

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**JUEGOS SERIOS PARA MEDIR CAPACIDADES
COGNOSCITIVAS USANDO DISPOSITIVOS
MOVILES**

**MARCOS MANUEL ABALLAY VALDENEGRO
LUIS JOAO ALARCÓN OLIVARES**

PROYECTO DE TÍTULO
PARA OPTAR AL TÍTULO PROFESIONAL DE
INGENIERO EN EJECUCIÓN EN INFORMÁTICA

DICIEMBRE 2012

PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA INFORMÁTICA

**JUEGOS SERIOS PARA MEDIR CAPACIDADES
COGNOSCITIVAS USANDO DISPOSITIVOS
MOVILES**

**MARCOS MANUEL ABALLAY VALDENEGRO
LUIS JOAO ALARCÓN OLIVARES**

Profesor Guía: **José Miguel Rubio León**
Profesor Co-referente: **Broderick Crawford Labrín**

Carrera: **Ingeniería en Ejecución en Informática**

DICIEMBRE 2012

Dedicatoria

Dedico este trabajo de título a mis familia, ya que gracias al esfuerzo de ellos y su sacrificio pude terminar mi carrera. Mi padre, quien trabajo duro durante todos estos años para poder entregarme el dinero y las cosas necesarias para poder vivir en Valparaíso, mi madre quien siempre me apoyo en mis decisiones y a pesar de tener días negros dentro de mi estancia en la universidad nunca dejo de estar presente y por último a mi hermano que siempre me entrego todas las cosas que mis padres no podían darme por falta de recursos o tiempo y que siempre estuvo ahí para darme consejos en momentos difíciles. A todos ellos y a mi familia en general, les dedico este trabajo de título.

Marcos Aballay

A mi familia por ser el pilar fundamental en todo lo que soy, en toda mi educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo. A Marcos que sin el equipo que formamos, no hubiéramos logrado esta meta.

Todo este trabajo ha sido posible gracias a ellos, gracias.

Luis Alarcón

Agradecimientos

Agradezco a mi profesor guía por darnos la oportunidad de llevar a cabo este proyecto que escapaba un poco a los estándares que se llevan dentro de la universidad, lo cual fue un poco arriesgado por nuestra parte, pero que al final logro frutos, tales como el segundo lugar en la feria del software 2012. También agradezco a mi compañero Luis, por soportar todos los problemas que tuvimos durante el desarrollo de nuestro proyecto y por entender en los momentos donde no tenía tanto tiempo como para trabajar. Para finalizar también agradezco a mi familia por creer en mí y darme la confianza para poder terminar mi carrera con éxito.

Marcos Aballay

Agradezco a mi familia por haberme apoyado en el transcurso de este proyecto, al profesor José Miguel Rubio por haber tenido la paciencia de corregir y evaluar los informes, a Diego Moraga por recordarme e indicarme los plazos para entregar esta tesis y a Sebastián Sandoval por ayudarme a corregir los errores de este proyecto.

Luis Alarcón

Índice

Resumen	v
Abstract	v
Índice de Figuras.....	vi
Índice de tablas	viii
1. Introducción.....	1
2. Descripción del problema.....	2
2.1. Estudio de inteligencias múltiples MIDAS.....	2
2.2. Estudio validez predictiva de la PSU sobre el éxito académico.....	7
3. Especificación de objetivos.....	14
3.1. Objetivo General.....	14
3.2. Objetivos Específicos.....	14
4. Plan de Trabajo	15
4.1. Fase Análisis del tema.....	15
4.2. Fase de diseño.....	16
4.3. Fase de implementación de prototipo	17
4.4. Fase de implementación.....	18
4.5. Fase de pruebas.....	19
5. Marco teórico.....	20
5.1. Juegos serios.....	20
5.2. Inteligencias Múltiples.....	23
5.3. Sistema operativo Android.....	25
5.4. Framework AndEngine	25
5.4.1. AndEngine Gles1	25
5.4.2. AndEngine Gles2.....	26
5.4.3. AndEngine Box2d extenxion	27
5.5. PhysicsEditor	28
5.6. Base de datos SQLite.....	28
6. Estudio de factibilidad	29
6.1. Estudio de factibilidad técnica.....	29
6.2. Estudio de factibilidad económica.....	30
6.2.1. Caracterización del consumidor	30

6.2.2.	Hábitos de consumo.....	31
6.2.3.	Cultura y sociedad inmersa	31
6.2.4.	Tasa de demanda del mercado y del proyecto actual y proyectado.....	31
6.2.5.	Competencias y ofertas del mercado actual	33
6.3.	Estudio de factibilidad operativa	34
6.4.	Estudio de factibilidad legal	34
7.	Estudio de inteligencias múltiples en alumnos de primer año.	35
7.1.	Selección de inteligencias múltiples	35
7.2.	Descripción del estudio	36
7.3.	Resultados del estudio.....	37
7.4.	Ponderación de puntaje de test	41
8.	Especificación de requerimientos	43
8.1.	Diagrama BPM	43
8.2.	Especificación de requerimientos funcionales	44
8.3.	Especificación de requerimientos no funcionales	47
9.	Análisis de requerimientos.....	48
9.1.	Enfoque utilizado.....	48
9.2.	Casos de uso	49
9.2.1.	Caso de uso general	49
9.2.1.1.	Caso de uso narrativo extendido para Smart Games.....	50
9.2.2.	Caso de uso juegos	51
9.2.2.1.	Caso de uso narrativo extendido para Juegos	52
9.2.3.	Caso de uso entrenamiento	53
9.2.3.1.	Caso de uso narrativo extendido para entrenamiento.....	54
9.2.4.	Caso de uso test.....	55
9.2.4.1.	Caso de uso narrativo extendido para test	55
9.2.5.	Caso de uso opciones	56
9.2.5.1.	Caso de uso narrativo extendido para Opciones	56
9.2.6.	Caso de uso puntajes	57
9.2.6.1.	Caso de uso narrativo extendido para puntajes.....	57
10.	Diagramas de actividad de los juegos	58
11.	Diseño.....	62
11.1.	Diagrama de clases.....	62
11.1.1.	Diagrama de clases principales	62

11.1.2.	Diagrama de clases test.....	64
11.1.3.	Diagrama de clases entrenamiento	66
11.1.4.	Diagrama de clase de juegos	67
11.1.5.	Diagrama de clases de puntajes	68
11.1.6.	Diagrama de clase matemáticas de figuras	68
11.1.7.	Diagrama de clase cañón de palabras	77
11.1.8.	Diagrama de clase mecanismo de escape	81
11.2.	Diagramas de secuencia.....	84
11.2.1.	Diagrama de secuencia realizar test.....	84
11.2.2.	Diagrama de secuencia realizar entrenamiento	85
11.2.3.	Diagrama de secuencia selección de opciones	86
11.2.4.	Diagrama de secuencia opción puntajes	87
11.2.5.	Diagrama de secuencia selección de juego	88
11.3.	Diseño de base de datos.....	89
11.4.	Diseño de interfaces	93
11.4.1.	Menú juegos	93
11.4.2.	Test.....	94
11.4.3.	Menú principal.....	95
11.4.4.	Menú opciones.....	96
11.4.5.	Puntajes	97
11.4.6.	Entrenamiento.....	98
12.	Diagrama de componentes.....	99
13.	Prototipo de la aplicación	101
13.1.	Menú principal y menú juegos.....	101
13.2.	Matemática de figuras	102
13.3.	Mecanismo de escape.....	102
13.4.	Cañón de palabras	103
13.5.	Menú opciones	103
13.6.	Test	104
13.7.	Puntaje	104
14.	Plan de Pruebas	105
14.1.	Propósito.....	105
14.2.	Entorno	105
14.3.	Alcance	105

14.4.	Visión general.....	105
14.5.	Sistema utilizado para las pruebas	106
14.6.	Pruebas a realizar	106
14.6.1.	Pruebas de interfaz.....	106
14.7.	Casos de prueba	107
14.8.	Resultado de las pruebas.....	110
15.	Conclusiones y trabajos futuros.....	113
16.	Bibliografía	115
Anexo	116
A:	Test de inteligencias	117

Resumen

El presente proyecto posee dos campos de investigación principales: los “Serious Games” y las inteligencias múltiples. Los “Serious Games” son una gama de juegos en que lo importante de ellos es adquirir algún tipo de conocimiento mientras se juega. La teoría de las inteligencias múltiples es propuesta por Howard Gardner y explica que la inteligencia está dividida en diferentes áreas independientes entre sí.

La idea principal de este proyecto es mezclar ambos conceptos y desarrollar una aplicación en el sistema operativo Android que ayude a cualquier persona que desee ingresar a una carrera de ingeniería; con el fin de estimar y apoyar su capacidad intelectual con respecto al ingreso a la universidad.

Se desarrolla una aplicación que constará con diferentes juegos que se utilizan para el apoyo de las capacidades intelectuales; además un test que separa y estima esas inteligencias, utilizando preguntas tipo PSU (Prueba de Selección Universitaria), dado que este tipo de preguntas tiene un factor predictivo sobre el éxito e ingreso a la universidad, lo cual es corroborado por diferentes estudios.

Palabras clave: Serious Games, inteligencias múltiples, capacidad intelectual, Framework AndEngine, Android.

Abstract

This project has two fields in which it focuses: Serious Games and multiple intelligences. Serious Games are a range of games it is important for them to acquire some knowledge while playing. The theory of multiple intelligences was proposed by Howard Gardner and explains that intelligence is divided into different areas independent.

The main idea of this project is to mix both concepts and develop an application on the Android operating system to help anyone wishing to enter a career in engineering, in order to estimate their intellectual capacity and support regarding university entrance.

It develops an application consisting of different games that are used to support the intellectual, plus a test that separates those intelligences and estimated using PSU (University Selection Test used in Chile) type questions, since the PSU score is a predictor of success and income the university, which is corroborated by several studies.

Keywords: Serious Games, multiple intelligences, intellectual capacity, AndEngine Framework.

Índice de Figuras

Figura 2.1 Gráfico nivel de significancia de los puntajes PSU en el promedio ponderado de notas con relación a los 4 años del estudio.....	12
Figura 2.2 Gráfico nivel de significancia de los puntajes PSU en el porcentaje de créditos aprobados con relación a los 4 años del estudio.	12
Figura 5.1 RescueSim. Videojuego para la formación de personal de rescate.....	22
Figura 5.2 Ejemplo uso de PhysicsEditor	28
Figura 7.1 Gráfico afinidad de inteligencia espacial en alumnos de primer año.....	37
Figura 7.2 Gráfico afinidad de inteligencia lingüística en alumnos de primer año.....	38
Figura 7.3 Gráfico afinidad de inteligencia lógico-matemática en alumnos de primer año.	39
Figura 7.4 Gráfico de rangos alumnos en el test de inteligencia espacial.....	40
Figura 7.5 Gráfico de rangos alumnos en el test de inteligencia lingüística.	40
Figura 7.6 Gráfico de rangos alumnos en el test de inteligencia lógico-matemático.	41
Figura 8.1 Diagrama de proceso de negocio a nivel general.....	43
Figura 9.1 Caso de uso general de la aplicación Smart Games.....	49
Figura 9.2 Caso de uso para la opción de juegos.....	51
Figura 9.3 Caso de uso para la opción Entrenamiento.....	53
Figura 9.4 Caso de uso para la opción Test.....	55
Figura 9.5 Caso de uso para las opciones del sistema.	56
Figura 9.6 Caso de uso para opción de puntajes.....	57
Figura 10.1 Diagrama de Cañón de palabras.....	58
Figura 10.2 Diagrama de Mecanismo de escape.	59
Figura 10.3 Diagrama de Matemática de las figuras	60
Figura 11.1 Diagrama de clases principales.	62
Figura 11.2 Diagrama de clases test.	64
Figura 11.3 Diagrama de clase entrenamiento.	66
Figura 11.4 Diagrama de clase menú juegos.....	67
Figura 11.5 Diagrama de clase puntaje	68
Figura 11.6 Diagrama de clases Matemática de las figuras.	68
Figura 11.7 Atributos Matemáticas de las figuras.	69
Figura 11.8 Métodos Matemática de las figuras.....	72
Figura 11.9 Clases pilas para figuras.	75
Figura 11.10 Clase figura	75
Figura 11.11 Diagrama de clase Countdown.....	76
Figura 11.12 Diagrama de clase cañón de palabras.	77
Figura 11.13 Atributos clase cañón de palabras.	78
Figura 11.14 funciones de clase cañón de palabras.	79
Figura 11.15 Diagrama de clase mecanismo de escape	81
Figura 11.16 Atributos de la clase mecanismo de escape.....	82
Figura 11.17 Metodos de la clase mecanismo_escape.....	83

Figura 11.18 Diagrama de secuencia realizar test.	84
Figura 11.19 Diagrama de secuencia realizar entrenamiento.....	85
Figura 11.20 Diagrama de secuencia selección de opciones.....	86
Figura 11.21 Diagrama de secuencia opción puntajes.....	87
Figura 11.22 Diagrama de secuencia selección de juego.....	88
Figura 11.23 Diseño prototipo del menú juegos.....	93
Figura 11.24 Diseño prototipo de test.....	94
Figura 11.25 Diseño prototipo de menú principal.	95
Figura 11.26 Diseño prototipo menú opciones.....	96
Figura 11.27 Diseño prototipo de puntajes.	97
Figura 11.28 Diseño prototipo de entrenamiento.	98
Figura 12.1 Diagrama de componente.	99
Figura 13.1 Prototipo de menú principal y menú juegos.	101
Figura 13.2 Prototipo matemática de figuras.	102
Figura 13.3 Prototipo mecanismo de escape.	102
Figura 13.4 Prototipo cañón de palabras.....	103
Figura 13.5 Prototipo menú opciones.	103
Figura 13.6 Prototipo test.....	104
Figura 13.7 Menú de puntaje y puntaje de test.....	104

Índice de tablas

Tabla 2.1 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Humanas parte 1.	4
Tabla 2.2 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Humanas parte 2.	4
Tabla 2.3 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Económicas.	5
Tabla 2.4 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ingeniería.	5
Tabla 2.5 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales.	6
Tabla 2.6 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Agronomía y Veterinaria.	6
Tabla 2.7 Alumnos de la muestra separados por género parte 1.	8
Tabla 2.8 Alumnos de la muestra separados por género parte 2.	9
Tabla 2.9 Tabla de nivel de significancia en el promedio ponderado de notas de NEM y puntajes PSU.	11
Tabla 2.10 Tabla de nivel de significancia en el porcentaje de créditos aprobados de NEM y puntajes PSU.	11
Tabla 4.1 Fechas y duración de actividades fase de análisis.	15
Tabla 4.2 Fechas y duración de actividades fase de análisis.	16
Tabla 4.3 Fechas y duración de actividades fase de implementación de prototipo.	17
Tabla 4.4 Fechas y duración de actividades fase de Diseño entre Julio y Septiembre.	18
Tabla 4.5 Fechas y duración de actividades fase de Diseño entre Septiembre y Noviembre.	19
Tabla 4.6 Fechas y duración de actividades fase de Pruebas.	19
Tabla 5.1 Géneros de video juegos.	20
Tabla 5.2 Taxonomía de juegos serios parte 1.	21
Tabla 5.3 Taxonomía de juegos serios parte 2.	22
Tabla 5.4 Inteligencias múltiples propuestas por Gardner.	24
Tabla 6.1 Costos de herramientas técnicas.	30
Tabla 6.2 Costos de horas de trabajo.	30
Tabla 6.3 Valor actual neto.	32
Tabla 6.4 Productos similares.	33
Tabla 7.1 Afinidad de alumnos por la inteligencia espacial.	37
Tabla 7.2 Afinidad de alumnos por la inteligencia lingüística.	38
Tabla 7.3 Afinidad de alumnos por la inteligencia lógico-matemática.	38
Tabla 7.4 Ponderación de puntajes de test.	42
Tabla 8.1 Requerimientos funcionales numero 1 a 3.3.	44
Tabla 8.2 Requerimientos funcionales numero 3.3.1 a 3.3.3.	45
Tabla 8.3 Requerimientos funcionales números 3.3.3.1 a 4.6.	46
Tabla 8.5 Requerimientos funcionales números 5 a 6.3.	47
Tabla 8.6 Requerimientos no funcionales.	47
Tabla 9.1 Caso de uso narrativo Smart Games.	50

Tabla 9.2 Caso de uso narrativo Juegos.....	52
Tabla 9.3 Caso de uso narrativo Entrenamiento.....	54
Tabla 9.4 Caso de uso narrativo Test.....	55
Tabla 9.5 Caso de uso narrativo Opciones.....	56
Tabla 9.6 Caso de uso narrativo Puntajes.....	57
Tabla 14.1 Documentos utilizados para realizar el plan de prueba.....	105
Tabla 14.2 Describe los componentes utilizados en el sistema en el que se realizan las pruebas	106
Tabla 14.3 Casos de prueba.....	107
Tabla 14.4 Casos de prueba de interfaz.....	108
Tabla 14.5 Casos de prueba de interfaz.....	109
Tabla 14.6 Resultado de casos de prueba.....	110
Tabla 14.7 Resultados de casos de prueba.....	111
Tabla 14.8 Resultados de los casos de prueba.....	112

1. Introducción

En este proyecto se presenta una aplicación para “Smartphone” Android que tiene como objetivo apoyar la capacidad de razonamiento de alumnos que deseen ingresar a una carrera de ingeniería a través del uso de la teoría de las inteligencias múltiples propuesta por Howard Gardner, para el desarrollo tanto de juegos que apoyan el razonamiento, como un test que permitirá comprobar tanto el éxito como el rendimiento en la Universidad.

Se llega a la conclusión de que la mejor manera para captar público es a través de un juego, ya que por medio del mismo se logra introducir ideas sobre las inteligencias y al mismo tiempo hacerlo atractivo para los usuarios.

El test contiene preguntas tipo PSU, las cuales, están comprobadas que afectan directamente tanto el éxito como el rendimiento universitario, lo anterior se incluye en un estudio que apoya esta hipótesis

Dos de los juegos se basan principalmente en la teoría de las inteligencias múltiples, mientras que el tercer juego está basado en preguntas tipo PSU.

Debido a la explosiva masificación de los dispositivos móviles, “Smartphones”, “Tablets” y otros dispositivos que utilizan el sistema Android, la aplicación se desarrolló para este sistema operativo en particular.

A lo largo de este informe se explica lo siguiente: el diseño de las clases a implementar; detallando cada una de ellas y el uso de diagramas de secuencia para mostrar la relación entre las clases y la secuencia de operaciones que se ejecutan al realizar una acción sobre la aplicación.

Además se muestran algunas imágenes del prototipo desarrollado y las funcionalidades implementadas.

Por último, se presentan las pruebas funcionales realizadas para comprobar si la aplicación cumple con los requisitos planteados, en caso que exista algún problema, se indica cómo se implementa la solución a ese problema.

2. Descripción del problema

La premisa principal es estimar y apoyar la capacidad de razonamiento de un alumno que desea entrar a la carrera de ingeniería, a través de la creación de juegos serios y la utilización de la teoría de inteligencias múltiples.

Para ello se desarrolla una aplicación que permite estimar y comprobar si un alumno tiene la capacidad intelectual necesaria para ingresar a una carrera de ingeniería a través de un test de inteligencias múltiples.

Identificar las inteligencias múltiples que tienen mayor influencia en la carrera de ingeniería y apoyar al usuario en su intento por ingresar a la universidad utilizando para ello serious games, con fundamento en la teoría de las inteligencias múltiples.

2.1. Estudio de inteligencias múltiples MIDAS.

Para comprobar si la teoría de las inteligencias múltiples es un factor viable y en caso de que lo sea, comprobar si una persona es apta para ingresar a una carrera de ingeniería, se realizó el siguiente estudio sobre alumnos de las carreras de ingeniería.

La teoría de inteligencias múltiples usa cuestionarios sobre gustos como por ejemplo el MIDAS (“Multiple Intelligences Developmental Assessment Scales”) escrito por Branton *Shearer*, el que permite conocer el perfil intelectual auto percibido de los sujetos, lo cual solo sirve para demostrar la afinidad que se tiene hacia una u otra inteligencia.

En este caso Daiana Yamila Rigo (Magíster en Psicología de la Educación) y Danilo Donolo (Doctor en Ciencias de la Educación) de la Universidad Nacional de Río Cuarto (Córdoba, Argentina) realizan un estudio aplicando el test MIDAS de adultos sobre alumnos de entre 20 y 47 años de la misma universidad.

El objetivo de su prueba es estudiar los perfiles de inteligencias múltiples en un grupo de alumnos que cursan distintas carreras.

Para diversificar aún más la muestra se seleccionan alumnos de diversos años, de los 638 alumnos 44 sujetos asisten a primer año, 163 a segundo año, 203 a tercer año, 110 a cuarto año, 90 a quinto año y 22 que cursaban sexto año; la edad cronológica del grupo fue de 22 años y 2 meses y por último el 53% de la muestra estuvo conformada por mujeres por lo que la muestra es relativamente homogénea.

La población estudiada estaba integrada por cinco facultades: Facultad de Ciencias Humanas 169 alumnos, Facultad de Ciencias Económicas 134 alumnos, Facultad de Ciencias

Exactas, Físico-Químico y Naturales 111 alumnos, Facultad de Ingeniería 95 y Facultad de Agronomía y Veterinaria 123 alumnos. En total se contó con 28 carreras a ser estudiadas.

La aplicación del test se realiza por escrito y de manera colectiva, en el cuestionario se toman 8 inteligencias múltiples (lógico-matemática, espacial, lingüística, intrapersonal, interpersonal y naturalista) y el formato de respuestas es de tipo Likert por lo que las preguntas tienen el siguiente formato:

¿Cuándo eras niño te gustaba escuchar música o asistir a clases de música?

- a) Totalmente en desacuerdo
- b) En desacuerdo
- c) Ni de acuerdo ni en desacuerdo
- d) De acuerdo
- e) Totalmente de acuerdo

Y usa la siguiente escala de calificaciones:

- 80-100 muy alto.
- 79-60 alto.
- 59-40 moderado.
- 39-20 bajo.
- 19-0 muy bajo.

La siguiente tabla muestra la media y desviación estándar del test aplicado en la Facultad de Ciencias Humanas de la Universidad Nacional de Río Cuarto, compuesta por alumnos de 11 carreras:

- Alumnos de licenciatura en psicopedagogía, en educación especial, políticas, abogacía, profesorado y licenciatura en educación inicial, comunicación, geografía e Historia, presentan preferencias sobre las inteligencias lingüística, interpersonal e intrapersonal.
- Alumnos de profesorado en ciencias jurídicas, con preferencia sobre las inteligencias lingüística, interpersonal y cinestésica.
- Alumnos de profesorado y licenciatura en inglés, con preferencias en la inteligencia lingüística, interpersonal y musical respectivamente.
- Alumnos de profesorado y la licenciatura en educación física destacándose en la inteligencia lingüística, interpersonal y cinestésica.
- Alumnos que cursan la Licenciatura en Enfermería, tienen preferencias por la inteligencia interpersonal, intrapersonal y musical.

Tabla 2.1 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Humanas parte 1.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-Personal	Intra-personal	Naturalista
Lic. En Psicopedagogía	M	36.41	38.20	40.55	41.77	51.66	63.95	50.86	39.89
	DT	15.11	17.13	14.72	15.37	13.10	10.83	11.70	15.66
Lic. en Educación Especial	M	39.53	40.06	38.41	36.12	49.65	63.06	51.59	40.76
	DT	13.35	18.40	11.81	13.36	13.47	11.69	10.61	13.73

Tabla 2.2 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Humanas parte 2.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-personal	Intra-personal	Naturalista
Abogacía	M	44.37	47.19	46.94	42.88	53.94	54.69	54.13	43.19
	DT	15.51	18.75	10.96	13.95	13.67	9.22	10.11	12.45
Prof. y Lic. En Cs de la Comunicación	M	43.30	36.90	38.70	38.60	58.50	63.60	49.60	41.60
	DT	13.90	11.99	11.20	15.61	10.95	11.88	10.97	20.03
Prof. y lic. En inglés	M	47.67	43.17	47.17	37.67	54.33	58.83	46.50	45.00
	DT	14.60	12.24	11.37	6.35	14.64	16.70	8.71	6.96
Prof. en Cs. Jurídicas	M	32.50	49.50	37.00	41.50	53.00	54.00	41.50	48.25
	DT	19.47	15.76	16.10	19.28	10.10	5.72	9.26	18.43
Lic. en Cs. Políticas	M	36.91	47.09	49.91	48.00	60.55	60.09	55.64	51.55
	DT	15.19	16.13	11.15	9.63	7.34	10.60	8.32	11.75
Prof. y Lic. En educación física	M	40.52	61.00	42.06	42.42	47.85	61.39	53.15	41.67
	DT	13.46	16.85	13.46	11.93	14.53	12.00	12.97	16.77
Lic. en enfermería	M	39.50	33.50	33.17	33.67	38.83	51.33	39.00	29.17
	DT	11.61	19.23	13.32	15.50	11.30	13.17	10.84	22.07

La siguiente tabla muestra los resultados del test midas aplicado a alumnos que cursan alguna carrera en la facultad de ciencias económicas. Estos presentan resultados similares a los alumnos de la facultad de ingeniería, prefiriendo las inteligencias lógico-matemática, interpersonal e intrapersonal.

Tabla 2.3 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Económicas.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-Personal	Intra-Personal	Naturalista
Lic. en Administración	M	37.64	45.19	48.97	43.17	47.58	57.69	55.19	38.81
	DT	16.23	17.34	11.75	13.54	15.62	15.80	11.73	15.00
Contador público	M	33.24	44.60	51.40	42.27	46.41	56.44	55.23	38.64
	DT	13.87	13.80	10.44	9.39	12.84	12.08	9.80	14.16
Lic. Economía	M	39.07	43.39	61.50	47.71	53.75	58.96	63.07	46.75
	DT	18.28	17.44	12.20	18.80	13.87	11.90	10.28	19.09

Tabla 2.4 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ingeniería.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-personal	Intra-personal	Naturalista
Ingeniería electricista	M	28.38	57.50	48.75	45.00	47.25	55.88	55.13	47.75
	DT	11.08	13.42	15.93	15.22	14.43	9.43	10.22	12.67
Ingeniería en química	M	32.56	39.95	56.37	47.07	45.22	52.98	56.68	42.27
	DT	18.67	15.35	11.62	13.25	16.41	13.49	12.09	17.03
Ingeniería mecánica	M	40.59	49.06	54.82	51.41	42.65	53.76	56.76	40.82
	DT	18.37	14.69	15.50	17.26	13.70	12.66	13.43	17.86
Ingeniería en telecomunicación	M	43.90	54.03	59.66	53.07	52.03	58.55	61.83	48.21
	DT	16.72	12.02	14.94	14.00	13.94	11.25	10.90	12.90

La siguiente tabla muestra la media y la desviación estándar de los resultados obtenidos por el test midas por los alumnos de la facultad de Ciencias Exactas, Físico-Químicas y Naturales donde:

Alumnos de Microbiología, Profesorado, Licenciatura en Geología y técnico de laboratorio tienen preferencia sobre la inteligencia interpersonal, intrapersonal y naturalista; en el caso de técnico de laboratorio también muestran preferencia por la inteligencia lógico-matemática.

En el caso de los alumnos de Profesorado y Licenciatura en Ciencias Biológicas muestran preferencia por inteligencias naturalista, lógico-matemática e intrapersonal.

