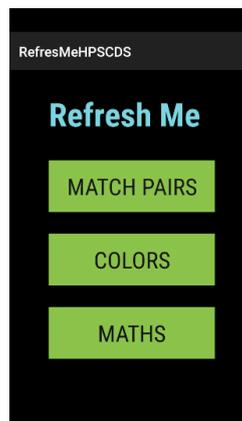


Figura B.2: Captura del menú de acceso

En las siguientes imágenes se mostrará el funcionamiento de los diferentes juegos. En primera instancia se presenta MatchPairs, destapamos un par de cartas, si estas no son iguales, ambas cartas se vuelven a tapar como se muestra al final de la secuencia.

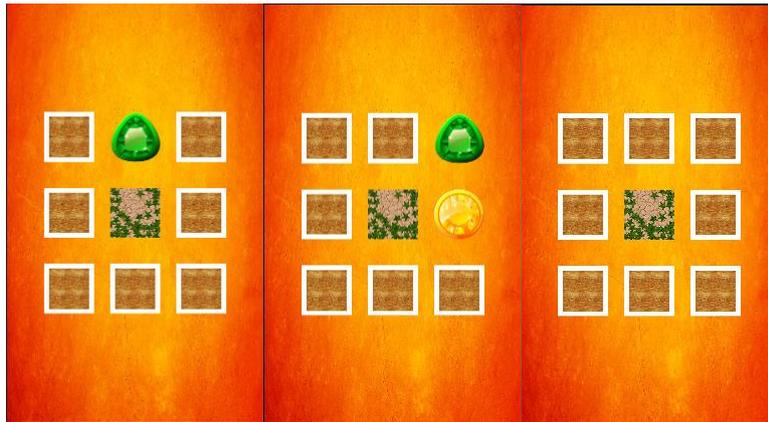


Figura B.3: Figuras no parejas

En caso de que acertemos la carta correspondiente estas dos quedan destapadas y seguimos con el procedimiento descubriendo el siguiente par de cartas. Si una vez hemos descubierto dos cartas y nos equivocamos en el siguiente par las ya adivinadas no se vuelven a tapar.

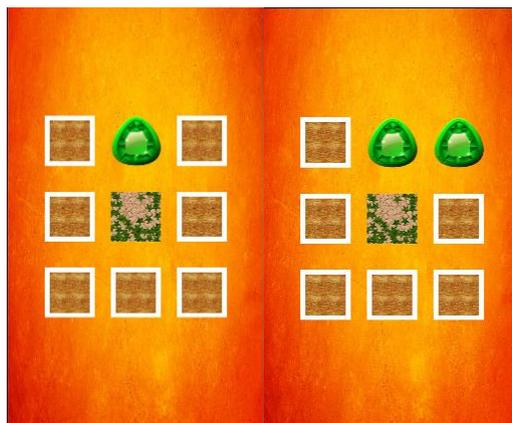


Figura B.4: Figuras parejas

En el juego de Colors, debemos ir memorizando la secuencia que nos muestran las figuras. La secuencia se marca por el inicio de la animación de una de las figuras acompañado de su sonido correspondiente, una vez acaba comienza la animación de la siguiente carta y sonido respectivamente.

La secuencia de figuras irá aumentando a medida que vayamos acertando, cada vez que acertemos la secuencia esta se incrementa en una figura a mayores, en cambio, si nos equivocamos la partida termina.



Figura B.5: Secuencia de figuras inicial en Colors

En Maths, se irán generando de manera aleatoria una serie de operaciones matemáticas, las cuales pueden ser sumas, restas o multiplicaciones y lo que debemos hacer es acertar la respuesta correcta, si acertamos acumulamos puntos y si no acertamos no los sumamos pasando a la siguiente operación.