Por último, los alumnos de las carreras de Profesorado y la Licenciatura en Matemáticas prefieren inteligencias interpersonal, lógico-matemática e intrapersonal.

Tabla 2.5 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Ciencias Exactas, Físico-Químicas y Naturales.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-personal	Intra-personal	Naturalista
Prof. Y Lic. Analista químico	M	44.09	56.26	64.22	54.22	53.43	62.65	65.39	59.74
	DT	13.20	17.94	13.11	11.18	15.80	10.94	14.09	14.96
Prof. Y Lic. En ciencias biológicas	M	38.88	40.76	54.76	49.41	48.41	52.00	55.59	72.12
	DT	18.43	13.38	15.76	12.07	8.81	9.25	14.09	9.71
Microbiología	M	41.52	49.19	54.31	47.98	53.67	58.79	57.02	55.60
	DT	17.68	12.30	12.75	14.65	12.83	11.12	10.05	17.76
Técnico en laboratorio	M	38.17	42.17	68.67	60.17	58.17	65.50	65.50	68.50
	DT	20.68	28.29	18.67	17.84	17.84	16.28	15.68	29.11
Prof. Y Lic. En geología	M	48.67	40.67	55.00	47.67	48.00	59.00	61.67	65.00
	DT	14.29	5.13	15.10	18.01	23.81	20.07	18.61	15.72
Analista en computación	M	31.80	47.53	45.80	35.53	39.47	49.33	48.20	35.93
	DT	13.73	12.12	11.01	10.01	12.26	12.93	10.87	16.74
Prof. Y Lic. En matemáticas	M	33.60	35.80	56.40	50.60	45.80	56.20	61.80	43.40
	DT	23.35	10.13	15.68	12.03	9.36	10.28	12.05	12.46

La última tabla muestra la media y desviación estándar de los resultados del test midas aplicados a alumnos de la facultad de agronomía y veterinaria; las carreras de Ingeniería Agronómica y medicina veterinaria poseen preferencias en las inteligencias naturalista, interpersonal e intrapersonal.

Tabla 2.6 Medias y desviaciones típicas para las ocho inteligencias según carreras de la Facultad de Agronomía y Veterinaria.

Carreras		Musical	Cinestésica	Lógico-Matemática	Espacial	Lingüística	Inter-personal	Intra-personal	Naturalista
Ingeniería agronómica	M	41.41	45.46	48.98	44.02	47.85	54.29	52.41	59.10
	DT	17.39	16.22	12.25	12.95	11.38	11.11	11.35	13.69
Medicina veterinaria	M	41.88	45.82	44.04	43.09	47.85	54.07	50.12	61.72
	DT	13.96	14.98	10.95	15.12	15.64	13.43	9.84	13.36

Lo que se deduce de este estudio es que en cada carrera, existe una preponderancia de las inteligencias intrapersonal e interpersonal, lo que indica que los alumnos pueden relacionarse mejor con las demás personas al comprenderlos mejor, además comprenderse mejor a sí mismo; por otro lado, el estudio demuestra que los alumnos de distintas carreras tienen mayor afinidad a inteligencias que tienen directa relación con sus carreras por ejemplo los alumnos de la carrera de técnico de laboratorio tienen mayor preferencia sobre inteligencias como lógico- matemática e inteligencia naturalista.

La dispersión mostrada por la desviación estándar indica la variabilidad de las preferencias de cada alumno para adecuar sus inteligencias.

Se destaca de este estudio que los resultados tienen validez con otros estudios realizados por el autor del test midas [3] Branton Shearer sobre estudiantes universitarios de los Estados Unidos de Norte América; el estudio hecho por Branton Shearer muestran altas preferencias sobre las inteligencias interpersonal e intrapersonal con un 60% y 56% respectivamente.

En el caso del estudio aquí presentado muestra un porcentaje de preferencias por las inteligencias interpersonal e intrapersonal de un 57% y 54% respectivamente. En otros países tales como España, Puerto Rico, Chile, Reino Unido, Islandia, Singapur, Irlanda, Corea, Canadá y Hong Kong los resultados fueron similares según se señala en el libro de Branton Shearer [3].

2.2. Estudio validez predictiva de la PSU sobre el éxito académico.

El estudio anterior demuestra que la mayoría de los alumnos tiene tendencia a utilizar o preferir las inteligencias intrapersonal e interpersonal, aquí es donde empiezan los problemas: ¿Cómo estimar estas inteligencias?, ¿estas inteligencias permiten tener éxito tanto como para entrar a la universidad y egresar de ella?

Si se estuviera hablando sobre el entorno laboral se diría: “si es necesario”, pero, en este caso la medición de éxito con las inteligencias interpersonal está limitada por diversos factores.

Del estudio anterior, se concluye que además de poseer esas dos inteligencias, los alumnos dependiendo de la carrera a la cual pertenecen tienen preferencia por una u otra inteligencia.

El factor con directa relación con lo anterior es el factor que principalmente imposibilita el ingreso a la universidad con solo las inteligencias intrapersonal e interpersonal, es llamada PSU.

La Prueba de Selección Universitaria que mide conocimientos en áreas de matemáticas, lenguaje y comunicación, historia y geografía, química, física y biología, aprendidos por los alumnos desde primero a cuarto medio.

El propósito fundamental de las pruebas es ordenar a los postulantes que aspiran a ingresar a las diversas carreras ofrecidas por las universidades del Consejo de Rectores; estas pruebas además deben predecir el éxito y rendimiento de los estudiantes de educación superior [2].

Estos estudios son realizados por El Comité Técnico Asesor del Consejo de Rectores de las Universidades Chilenas en su documento “Estudio acerca de la Validez Predictiva de los Factores de Selección a las Universidades del Consejo de Rectores” y la memoria de Título “La PSU y otros factores de rendimiento y éxito académico universitario: El caso de la Pontificia Universidad Católica de Valparaíso” [8].

En este apartado se tratará de los resultados arrojados por el estudio de la memoria de título el cual tuvo como objetivo general “identificar las principales variables que influyen en el rendimiento y éxito académico de los alumnos de la PUCV ingresados entre los años 2004 a 2007”.

En el estudio se utilizó una muestra de 11.200 alumnos que ingresaron a la universidad entre los años 2004 a 2007, de lo cual se cubre desde el primer año de aplicación de la PSU, evaluando el rendimiento académico entre primero a cuarto año, de forma en que se puedan evaluar la influencia de los distintos factores a través del tiempo.

La muestra usada para el estudio está compuesta por 11.200 alumnos de la PUCV de 51 carreras impartidas, la cantidad de alumnos y las carreras a las que pertenecen se muestra en la siguiente tabla:

Tabla 2.7 Alumnos de la muestra separados por género parte 1.

Carrera	Estudiantes		
	Masculinos	Femeninos	Total
Agronomía	54,60%	45,40%	463
Bachillerato en ciencias	44,20%	55,80%	190
Biología y ciencias naturales	28,20%	71,80%	177
Biólogo	42,90%	57,10%	156
Bioquímica	41,90%	58,10%	203
Castellano y comunicación	29,60%	70,40%	233
Ciencias religiosas	50,00%	50,00%	22
Contador auditor	42,60%	57,40%	305
Derecho	48,80%	51,20%	685
Educación diferencial	5,60%	94,40%	214
Educación física	58,30%	41,70%	230
Educación general básica	18,10%	81,90%	193
Educación musical	66,70%	33,30%	126
Educación parvulario	0,60%	99,40%	169
Estadística	55,70%	44,30%	131
Filosofía	61,80%	38,20%	123
Física	71,00%	29,00%	131
Geografía	58,50%	41,50%	171
Historia, geografía y ciencias sociales	47,70%	52,30%	260

Tabla 2.8 Alumnos de la muestra separados por género parte 2.

Carrera	Estudiantes		
	Masculinos	Femeninos	Total
Ingeniería civil	79,50%	20,50%	210
Ingeniería civil bioquímica	46,60%	53,40%	277
Ingeniería civil eléctrica	93,90%	6,10%	165
Ingeniería civil electrónica	96,60%	3,40%	328
Ingeniería civil industrial	67,10%	32,90%	593
Ingeniería civil informática	87,70%	12,30%	308
Ingeniería civil mecánica	95,30%	4,70%	279
Ingeniería civil metalurgia extractiva	82,00%	18,00%	122
Ingeniería civil química	61,80%	38,20%	204
Ingeniería de alimentos	27,40%	72,60%	230
Ingeniería de ejecución en bioprocesos	47,30%	52,70%	129
Ingeniería en ejecución en informática	88,60%	11,40%	342
Ingeniería de transporte	79,50%	20,50%	171
Ingeniería eléctrica	94,20%	5,80%	190
Ingeniería electrónica	95,80%	4,20%	264
Ingeniería en acuicultura	57,00%	43,00%	149
Ingeniería en construcción	81,60%	18,40%	304
Ingeniería en mecánica	96,10%	3,90%	232
Ingeniería pesquera	66,70%	33,30%	114
Inglés	28,80%	71,20%	205
Interpretación- traducción inglés al español	36,50%	63,50%	222
Kinesiología	36,10%	63,90%	227
Licenciatura en arte	26,50%	73,50%	102
Matemática	56,20%	43,80%	242
Oceanografía	46,70%	53,30%	90
Óptica	32,60%	67,40%	95
Periodismo	39,50%	60,50%	185
Psicología	37,80%	62,20%	225
Química Industrial	55,40%	44,60%	130
Química y ciencias naturales	31,30%	68,70%	112
Trabajo social	12,00%	88,00%	191
Totales	56,30%	43,70%	11200

La muestra no incluye a alumnos que aunque tengan el puntaje necesario para ingresar a la PUCV, no ingresan; ni tampoco a los alumnos que no tuvieron el puntaje necesario.

Como información para el estudio se utilizaron los siguientes datos:

- Puntajes obtenidos en la rendición de la PSU.
- Promedio de notas de enseñanza media,
- Carrera en la que se matricularon, año de ingreso.
- Puntaje ponderado (según requerimiento de cada carrera).
- Promedios simples y ponderados, con y sin ramos aprobados cada semestre.
- Créditos inscritos y créditos aprobados.
- Situación académica del alumno (sancionado, eliminado).
- Año de egreso de enseñanza media.
- Información laboral de los alumnos.

Y otras variables socioeconómicas tales como:

- Género.
- Rama educacional.
- Tipo de establecimiento medio de egreso (liceo técnico o científico humanista).
- Nivel educacional de los padres.
- Ingreso familiar.
- Lugar de residencia.

Además de utilizar el promedio ponderado de notas y el porcentaje de créditos aprobados para medir el rendimiento académico del alumno, se usa la variable sancionados, como representativa del éxito académico; la variable “sancionado” se define como los casos planteados en el reglamento general de estudios de pregrado de la PUCV en los artículos 28°, 33° y 45°.

Estos datos se usan como variables dependientes en el método de estimación que se utiliza para calcular la variable de éxito y rendimiento académico; ya que indica que si el alumno logra avanzar en el plan de estudios, y mantiene un ritmo de avance curricular no inferior a lo estipulado en el reglamento general de estudios de pregrado de la PUCV.

Como variables independientes se utilizan:

- NEM.
- Puntajes PSU:
 - Matemática.
 - Lenguaje y comunicación
 - PSU Optativa.
- Promedio PSU.
- Puntaje ponderado.

Se utiliza un modelo de regresión lineal múltiple, el cual permite controlar las diversas variables que influyen sobre la variable dependiente, además ya ha sido usado en otros

estudios tanto a nivel nacional como internacional; para las variables dependientes se utiliza la siguiente ecuación general:

$$= + + + +$$

$$= 1, \dots,$$

Donde es la variable de éxito y/o rendimiento académico, variables asociadas a la prueba de selección universitaria y notas de enseñanza media, corresponde a variables individuales, familiares y de entorno, corresponde al vector de variables de control y representa el error.

Para indicar cuanta relevancia tiene una variable respecto al promedio ponderado de notas (en la universidad) y a los créditos aprobados se usa el test de significancia el cual permite ver cuando una variable influye sobre otra y no es probable que esa relación se deba al azar, el cual se muestra en la siguiente tabla:

Tabla 2.9 Tabla de nivel de significancia en el promedio ponderado de notas de NEM y puntajes PSU.

	1er Año		2do Año		3er Año		4to Año	
	Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo
NEM	0,63374	0,68162	0,66420	0,71928	0,67186	0,73324	0,67014	0,72307
PSU Lenguaje	0,00035	0,00042	0,00033	0,00040	0,00035	0,00039	no significativa	no significativa
PSU Matemática	0,00589	0,00619	0,00514	0,00547	0,00481	0,00514	0,00470	0,00507
Prueba optativa	0,00170	0,00223	0,00172	0,00216	0,00180	0,00220	0,00143	0,00180

Tabla 2.10 Tabla de nivel de significancia en el porcentaje de créditos aprobados de NEM y puntajes PSU.

	1er Año		2do Año		3er Año		4to Año	
	Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo	Mínimo	Máximo
NEM	0,18098	0,19603	0,18440	0,20070	0,18928	0,20734	0,19145	0,20732
PSU Lenguaje	0,00012	0,00014	0,00009	0,00011	0,00010	0,00011	no significativa	no significativa
PSU Matemática	0,00183	0,00191	0,00151	0,00166	0,00141	0,00150	0,00138	0,00149
Prueba optativa	0,00054	0,00070	0,00053	0,00066	0,00057	0,00069	0,00048	0,00059

Lo que indica la primera tabla es por ejemplo en el caso del máximo de primer año; el puntaje de PSU matemáticas (0,00619) un aumento en 100 puntos en esta prueba permite un aumento aproximado de 6 décimas al promedio ponderado de notas a la universidad.

En el caso de la segunda tabla en el mismo caso anterior (0,00191), 100 puntos más en la PSU de matemática permite un aumento aproximado del 19% de créditos aprobados.

En el caso de del NEM tener un punto más en el promedio de enseñanza media produce un aumento entre 6 a 7 décimas en promedio ponderado de notas y entre 18% y 20% de aumento de créditos aprobados.

Los siguientes gráficos muestran los datos de las tablas anteriores en relación a los puntajes PSU:

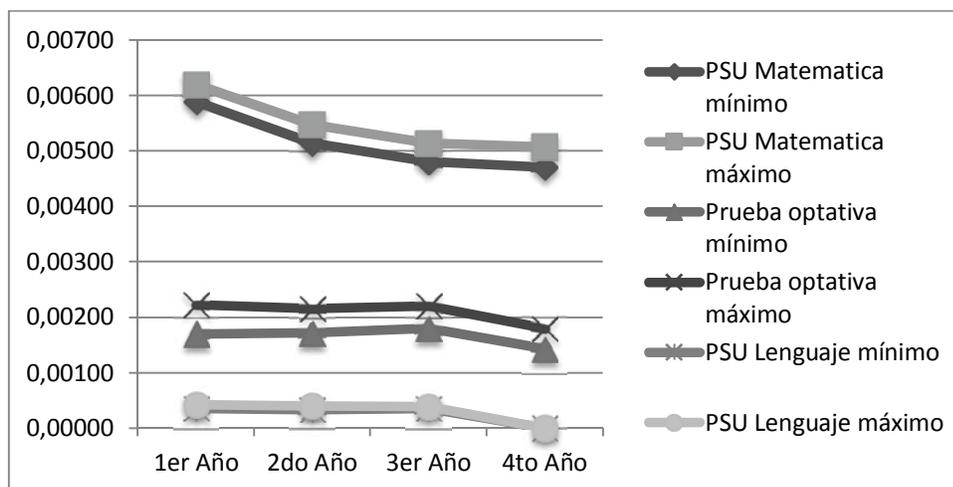


Figura 2.1 Gráfico nivel de significancia de los puntajes PSU en el promedio ponderado de notas con relación a los 4 años del estudio.

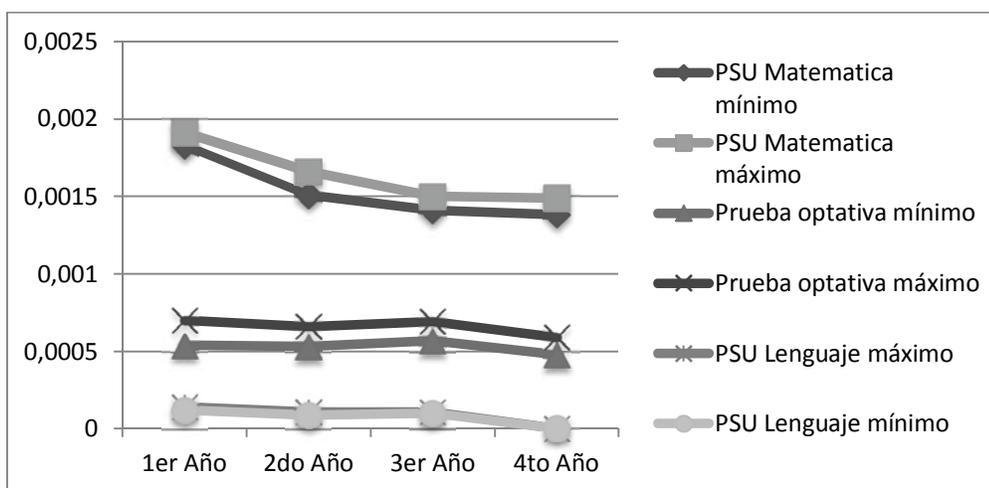


Figura 2.2 Gráfico nivel de significancia de los puntajes PSU en el porcentaje de créditos aprobados con relación a los 4 años del estudio.

De los gráficos se puede concluir que la significancia de la PSU matemática es mucho mayor que la de las PSU optativa y lenguaje, en el cuarto año el valor de la PSU de lenguaje se hace insignificante; se concluye que a medida que aumentan los años en la universidad tanto los puntajes de la PSU como NEM disminuyen hasta hacerse insignificantes.

Junto a los gráficos y tablas mostrados, la conclusión a la que se llegó en el estudio de los datos anteriormente mencionados es que el factor más importante es el NEM, ya que tiene mayor poder predictivo sobre éxito y rendimiento académico, seguido por el puntaje PSU matemáticas, PSU optativa y por último PSU lenguaje, aunque cabe destacar que estas variables son en parte resultado de variables, individuales (género), familiares (nivel de educación de los padres) y el entorno del alumno (lugar de residencia, ingreso familiar), las cuales también fueron medidas en el estudio pero no son incluidas.

Para finalizar el estudio de inteligencias múltiples MIDAS y el estudio validez predictiva de la PSU sobre el éxito académico fueron de gran ayuda ya que dieron respaldo a muchas premisas que son usadas en este proyecto, tanto por la parte de las inteligencias múltiples o por el tipo de preguntas que se agregaron en el test y el en juego cañón de palabras.

3. Especificación de objetivos

3.1. Objetivo General

Crear un juego para dispositivos móviles que permita estimar el nivel de inteligencia lógico-matemática, lingüística y espacial que posee una persona que desee ingresar a una carrera de ingeniería, apoyándose en un test de inteligencias para corroborar los resultados.

3.2. Objetivos Específicos

- Crear un test que permita estimar las inteligencias lógico-matemática, lingüística y espacial, basado en el estudio MIDAS y el estudio de validez predictiva de la PSU sobre el éxito académico.
- Realizar un muestreo con el test realizado anteriormente; de sus resultados crear una escala que permita diferenciar en rangos los diferentes niveles de inteligencia.
- Diseñar de forma análoga puntajes que puedan ser comparables a esos rangos
- Diseñar juegos que permitan estimar la capacidad de razonamiento (capacidad de resolver un problema) de los jugadores a través de esos puntajes.
- Crear un test dentro de la aplicación que permita estimar capacidades de razonamiento del usuario.
- Indicar el juego que permitirá fortalecer cada una de las capacidades de razonamiento a de las inteligencias nombradas en el objetivo general.

4. Plan de Trabajo

En este trabajo, se expone el desarrollo de una aplicación Android que a través de juegos permita tanto la estimación de las inteligencias múltiples (los cuales son apoyados con un test escrito), como el desarrollo de la capacidad de comprender conceptos referidos a las áreas de inteligencia (aprender con mayor facilidad algunas materias).

4.1. Fase Análisis del tema

Aquí se recopila la información necesaria para realizar el proyecto, la factibilidad de realizar el proyecto incluyendo los costos asociados y tanto los objetivos como los requerimientos para desarrollar la aplicación.

La siguiente tabla indica los plazos asociados a cada uno de los anteriores puntos:

Tabla 4.1 Fechas y duración de actividades fase de análisis

Nombre de Tarea	Duración	Comienzo	Fin
Objetivos generales	4 días	lun 12-03-12	jue 15-03-12
Objetivos específico	4 días	lun 12-03-12	jue 15-03-12
Investigación	15 días	lun 12-03-12	vie 30-03-12
Investigación de juegos serios	9 días	lun 12-03-12	jue 22-03-12
Investigación de inteligencias múltiples	9 días	lun 12-03-12	jue 22-03-12
Confección del test aplicar	3 días	mié 28-03-12	vie 30-03-12
Aplicación y muestreo de test	5 días	lun 02-04-12	vie 06-04-12
Especificación de requerimientos	5 días	lun 02-04-12	vie 06-04-12
Requerimientos funcionales	5 días	lun 02-04-12	vie 06-04-12
Requerimientos no funcionales	5 días	lun 02-04-12	vie 06-04-12
Definición de los juegos	5 días	mié 04-04-12	mar 10-04-12
Definición de arquitectura	4 días	mié 04-04-12	lun 09-04-12
Estudio de factibilidad	3 días	lun 09-04-12	mié 11-04-12
Definición de prototipos de interfaz	7 días	mar 10-04-12	mar 17-04-12
Casos de uso	9 días	lun 09-04-12	Jue 19-04-12

Además en esta fase se realiza un estudio a través de la utilización de un test escrito; con el fin de tomar una muestra de las capacidades de razonamiento de los alumnos de primer año tanto de ingeniería civil informática como ingeniería en ejecución informática.

4.2. Fase de diseño

Luego de realizar las tareas indicadas anteriormente se procede a especificar una serie de puntos que permitan completar la implementación de la aplicación; tales como el diseño de escenarios en los casos de uso, diagramas de secuencia y actividad; la definición de prototipos de interfaz, los modelos de datos a utilizar y el diseño de un prototipo a implementar, y por último la forma en la cual se comparan los puntajes del test realizado con los puntajes de los juegos.

En la siguiente tabla se muestran las fechas y duración de las tareas de la fase de diseño:

Tabla 4.2 Fechas y duración de actividades fase de análisis.

Nombre tarea	Duración	Comienzo	Fin
Estimación del puntaje de los juegos y rangos de puntajes	5 días	vie 20-04-12	jue 26-04-12
Comparación de puntaje en test con puntaje del juego	5 días	jue 26-04-12	mié 02-05-12
Diagrama de clases	5 días	vie 20-04-12	jue 26-04-12
Modelo relacional	5 días	vie 20-04-12	jue 26-04-12
Diagrama de actividad de los juegos	4 días	mié 25-04-12	lun 30-04-12
Diagramas de secuencia	4 días	mié 25-04-12	lun 30-04-12
Diseño de prototipo	5 días	mar 01-05-12	lun 07-05-12

Al comparar los puntajes del test con los puntajes del juego permite realizar una estimación entre ambos, lo cual permite estimar una diferencia entre la capacidad de razonamiento y poder así guiar al usuario a cual entrenamiento debe realizar, así como también realizar nuevamente el test para ver si se ha avanzado.

4.3. Fase de implementación de prototipo

En esta fase se realiza la codificación que se plantea en la fase de diseño sobre la aplicación Android.

Se detalla la implementación de prototipo, indicando las funcionalidades implementadas.

Tabla 4.3 Fechas y duración de actividades fase de implementación de prototipo.

Nombre de tarea	Duración	Comienzo	Fin
Codificación del prototipo	32 días	lun 07-05-12	mar 19-06-12
Codificación de menú principal	3 días	lun 07-05-12	mié 09-05-12
Codificación de menú juegos	3 días	lun 07-05-12	mié 09-05-12
Codificación de juego cañón de palabras	30 días	mié 09-05-12	mar 19-06-12
Implementación de “sprites”	13 días	mié 09-05-12	vie 25-05-12
Implementación de movimiento	7 días	vie 25-05-12	lun 04-06-12
Implementación de LinkedList y clases que agregan elementos	7 días	vie 25-05-12	lun 04-06-12
Implementación de base de datos locales	6 días	mar 05-06-12	mar 12-06-12
Implementación de sonido	6 días	mar 05-06-12	mar 12-06-12
Codificación de menú opciones	6 días	mar 12-06-12	mar 19-06-12
Codificación de menú test	6 días	mar 12-06-12	mar 19-06-12

La implementación de LinkedList y de clases que agregan elementos, permite la gestión eficiente de memoria, ya que existen algunos juegos como cañón de palabras que crean muchos elementos, los cuales al cumplir su cometido permanecían en pantalla, causando fallos graves tanto en la aplicación (cierre forzoso), como en el funcionamiento del “Smartphone” (queda en negro y la única acción que se puede hacer es reiniciar el dispositivo).

La implementación de la base de datos locales permite el almacenamiento de datos tales como preguntas y puntajes usados tanto por el test como los juegos.

4.4. Fase de implementación

En esta fase se hace uso del prototipo anterior, y se modifican algunos elementos tales como agregar los niveles y fases al cañón de palabras, agregar bases de datos locales que permitirán guardar opciones del juego aun cuando el juego se ha cerrado, además de los puntajes, niveles y fases superados.

Y además se agregan nuevos elementos tales como la opción de revisar los puntajes, tanto de juegos como test, como la implementación de los juegos restantes. En la siguiente tabla se detallan las fechas:

Tabla 4.4 Fechas y duración de actividades fase de Diseño entre Julio y Septiembre.

Nombre de tarea	Duración	Comienzo	Fin
Implementación de interfaz fases de juego	3 días	mié 18-07-12	vie 20-07-12
Implementación interfaz de niveles de juego	3 días	jue 19-07-12	lun 23-07-12
Implementación de base de datos de puntajes	1 día	mar 24-07-12	mar 24-07-12
Implementación de bases de datos de puntaje de test	1 día	mié 25-07-12	mié 25-07-12
Implementación de base de datos de fases y niveles	1 día	jue 26-07-12	jue 26-07-12
Implementación de base de datos de opciones	1 día	vie 27-07-12	vie 27-07-12
Implementación de niveles, fases y dificultad cañón de palabras	3 días	lun 30-07-12	mié 01-08-12
Implementación de entrenamiento cañón de palabras	3 días	jue 02-08-12	lun 06-08-12
Implementación de menú en cañón de palabras, al pausar, ganar o perder.	5 días	mar 07-08-12	lun 13-08-12
Implementación de opción de puntajes	2 días	mar 14-08-12	mié 15-08-12
Implementación de juego matemáticas de figuras	35 días	jue 16-08-12	mié 03-10-12
Implementación de figuras	1 día	jue 16-08-12	jue 16-08-12
Implementación de colisiones	14 días	vie 17-08-12	mié 05-09-12
Implementación de delimitadores	2 días	jue 06-09-12	vie 07-09-12
Implementación de pilas de figuras ("LinkedList")	3 días	lun 10-09-12	mié 12-09-12
Condiciones de victoria y derrota	2 días	jue 13-09-12	vie 14-09-12
Implementación de puntaje y tiempo	1 día	lun 17-09-12	lun 17-09-12
Menú de pausa, victoria y derrota	2 días	mar 18-09-12	mié 19-09-12
Implementación de niveles y fases matemática de figuras	3 días	jue 20-09-12	lun 24-09-12
Implementación de entrenamiento matemática de figuras	3 días	mar 25-09-12	jue 27-09-12
Implementación de mecanismo de escape	30 días	vie 28-09-12	jue 08-11-12

Tabla 4.5 Fechas y duración de actividades fase de Diseño entre Septiembre y Noviembre.

Nombre	Duración	Comienzo	Fin
Implementación de figuras	1 día	vie 28-09-12	vie 28-09-12
Implementación de colisiones	5 días	vie 28-09-12	jue 04-10-12
Implementación de delimitadores	1 día	jue 04-10-12	jue 04-10-12
Implementación de condición de victoria y derrota	4 días	vie 05-10-12	mié 10-10-12
Implementación de puntaje, intentos y tiempo	4 días	lun 08-10-12	jue 11-10-12
Implementación de menú en pausa, victoria o derrota	4 días	vie 12-10-12	mié 17-10-12
implementación de niveles y fases	5 días	jue 18-10-12	mié 24-10-12
Implementación de mecanismo de escape en entrenamiento	5 días	lun 22-10-12	vie 26-10-12

4.5. Fase de pruebas

Se realizan pruebas de funcionalidad con el fin de comprobar si se cumple con los requisitos.

Tabla 4.6 Fechas y duración de actividades fase de Pruebas.

Nombre de la tarea	Duración	Comienzo	Fin
Realización de pruebas unitarias sobre aplicación.	5 días	lun 29-10-12	vie 02-11-12

5. Marco teórico

5.1. Juegos serios

En la actualidad, existe una vasta gama de video juegos que abarcan una gran cantidad de escenarios y formatos; así como también, una gran cantidad de la población (personas entre 10 a 34 años) e influyen en las conductas, pensamientos y creencias de estas personas. En la tabla 5.1 se muestran los diferentes géneros de video juegos con los que se clasifican a estos en la actualidad.

Tabla 5.1 Géneros de video juegos.

Géneros de video juego	Descripción
Aventura	Son videojuegos en los que el protagonista debe avanzar en la trama interactuando con diversos personajes y objetos.
Disparos	El jugador tiene que resolver un conflicto disparando a sus oponentes. Pueden ser estáticos o de desplazamiento.
Serios	Videojuegos cuyo objetivo es dar a conocer al usuario algún tipo de conocimiento.
Estrategia	Los jugadores controlan los aspectos económicos y militares de un ejército o de una población. Deben tomar decisiones estratégicas rápidas.
Lucha	Un videojuego de lucha o pelea, es un videojuego que se basa en manejar un personaje peleador, ya sea dando golpes, usando poderes mágicos o aplicando llaves
Survival horror	Es un sub género de aventuras el cual es mezclado con suspenso dentro de una atmósfera o ambientación de terror psicológico donde lo que prima es la supervivencia.
Plataformas	El jugador debe desplazarse en un espacio en el que tiene que avanzar por plataformas (de ahí el nombre). Este tipo de videojuegos suelen usar vistas de desplazamiento horizontal hacia la izquierda o hacia la derecha.
Rol	Los jugadores encarnan uno de los personajes de ficción. El personaje tiene varias características que pueden evolucionar a lo largo del juego, como salud, fuerza u otras habilidades.
Musicales	La jugabilidad está orientada significativamente, o incluso totalmente, a la interacción del jugador con la música, la banda sonora o los efectos de sonido del videojuego.
Party games	En este género los jugadores habrán de ir avanzando por turnos en un tablero virtual e ir superando diversas pruebas de tipos muy diversos en los que compiten entre sí por llegar lo antes posible a la meta, o conseguir la máxima cantidad posible de puntos
Simulación	Intentan recrear situaciones de la vida real.
Deportivos	Estos juegos reproducen deportes conocidos como el fútbol, el golf o el baloncesto. Están disponibles en 2D o 3D.
Carreras	Un videojuego de carreras es un videojuego en el que se imitan competencias entre vehículos. Usualmente el objetivo es recorrer cierta distancia o ir de un sitio hacia otro en el menor tiempo posible
No Lineal	Videojuegos en los que el jugador puede elegir el orden de las misiones o viajar libremente por el mapa del videojuego, e interactuar con casi todo lo que este a su disposición

En este sentido la investigación se enfocara en los denominados “*juegos serios*”. Los juegos serios (o “*Serious Games*” en inglés), son una categoría de video juegos que ha entrado en auge en los últimos años y, a diferencia de la mayoría de los juegos convencionales, tienen un trasfondo ya que su misión, además de entretener al jugador, es dejarle una enseñanza o ayudarlos con alguna capacidad. Para definirlo usaremos la siguiente cita: “*una prueba mental, llevada a cabo frente a una computadora de acuerdo con unas reglas específicas, que usa la diversión como modo de formación gubernamental o corporativo, con objetivos en el ámbito de la educación, sanidad, política pública y comunicación estratégica*”.

En estos últimos años, este tipo de juego a obtenido mayor popularidad, gracias al avance que ha tenido la industria de los video juegos, ya no solo enfocándose en el ámbito comercial y masivo, sino también en juegos educacionales (ejemplo Big Brain Academy), que sirvan tanto en las aulas, como en rutinas de ejercicios dentro de empresas o en la milicia.

La “*serious game initiative*”, por medio de sus colaboradores Ben Sawyer y Peter Smith, han creado una taxonomía para los juegos serios que se divide en 7 modalidades y que abarcan 7 sectores de interés. A continuación se muestra una tabla con las modalidades y sectores:

Tabla 5.2 Taxonomía de juegos serios parte 1.

	Juegos para la Salud	Juegos publicitarios	Juegos para la Formación	Juegos para la Educación
Gobiernos y ONGs	Educación para la Salud - Respuesta a problemas de salud masivos.	Juegos políticos (campañas de partidos políticos)	Formación de empleados	Información pública
Defensa	Rehabilitación y Bienestar psicológico	Reclutamiento y propaganda	Formación de apoyo a los soldados	Educación en la escuela y en el hogar
Sistemas de salud	Ciberterapia y Videojuegos para hacer deporte o ejercicio físico	Política de Salud Pública - Campañas de concienciación Social	Juegos formatos para profesionales de la salud	Juegos para educación de los pacientes y para la gestión de la enfermedad
Marketing y comunicaciones	Publicidad de tratamientos médicos	Publicidad. marketing con juegos, publicidad indirecta (publicidad por emplazamiento)	Uso de productos	Información de productos
Educación	Informar sobre enfermedades y riesgos sanitarios	Juegos sobre temática social	Formación de Profesorado - Entrenamiento de competencias específicas	Aprendizaje
Empresas	Infamación a empleados del sistema sanitario - bienestar para los empleados	Educación y Concienciación del cliente	Formación de empleados	Formación Continua - Cualificación profesional
Industria	Prevención de riesgos laborales	Ventas y contratación	Formación de empleados	Formación profesional

Tabla 5.3 Taxonomía de juegos serios parte 2.

	Juegos para la Ciencia y la Investigación	Producción	Juegos como Empleo
Gobiernos y ONGs	Recogida de datos - Planificación	Planificación de políticas y estrategias	Diplomacia - estudios de opinión
Defensa	Juegos de guerra- Planificación	Planificación de la guerra e investigación armamentístico	Mando y control
Sistemas de salud	Visualización y epidemiología	Diseño y fabricación de biotecnologías	Planificación y Logística de planes de salud pública
Marketing y comunicaciones	Estudios de opinión	“Machinima” (corto de animación que usa un video juego)	Estudios de opinión
Educación	Ciencias de la Computación y reclutamiento	Aprendizaje P2P Constructivismo	Formación on-line
Empresas	Publicidad - Visualización	Planificación estratégica	Mando y control
Industria	Procesos de optimización mediante simulación	Diseño nano biotech	Mando y control

Usando esta terminología, el proyecto es enfocado en juegos destinados al aprendizaje y tendrán como fin estimar las capacidades intelectuales de jóvenes y adultos que quieran ingresar a carreras de ingeniería.

Los juegos para la educación han sido, por largo tiempo, una herramienta para ayudar a los educadores y profesores en su tarea de enseñar a las personas, ya que tienden a generar mayor interés en ciertos campos motivando a los jugadores con el desafío, la curiosidad, la belleza, la fantasía y el reconocimiento social entre sus pares. Ejemplos de estos tipos de juegos son: “DimensionM”, “¿dónde está Carmen san diego?”, “el conejo matemático”, entre otros.

Mediante estos juegos, el usuario aprende a pensar críticamente acerca de la situación, mientras, al mismo tiempo gana conocimientos interactuando con el entorno del juego. También proporcionan a los jugadores la oportunidad de aprender con la práctica y ganar experiencia en ciertas áreas.



Figura 5.1 “RescueSim”. Videojuego para la formación de personal de rescate.

5.2. Inteligencias Múltiples

Para ayudar el desarrollo de este proyecto utilizamos el enfoque de Howard Gardner de las inteligencias múltiples. Lo que dice este científico es que la inteligencia no es una sola como se pensaba en la antigüedad si no que esta se divide en 7 diferentes tipos que van acorde a diferentes regiones en el cerebro y diferentes ámbitos. De su libro estructuras de la mente se utiliza la siguiente cita: “la inteligencia es la capacidad de resolver problemas, o de crear productos, que sean valiosos en uno o más ambientes culturales” [1].

Los ámbitos para describir y elegir cada tipo de inteligencia utilizados por Gardner son los siguientes:

- Posible aislamiento por daño cerebral. (Por ejemplo si se daña cierta parte del cerebro se puede perder la capacidad de hablar fluidamente).
- Existencia de genios y otros individuos con habilidades excepcionales en ciertos o muchos campos.
- Una operación medular o conjunto de operaciones identificables:

Al realizar ciertas actividades una persona puede comenzar a adquirir conocimientos sobre una materia como si estuviera programado para ello, por ejemplo cuando un niño comienza a imitar los movimientos de sus padres.

- Una historia distintiva de desarrollo, junto con un conjunto definible de desempeños expertos de "estado final". (Existen distintos niveles de dominio del campo desde novato a experto).
- Una historia evolucionista y la evolución verosímil.
- Apoyo de tareas psicológicas experimentales.
- Apoyo de hallazgos psicométricos.
- Susceptibilidad a la codificación en un sistema simbólico. (El campo tiene un conjunto de símbolos propios para expresarse).

Es así como este psicólogo diferencia la inteligencia en 7 tipos:

Tabla 5.4 Inteligencias múltiples propuestas por Gardner.

Inteligencia	Descripción	Profesiones relacionadas
Lógico-matemática	Capacidad para razonar con números y resolver operaciones abstractas. Incluye el uso del pensamiento lógico, deductivo y secuencial.	Matemáticos, físicos, ingenieros, contadores, economistas
Verbal-lingüística	Vinculada con la habilidad para el lenguaje oral y escrito, la relación de ideas, y la posibilidad de expresar con claridad pensamientos y sentimientos a través de la palabra.	Poetas, escritores, periodistas y oradores
Espacial	Visualizar imágenes, formas y colores; las transforma, relaciona y las traduce en esquemas gráficos o conceptuales.	Arquitectos, cirujanos, escultores y pintores
Musical	Consiste en la habilidad para expresarse por medio de la música, interpretar temas musicales a través de instrumentos o de la voz, y también para componer.	Músicos, cantantes y compositores
Corporal-kinestésica	Habilidad para expresarse con el cuerpo con plasticidad y armonía, coordinar movimientos generales y manuales.	Bailarines, atletas, cirujanos, escultores y artesanos
Intra-personal	Capacidad para conectarse consigo mismo, reconocer los estados interiores y tener una imagen realista de sí.	Filósofos, teólogos y psicólogos
Inter-personal	Habilidad para establecer contacto con otras personas, relacionarse y trabajar cooperativamente en equipo.	psicólogos, médicos, maestros, políticos, cineastas, vendedores, comunicadores en general

La inteligencia naturalista, la cual, se describe como la capacidad de percibir las relaciones entre las especies, grupos de objetos y personas reconociendo las posibles diferencias o semejanzas entre ellos se ha descartado, pues es incluida por las inteligencias lógico-matemática y la inteligencia espacial.

En la actualidad la teoría de las inteligencias múltiples ha sido aplicada en la enseñanza, a tal punto que existen escuelas que trabajan con esta teoría como base metodológica, como lo hace por ejemplo la New City School en St. Louis, Missouri.

Esta teoría tiene muchos detractores desde su proposición en los años 80, ya que la mayoría de los psicólogos y expertos en el tema de la inteligencia plantean la idea de una sola inteligencia y que se expresaba de diferentes formas para cada persona. Hasta el día de hoy

Gardner no ha entregado una definición concreta sobre inteligencia solamente se refiere a capacidades, además existe muy poca trabajo practico sobre el tema.

5.3. Sistema operativo Android

Como tercer pilar fundamental del proyecto esta Android, el sistema operativo open source basado en Linux para dispositivos móviles. Android es desarrollado por la Open Handset Alliance desde el 2005 y su principal contribuidor es el gigante Google. Este sistema operativo ha tenido un gran auge desde su aparición a mediados de la década pasada y en la actualidad es parte de la mayoría de los dispositivos móviles de empresas como Samsung, Motorola, Lenovo, Acer, entre otras. El software está básicamente programado en Java, con líneas el XML, C y C++ y tiene su propia extensión para sus aplicaciones, la extensión “.apk”.

Como la mayoría de los dispositivos de última generación tienen este sistema operativo instalado, se opta por usarlo para el soporte de los juegos, además de que al ser open source, los programas y las herramientas necesarias para realizar la aplicación son más fáciles de encontrar y utilizar.

5.4. Framework AndEngine

5.4.1. AndEngine Gles1

Es un “framework” para el desarrollo de juegos llamado “AndEngine”. Un “framework” es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, con base a la cual otro proyecto de software puede ser más fácilmente organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Este completo “framework” para el desarrollo de juegos para el sistema operativo Android entrega todas las herramientas necesarias para crear juegos en dos dimensiones, mediante una compilación de librerías llamada “Andengine.jar” y el proyecto “AndEngine”.

Las principales características de AndEngine son:

- **Actividad principal:** toda clase que quiera implementarse como un juego debe ser de extensión BaseGameActivity o y será posible re implementar los eventos principales. Algunos de estos eventos son los típicos de cualquier Activity, como comenzar con la aplicación, parar la aplicación, reproducir sonido, etc.

- **Motor:** El motor es la pieza encargada de hacer que el juego funcione. Dispone de un hilo que refresca la ventana cada x milisegundos de tiempo. También se encarga de sincronizar los refrescos en pantalla y actualizar la escena.
- **Escena:** La clase “scene” es el contenedor de los “sprites” (imágenes) que se deben visualizar en pantalla. Las escenas pueden tener capas (Layers) y entidades (Entities). Se puede encontrar también con “sprites” estáticos en la pantalla que servirán para visualizar menús, puntuaciones, energía, etc.
- **Texturas:** Las texturas son imágenes en la memoria. Estas imágenes serán utilizadas para visualizar “sprites” o fondos. Por ello, en el momento de desarrollar siempre hay que tener en mente que por una parte se gestiona la información lógica de los objetos y por otra parte el aspecto visual que tiene o la textura.
- **Sprites:** Los “sprites” son los objetos que se visualizan en el juego y en la mayoría de los casos son interactivos o animados. Existen varios tipos de “sprites”: “titles” en el que el juego tiene forma matricial, animados en los que el sprite dispondrá de varias imágenes o fotogramas, y de “background” que se utilizarán para pintar fondos.
- **Física y colisiones:** se disponen las funciones necesarias para detectar colisiones e incluso aplicarles física para que tengan un comportamiento más real. Un ejemplo de estas funciones es `collidesWith()` que permite saber cuándo dos “sprites” chocan entre sí.
- **Música y efectos de sonido:** AndEngine consta de herramientas para la reproducción de archivos de sonido, tales como mp3, aac, ogg y midi entre otros. Igualmente, es posible controlar características y eventos de la reproducción como subir y bajar el volumen, repetir el sonido, conocer si se está reproduciendo, situarse en una posición concreta del audio y algunas funciones más.

5.4.2. AndEngine Gles2

Es la última versión de AndEngine la cual es más eficiente por las siguientes características:

- Para agregar imágenes en la versión anterior se debía utilizar como posición inicial para esas figuras números de potencias de 2, lo cual era un gasto innecesario de memoria, se corrige en esta versión.
- Uso de `VertexBufferObject`: es una mejora de OpenGL que entrega los beneficios de arreglos, vértices y listas desplegables, evitando los inconvenientes de su implementación; también permite al arreglo de vértices ser guardado en memorias de

alto rendimiento en el lado del servidor para promover la transferencia de datos eficiente.

- Uso de fuente pre-renderizadas.

Su inconveniente es que su uso está limitado para versiones superiores o iguales a Android 2.2, pero soportado por el 89% de los dispositivos Android activos y es un poco más lenta que la versión gles1.

Con relación a las clases utilizadas:

- BaseGameActivity pasa a ser SimpleBaseGameActivity.
- Para crear “sprites” y textos se requiere el paso del VertexBufferObject.
- Existe un EngineOptions el cual se encarga de guardar la información con respecto al motor.

5.4.3. AndEngine Box2d extencion

Para los juegos matemáticas de figuras y mecanismo de escape se requiere el uso del acelerómetro y el uso de colisiones entre figuras.

Para ello usamos la extensión de AndEngine Box2d basado en la librería para simulación de cuerpos rígidos Box2d en motores de juegos.

En los juegos esta librería es utilizada para darle a los “sprites” y otros elementos (como líneas) gravedad, peso, tamaño, impulso, torque e inercia.

Algunos elementos utilizados de esta librería son:

- PhysicsWorld: Permite implementar constantes físicas al mundo como por ejemplo la gravedad, además es quien almacena “body” y “fixtures”.
- Body: es el cuerpo de un objeto, posee masa, peso, tamaño y “fixture”, además es quien permite la colisión entre cuerpos, puede ser tanto estático (ninguna fuerza aplicada lo mueve) o dinámico (toda fuerza aplicada lo mueve).
- Fixture: entrega al cuerpo atributos como densidad, fricción y resolución, además permite implementar categorías y máscaras de colisión, las cuales indican con que objetos colisionar y con cuáles no.

5.5. PhysicsEditor

Es un programa que al entregarle una imagen nos devuelve los vértices del objeto en formato xml. Es usado en la aplicación para asignarle a los “sprites” los vértices para verificar las colisiones usadas, ya que AndEngine al instanciar un “sprite” se instancia usando la imagen original sin tener en cuenta si la imagen utiliza todo el espacio asignado, para explicarlo mejor se usará el siguiente ejemplo:

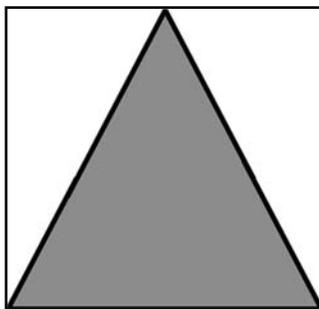


Figura 5.2 Ejemplo uso de PhysicsEditor

Al instanciar esta imagen en AndEngine lo que se instancia no es solo el triángulo, sino también el fondo sea este transparente o no, el problema es que para trabajar con Box2d se necesitan los vértices de la figura para implementar las colisiones, el cual es el uso de PhysicsEditor que calcula por sí solo los vértices y genera un archivo XML, luego es llamado usando la clase PhysicsEditorLoader e indicando donde se encuentra el archivo XML a leer, por último se crea el “body” utilizando los vértices especificados en el archivo XML.

5.6. Base de datos SQLite

SQLite es un sistema de gestión de base de datos relacionales que está contenida en una biblioteca muy pequeña (275 kiB) y está escrita en c. A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción.

6. Estudio de factibilidad

6.1. Estudio de factibilidad técnica

Se utilizaron las siguientes herramientas de desarrollo:

- Eclipse.
- Power Designer.
- StartUML
- Java developer kit 32bits.
- Motor de desarrollo de juegos AndEngine.
- Android SDK 2.3
- Base de datos SQLite.
- Microsoft Word.
- Microsoft Visio.
- Microsoft Excel.
- Microsoft Project.
- Balsamic Mockups.

Herramientas hardware:

Computador para el desarrollo con las siguientes características:

- Procesador: 2.6 GHz Intel Pentium IV o equivalente.
- Memoria: 1 GB.
- Espacio disponible en el disco: 3.5 GB espacio libre en el disco.

6.2. Estudio de factibilidad económica

Estos costos fueron asumidos, las herramientas de desarrollo son en su mayoría gratuitas o proporcionadas por la escuela. El costo de software operativo se incluye en el costo de los computadores.

Tabla 6.1 Costos de herramientas técnicas.

Dispositivos	Cantidad	Precio	Subtotal
Computadores	2	300.000	600.000
Dispositivo de almacenamiento	1	5.000	5.000
Servicio de Internet	1	27.000	27.000
		Total (pesos)	632.000

La siguiente tabla muestra los costos relacionados al trabajo producido expresado en horas hombre y el precio de esas horas, como todos son considerados programadores el costo por este no se toman en cuenta, por ultimo como solo somos dos personas no existe un jefe de proyecto.

Tabla 6.2 Costos de horas de trabajo.

Cargo	Horas semanales	Semanas	Precio	Total
Analista	20	40	2.000	1.600.000
Diseñador	20	40	1.500	1.200.000

6.2.1. Caracterización del consumidor

Los consumidores del producto son personas de una edad entre 16 y 20 años que tengan el interés de entrar en una carrera de ingeniería, además poseen “Smartphone” con sistema operativo Android 2.3 o superior.

6.2.2. Hábitos de consumo

Debido al auge que posee la Internet, la masividad de la tecnología actual y el uso de redes sociales, las personas invierten más dinero en mantenerse al tanto de las nuevas tecnologías.

En la actualidad, el avance de la tecnología ha hecho posible que los teléfonos celulares posean similares características que los computadores, y con el paso del tiempo su costo ha comenzado a disminuir a tal punto que cualquier persona con un ingreso mínimo puede obtener uno.

Los consumidores están pensando en obtener un “Smartphone” considerando el costo y el beneficio que traerá el teléfono, además, están apoyados por la información que extraen de internet, por lo que se puede concluir que es un usuario relativamente informado, tanto en el producto sugerido como en demás productos de su interés.

Las personas además de buscar una forma de estar conectadas, buscan entretención, la cual mediante la computación es brindada mediante juegos, archivos multimedia y acceso a medios de comunicación.

6.2.3. Cultura y sociedad inmersa

En la cultura chilena, en el rango de edades sugerido (y también personas de mayor edad), las personas suelen elegir su futura profesión, la cual depende de diversos factores (tanto recursos económicos como gustos y preferencias de la persona que quiere ser profesional).

El producto creado tiene como fin ese tipo de personas, las cuales tienen como preferencia profesiones como carreras de ingeniería, esta aplicación les ayudara a orientarse y les permitirá decidir si tienen su inteligencia tan desarrollada como para estudiar en alguna carrera de ingeniería.

6.2.4. Tasa de demanda del mercado y del proyecto actual y proyectado

Un usuario estaría dispuesto a comprar el producto por la suma de 1,5 dólares (aproximadamente 740 pesos chilenos), cada usuario compraría solo una unidad del producto. La fórmula ingreso está dada por:

$$VAN = (740 * N)(P/F, 0, 2, 10) - 2.800.000 - 632.000$$

Siendo N el número de personas que compran el producto, la siguiente tabla muestra las estimaciones:

Tabla 6.3 Valor actual neto.

Pesimista (N=10000)	Esperado(N=25000)	Optimista(N=30000)
-2.156.150	-444.250	153.300

Los consumidores son personas entre 16 y 20 años, que quieren estudiar en una carrera de ingeniería y desean saber si su capacidad intelectual es suficiente para estudiar en dichas carreras, o cualquier persona que quiera saberlo.

En Chile la cantidad de personas en el rango entre los 16 y 20 años es alrededor de 1.200.000, de los cuales entre 500 y 400 mil personas ingresaron a universidades (estudio realizado hace 2 años).

De la anterior cantidad de personas se deduce a entre unas 150 y 250 mil personas que ingresan a estudiar a una carrera de ingeniería en Chile.

Este mercado equivale a todas las personas que posiblemente puedan y quieran estudiar alguna ingeniería que tenga una edad entre 16 y 20 años (no se incluye la gente mayor a 20 años), la cantidad se reducirá aún más si solo se considera a la gente que además tenga un “Smartphone”.

Se espera a futuro que la disminución de los precios de teléfonos inteligentes aumente la cantidad de unidades vendidas del producto; el año pasado la cantidad de “Smartphone”'s supero varias veces la cantidad de computadores vendidos.

6.2.5. Competencias y ofertas del mercado actual

Tabla 6.4 Productos similares.

Oferente	Distribución y tipología	Objetivo	Competitividad con el producto
Glowium-BrainChallenge	Es un puzzle muy complejo, en el cual se deben mover las líneas de tal manera que formen la figura de indicada por las líneas blancas. Es un juego gratis y hasta ahora posee más de 50.000 descargas oficiales, soportado desde la versión 2.1 de Android.	Se basa en cautivar a los consumidores con los complicados puzzles.	Es un producto sustituto de la aplicación desarrollada, el cual está más enfocado en la entretención que la enseñanza.
Treasures of Mystery Island	Juego en el cual se aprende cultura aborigen mientras se realizan intrincados puzzles, en diciembre del 2011 alcanzo las 250000 descargas a U\$0,99 dólares.	Fomenta el aprendizaje cultural en base de entretenimiento.	Es un producto complementario del producto presentado en este documento.
LanguageExamTrainer	Puedes descargar de forma gratuita algunos archivos básicos de vocabulario: Castellano - Catalán Catalán - Castellano Castellano - Gallego Gallego - Castellano Castellano - Euskera Euskera - Castellano Es muy sencilla de utilizar y que puedes añadir tus propios ficheros de vocabulario desde tu ordenador. Tiene un valor de U\$0,99, ya lo han descargado más de 30.000 personas.	Ayuda a personas que necesitan tanto traducir de un idioma a otro como a gente que quiere aprender algún idioma.	Es un producto complementario del producto presentado en este documento.
Grockit	Es una red social con profesores online que ayudan a quien quiera aprender, además las personas pueden utilizar los recursos de la red como Wikipedia y Youtube para ayudarse entre sí. Se puede inscribir a diferentes programas de estudio, por ejemplo calculo, registrándose en la página o utilizando una cuenta de Facebook. El costo de este sistema es de U\$29.99 dólares el por el acceso y U\$50 dólares por tutoría.	Sirve como ayuda a los estudiantes que tienen ciertos problemas con los estudios y no pueden encontrar tutores que los ayuden.	Este producto es un complemento del producto presentado en este documento, mientras que este producto se encarga de enseñar, el producto presentado en este documento se encarga de potenciar las capacidades intelectuales de la persona.

6.3. Estudio de factibilidad operativa

El mobiliario necesario para el correcto desarrollo del proyecto son sillas, mesas o escritorios y las herramientas hardware necesarias tales como impresora, computadores (notebook).

Los miembros del proyecto se encargarán de disponer un entorno de desarrollo adecuado, libre de distracciones que puedan entorpecer el trabajo, para que su equipo pueda hacer un correcto desarrollo del software.

El lenguaje de programación fue escogido por que es conocido por los integrantes del grupo y no requieren mayor preparación para utilizarlo.