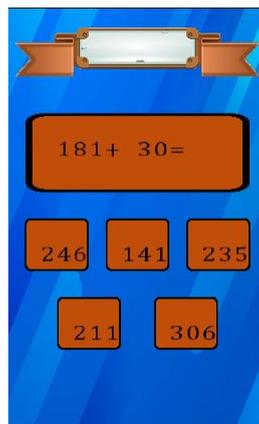


Figura B.6: Ejemplo de operación aritmética en Maths

Apéndice C: Generación de la dificultad de resultados en Maths

Aquí volvemos a tener cuatro niveles de dificultad posibles, estos niveles de dificultad se generan dependiendo del número de cifras que tendrán los números y el tamaño del resultado.

En la tabla C.1 se explica cómo se generan los dos operandos A y B que intervienen en la correspondiente operación.

Operación	Dificultad			
	Easy	Medium	High	Hard
Suma	A = [0, 9] B = [0, 9]	A = [0, 99] B = [0, 9]	A = [0, 99] B = [0, 99]	A = [0, 999] B = [0, 99]
Resta	A = [0, 9] B = [0, 9]	A = [0, 99] B = [0, 9]	A = [0, 99] B = [0, 99]	A = [0, 999] B = [0, 99]
Multiplicación	A = [1, 4] B = [1, 4]	A = [1, 6] B = [1, 6]	A = [1, 8] B = [1, 8]	A = [1, 99] B = [1, 5]

Tabla C.1: Tamaño de los operandos A y B generados para cada operación

En las siguientes tablas C.2, C.3 y C.4, K se corresponde al resultado correcto de la operación aleatoria que se haya generado, ya sea suma, resta o multiplicación. Emplearemos este resultado K para generar las posibles respuestas que se mostrarán al usuario. Estas respuestas se generan sumando o restando una cifra P (variable) que pertenecerá a un determinado intervalo. No se indica en esta tabla, pero se controla que no se generen resultados iguales.

Operación	Dificultad	Longitud de cifra resultado			
		1	2	3	4
Suma	<i>Easy</i>	$X_0 = K \pm P \in [1, 9]$ $X_1 = K + P \in [1, 9]$	No observable	No observable	No observable
	<i>Medium</i>	$X_0 = K \pm P \in [1, 99]$ $X_1 = K + P \in [1, 9]$	$X_0 = K \pm P \in [1, 99]$ $X_1 = K + P \in [1, 9]$	$X_0 = K \pm P \in [1, 99]$ $X_1 = K + P \in [1, 9]$	No observable
	<i>High</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 9]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	No observable
	<i>Hard</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 999]$

Tabla C.2: Generación de resultados para la operación suma

Operación	Dificultad	Longitud de cifra			
		1	2	3	4
Resta	<i>Easy</i>	$X_0 = K \pm P \in [1, 99]$ $X_1 = K \pm P \in [1, 99]$			
	<i>Medium</i>				
	<i>High</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$	No observable	No observable
	<i>Hard</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 9]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 9]$	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 9]$	No observable

Tabla C.3: Generación de resultados para la operación la operación resta

Operación	Dificultad	Longitud de cifra			
		1	2	3	4
Multiplicación	<i>Easy</i>	$X_0 = K \pm P \in [1, 99]$ $X_1 = K \pm P \in [1, 99]$			
	<i>Medium</i>				
	<i>High</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 99]$			
	<i>Hard</i>	$X_i, i = [0, 3]$ donde $X_i = K \pm P \in [1, 9]$			

Tabla C.4: Generación de resultados para la operación multiplicación

Como se puede ver en la tabla C.2, C.3 y C.4, para las dos primeras dificultades, *easy* y *medium*, el número de resultados posibles generados es 2, mientras que cuando aumentamos la dificultad a *high* o *hard* este número se eleva hasta 4, pero en ambos casos sólo hay un resultado correcto el cual no se genera, sino que ya está previamente calculado y se incluye en el listado de resultados finales mostrados al usuario.

Apéndice D: Contenido del CD

El contenido del CD-ROM que acompaña a esta memoria contiene:

- Fichero .zip del proyecto exportado
- Ejecutable de instalación APK
- Memoria del trabajo en formato PDF