El equipo deberá estar capacitado para utilizar las herramientas de desarrollo que se utilizarán, además se deberá capacitar a los usuarios sobre el uso del programa.

6.4. Estudio de factibilidad legal

- El test de inteligencia es anónimo y no se indica a nadie que ha realizado este test sus resultados.
- No realizar operaciones que alteren o modifiquen acciones del programa o el programa mismo.
- No reproducir la aplicación, ni realizar copias sin la autorización de los autores.

7. Estudio de inteligencias múltiples en alumnos de primer año.

En la aplicación, la teoría de las inteligencias múltiples tiene el fin de separar los tipos de inteligencia que se fueron nombradas anteriormente, las que se consideran como las más importantes para tener éxito universitario en carreras ingenieriles.

7.1. Selección de inteligencias múltiples

Para el siguiente estudio, y con relación al estudio MIDAS, se seleccionan las 3 inteligencias que se consideran más importantes para el ingreso a la universidad en una carrera de ingeniería, las cuales son:

- **Inteligencia lógico-matemática:** Utiliza el pensamiento lógico para entender causa y efecto, conexiones y relaciones entre acciones objetos e ideas. Además la habilidad de resolver operaciones complejas tanto lógicas como matemáticas.
- **Inteligencia espacial:** Permite visualizar mejor los problemas y sus posibles soluciones, permite comprender la magnitud de un problema y diseñar soluciones con un mayor acotamiento.
- **Inteligencia lingüística:** Permite una mejor comprensión de los problemas, al comprender con mayor claridad el significado de cada palabra, frase e intención de cada una de ellas. Además permite expresar con mayor exactitud deseos o ideas.

Pero, según el estudio de la relación entre éxito académico en la universidad y la PSU, se llega a la conclusión que el resultado del estudio MIDAS solo es aplicable utilizando inteligencias que tienen directa relación con sus carreras y aplicando una escala similar a la utilizada por la PSU.

En este caso sobre las inteligencias lógico-matemática que tiene relación con el puntaje PSU de matemáticas y la inteligencia lingüística con el puntaje de la PSU de lenguaje y comunicación.

Por lo planteado en el test se usan preguntas tipo PSU tanto en las preguntas de tipo lógico-matemáticas como en las lingüísticas.

En el caso de la inteligencia espacial, se mantiene en la estimación del test, ya que a diferencia de las otras inteligencias evaluadas en el test MIDAS y que tienen mayor media; esta es considerada más importante por las razones entregadas anteriormente y porque es medible a diferencia de, por ejemplo, la inteligencia cinestesia. La cual en el caso de carreras

de ingeniería tiene más importancia para los alumnos que la inteligencia espacial, pero es muy difícil de estimar con algún test.

Con la información recopilada por los dos estudios anteriormente mencionados (MIDAS y estudio sobre PSU), podemos plantear varias premisas que ayudan a conformar una base para este proyecto:

- 1) Las inteligencias múltiples pueden ser relacionadas con diferentes tipos de perfiles universitarios, es así el caso de que un estudiante de ingeniería lo más probable es que tenga muy desarrollada su inteligencia lógico-matemática o un estudiante de carreras como derecho o pedagogía tenga muy desarrollada su inteligencia lingüística.
- 2) Cada inteligencia puede ser evaluada y cuantificada pero no de forma muy exacta, es por esto que existen test para estimar la afinidad o el gusto que tienen las personas con cada inteligencia, siendo uno de estos test el utilizado a comienzos del proyecto para verificar si el resultado de las preguntas realizadas a los alumnos de primer año tenían relación con la inteligencia que más afinidad tenían.
- 3) La PSU, como prueba, predice de una manera muy acertada la proyección de un estudiante al momento de ingresar a la universidad, tanto así que mientras más puntaje obtenga la persona en el área (relacionada con la inteligencia lógico-matemática y lingüística) que corresponde con la carrera que el ingresará mejor serán sus notas y menor será su cantidad de ramos reprobados dentro de su vida universitaria, aunque pueden haber algunas excepciones a esto.
- 4) Como la PSU es una buena herramienta y coincide con los objetivos de la aplicación desarrollada, los tipos de preguntas que se utilizaron dentro del software son del tipo preguntas PSU, sacados de facsímiles de años anteriores, creados de acuerdo a los estándares de esta prueba o relacionados directamente con la inteligencia lingüística.

7.2. Descripción del estudio

Para comprobar la validez del estudio Midas sobre alumnos de alguna carrera de ingeniería, se realiza un test sobre 53 alumnos de primer año de la escuela de ingeniería informática, tanto de la carrera de ingeniería civil en informática (31 alumnos) como la carrera de ingeniería en ejecución en informática (22 alumnos); el cual consiste en dos partes:

- Estimar la afinidad a cierta inteligencia.(15 preguntas)
- Estimar la capacidad intelectual en ciertas áreas.(21 preguntas)

Las áreas a estimar fueron:

- Lingüística. (7 preguntas)
- Lógico-matemática. (7 preguntas)
- Espacial. (7 preguntas)

A cada alumno se le entrega un tiempo aproximado de 20 minutos para contestar este test, el cual se superviso de tal forma que no existieran copias y se resolvieran las dudas de los alumnos al realizar el test; por otra parte el test es de carácter anónimo, por lo que no se guardan registro de las personas que tomaron el test.

La primera parte de afinidad consta de quince preguntas las cuales tratan sobre la forma en que el alumno prefiere hacer las cosas (similar a una encuesta vocacional), el alumno responde con una V si se identifica con la afirmación, F si no se identifica o lo deja nulo si no está seguro; de los resultados se estima a cual inteligencia tiene mayor afinidad o si es apático respecto a esa inteligencia.

7.3. Resultados del estudio

La siguiente tabla muestra la afinidad de los alumnos en base a la inteligencia espacial:

Tabla 7.1 Afinidad de alumnos por la inteligencia espacial

	Espacial
Afín	7
Normal	35
Apatía	11
Total	53

Lo que esta muestra indica es que existe poca cantidad de alumnos que sienten afinidad por la inteligencia espacial y pocas personas que sienten apatía por la inteligencia lo que permite concluir que los alumnos de ingeniería al ingresar no le complican o no le parece difícil en mayor medida realizar dibujos, crear mapas conceptuales, entender gráficos.

El siguiente gráfico representa a la tabla anterior:

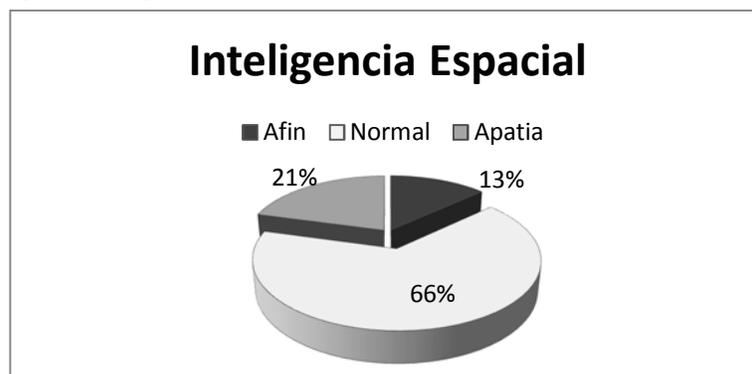


Figura 7.1 Gráfico afinidad de inteligencia espacial en alumnos de primer año.

La siguiente tabla muestra la cantidad de alumnos que tienen afinidad hacia la inteligencia lingüística:

Tabla 7.2 Afinidad de alumnos por la inteligencia lingüística.

	Lingüística
Afín	2
Normal	39
Apatía	12
Total	53

Existe muy poca gente que se siente afín a esta inteligencia, comparado con la cantidad de personas que sienten apatía a la inteligencia, pero al igual que con la inteligencia espacial, la muestra indica que a las personas no les molesta o no les resulta muy difícil relacionar ideas, expresar claridad al expresarse de forma verbal o escrita y entender conceptos.

El siguiente gráfico representa los datos de la tabla anterior:

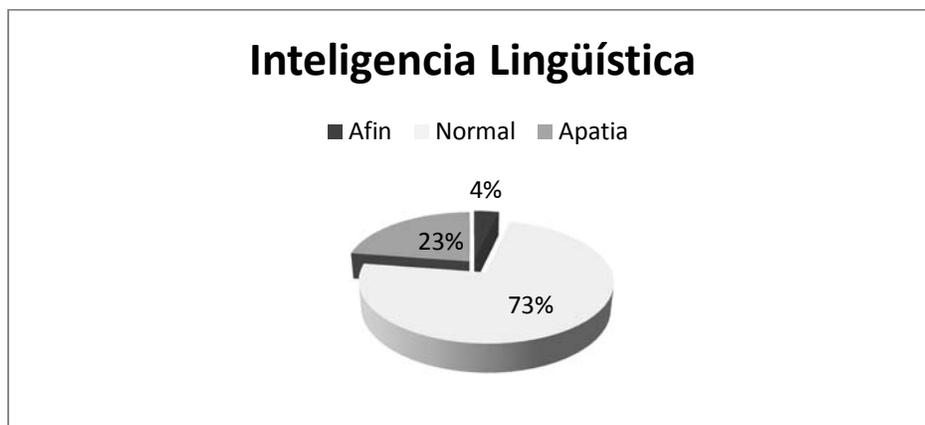


Figura 7.2 Gráfico afinidad de inteligencia lingüística en alumnos de primer año.

La siguiente tabla muestra la cantidad de alumnos que tienen afinidad hacia la inteligencia lógico-matemática:

Tabla 7.3 Afinidad de alumnos por la inteligencia lógico-matemática.

	Lógico-matemático
Afín	25
Normal	25
Apatía	3

La cantidad de alumnos afines hacia la inteligencia lógico-matemática muestra que la mayoría de alumnos de primer año de ingeniería les gusta o no tiene complicaciones resolviendo problemas, tanto de cálculo como lógico, realizando deducciones.

El siguiente gráfico representa los datos de la tabla anterior:

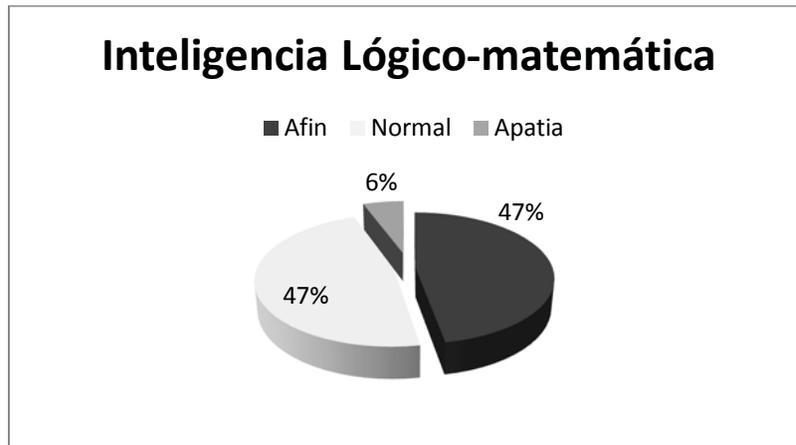


Figura 7.3 Gráfico afinidad de inteligencia lógico-matemática en alumnos de primer año.

La segunda parte trata de un conjunto de preguntas que tienen directa relación con el uso de las inteligencias; cada pregunta posee varias alternativas, de las cuales solo una es correcta.

En el caso de las preguntas sobre inteligencias lingüísticas y lógico-matemáticas fueron basadas en preguntas tipo PSU, por lo que se aplica el estudio PSU sobre factores de éxito en ingreso y permanencia en la universidad.

Este test tiene como fin estimar la capacidad intelectual de la persona, por cada pregunta, el alumno obtiene un puntaje, el cual al responder todas las preguntas se califica al alumno en alguna de las inteligencias en uno de los siguientes rangos:

- Inferior: si el alumno no contesta correctamente más de la mitad de las preguntas.
- Normal: si el alumno contesta más de la mitad de las preguntas correctamente
- Superior: si el alumno contesta todas las preguntas correctas

La siguiente figura se muestra los rangos de los alumnos luego de contestar las preguntas de la inteligencia espacial:

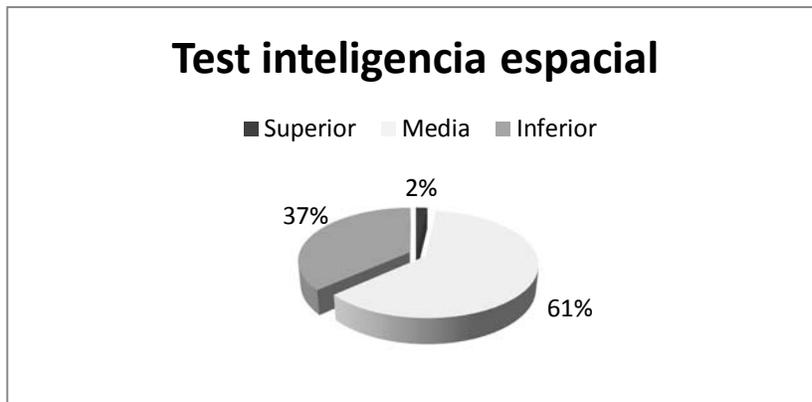


Figura 7.4 Gráfico de rangos alumnos en el test de inteligencia espacial.

En cantidades, solo existe 1 alumno en el rango superior, mientras que los alumnos que están en el rango medio es de 32 alumnos, de lo que se concluye que la mayoría de los alumnos de primer año de ingeniería están en un rango de inteligencia espacial medio, por lo que no sobresalen ni tienen problemas con las actividades que se relacionan a la inteligencia espacial.

El gráfico a continuación muestra los rangos de inteligencia de los alumnos luego de realizar el test de inteligencia lingüística:

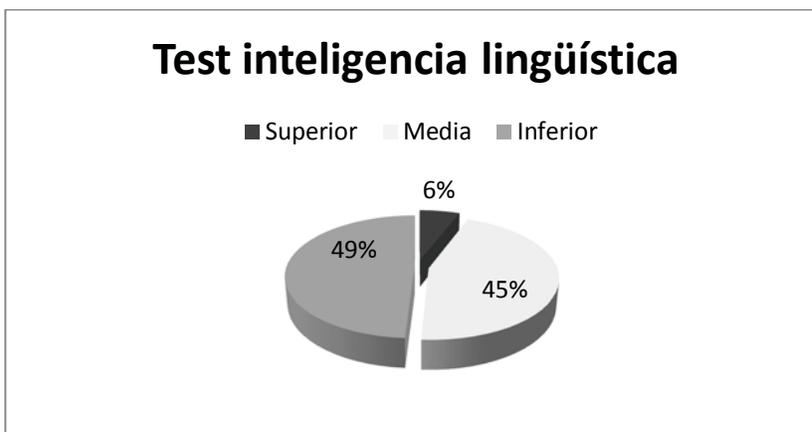


Figura 7.5 Gráfico de rangos alumnos en el test de inteligencia lingüística.

El gráfico anterior muestra que la cantidad de alumnos que están en un rango inferior de inteligencia es muy alto (en cantidad fueron 25 alumnos), esto se debe a que era el último ítem presentado en el test y no todos los alumnos alcanzan a terminarlo en el tiempo dado (20 minutos), por lo que hace que esta muestra sea poco representativa.

Por otro lado en el rango superior, si lo comparamos con la cantidad de personas que están dentro del rango, en la inteligencia espacial la cantidad es mayor en el área de la inteligencia lingüística (3 alumnos en el rango superior de la inteligencia lingüística).

Por conclusión la muestra indica que la mayoría de los alumnos de ingeniería no sobresalen mucho al aplicar la inteligencia lingüística ni en ningún área en la que tenga que utilizarse este tipo de inteligencia.

El siguiente gráfico muestra los rangos de inteligencia de los alumnos luego de realizar el test lógico-matemático:

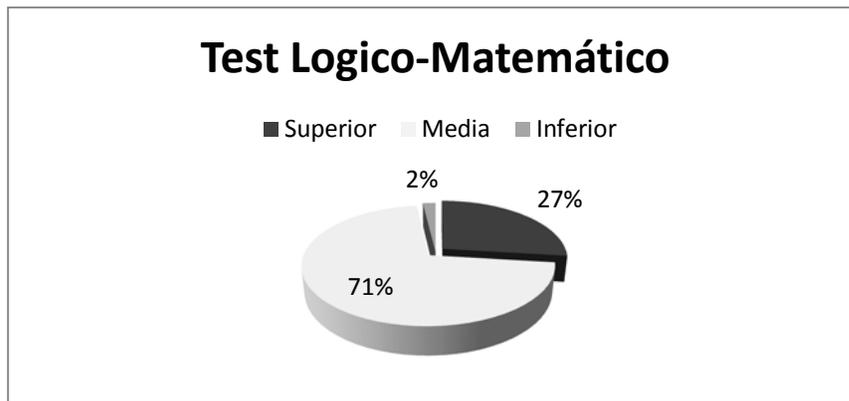


Figura 7.6 Gráfico de rangos alumnos en el test de inteligencia lógico-matemático.

La muestra indica que existe una gran cantidad de personas que sobresalen en este tipo de inteligencia, lo que es lógico pensando que en carreras de ingeniería las personas suelen utilizar mucho este tipo de inteligencia; en conclusión, la muestra indica que la mayoría de los alumnos de primer año de una carrera de ingeniería tienen una alta capacidad de razonamiento en el área lógico-matemática.

Con este estudio se llega a la conclusión, es de que este test muestra un resultado similar al entregado por el test MIDAS; lo que indica que el test MIDAS tiene validez sobre alumnos de carreras de ingeniería dentro de la universidad, aunque la muestra podría considerarse como no representativa, la cantidad de estudios hechos en diferentes países demuestran un resultado similar.

7.4. Ponderación de puntaje de test

En este apartado se explica las ponderaciones y la forma en que se mide el puntaje obtenido al realizar el test. En el test de la aplicación se tiene una medida de puntos con los cuales se estima las capacidades del usuario. La cantidad máxima de puntos que se podrá obtener en el test es de 100 puntos, repartidos en 25 preguntas diferentes, de lo cual se infiere que cada pregunta tiene 4 puntos. Las preguntas se dividen en 10 preguntas para la inteligencia lógico-matemática, 8 preguntas para la inteligencia lingüística y 7 preguntas para la inteligencia espacial.

La cantidad de preguntas a contestar en el test y el número de preguntas por cada inteligencia se toma teniendo cuenta que la inteligencia lógico-matemática es la más importante para las personas que quieran entrar a carreras de ingeniería.

Luego sigue la inteligencia lingüística que entrega las capacidades de entender los problemas planteados.

Por último la inteligencia espacial para poder abstraer los problemas en los cuales hay objetos o escenarios y poder analizarlos mentalmente. La cantidad de puntaje máximo y la cantidad se especifican para poder tener un mejor tipo de medición.

A continuación se muestra una tabla donde se explica la ponderación de puntajes y los porcentajes en cuanto a puntos de estos mismos:

Tabla 7.4 Ponderación de puntajes de test.

			Tipos de inteligencias					
			Lógico -matemática		Lingüística		Espacial	
	Porcentaje	Puntos	Puntaje	Respuestas	Puntaje	Respuestas	Puntaje	Respuestas
	100%	100	40	10 correctas	32	8 correctas	28	7 correctas
	75%	76	32	8 correctas	24	6 correctas	20	5 correctas
	50%	52	20	5 correctas	16	4 correctas	16	4 correctas
	25%	28	12	3 correctas	8	2 correctas	8	2 correctas
			Ponderación de puntajes					
Nivel	Porcentaje	Puntos	Lógico -matemática		Lingüística		Espacial	
Deficiente	0% - 20%	0 a 20	----		----		----	
Bajo	20% - 40%	20 a 40	mínimo 25% correctas		mínimo 25% correctas		----	
Medio	40% - 60%	40 a 60	mínimo 50% correctas		mínimo 25% correctas		mínimo 25% correctas	
Normal	60% - 80%	60 a 80	mínimo 75% correctas		aprox 50% correctas		mínimo 25% correctas	
Excelente	80% - 100%	80 a 100	mínimo 100% correctas		mínimo 50% correctas		mínimo 50% correctas	

8. Especificación de requerimientos

8.1. Diagrama BPM

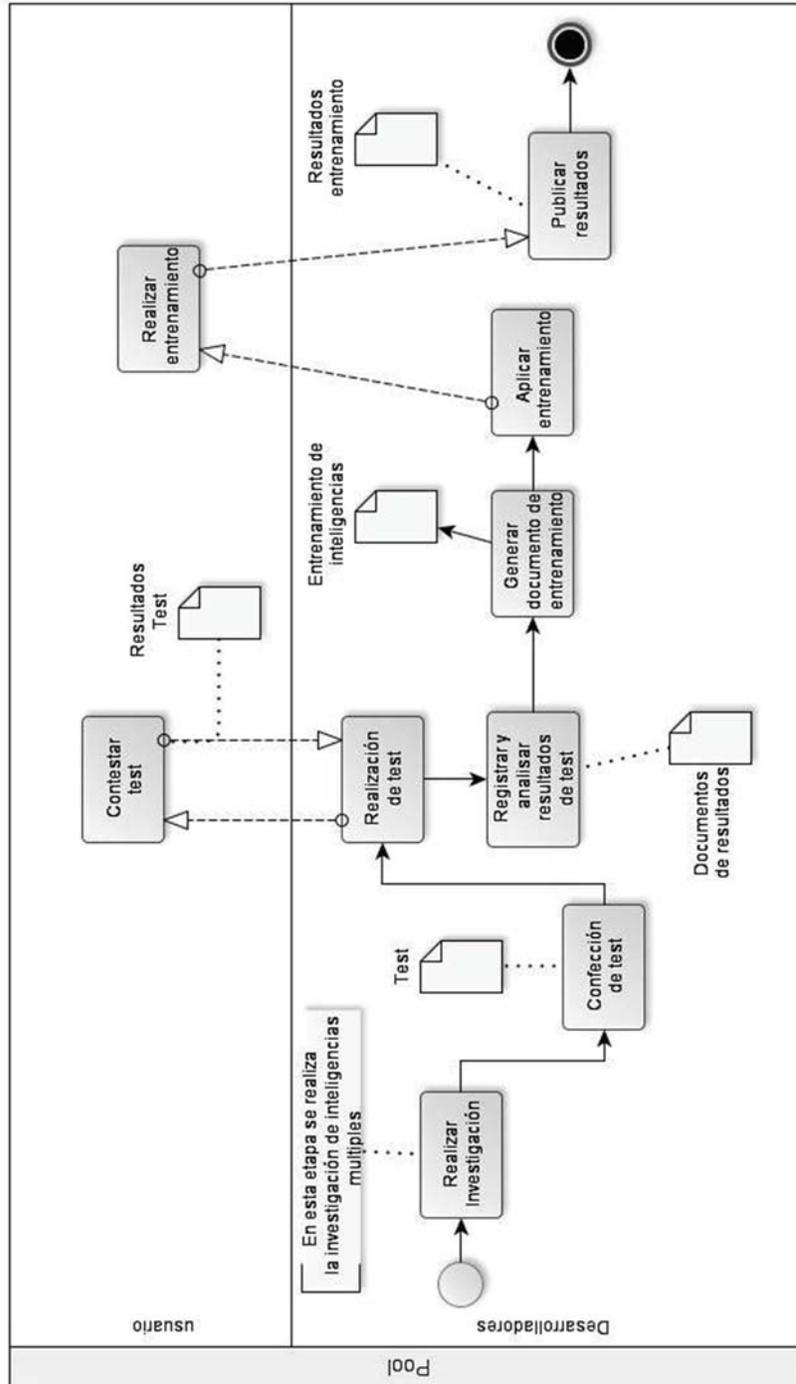


Figura 8.1 Diagrama de proceso de negocio a nivel general.

En la figura anterior se muestran los procesos que ocurren dentro del sistema planteado, de una forma general, cada subproceso será detallado dentro de este documento, mediante casos de uso, diagramas de actividad y diagramas de secuencia.

Del diagrama anterior se realiza la siguiente especificación de requerimientos, la cual además toma en cuenta el estudio realizado y crear tres juegos, cada uno basado en una de las inteligencias múltiples seleccionadas para el test aplicado.

8.2. Especificación de requerimientos funcionales

Tabla 8.1 Requerimientos funcionales numero 1 a 3.3.

Número	Definición	Estado
1.	La aplicación tendrá una pantalla principal con opciones. <i>-Las opciones serán: Test, juegos, entrenamiento, puntajes, opciones y salir</i>	Implementado
2.	Test <i>-En el juego se podrá hacer un test de inteligencias múltiples</i>	Implementado
2.1.1	El test evaluará el potencial de la persona. <i>-Potencial que tiene la persona para ingresar a una carrera ingenieril</i>	Implementado
2.1.2	El test evaluará 3 tipos de inteligencias. <i>-La inteligencia lógico – matemática (10 preguntas), la inteligencia espacial (8 preguntas) y la inteligencia lingüística (7 preguntas).</i>	Implementado
2.2	El test puede ser dado varias veces.	Implementado
2.2.1	Las preguntas variaran cada vez que se realice el test. <i>-Existe una pequeña probabilidad que las preguntas se repitan</i>	Implementado
2.3	El juego guardará el resultado de los test que aplique el usuario. <i>-Si se tiene conexión a internet se subirán al servidor del juego</i>	Implementado
2.4	El test indicará los puntajes obtenidos luego de su realización.	Implementado
2.5	Se mostrará el puntaje del test anterior. <i>-Solo si se ha realizado anteriormente un test</i>	Implementado
3.	La aplicación tendrá diferentes tipos de juegos <i>-Los juegos serán relacionados a diferentes tipos de inteligencias.</i>	Implementado
3.1	Cada uno de los juegos podrá ser jugado libremente por el usuario.	Implementado
3.2	Con los puntajes de los juegos se estimará la capacidad intelectual.	Implementado
3.3	Existirán 3 distintos juegos. <i>-Los juegos son: la matemática de las figuras, el cañón de palabras y mecanismo de escape</i>	Implementado

Tabla 8.2 Requerimientos funcionales numero 3.3.1 a 3.3.3.

Número	Definición	Estado
3.3.1	La matemática de las figuras. <i>-Juego basado en inteligencia matemática y un poco de lógico</i>	Implementado
3.3.1.1	Juego en que se tiene que armar figuras con diferentes piezas.	Implementado
3.3.1.2	Cada pieza tendrá un número.	Implementado
3.3.1.3	La Figura tiene un número que se debe igualar. <i>-La figura que se tiene que armar o desarmar tiene un número determinado que se tiene que igualar con la suma de los números de las piezas</i>	Implementado
3.3.1.4	Para superar un nivel se tiene que Armar la figura y coincidir con el número en las piezas.	Implementado
3.3.1.5	El tiempo en cada nivel irá disminuyendo a medida que se avance en el juego. <i>-Mientras más se avance menos tiempo, así como la dificultad también ira aumentando</i>	Implementado
3.3.1.6	El juego tendrá un botón de funciones matemáticas. <i>-El botón contendrá las funciones efectuara la operación sobre la figura al agregar cada pieza</i>	Implementado
3.3.1.6.1	Función de sumar. <i>-Suma el número de la pieza que se agrega a la figura</i>	Implementado
3.3.1.6.2	Función de restar. <i>-Resta el número de la pieza que se agrega a la figura</i>	Implementado
3.3.2	Mecanismo de escape. <i>-Juego que trabaja la inteligencia espacial y lógico matemática</i>	Implementado
3.3.2.1	El juego se tiene que llevar una bola a través de un laberinto. <i>-La bola en un principio está encerrada en una esquina</i>	Implementado
3.3.2.2	Se deben insertar engranajes en ciertas partes del laberinto. <i>-Los engranajes de deben insertar en diferentes ranuras que tienen el tamaño y color de los engranajes</i>	Implementado
3.3.2.3	Los engranajes se mueven a través del laberinto. <i>-Todos los engranajes se mueven a la vez dentro del laberinto</i>	Implementado
3.3.2.4	Cuando se insertan todos los engranajes se libera la bola.	Implementado
3.3.2.5	Cuando la bola sale de su posición inicial aparece una bola de color negro.	Implementado
3.3.2.6	El juego se termina cuando se lleva la bola blanca a la posición final que se mostró en un principio en el juego.	Implementado
3.3.2.7	Si la bola de color negro toca a la de color blanco, el juego termina.	Implementado
3.3.2.8	El juego tendrá 3 intentos por nivel. <i>-Cada vez que se toquen dos engranajes se pierde un intento</i>	Implementado
3.3.2.9	Si se pierden los 3 intentos, termina el juego	Implementado
3.3.3	Cañón de palabras	Implementado

Tabla 8.3 Requerimientos funcionales números 3.3.3.1 a 4.6.

Número	Definición	Estado
3.3.3.1	El juego se dispara un cañón hacia un lote de palabras. <i>-El lote de palabras baja por la pantalla y cada fila del lote contiene 4 palabras.</i>	Implementado
3.3.3.2	En el juego aparece una pregunta. <i>-Esta pregunta es la que guía al jugador para disparar al lote de palabras que cae por pantalla</i>	Implementado
3.3.3.3	La palabra disparada elimina la fila si responde la pregunta. <i>-Si la palabra golpeada responde la pregunta se elimina la fila, tomando en cuenta preguntas como antónimos, sinónimos, grupos de palabras, etc.</i>	Implementado
3.3.3.4	El lote de palabras cae a una velocidad constante en cada nivel. <i>-El margen del cual comienzan a caer el lote de preguntas, varía según la dificultad.</i>	Implementado
3.3.3.5	Si se equivoca en lanzar la palabra habrá una penalización. <i>-Se disminuye el margen en el cual bajan las palabras.</i>	Implementado
3.3.3.6	El juego termina cuando una barra colisiona con el cañón. <i>-La pantalla se va llenando de fila de palabras, las cuales bajan a medida que pasa el tiempo.</i>	Implementado
3.3.3.7	Para ganar el juego hay que eliminar la cantidad indicada de palabras. <i>-Cada vez que se responde la pregunta correctamente se elimina una fila y el contador aumenta en 1.</i>	Implementado
3.4	Cada juego poseerá diferentes fases.	Implementado
3.4.1	Cada fase se dividirá en niveles. <i>-Cada fase de cada juego se dividirá entre 4 o 6 niveles por fases</i>	Implementado
3.4.2	Al completar todos los niveles de una fase se desbloquea la siguiente.	Implementado
3.4.3	Cada juego, con excepción de cañón de palabras, tendrá un límite de tiempo. <i>-Luego de ese límite de tiempo, el juego concluirá</i>	Implementado
3.4.3.1	Los juegos concluirán si se termina el tiempo y no se completa el objetivo del nivel.	Implementado
3.4.3.2	Mientras menor sea el tiempo en superar un nivel mayor será el puntaje dado.	Implementado
4.	Los juegos tendrán un modo de entrenamiento.	Implementado
4.1	La aplicación sugerirá juegos al usuario. <i>-Al utilizar los juegos sugeridos, permite potenciar las inteligencias evaluadas en el test</i>	Propuesto
4.2	El entrenamiento tendrá niveles de los 3 juegos pero con leves modificaciones.	Implementado
4.2.1	Muestra el resultado del test y el juego asociado <i>-Se muestran los puntajes del test y el juego que el usuario puede utilizar para fortalecer esa habilidad.</i>	Implementado /Cambiado
4.3	El usuario puede potenciar cualquier inteligencia.	Implementado
4.4	El nivel de dificultad es ajustado automáticamente.	Implementado
4.5	Se mostrara la diferencia de puntajes cada vez que juegue el entrenamiento. <i>-La diferencia de puntaje cada vez que se entrena indicara un pequeño porcentaje de aumento de esa inteligencia.</i>	Propuesto
4.6	Los juegos de entrenamiento terminan cuando el usuario pierde. <i>-En caso del mecanismo de palabras, no posee límite de tiempo, los demás juegos sí.</i>	Implementado

Tabla 8.4 Requerimientos funcionales números 5 a 6.3.

Número	Definición	Estado
5.	Opciones.	Implementado
5.1	Se puede ajustar la dificultad de los juegos. <i>-La dificultad es igual para cada juego.</i>	Implementado
5.2	Se podrá activar y desactivar el sonido de la aplicación.	Implementado
6.	Puntajes.	Implementado
6.1	Se mostraran los diferentes juegos y sus puntajes.	Implementado
6.2	Se mostraran los 3 mejores puntajes por cada juego	Implementado
6.3	Se podrán ver los puntajes globales. <i>-Para esta opción se necesitara tener conexión a internet.</i>	Implementado

8.3. Especificación de requerimientos no funcionales

Tabla 8.5 Requerimientos no funcionales.

Número	Descripción y notas	Estado
1.	Antes de jugar cada juego por primera vez existirá una breve explicación del juego en sí. <i>-Se aplica lo mismo con el entrenamiento</i>	Implementado
2.	Al jugar un juego por primera vez no se podrán elegir fases o niveles. <i>-Luego de pasar cierta fase y cierto nivel, se podrán seleccionar en el juego.</i>	Implementado
3.	Solo existirá un usuario para cada dispositivo móvil. <i>-Los puntajes serán asociados a ese usuario</i>	Implementado
4.	El programa será implementado en un dispositivo móvil bajo la plataforma Android.	Implementado
5.	La aplicación funcionará bajo la plataforma Java.	Implementado
6.	Se utilizara el motor de juegos AndEngine.	Implementado
7.	La aplicación se implementara sobre la versión Android SDK 2.3.3 correspondiente a la versión de api 10.	Implementado
8.	Se utilizara la plataforma de desarrollo eclipse en Windows 7.	Implementado

9. Análisis de requerimientos.

9.1. Enfoque utilizado

Como se programó en Android, el cual utiliza como lenguaje de programación java, por lo que se usó orientación a objetos, además al utilizar eclipse se pudo complementar el desarrollo de las interfaces con el metalenguaje XML.

La gran ventaja que da esta mezcla entre XML y JAVA es la versatilidad que se tiene al momento de programar dada la gran cantidad de información, el código existente en línea y herramientas que ayudan a desarrollar en este tipo de ambiente.

La programación orientada a objetos proporciona las siguientes ventajas:

Uniformidad: ya que es la representación de los objetos lleva implícito tanto el análisis como el diseño y la codificación de los mismos.

Comprensión: tanto los datos que componen los objetos, como los procedimientos que los manipulan, están agrupados en clases, que se corresponden con las estructuras de información que el programa usa.

Flexibilidad: al tener relacionados los procedimientos que manipulan los datos con los datos a tratar, cualquier cambio que se realice sobre ellos quedará reflejado automáticamente en cualquier lugar donde estos aparezcan.

Estabilidad: dado que permite un tratamiento diferenciado de aquellos objetos que permanecen constantes en el tiempo sobre aquellos que cambian con frecuencia permite aislar las partes del programa que permanecen inalterables en el tiempo.

9.2. Casos de uso

Se muestran los casos de uso de los requerimientos.

9.2.1. Caso de uso general

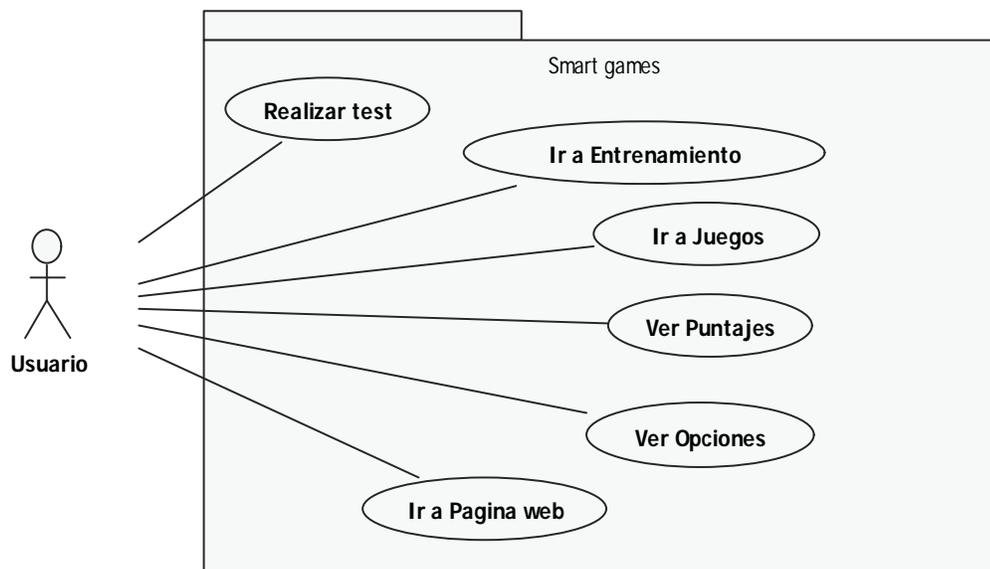


Figura 9.1 Caso de uso general de la aplicación Smart Games.

9.2.1.1. Caso de uso narrativo extendido para Smart Games

Tabla 9.1 Caso de uso narrativo Smart Games.

Caso de uso:		Smart Games	
Descripción:		Caso de uso general de la aplicación, donde se muestran las opciones generales del sistema	
Actores:		Usuario	
Pre-condiciones:			
Post-condiciones:			
	Actor		Sistema
1°	El usuario puede elegir entre las siguientes opciones: 1 - Realizar test. 2 - Ir a Entrenamiento. 3 - Ir a Juegos. 4 - Ver Puntajes. 5 - Ver Opciones.		
		2°	El programa ejecutara las siguientes acciones dependiendo de la opción elegida: 1 - Entra a la pantalla para realizar el test. 2 - Entra a la sección de juegos de entrenamiento. 3 - Entra a la sección de juegos, donde podrá elegir entre los 3 juegos de la aplicación. 4 - Entra a la pantalla de puntajes, donde se observan los mejores puntajes de los juegos. 5 - Entra a la pantalla de opciones del juego

9.2.2. Caso de uso juegos

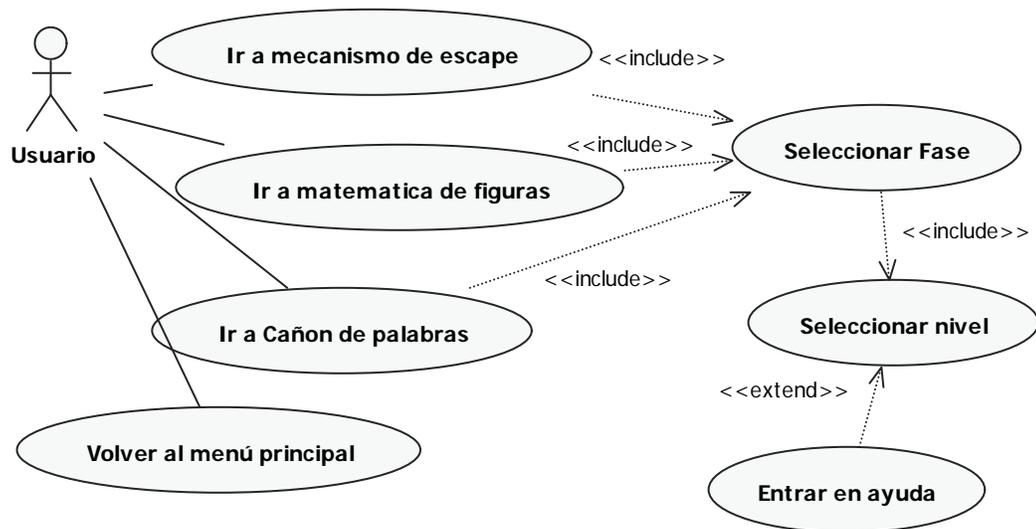


Figura 9.2 Caso de uso para la opción de juegos.

9.2.2.1. Caso de uso narrativo extendido para Juegos

Tabla 9.2 Caso de uso narrativo Juegos.

Caso de uso:		Juegos	
Descripción:		Caso de uso donde se muestran las opciones que tendrá el usuario en el menú de juegos	
Actores:		Usuario	
Pre-condiciones:		Seleccionar la opción juegos en el menú principal	
Post-condiciones			
	Actor		Sistema
1°	El usuario puede elegir las siguientes opciones: 1 - Jugar Mecanismo de escape. 2 - Jugar Matemática de las figuras. 3 - Jugar Cañón de palabras. 4 - Volver al menú principal.		
		2°	El sistema mostrara la pantalla de fases donde el usuario puede elegir las siguientes opciones: 1 – Fase novato. 2 – Fase media. 3 – Fase Avanzado. 4 – Fase Experto.
3°	El usuario debe ingresar una opción de fase		
5°	El usuario elige una fase		
		4°	Luego de que el usuario elige la fase, tiene que seleccionar cualquier de los niveles de dicha fase.
6°	El usuario elige nivel		
		7°	Si la ayuda esta activada, el usuario accede a la pantalla de ayuda de cada juego.
8°	El usuario presiona sobre la pantalla.		
		9°	Se inicia el juego en el nivel seleccionado.

9.2.3. Caso de uso entrenamiento

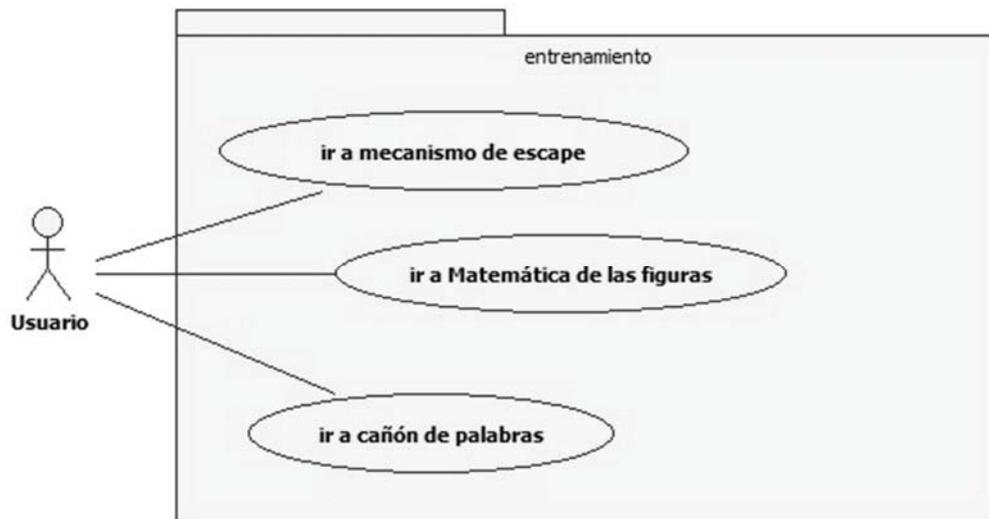


Figura 9.3 Caso de uso para la opción Entrenamiento.

9.2.3.1. Caso de uso narrativo extendido para entrenamiento

Tabla 9.3 Caso de uso narrativo Entrenamiento.

Caso de uso:		Entrenamiento	
Descripción:		Caso de uso donde se muestran las opciones que tendrá el usuario en el menú de entrenamiento	
Actores:		Usuario	
Pre-condiciones:		seleccionar la opción entrenamiento en el menú principal	
Post-condiciones			
	Actor		Sistema
1°	El usuario puede elegir las siguientes opciones 1 – Ir a mecanismo de escape 2 – Ir a matemática de figura 3 – Ir a cañón de palabras		
		2°	El programa ejecutara las siguientes acciones dependiendo de la opción elegida: 1 – mecanismo de escape. 2 –matemáticas de figuras. 3 -cañón de palabras.
3°	1 – El usuario comienza a jugar mecanismo de escape y finaliza la partida. 2 – El usuario comienza a jugar matemáticas de figuras y finaliza la partida. 3 – El usuario comienza a jugar cañón de palabras y finaliza la partida.		
		4°	1 –El sistema muestra el puntaje de mecanismo de escape. 2 – El sistema muestra el puntaje de matemáticas de figuras. 3– El sistema muestra el puntaje de cañón de palabras.
5°	El usuario presiona el botón volver.		
6°			El sistema vuelve a la pantalla de entrenamiento.

9.2.4. Caso de uso test

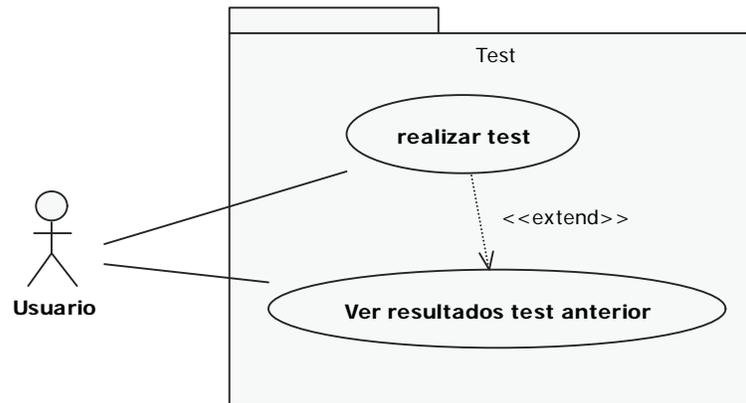


Figura 9.4 Caso de uso para la opción Test.

9.2.4.1. Caso de uso narrativo extendido para test

Tabla 9.4 Caso de uso narrativo Test.

Caso de uso:		Test	
Descripción:		Caso de uso donde se muestran las opciones que tendrá el usuario en el menú de test	
Actores:		Usuario	
Pre-condiciones:		seleccionar la opción test en el menú principal	
Post-condiciones			
	Actor		Sistema
1°	El usuario puede elegir las siguientes opciones: 1 - Realizar test. 2 - Ver resultados test anterior		
		2°	El programa ejecutara las siguientes acciones dependiendo de la opción elegida: 1 - Se ingresa a la pantalla del test, donde se tendrán que responder las preguntas 2 - Se ingresa a la pantalla donde se muestran los puntajes del test anterior.
3°	Si el usuario ingresa a la pantalla de los resultados, se mostraran la opción de iniciar al test.		
		4°	Si el usuario selección iniciar test, se ingresa a la pantalla donde se tendrán que responder las preguntas del test.

9.2.5. Caso de uso opciones

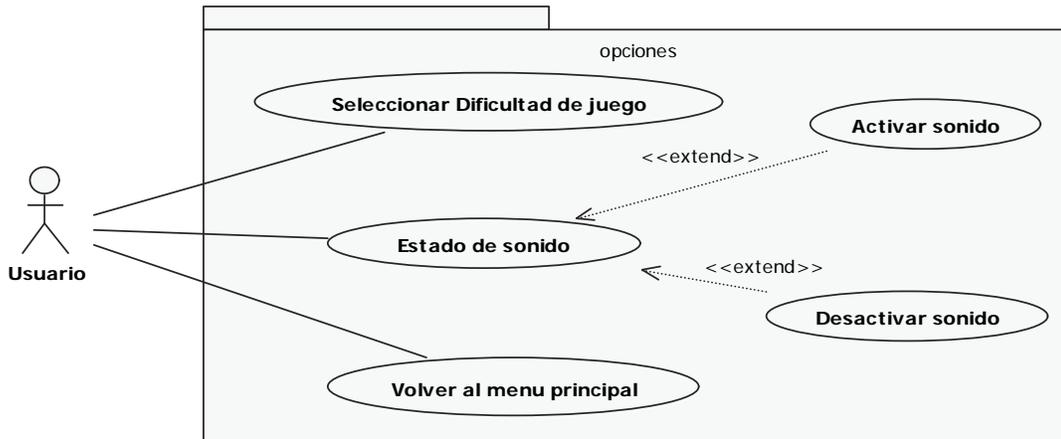


Figura 9.5 Caso de uso para las opciones del sistema.

9.2.5.1. Caso de uso narrativo extendido para Opciones

Tabla 9.5 Caso de uso narrativo Opciones.

Caso de uso:		Opciones	
Descripción:		Caso de uso donde se muestran las opciones que tendrá el usuario en el menú de opciones	
Actores:		Usuario	
Pre-condiciones:		seleccionar la opción entrenamiento en el menú principal	
Post-condiciones			
	Actor		Sistema
1°	El usuario puede elegir las siguientes opciones: 1 - Seleccionar dificultad de juego. 2 - Estado de sonido. 3 - volver al menú principal.		
		2°	El programa ejecutara las siguientes acciones dependiendo de la opción elegida: 1 - aparece una ventana donde el usuario puede elegir la dificultad del juego. 2 - Se activa o se desactiva el sonido dependiendo de su estado anterior 3 - Se muestra el menú principal
3°	Si el usuario eligió Seleccionar dificultad de juego puede elegir entre las siguientes opciones: 1 - Fácil. 2 - Medio 3 - Difícil.		
		4°	Dependiendo de la opción que ingrese el usuario se modificara la dificultad de los juegos.

9.2.6. Caso de uso puntajes

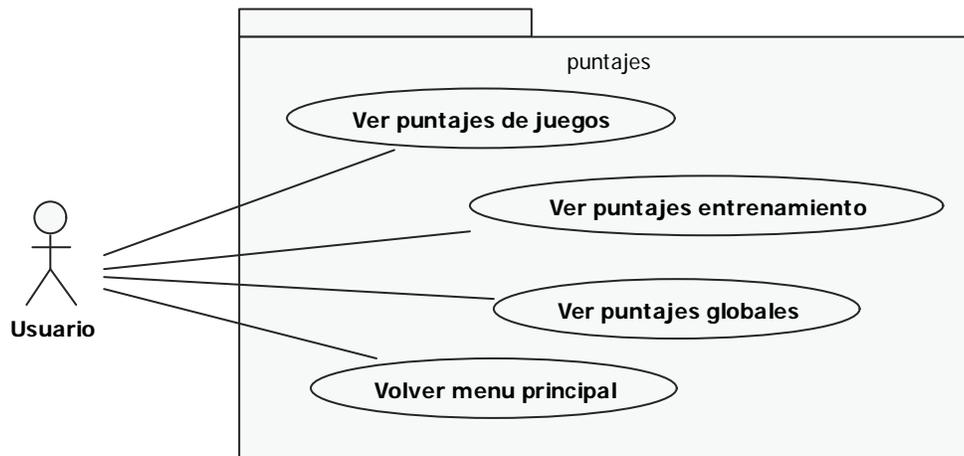


Figura 9.6 Caso de uso para opción de puntajes.

9.2.6.1. Caso de uso narrativo extendido para puntajes

Tabla 9.6 Caso de uso narrativo Puntajes.

Caso de uso:		Puntajes	
Descripción:		Caso de uso donde se muestran las opciones que tendrá el usuario a la pantalla de puntajes.	
Actores:		Usuario	
Pre-condiciones:		Seleccionar la opción puntajes en el menú principal.	
Post-condiciones			
	Actor		Sistema
		1°	El sistema muestra por pantalla los puntajes de todos los juegos
2°	El usuario puede elegir las siguientes opciones: 1 - Ver puntajes globales 2 - Ver puntajes entrenamiento 3 - Volver al menú principal		
		2°	El programa ejecutara las siguientes acciones dependiendo de la opción elegida: 1 - Se muestran los mejores puntajes a nivel global para cada juego. 2 - Se muestran los puntajes totales de los entrenamientos en cada área. 3 - Se vuelve al menú principal.

10. Diagramas de actividad de los juegos

Se realizan diagramas de actividad para representar de manera gráfica el proceso que se lleva a cabo en cada juego. A continuación se muestran los diagramas para los 4 juegos y se entrega una pequeña explicación de cada imagen:

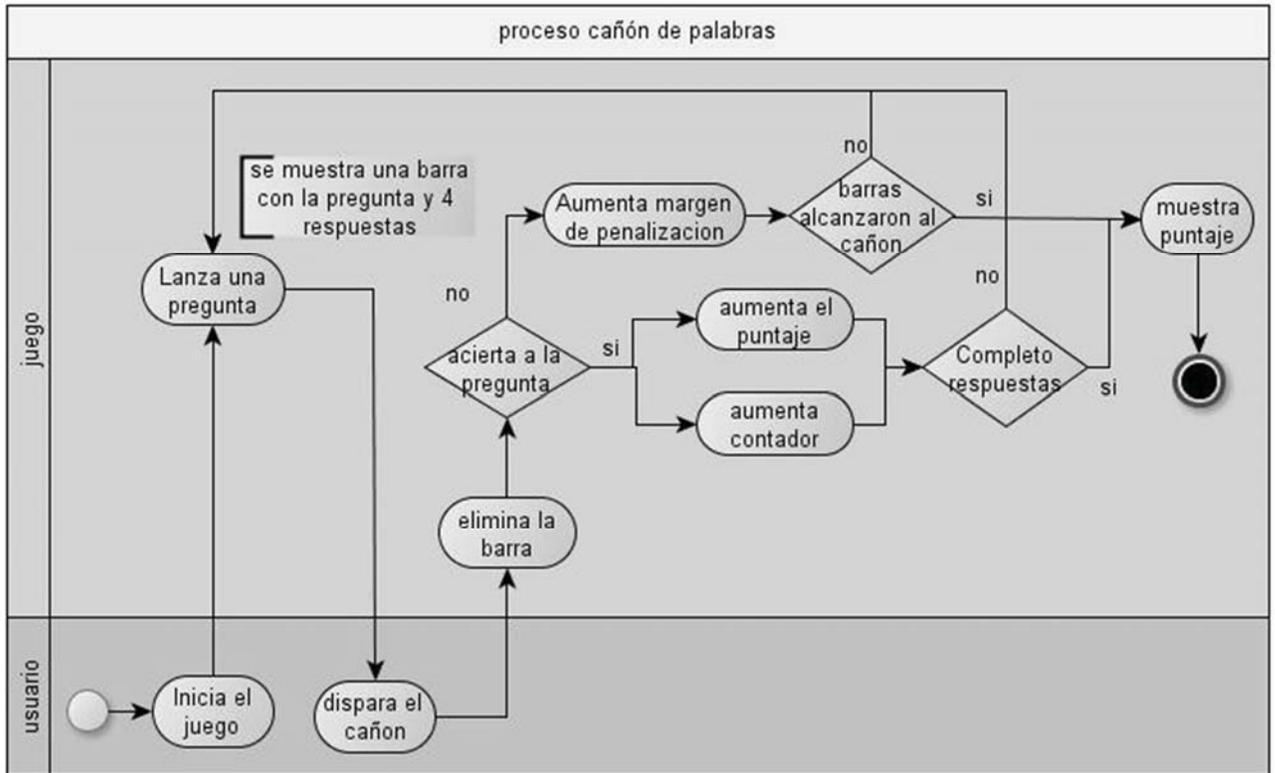


Figura 10.1 Diagrama de Cañón de palabras.

- **Inicia el juego:** el juego comienza cuando el usuario accede desde la pantalla de selección de juegos a “Cañón de palabras”. Luego se cargan las imágenes de fondo, el marcador de puntaje, el marcador de respuestas acertadas y los demás “sprites”.
- **Lanza una pregunta:** el juego lanza una pregunta desde la parte superior de la pantalla.
- **Dispara el cañón:** Luego de que están todos los “sprites” cargados y se lanza la primera pregunta el usuario puede disparar el cañón que se mueve de izquierda a derecha y viceversa.
- **Elimina la barra:** cuando la bala golpea a la barra, esta se elimina.

- **Acierta a la pregunta:** Si la bala disparaba por el cañón choca con una palabra que responde la pregunta planteada se **incrementa el puntaje** y se **incrementa el contador** de respuestas correctas, en el caso contrario, si no responde la pregunta, se **incrementa el margen de penalización** en la pantalla.
- **Completo respuestas:** Si al momento de que se contesta correctamente la pregunta se llega al número de respuestas dado por nivel, se termina el juego y se **muestra el puntaje**.
- **Barras alcanzaron el cañón:** si alguna barra de preguntas llega a tocar al cañón, el juego se termina y se **muestra el puntaje**.

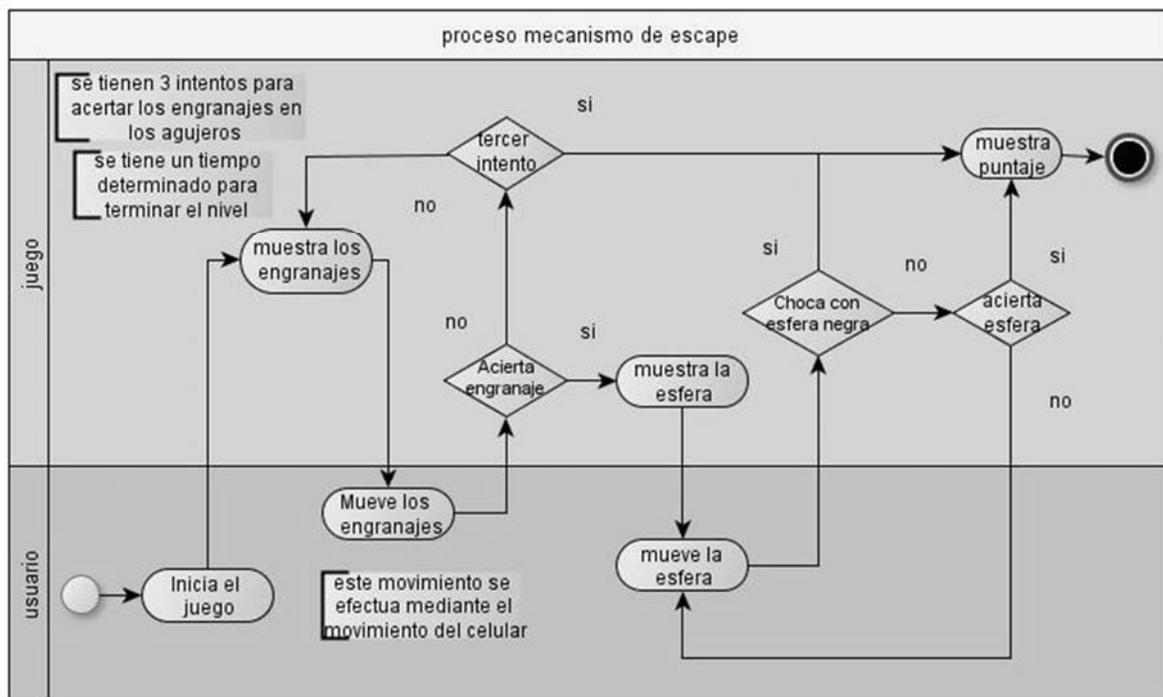


Figura 10.2 Diagrama de Mecanismo de escape.

- **Inicia el juego:** el juego comienza cuando el usuario accede desde la pantalla de selección de juegos a "Mecanismo de escape". Luego se cargan las imágenes de fondo, el marcador de puntaje, el marcador de respuestas acertadas y los demás "sprites".
- **Muestra los engranajes:** después de cargar el laberinto y los demás "sprites" se muestran los engranajes que deben ser encajados en los diferentes orificios para activar el mecanismo que destraba la esfera.
- **Mueve los engranajes:** el usuario puede mover los engranajes haciendo movimientos con el dispositivo, en esta sección se usa acelerómetro.

- **Acierta engranaje:** Si el usuario acierta los engranajes que se muestran en la pantalla, se **muestra la esfera**, aparece la esfera negra y se puede continuar con el juego. En el caso de que choque engranajes y se excede el **tercer intento** para colocar los engranajes en sus posiciones, **se muestra el puntaje** y el juego termina
- **Mueve la esfera:** cuando se aciertan los engranajes se puede mover la esfera por el laberinto.
- **Choca con esfera negra:** Si al momento de mover la esfera esta choca con la esfera negra, se termina el juego y **se muestra el puntaje**.
- **Acierta esfera:** si el usuario acierta la esfera en el lugar señalado el juego termina y **se muestra el puntaje** obtenido, en caso contrario se puede seguir moviendo la esfera.

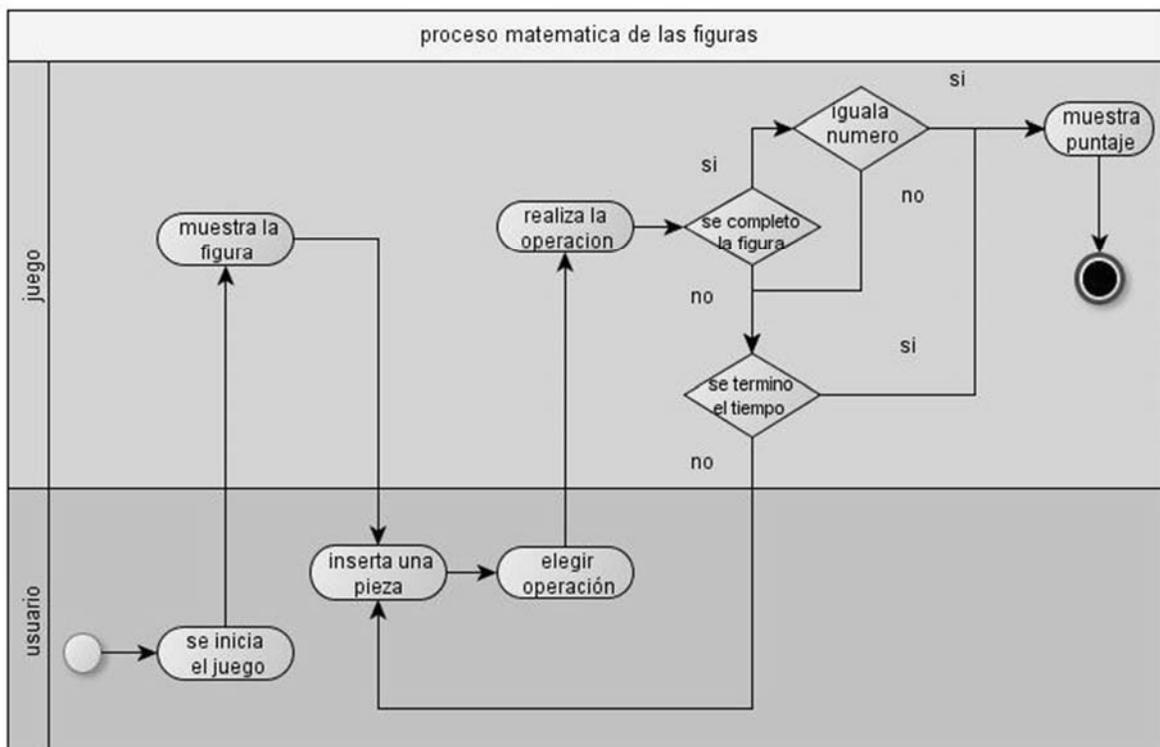


Figura 10.3 Diagrama de Matemática de las figuras

- **Inicia el juego:** el juego comienza cuando el usuario accede desde la pantalla de selección de juegos a “Matemática de las figuras”. Luego se cargan las imágenes de fondo, el marcador de puntaje, el marcador de respuestas acertadas y los demás “sprites”.

- **Muestra la figura:** se muestra una sombra de la figura que se debe construir con las piezas disponibles y se muestra el número al cual se debe llegar.
- **Insertar una pieza:** esto sucede cuando el usuario arrastra una pieza desde cinta de piezas a la figura y luego de esto se tiene que **elegir la operación** a realizar cuando esta se agrega a la figura final.
- **Realizar la operación:** acá se realiza la operación seleccionada sobre el contador que se lleva por todas las otras piezas ingresadas anteriormente.
- **Se completó la figura:** si se completa la figura mostrada como sombra con las piezas dadas se pasa a preguntar si se **iguala el número** presentado al principio, si es así el juego se completa y **se muestra el puntaje** obtenido.
- **Se terminó el tiempo:** si no se consigue armar la figura o no se consigue llegar al número determinado se termina el juego y se muestra el puntaje obtenido hasta ese momento.

11. Diseño

11.1. Diagrama de clases

Al utilizar Android cada interfaz representa una clase, el diagrama de clases es muy grande como para incluirlo completo en una sola hoja, por lo cual es dividido en las siguientes partes:

11.1.1. Diagrama de clases principales

Estas clases son las que se implementaran las principales funcionalidades del programa, algunas de las cuales, se relacionan con otras clases para cumplir con su funcionalidad.

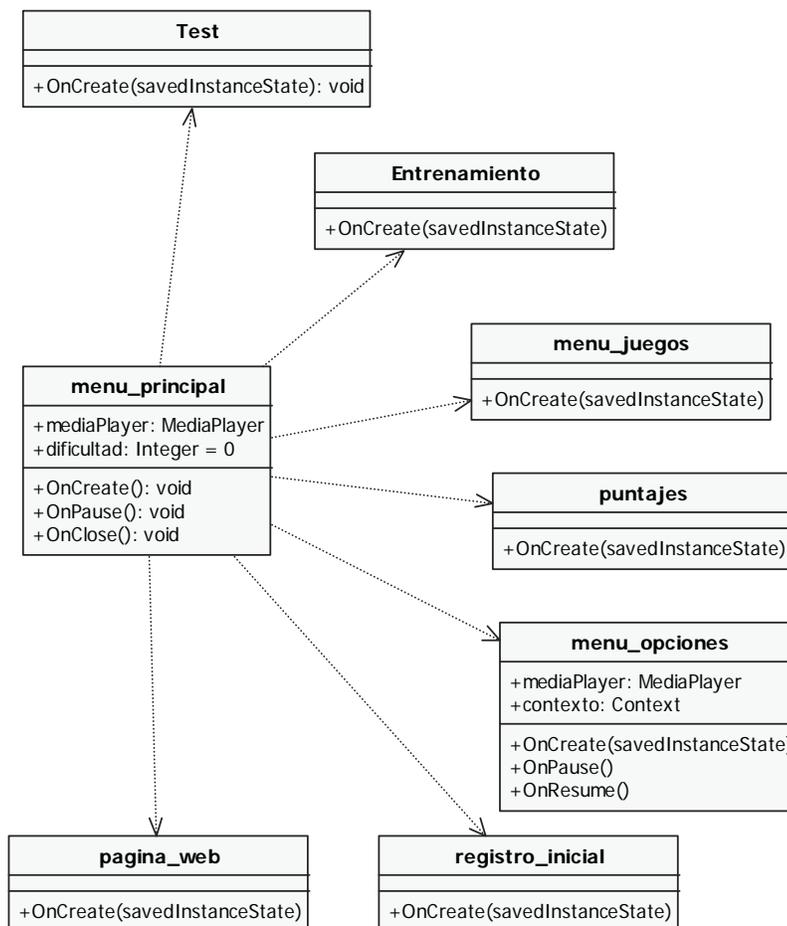


Figura 11.1 Diagrama de clases principales.

- La clase `menu_principal`, que es la primera con la que interactúa el usuario y esta posee los siguientes métodos y atributos:
 - El objeto de tipo `MediaPlayer` es el objeto que se encarga de almacenar y reproducir la música de fondo, en el caso de las actividades normales.
 - `OnCreate()`: Contiene las acciones que se realizarán cuando se cree la actividad.
 - `OnPause()`: Contiene las acciones que se realizarán cuando se pause la actividad, esto pasa, por ejemplo cuando en el “Smartphone” se presiona el botón “HOME”.

- La clase `registro_inicial` se ejecuta la primera vez que se ejecute la aplicación, permitirá el registro del usuario en la aplicación.

- En el caso de las demás clases, la función `OnCreate(savedInstanceState)` se encarga de pasar el control a el objeto instanciado de la clase luego de que sea llamado, con la posibilidad de entregar algunos parámetros del objeto que llamo a la clase.

- Como nota se puede agregar que al utilizar actividades en Android, la interfaz anterior es la que instancia a la siguiente, por lo que se utilizaron mayoritariamente dependencias entre clases, de tal forma que el diagrama de clase completo representa un árbol jerárquico con el `menu_principal` como nodo raíz.

11.1.2. Diagrama de clases test

Son las clases que se implementaran para realizar los test en la aplicación.

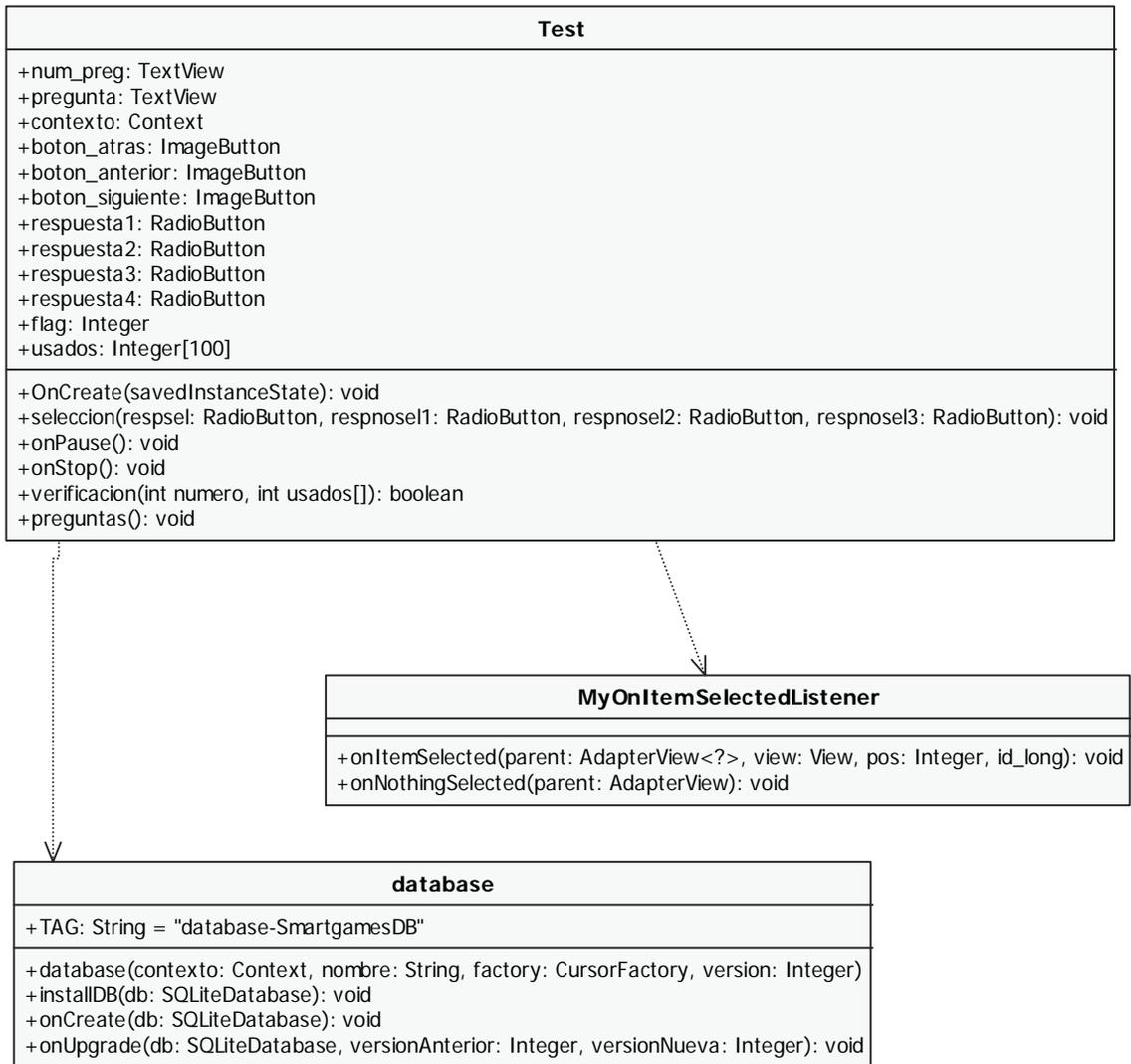


Figura 11.2 Diagrama de clases test.

11.1.2.1. Clase Test.

- En el diagrama de clases test los atributos poseen las siguientes clases de Android:
 - La clase TextView permite instanciar mensajes de caracteres.
 - La clase Context permite crear un objeto que contenga el contexto de la aplicación la cual guarda información global acerca del entorno de la aplicación.
 - La clase ImageButton permite crear un botón que posee una imagen en su interior.
 - La clase RadioButton, similar a los radio en html, permiten seleccionar solo una alternativa.
- El atributo flag es usado para indicar si una respuesta ha sido respondida o no.
- El vector de enteros usados contiene las preguntas que ya se han hecho, para que no se repitan esas preguntas.
- La función selección revisa si se ha seleccionado un radio button luego de pulsar el botón siguiente.
- La función onStop() permite realizar acciones luego de que la aplicación se detenga.

11.1.2.2. Clase database

En el caso de la clase database, es la clase que contiene la base de datos local usada para almacenar los puntajes y las preguntas del test; contiene los siguientes atributos y funciones:

- El String TAG contiene el nombre de la base de datos por defecto.
- La función installDB se encarga de ejecutar las consultas que crearan la base de datos local.
- La función onCreate se encarga de asignar las consultas a la variable de base de datos.
- La función onUpgrade se encarga de actualizar la base de datos.

11.1.2.3. Clase MyOnItemSelectedListener

Se encarga de implementar el funcionamiento de elección un RadioButton.

- La función onItemSelected se encarga dar atributo al RadioButton cuando se selecciona.
- La función onNothingSelected de definir la acción cuando ningún RadioButton ha sido seleccionado.

11.1.3. Diagrama de clases entrenamiento

Son las clases que se implementaran para realizar los entrenamientos de inteligencias.

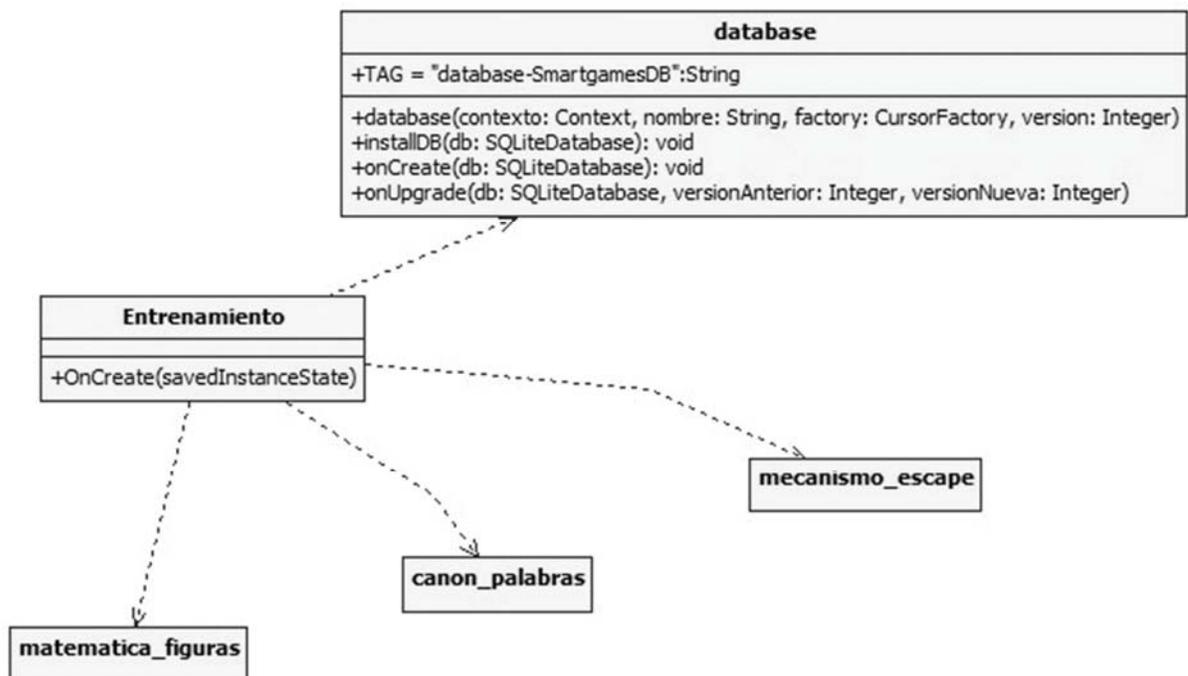


Figura 11.3 Diagrama de clase entrenamiento.

Esta clase permite acceder a los juegos en modo entrenamiento, los cuales no tienen fases ni niveles.

11.1.4. Diagrama de clase de juegos

Son las clases que se implementaran para jugar lo juegos disponibles.

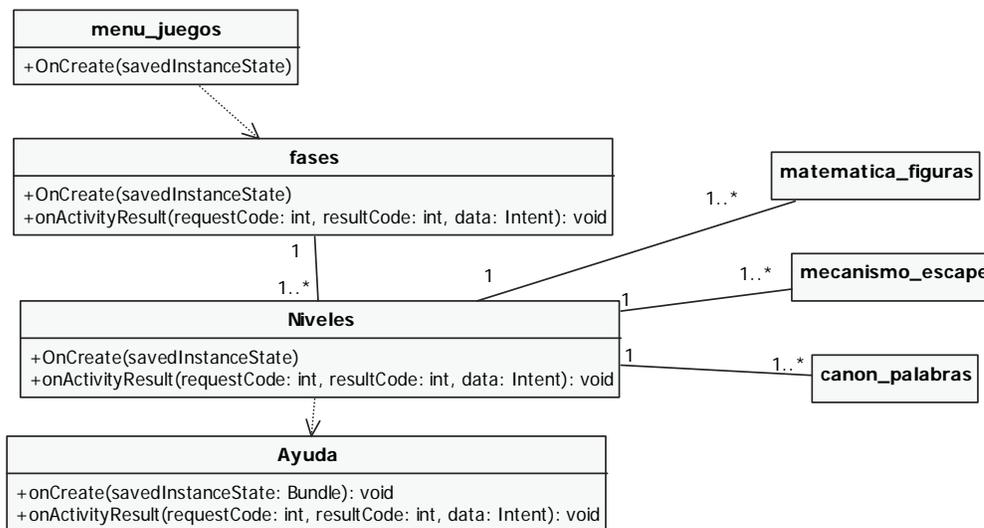


Figura 11.4 Diagrama de clase menú juegos.

11.1.4.1. Clase menú juegos

La clase menú juego contiene el acceso a los diferentes juegos, luego de elegir el juego, se debe seleccionar la fase; luego seleccionar el nivel para ver en detalle la secuencialidad de las acciones ver el diagrama de secuencia seleccionar archivo.

- El método onCreate se encarga de realizar acciones cuando se crea una nueva actividad.
- El método onActivityResult se encarga realizar acciones cuando la actividad llamada por la actividad actual termina y entrega un resultado.

11.1.4.2. Clase fases

Se encarga de rescatar la elección de fase de usuario para luego pasar a la fase de niveles.

11.1.4.3. Clase niveles

Se encarga de rescatar la elección de nivel de usuario y junto con el parámetro de fase, será enviado al juego para allí ajustar la dificultad del juego.

11.1.4.4. Clase ayuda

Si la opción de ayuda esta activada se encarga de mostrar la ayuda de cada juego.

11.1.5. Diagrama de clases de puntajes

Son las clases que se implementaran para que el usuario pueda revisar sus mejores puntajes.

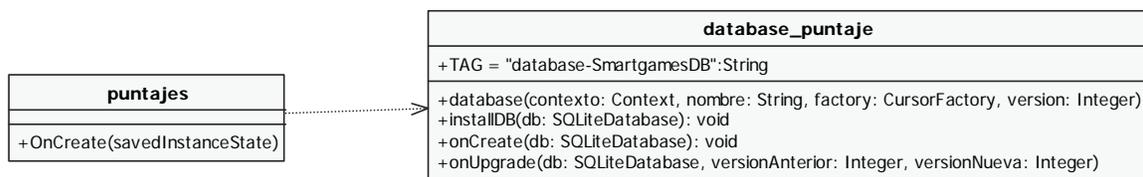


Figura 11.5 Diagrama de clase puntaje

11.1.5.1. Clase puntajes

La clase puntaje se encarga de crear un objeto de tipo database_puntaje y con el extraer los puntajes del usuario, para más información revisar el diagrama de secuencia.

11.1.6. Diagrama de clase matemáticas de figuras

Este diagrama indica como se implementaran las clases del juego matemáticas de figuras.

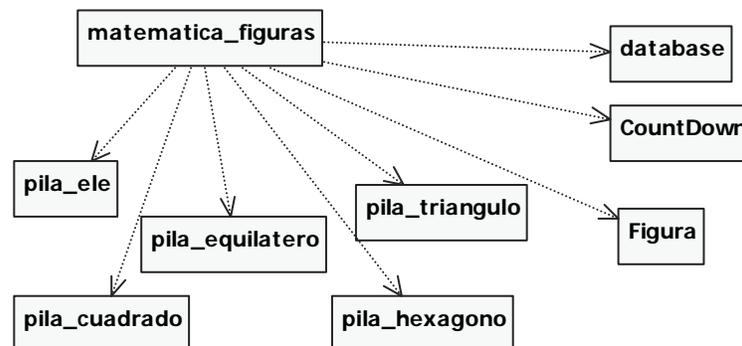


Figura 11.6 Diagrama de clases Matemática de las figuras.

11.1.6.1. Atributos de la clase `matematica_figuras`

<code>matematica_figuras</code>
<code>+contexto: Context</code>
<code>+mScene: Scene</code>
<code>+mBitmapTextureAtlas: BitmapTextureAtlas</code>
<code>+MenuAtlas: BitmapTextureAtlas</code>
<code>+fuente: Font</code>
<code>+background: TextureRegion</code>
<code>+puntaje: Text</code>
<code>+tiempo_limite: Text</code>
<code>+MenuScene: MenuScene</code>
<code>+WinScene: MenuScene</code>
<code>+LoseScene: MenuScene</code>
<code>+mMouseJointActive: MouseJoint</code>
<code>+splash: Scene</code>
<code>+splash_TR: TextureRegion</code>
<code>+figura_contenedora: TextureRegion</code>
<code>+asigna_memoria_figuras: crea_figuras</code>
<code>+lista_figuras: LinkedList</code>
<code>+figuras_agregadas: LinkedList</code>
<code>+musica_fondo: Music</code>
<code>+victory: Sound</code>
<code>+fail: Sound</code>
<code>+puntaje: Integer</code>
<code>+mPhysicsWorld: PhysicsWorld</code>
<code>+mGravityX: float</code>
<code>+mGravityY: float</code>
<code>+CATEGORY_FIGURE: short</code>
<code>+CATEGORY_SCENERY: short</code>
<code>+CATEGORY_TARGET: short</code>
<code>+CATEGORY_SELECTED: short</code>
<code>+CATEGORY_P_SELECTED: short</code>
<code>+CATEGORY_BSHAPE: short</code>
<code>+CATEGORY_POLYGON: short</code>
<code>+MASK_SELECTED: short</code>
<code>+MASK_P_SELECTED: short</code>
<code>+MASK_POLYGON</code>
<code>+MASK_BSHAPE: short</code>
<code>+CAMERA_WIDTH: int</code>
<code>+CAMERA_HEIGHT: int</code>
<code>+contador: int</code>
<code>+contacto: int</code>
<code>+fixm: FixtureDef</code>
<code>+aplicado: boolean[]</code>
<code>+eliminar: boolean</code>
<code>+genera_cuadrados: pila_cuadrado</code>
<code>+genera_triangulos: pila_triangu</code>
<code>+genera_equilateros: pila_equilateros</code>
<code>+genera_eles: pila_ele</code>
<code>+genera_hexagonos: pila_hexagonos</code>
<code>+onpause: boolean</code>
<code>+detect: IUpdateHandler</code>
<code>+countTimer: Countdown</code>

Figura 11.7 Atributos Matemáticas de las figuras.

La clase matemática de figuras se encarga de implementar el juego matemáticas de las figuras y posee los siguientes atributos:

- La clase EngineOptions entrega estados a la clase Engine la cual es el motor interno del juego, se encarga de ir dibujando en pantalla y actualizando objetos en la escena.
- La clase “scenes” es el contenedor para todos los objetos que se van a dibujar en la escena, puede tener un tamaño.
- La Clase camera indica el área de enfoque sobre “Scene”, el enfoque sobre “Scene” puede ser movido (dependiendo del tamaño de la “Scene”) y hacer zoom (por ejemplo con la clase SmoothCamera o ZoomCamera).
- SurfaceScrollDetector se usa para realizar una acción cuando se arrastra algún elemento en pantalla.
- Para explicar BitmapTextureAtlas y TextureRegion se necesita saber sobre Texture, que es una imagen guardada en memoria.
- Una TextureRegion define un rectángulo en una imagen. Las TextureRegion se usan por “Sprites” los cuales son los que se agregan a “scene” y representan un elemento del juego.
- En el caso de BitmapTextureAtlas la misma es una imagen que es cargada desde el disco hacia el chip gráfico. Hay que tener en cuenta, cuando se instancie, que en OpenGL el valor del ancho y alto deben ser una potencia de dos.
- Una Font es la representación de una fuente de letra, se debe agregar a algún BitmapTextureAtlas para cargarla y para utilizarla a algún Text.
- La clase ChangeableText(Gles1 en Gles2 es llamada Text) como su nombre lo indica es un tipo de texto que es agregado a “scene” y puede ser cambiado en cualquier momento, un objeto de este tipo debe tener alguna fuente.
- Las clases Music y Sound son usadas para utilizar la reproducción de sonidos.
- Context: contiene el contexto de la aplicación.

- FixtureDef: se encargan de definir elasticidad, densidad y fricción de las figuras, además permite definir el contacto.
- Las variables category es la clasificación que posee cada figura e indica con que esta colisionando son usadas por el fixtureDef.
- Las variables mask indica con que categorías de figuras puede colisionar la figura con esa mask; -1 indica que colisiona con todo 0 indica que es atravesado por todo, además una máscara puede ser definida como una suma de categorías indicando así con que categorías puede colisionar.
- Aplicado indica las posiciones en la barra de figuras que están ocupadas o libres.
- MouseJoint: es un enlace creado entre el dedo y la figura seleccionada el cual la permite mover.
- MenuScene: son usados para crear menus al pausar, ganar o perder; contienen las imágenes necesarias para crear los menus y se encarga de mostrarlos.
- IUpdateHandler: es una hebra que se ejecuta continuamente y gracias a ella se puede revisar el estado de los elementos de la “scene”, es la que permite eliminar o modificar los elementos de la “scene”.
- Body es una clase que permiten implementar gravedad y colisiones a una imagen.
- PhysicWorld: permite implementar la gravedad y colisiones al “engine” y luego crear “body” y fixtureDef para las demás figuras.

11.1.6.2. Métodos de la clase `matematica_figuras`

<code>matematica_figuras</code>
<pre>+onCreateEngineOptions(): EngineOptions +onCreateResources(): void +onCreateScene(): Scene +onPauseGame(): void +creaMenuPausa(): void +creaMenuVictoria(): void +creaMenuDerrota(): void +onMenuItemClicked(pMenuScene: MenuScene, pMenuItem: IMenuItem, pMenuItemLocalX: float, pMenuItemLocalY: float): boolean +onResumeGame(): void +onKeyDown(pKeyCode: Integer, pEvent: KeyEvent): boolean +createMouseJoint(pFace: Sprite, pTouchAreaLocalX: float, pTouchAreaLocalY: float): MouseJoint +addFace(pX: float, pY: float, fig: figura): Sprite +onSceneTouchEvent(pScene: Scene, pSceneTouchEvent: TouchEvent): boolean +gameToast(contexto: context): void +onAccelerationAccuracyChanged(pAccelerationData: AccelerationData): void +onAccelerationChanged(pAccelerationData: AccelerationData): void +onAreaTouched(pSceneTouchEvent: TouchEvent, pTouchArea: ITouchArea, pTouchAreaLocalX: float, pTouchAreaLocalY: float): boolean +beginContact(contact: Contact): void +endContact(contact: Contact): void +preSolve(contact: Contact, oldManifold: Manifold): void +postSolve(contact: Contact, impulse: ContactImpulse): void +cambia_estado(a_cambiar: Sprite): void +a_eliminar(a_cambiar: Sprite): void +agregar(): void +busca_body(face: Sprite): Sprite +numero_aleatorio(): void +figura_agregada(): void +cambia_figura(): void +cambia_fixtures(b: Body, pos: int): void +accionFinalizar(): void +guardar_puntaje(): void +guardar_puntaje_entrenamiento(): void +guarda_nvl(): void +guarda_fases(): void +createExplosion(posX: float, posY: float, target: IEntity, activity: SimpleBaseGameActivity): void +guarda_puntaje_nvl(): void +interior_figura(a_comprobar: Body): figura +restart(flag: int): void</pre>

Figura 11.8 Métodos Matemática de las figuras.

La clase `matematica_figuras` se encarga de implementar el juego matemática de las figuras, posee los siguientes métodos:

- La función `OnCreateEngineOptions` se ejecuta al comienzo, se encarga de configurar aspectos de la cámara, orientación, resolución de pantalla, etc.
- La función `OnCreateResources` se encarga de cargar imágenes, música y otros elementos que necesite el juego.
- La función `OnCreateScene` se encarga de ejecutar el juego en sí, cargado todos elementos necesarios en pantalla.
- La función `OnLoadComplete` se ejecuta cuando el juego ya ha cargado.

- `onKeyDown` se encarga de ejecutar acciones al presionar ciertos botones.
- `onSceneTouchEvent` se encarga de ejecutar acciones cuando se pulsa en algún lugar del “scene”.
- La función `gameToast` se encarga de enviar mensajes por pantalla al usuario.
- Las funciones `onAccelerationChanged` y `onAccelerationAccuracyChange`, permiten establecer acciones al realizar algún movimiento que afecte al acelerómetro.
- La función `agregar` se encarga de agregar figuras.
- Las funciones `guardar` se encargan de guardar la respectiva información en la base de datos local.
- La función `createExplosion` se encarga de crear la animación utilizada al eliminar alguna figura de pantalla.
- Las funciones `crearMenu` se encargan de los elementos visuales utilizados en los menus de pausa, victoria y derrota.
- `onMenuClicked`: se encarga de realizar una acción al presionar elemento del menú.
- `onPauseGame` y `onResumeGame` se encarga de realizar acciones al pausar o reanudar el juego.
- `createMouseJoint`: se encarga de crear los `mousejoint` los cuales permiten mover las figuras con el dedo.
- `onAreaTouched`: se encarga ejecutar alguna acción al pulsar sobre algún elemento en pantalla.
- `beginContact`, `endContact`, `postSolve` y `preSolve` se encargan de ejecutar acciones cuando dos figuras hacen contacto.
- `Cambia_estado`: se encarga de cambiar el estado de una figura para indicar que la figura ha sido agregada a la figura a rellenar.

- **A_eliminar:** se encarga de cambiar el estado de una figura indicado que ha sido eliminada.
- **Busca_body:** en caso de que no se encuentre el cuerpo de una figura esta función se encarga de buscarlo en la lista.
- **Numero_aleatorio:** se encarga de modificar los resultados entregados por el generador de números aleatorios utilizado para generar los números de las figuras.
- **Figura_agregada:** se encarga de revisar si una figura ya ha sido aplicada en el juego con el fin de comprobar si existe una posición para agregar una nueva figura.
- **Cambia_figura:** se encarga de modificar los resultados entregados por el generador de números aleatorios utilizado para generar el número que indica el tipo de figura a agregar.
- **Cambia_fixtures:** permite cambiar las categorías de las figuras; para una figura existen 3 categorías:
 - Selected que permite atravesar la línea para agregar las figuras
 - selected_p que permite atravesar solo las líneas de la figura a rellenar
 - figure que colisiona con todo
- **accionFinalizar:** al terminar el juego se encarga de decidir entre victoria y derrota.
- **Interior_figura:** comprueba si una figura pequeña esta al interior de la figura a rellenar.
- **Restart:** para seguir al siguiente nivel o para reintentar un nivel se finaliza la actividad y se envía a la actividad nivel los datos necesarios para abrir una nueva actividad flag==1 indica el paso a un siguiente nivel y flag==0 indica el reinicio de un nivel.

Los atributos y funciones anteriormente explicadas se usan también en otros juegos, por lo que a la mayoría de ellos se aplica la misma explicación.

11.1.6.3. Clase pila_cuadrado, pila_triangulo, pila_ele, pila_equilatero, pila_hexagono.

pila_equilatero	pila_ele	pila_triangulo
+tr: TextureRegion +posicion: int[4] +vertex: VertexBufferObjectManager +aplicado: boolean[4]	+tr: TextureRegion +posicion: int[4] +vertex: VertexBufferObjectManager +aplicado: boolean[4]	+tr: TextureRegion +posicion: int[4] +vertex: VertexBufferObjectManager +aplicado: boolean[4]
+pila_equilatero() +onAllocatePoolItem() +cambiar_posicion() +onHandleRecycleItem()	+pila_ele() +onAllocatePoolItem() +cambiar_posicion() +onHandleRecycleItem()	+pila_triangulo() +onAllocatePoolItem() +cambiar_posicion() +onHandleRecycleItem()

pila_cuadrado	pila_hexagono
+tr: TextureRegion +posicion: int[4] +vertex: VertexBufferObjectManager +aplicado: boolean[4]	+tr: TextureRegion +posicion: int[4] +vertex: VertexBufferObjectManager +aplicado: boolean[4]
+pila_cuadrado() +onAllocatePoolItem() +cambiar_posicion() +onHandleRecycleItem()	+pila_hexagono() +onAllocatePoolItem() +cambiar_posicion() +onHandleRecycleItem()

Figura 11.9 Clases pilas para figuras.

Similar a la clase crea_numeros, esta clase se encarga de crear las figuras utilizando un TextureRegion (llamado mTextureRegion) y con el crear “sprites”. Como en el juego se utilizaran muchas figuras, se debe hacer un eficiente uso de memoria, para ello se utiliza esta clase, junto con las LinkedList (lista_figuras, figuras_agregadas) declaradas como variables globales en la clase matematica_figuras. Estas clases se separan dependiendo de la figura que se quiere crear ya que la función onHandleRecycleItem se encarga de que luego de salir un “sprite” del “scene” este pueda ser reutilizado y no se produzcan confusiones sobre el “sprite” a utilizar. Las funciones realizan las mismas acciones que las de la clase crear_numero.

11.1.6.4. Clase figura

Figura
+shape: Sprite +eliminado: boolean +utilizada: boolean +ispresed: boolean +on: false +posicion: int +tipo_figura: int +numero: Text +signo: boolean

Figura 11.10 Clase figura

Se encarga de guardar los estados que puede tener cada figura tales como si una figura es eliminada o si es aplicada, se implementa una lista de esta clase.

11.1.6.5. Clase Countdown

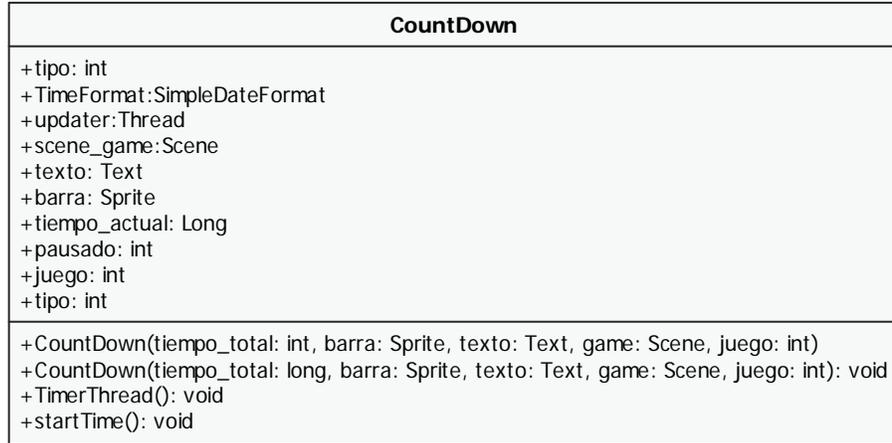


Figura 11.11 Diagrama de clase Countdown.

Se encarga de implementar el tiempo a los juegos mecanismo de escape y matemática de figura, posee 2 constructores pues uno es utilizado para los juego normales donde la cantidad de tiempo total no cambia y en los juegos de entrenamiento donde el tiempo se traspasa entre niveles y fases; donde el tiempo comienza desde tiempo_total y disminuye hasta 0.

11.1.7. Diagrama de clase cañón de palabras

Este diagrama muestra cómo se implementan las clases del juego cañón de palabras.

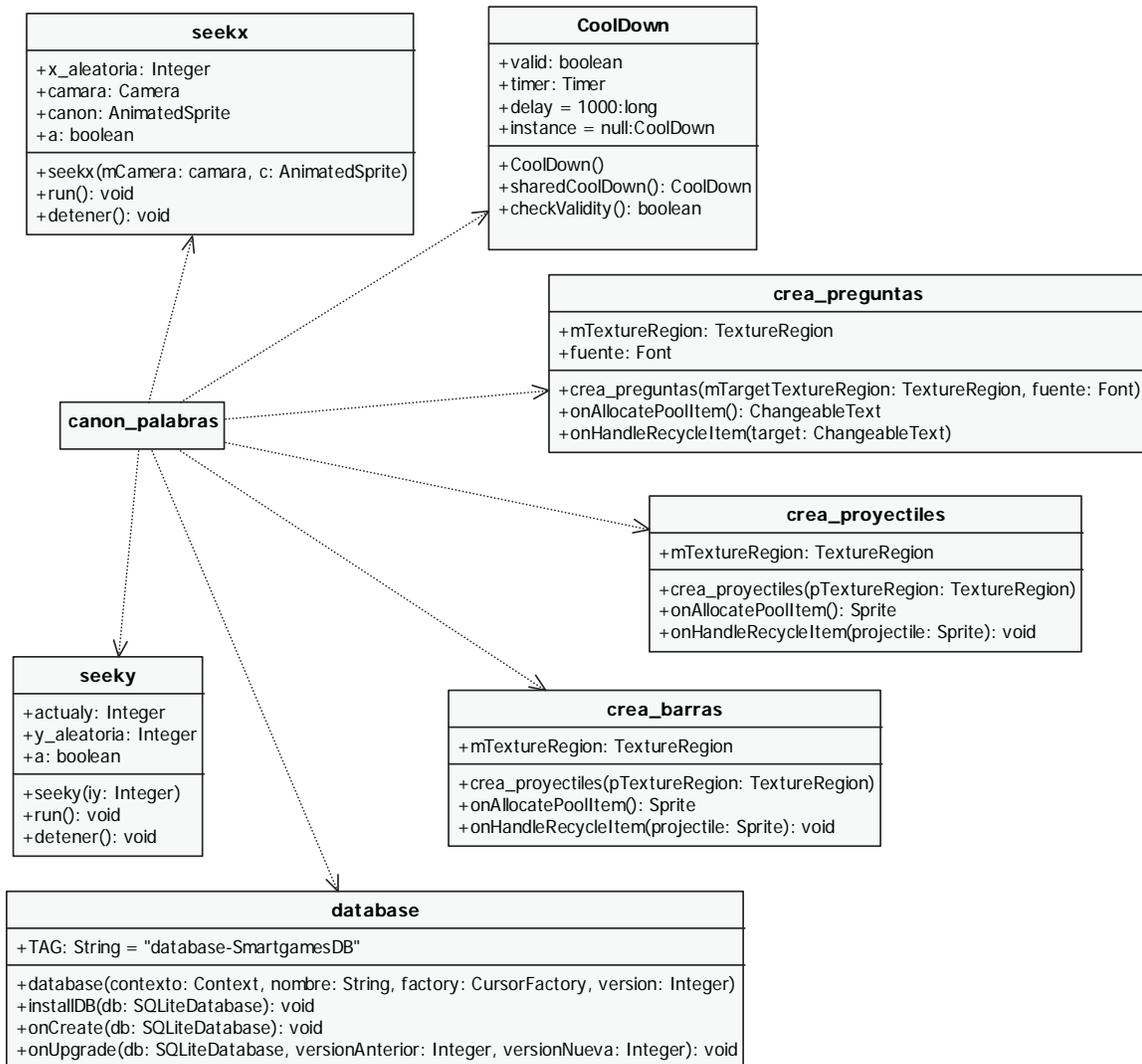


Figura 11.12 Diagrama de clase cañón de palabras.

11.1.7.1. Clase cañón de palabras

11.1.7.1.1. Atributos clase cañón de palabras

canon_palabras
+mEngine: Engine
+mScrollDetector: SurfaceScrollDetector
+randomy: seeky
+randomx: seekx
+contexto: Context
+scene: Scene
+mBitmapTextureAtlas: BitmapTextureAtlas
+back: BitmapTextureAtlas
+mfuente: BitmapTextureAtlas
+mfuente_puntaje: BitmapTextureAtlas
+fuente1: Font
+fuente2: Font
+player: TextureRegion
+background: TextureRegion
+pewpew: TextureRegion
+fondop: TextureRegion
+pause: TextureRegion
+barra_p: TextureRegion
+barra_comp: TextureRegion
+word: ChangeableText
+puntaje: ChangeableText
+pausar: CameraScene
+area_cortada: BitmapTextureAtlas
+mCanon: TiledTextureRegion
+mCamera: Camera
+proyectiles: LinkedList
+proyectiles_agregados: LinkedList
+lista_palabras: LinkedList
+palabras_agregadas: LinkedList
+fondo_palabras: LinkedList
+fondos_agregados: LinkedList
+asigna_memoria_preguntas: crea_preguntas
+asigna_memoria_balas: crea_proyectiles
+asigan_memoria_balas: crea_barras
+margen = 80: Integer
+respuestas: Vector
+usados: Integer[100]
+cant_resp = 0: Integer
+points = 0: Integer
+mWinTextureRegion: TextureRegion
+mFailTextureRegion: TextureRegion
+mResultScene: CameraScene
+pausado = 0: Integer
+disparo: Sound
+musica_fondo: Music
+detect: IUpdateHandler
+velocidad_barra: integer
+retardo_barra: float
+splash: Scene
+splash_BTA: BitmapTextureAtlas
+splash_TR: TextureRegion
+correct: Sound
+incorrect: Sound
+fanfarre: Sound
+lose: Sound
+cantidad_preguntas: Integer
+cantidad_preg: ChangeableText
+respondidas: ChangeableText
+cantidad_preg: ChangeableText

Figura 11.13 Atributos clase cañón de palabras.

Se destacan atributos como margen (que indica el margen de caída de las preguntas), cant_resp (indica la cantidad de respuestas contestadas correctamente), points (indica el puntaje obtenido), cantidad de preguntas (indica la cantidad de preguntas que se debe responder para pasar al siguiente nivel).

Los atributos velocidad_barra y retardo_barra son usados para cambiar la dificultad del juego, mientras más rápida es la velocidad de barra, más fácil es perder y mientras menos retardo de barra, mayor es la cantidad de preguntas que aparecen en pantalla.

11.1.7.2. Funciones clase cañón de palabras

canon_palabras
<pre> +onLoadEngine(): Engine +onLoadResources(): void +onLoadScene(): Scene +shootProyectile(px: float, py: float): void +pauseGame(): void +unPauseGame(): void +pauseMusic(): void +resumeMusic(): void +onKeyDown(pKeyCode: Integer, pEvent: KeyEvent): boolean +createSpriteSpawnTimeHandler(): void +addTarget(): void +verificacion(): boolean +preguntas(numero: Integer, resp1: String, resp2: String, resp3: String, resp4: String, pregunta: String): void +fail(): void +win(): void +removeSprite(): void +removeText(): void +onLoadComplete(): void +onSceneTouchEvent(arg0: Scene, arg1: TouchEvent): boolean +gameToast(contexto: context): void </pre>

Figura 11.14 funciones de clase cañón de palabras.

- La función shootProyectile, es la que al presionar algún área en pantalla permite disparar.
- Las funciones de fail y win se encargan de ver las acciones bajo los casos de derrota o victoria en el juego.
- La función removeSprite se encarga de quitar elementos de “scene” como los “sprites” innecesarios, mientras la función removeText se encarga de quitar objetos de la clase ChangeableText.

- La función createSpriteSpawnTimeHandler permite que cada cierto tiempo (dado por retardo_barra), se creen y agreguen a la “scene” nuevas preguntas, llamando a la función addTarget, la cual se encarga de crear las preguntas.
- La función add se encarga de agregar figuras.
- La función verificación se encarga de asegurarse que no se repitan las preguntas.
- La función preguntas se encarga de buscar las preguntas en la base de datos.
- pauseGame, pauseMusic, UnPauseGame, resumeMusic; se encargan de pausar o resumir el juego y la música del juego.

11.1.7.3. Clases usadas por la clase cañón de palabras

- La clase CoolDown se encarga de disminuir la cantidad de disparos que se pueden dar en un lapso de tiempo determinado.
- Las clases seeky y seekx indica la posición a la bala se va a disparar, la posición destino es una posición generada pseudo aleatoriamente dando el objeto de seekx la posición horizontal y el objeto seeky la posición vertical.
- La función crea_preguntas, funciona de manera similar a la función crea_numeros.
- Tanto las funciones crea_proyectiles, como la función crea_barras trabajan similar a la función crea_figuras.

11.1.8. Diagrama de clase mecanismo de escape

Este diagrama muestra cómo se implementa el juego mecanismo de escape.

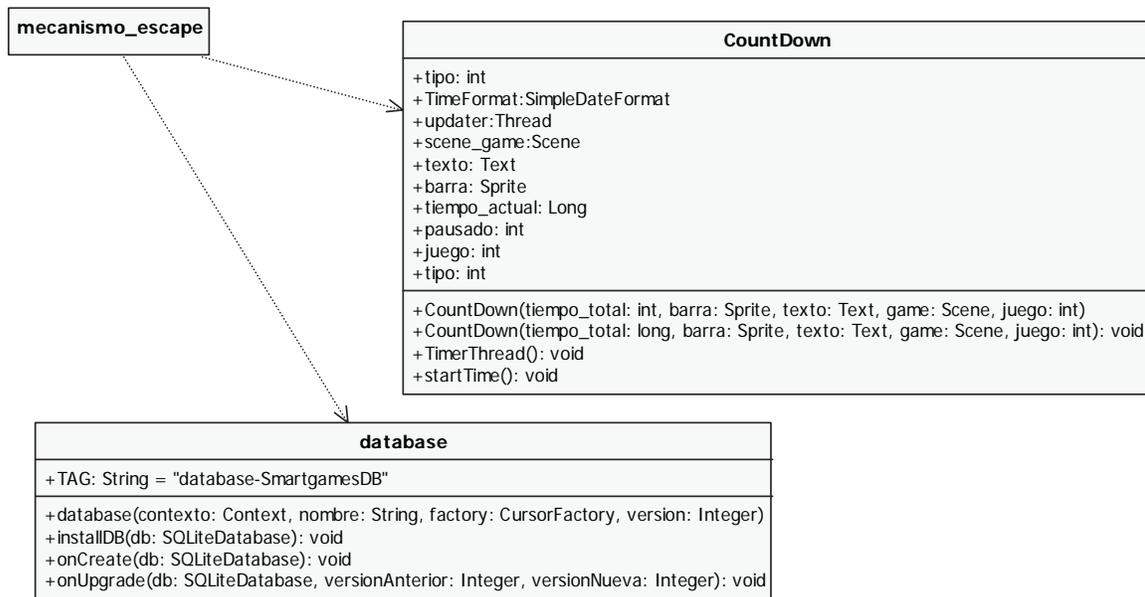


Figura 11.15 Diagrama de clase mecanismo de escape

11.1.8.1. Clase mecanismo de escape

Esta clase posee atributos y funciones similares a la clase matemática de figuras, pero la funcionalidad al interior es lo que hace distinta a cada clase, está no requiere implementación de LinkedList, ya que no posee muchos elementos como las demás.

Como en este juego se hace uso del acelerómetro, se agregan unas variables que indican la gravedad (`gravedadx` y `gravedady`), además se crea la función `onAccelerometerChanged` que permite la gravedad de un elemento al mover el “Smartphone”.

11.1.8.1.1.

Clase mecanismo de escape atributos

mecanismo_escape	mecanismo_escape
<pre> +CAMERA_WIDTH: private static final int +CAMERA_HEIGHT: private static final int +mCamera: Camera +mGameRunning: boolean +mBitmapTextureAtlas: BitmapTextureAtlas + esfera_negra: ITextureRegion +esfera_blanca: ITextureRegion +engranaje_rojo: ITextureRegion +engranaje_azul: ITextureRegion +engranaje verde: ITextureRegion +fondo_rojo: ITextureRegion +fondo_azul: ITextureRegion +fondo_verde: ITextureRegion +fondo_esfera: ITextureRegion +fondo_negro_esfera: ITextureRegion +fondo_engranaje: ITextureRegion +menu_pausa: ITextureRegion +menu_win: ITextureRegion +menulose: ITextureRegion +reintentar: ITextureRegion +fases: ITextureRegion +resumir: ITextureRegion +principal: ITextureRegion +siguiente: ITextureRegion +fondo_tiempo: ITextureRegion +fondo: ITextureRegion +mMenuResetTextureRegion: ITextureRegion +mMenuQuitTextureRegion: ITextureRegion +mPausedTextureRegion: ITextureRegion +body_engranaje_rojo: Body +body_engranaje_azul: Body +body_engranaje_verde: Body +body_esfera_blanca: Body +body_esfera_negra: Body +body_fondo_rojo: Body +body_fondo_azul: Body +body_fondo_verde: Body +body_fondo_blanco: Body +WinScene: MenuScene +LoseScene: MenuScene +sprite_engranaje_rojo: Sprite +sprite_engranaje_azul: Sprite +sprite_engranaje_verde: Sprite +sprite_esfera_blanca: Sprite +sprite_esfera_negra: Sprite +sprite_fondo_rojo: Sprite +sprite_fondo_azul: Sprite +sprite_fondo_verde: Sprite +sprite_fondo_blanco: Sprite +sprite_fondo_tiempo: Sprite +laberinto1[]: int +posiciones_blanca_x: int +posiciones_blanca_y: int +posiciones_negra_x: int +posiciones_negra_y: int +posiciones_rojo_x: int +posiciones_rojo_y: int </pre>	<pre> +posiciones_verde_x: int +posiciones_verde_y: int +contador_atlas: int +nvl: int +numero_laberinto: int +dificultad: int +intentos: int +numero_engranajes: int +puntaje: int +figura: insertada: int +insertado: boolean +perdiste: boolean +choque: boolean +texto_puntaje: Text +texto_intentos: Text +texto_tiempo: Text +color_barras: Color +musica_fondo: Music +eliminarS: Sound +victory: Sound +fail: Sound +mPhysicWorld: PhysicWorld +mGravityX: float +mGravityY: float +mMenuTexture: BitmapTextureAtlas +mHandler: IUpdateHandler +contexto: Context +contador_atlas2: int +fase: int +niveles_sup: int +minimo_figuras: int +minutos_restantes: int +fuentes: Font +count_timer: prototipo.juego.CountDown +MENU_RESUMIR: int +MENU_FASE: int +MENU_PRINCIPAL: int +MENU_SIGUIENTE: int +MENU_REINTENTAR: int </pre>

Figura 11.16 Atributos de la clase mecanismo de escape.

11.1.8.1.2.

Clase mecanismo de escape métodos

mecanismo_escape
+onCreateEngineOptions(): EngineOptions() +onCreateResources(): void() +onCreateScene(): void() +gameToast(String): void() +creaMenuPausa(): void() +creaMenuVictoria(): void() +creaMenuDerrota(): void() +onAreaTouched(TouchEvent, ITouchArea, float, float): boolean() +onAccelerationAccurazyChanged(AccelerationData): void() +onAccelerationChanged(final AccelerationData pAccelerationData): void() +onResumeGame(): void() +onPauseGame(): void() +onKeyDown(int, KeyEvent): boolean() +onMenuItemClicked(MenuScene, IMenuItem, float, float): boolean() +restart(int): void() +onDestroy: void() +onSceneTouchEvent(Scene, TouchEvent): boolean() +cargar_laberinto_bd(): void() +addEsfera(): void() +addEngranaje(): void() +reiniciar_laberinto(): void() +eliminar_objeto(Body, Sprite): void() +beginContact(Contact): void() +endContact(Contact): void() +guardar_puntaje(): void() +guarda_nvl(): void() +guarda_fase(): void() +preSolve(Contact, Manifold): void() +postSolve(Contact, ContactImpulse): void() +accion_finalizar(): void() +guadar_puntajes_entrenamiento(): void() +guarda_puntaje_nvl(): void()

Figura 11.17 Metodos de la clase mecanismo_escape.

- Los métodos addEsfera y addEngranaje se encargan de crear los “sprites” y “bodys” para los engranajes y para las esferas.
- onDestroy es la acción a realizar cuando se finaliza o cierra el juego.
- Eliminar_objeto: se encarga de eliminar engranajes y esferas cuando se termina su uso.
- Carga_laberinto_bd: consulta a la base de datos sobre los muros a utilizar en el laberinto, y los crea para que puedan ser visualizados en pantalla.

11.2. Diagramas de secuencia

Los siguientes diagramas de secuencia se basan en las principales funcionalidades del sistema.

11.2.1. Diagrama de secuencia realizar test.

El siguiente diagrama muestra la secuencia de pasos para realizar el test.

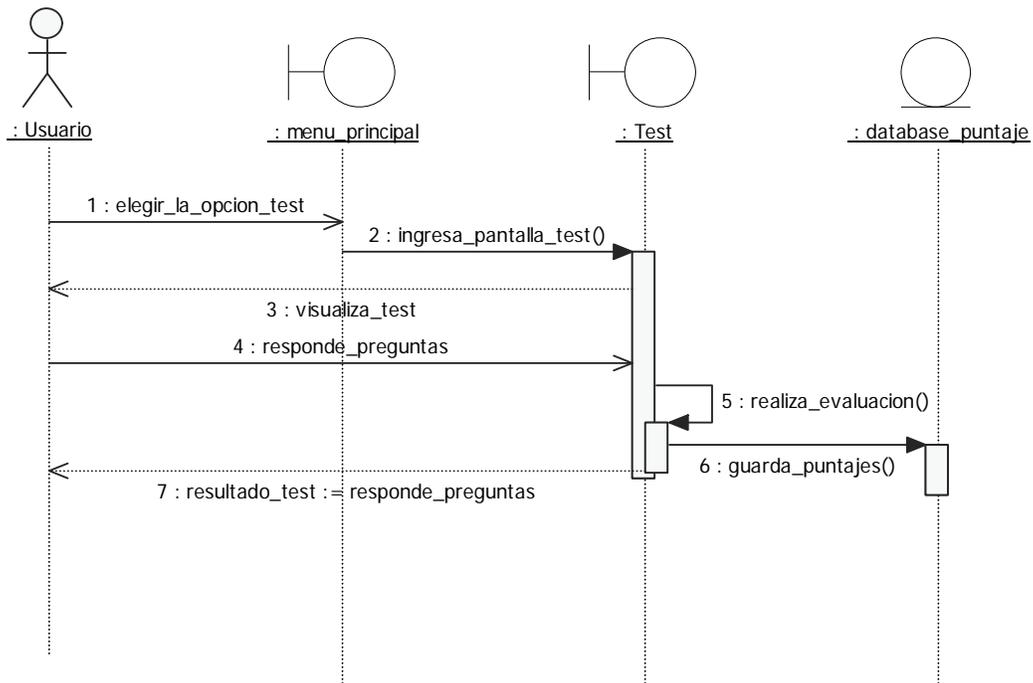


Figura 11.18 Diagrama de secuencia realizar test.

11.2.2. Diagrama de secuencia realizar entrenamiento

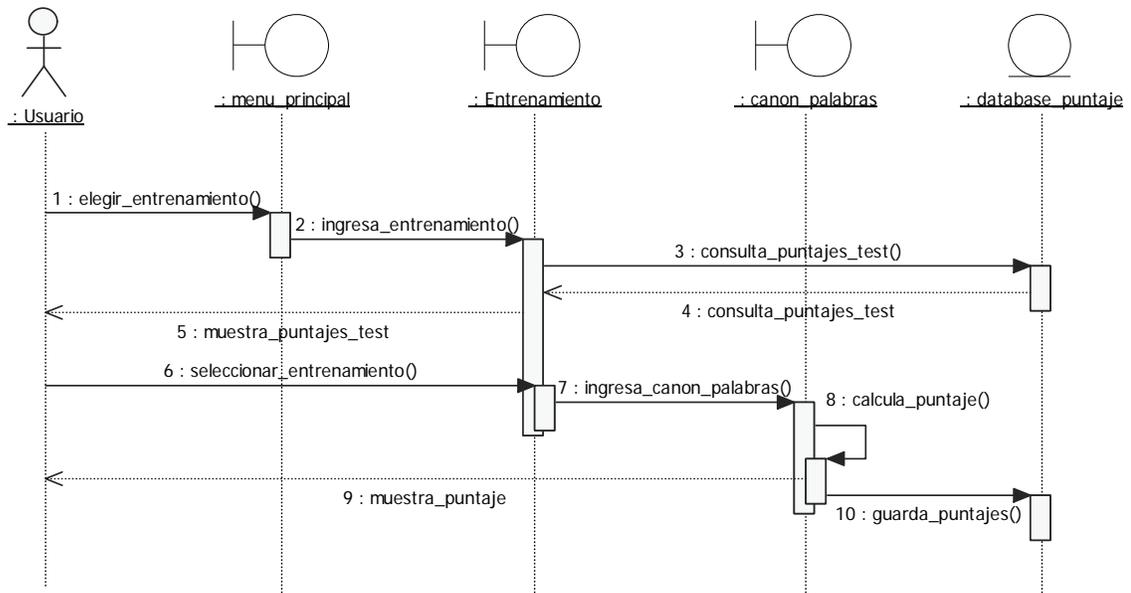


Figura 11.19 Diagrama de secuencia realizar entrenamiento.

Se encarga de mostrar puntajes del test y los puntajes de entrenamientos anteriores.

11.2.3. Diagrama de secuencia selección de opciones

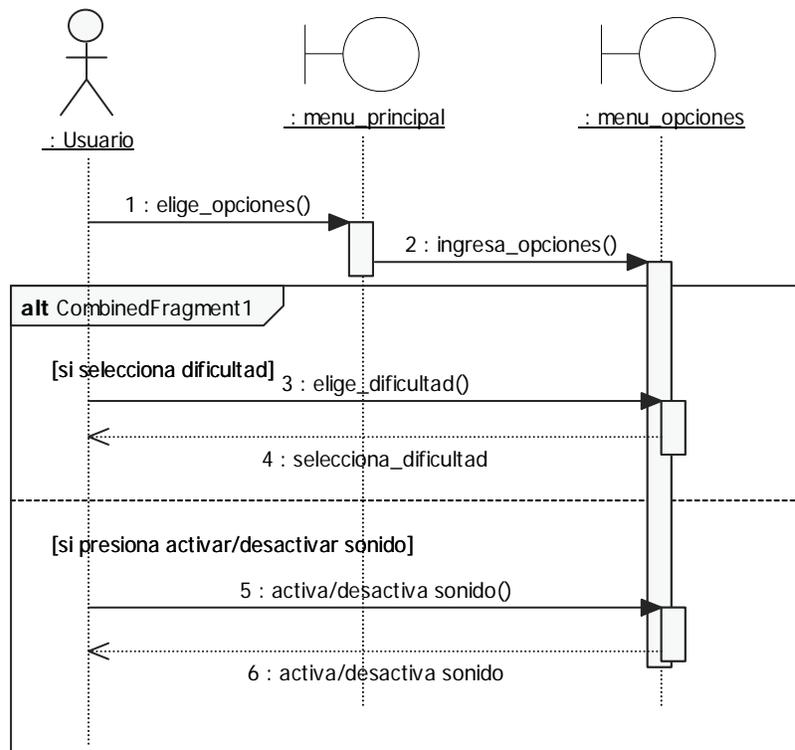


Figura 11.20 Diagrama de secuencia selección de opciones.

11.2.4. Diagrama de secuencia opción puntajes

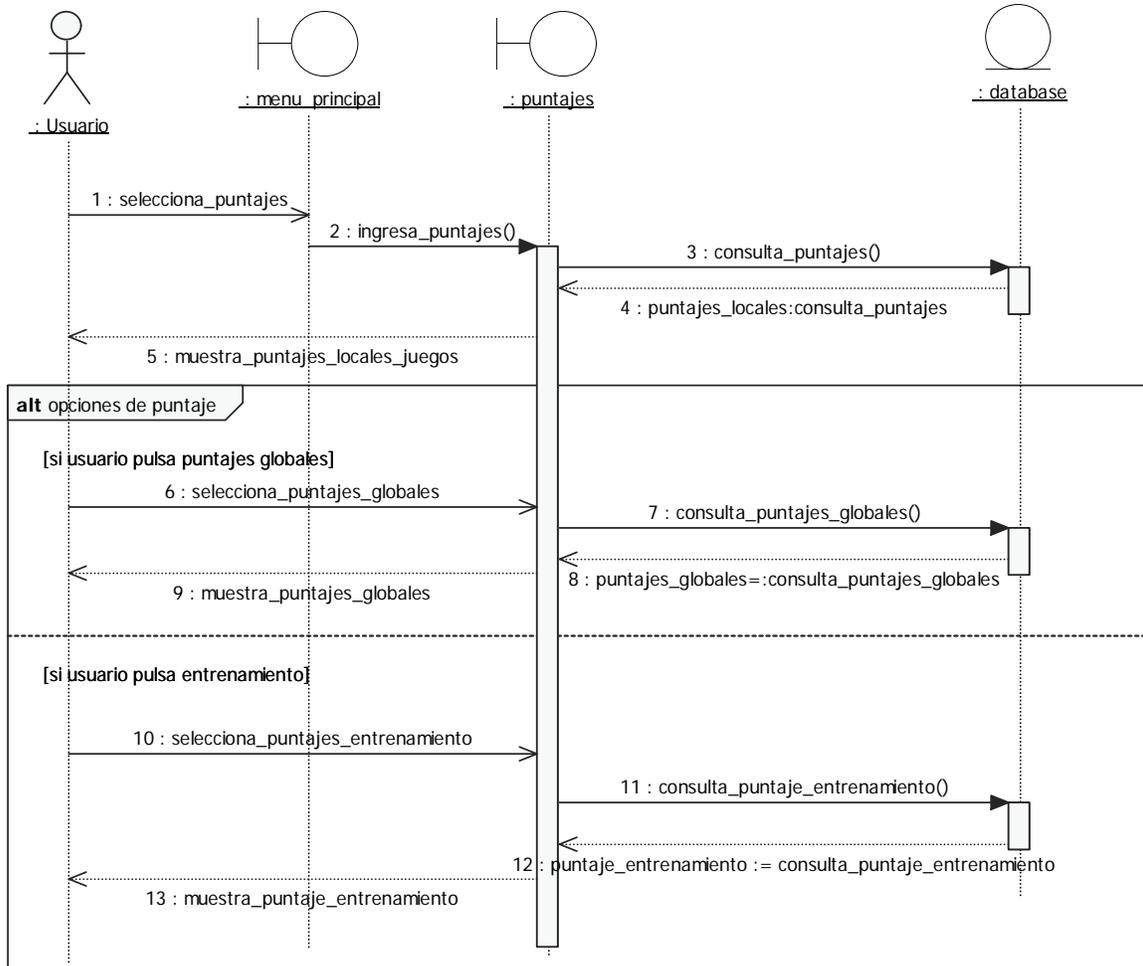


Figura 11.21 Diagrama de secuencia opción puntajes.

11.2.5. Diagrama de secuencia selección de juego

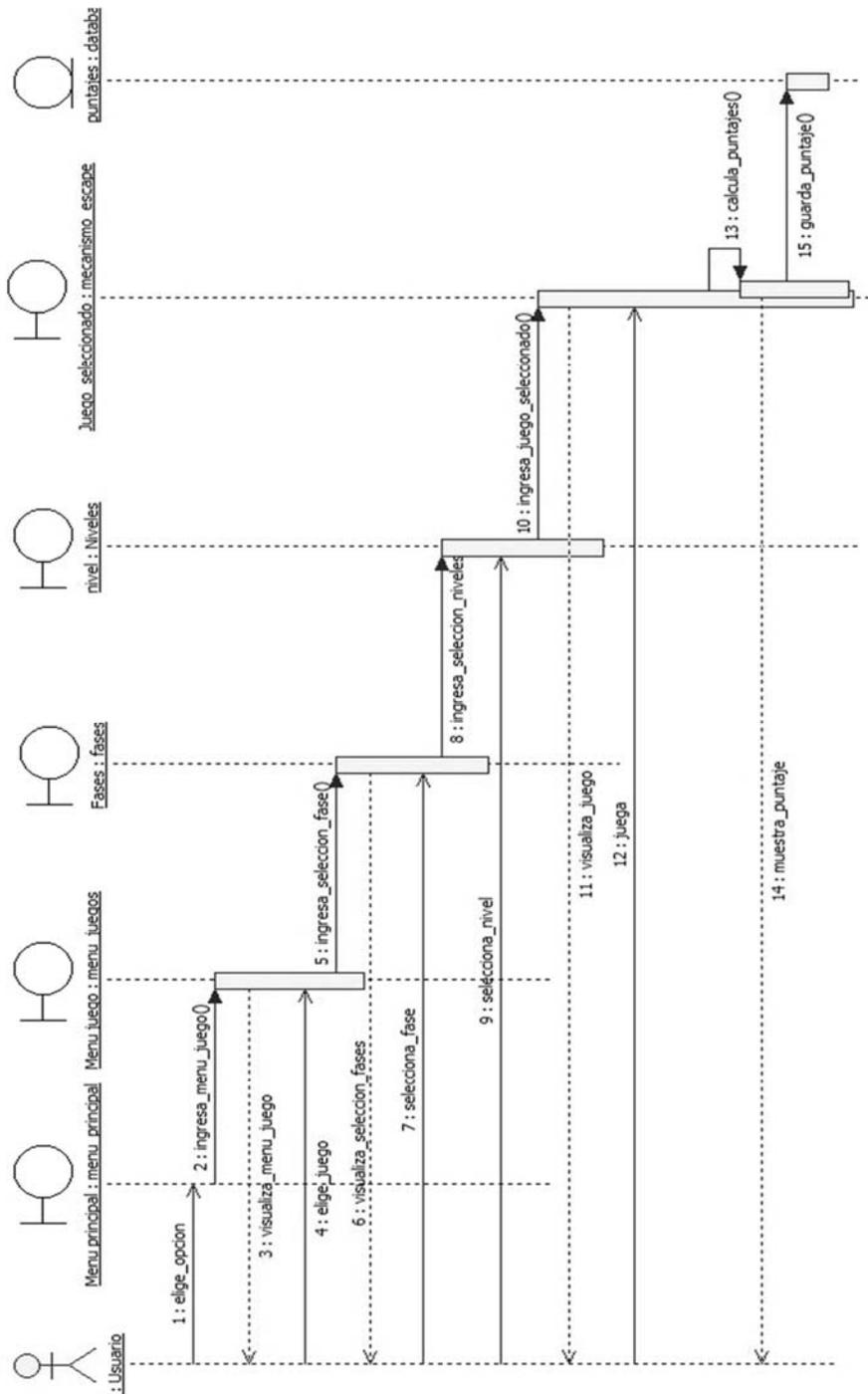


Figura 11.22 Diagrama de secuencia selección de juego.

11.3. Diseño de base de datos

Dentro de la aplicación existe una base de datos SQLite local llamada *database-SmartgamesDB*, en esta se guardan diferentes tipos de información separados en tablas.

A continuación se explicaran las diferentes tablas utilizadas:

- Preguntas: esta tabla contiene las preguntas del juego Cañón de palabras y se conforma por las siguientes columnas:
 - Id (INTEGER): esta columna contiene el id de cada pregunta.
 - Pregunta1 (TEXT): esta columna contiene la pregunta que se mostrara por pantalla en el juego.
 - Resp1, resp2 resp3 y resp4 (TEXT): son las 4 diferentes respuestas para la pregunta señalada arriba.
 - Correcta (TEXT): acá se almacena la respuesta correcta a la pregunta.

- Test: esta tabla contiene las preguntas que aparecerán en el test de inteligencias que contendrá la aplicación y se conforma por las siguientes columnas:
 - Id (INTEGER): esta columna contiene el id de cada pregunta.
 - Tipo (TEXT): acá se almacena el tipo de cada pregunta siendo estos LM (lógico-matemático), LE (lingüística) y ES (espacial)
 - Pregunta1 (TEXT): esta columna contiene la pregunta que se mostrara por pantalla en el juego.
 - Resp1, resp2 resp3 y resp4 (TEXT): son las 4 diferentes respuestas para la pregunta señalada arriba.
 - Correcta (TEXT): acá se almacena la respuesta correcta a la pregunta.

- Puntaje: esta tabla contiene los puntajes obtenidos en los tres diferentes tipos de juegos y conforma por las siguientes columnas:
 - Juego (TEXT): en esta columna se almacena el nombre del juego en particular.
 - Puntaje (INTEGER): como su nombre lo dice, guarda el puntaje obtenido en el juego mencionado en la columna anterior.
 - Dificultad (TEXT): en esta columna se guardara la dificultad en la que esté jugando el usuario.
 - Fase (INTEGER): en esta columna se almacena la fase en la que el usuario obtuvo el puntaje.
 - Nivel (INTEGER): al igual que arriba se guarda el nivel en el que el usuario obtuvo el puntaje.

- Fecha (TEXT): por último, pero no menos importante, se almacena la fecha en la que el usuario obtuvo el puntaje.
- Puntaje_nivel: esta tabla contiene los puntajes obtenidos al superar un nivel en cada juego los puntajes se ordenan en forma de pila donde en la cima se encuentra el ultimo puntaje registrado y en el fondo se encuentra el puntaje más antiguo.
 - Posicion (INTEGER): se almacena un id para cada puntaje.
 - Juego (TEXT): en esta columna se almacena el nombre del juego en particular.
 - Puntaje (INTEGER): como su nombre lo dice, guarda el puntaje obtenido en el juego mencionado en la columna anterior.
 - Dificultad (TEXT): en esta columna se guardara la dificultad en la que esté jugando el usuario.
 - Fase (INTEGER): en esta columna se almacena la fase en la que el usuario obtuvo el puntaje.
 - Nivel (INTEGER): al igual que arriba se guarda el nivel en el que el usuario obtuvo el puntaje.
 - Fecha (TEXT): por último, pero no menos importante, se almacena la fecha en la que el usuario obtuvo el puntaje.
- Puntaje_t: esta tabla contiene los puntajes obtenidos por el usuario al momento de realizar el test de inteligencias múltiples y se conforma de las siguientes columnas:
 - L_matematico (INTEGER): en esta columna se guarda el puntaje obtenido en las preguntas de tipo lógico-matemático.
 - Linguistico (INTEGER): en esta columna se guarda el puntaje obtenido en las preguntas de tipo lingüístico.
 - Espacial (INTEGER): en esta columna se guarda el puntaje obtenido en las preguntas de tipo Espacial.
 - Total (INTEGER): es esta columna se guarda el puntaje total obtenido en todo el test.
 - Fecha (TEXT): acá se guarda la fecha donde el usuario realizo el test.
- Puntaje_e: esta tabla contiene los puntajes que obtiene el usuario al momento de realizar el modo entrenamiento de los juegos. Tiene las siguientes columnas:
 - Juego (TEXT): guarda el nombre del juego en el cual se está realizando el entrenamiento.
 - Dificultad (INTEGER): en esta columna se guarda la dificultad en la que se estaba jugando el entrenamiento.
 - Fecha (INTEGER): acá se guarda la fecha donde el usuario obtuvo el puntaje.

Además de las tablas relacionadas a los juegos y al test, existen otras 3 tablas que contienen información de control o para guardar opciones del sistema, estas tablas son:

- Estado: en esta tabla se guarda el estado de los diferentes juegos para que cuando el usuario cierre la aplicación y vuelva a abrirla los niveles y fases superados se reflejen en la aplicación. Esta tabla contiene las siguientes columnas:
 - Juego (INTEGER): esta columna representa el juego del cual se están guardando los niveles y fases.
 - Fase (INTEGER): acá se guarda la fase que alcanzo el usuario antes de cerrar la aplicación.
 - Nivel (INTEGER) acá se guarda el nivel que alcanzo el usuario antes de cerrar la aplicación.
- Opciones: en esta tabla se guardan las diferentes opciones que trae la aplicación en cuanto a su uso y estas son:
 - Sonido (INTEGER): acá se guarda la opción del sonido de la aplicación, siendo sonido activo (un 1) o sonido desactivado (un 0).
 - Dificultad (INTEGER): en esta columna se guardan las preferencias de dificultad de los diferentes jugos.
 - Ayuda (INTEGER): en esta columna se guarda la información si es que el usuario quiere o no quiere la ayuda en los juegos.
- Laberintos: en esta tabla se almacenan las posiciones utilizadas para los muros que conforman los laberintos utilizados en el juego laberinto de figuras. Posee las siguiente columnas:
 - Numero_laberinto (INTEGER): Almacena el número de laberinto, el cual es definido como la multiplicación entre nivel y fase correspondiente.
 - Posicion_x (INTEGER): Almacena la posicion inicial horizontal.
 - Posicion_y (INTEGER): Almacena la posicion inicial vertical.
 - Largo_x (INTEGER): Almacena el ancho del muro.
 - Largo_y (INTEGER): Almacena el largo del muro.
- Posiciones_laberinto: se encarga de almacenar las posiciones de los engranajes y esferas que serán agregadas al juego:
 - Numero_laberinto (INTEGER): Almacena el número de laberinto, el cual es definido como la multiplicación entre nivel y fase correspondiente.
 - Tipo_pieza (INTEGER): diferencia cada pieza tal como si es un engranaje de qué color es o si es una esfera y su color.
 - Posicion_x(INTEGER): la ubicación en el eje x que tendrá.
 - Posicion_y (INTEGER): la ubicación en el eje y que tendrá.

- Posicion_fondo_x (INTEGER): indica la posición en el eje que tendrá agujero donde deberá llegar el engranaje, utilizado por los engranajes y la esfera blanca.
- Posicion_fondo_y (INTEGER): indica la posición en el eje que tendrá agujero donde deberá llegar el engranaje, utilizado por los engranajes y la esfera blanca.

11.4. Diseño de interfaces

11.4.1. Menú juegos

Este menú contiene a los siguientes juegos explicados en la siguiente imagen:

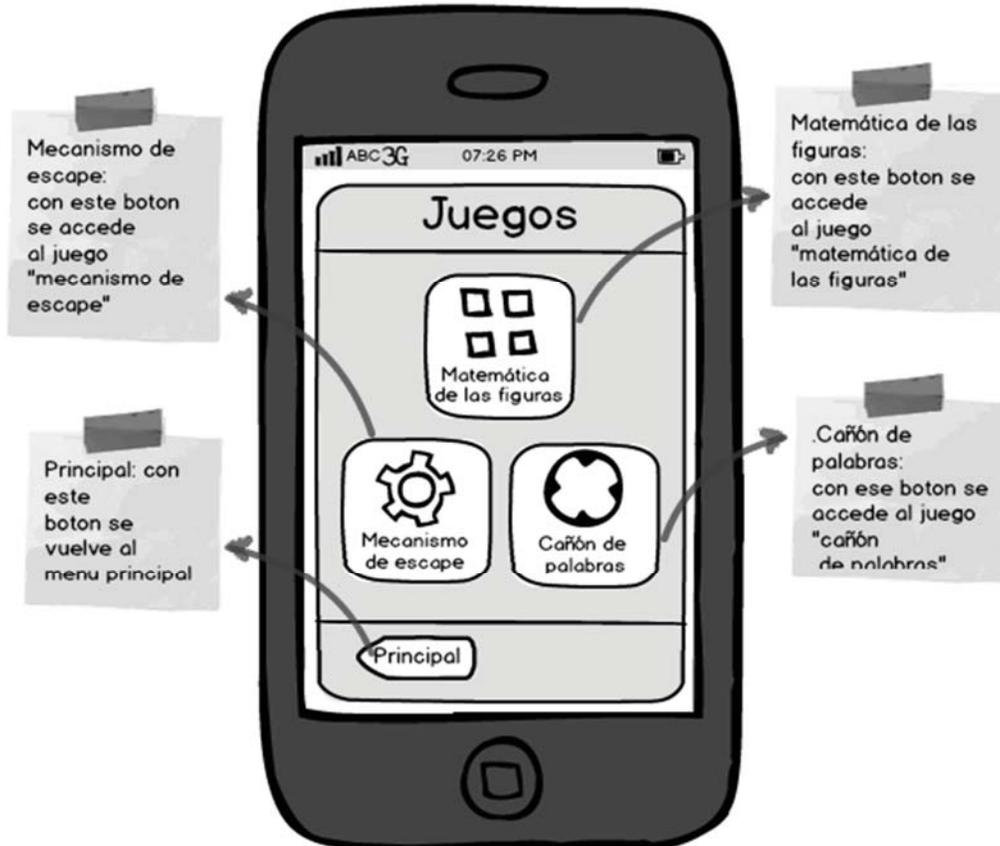


Figura 11.23 Diseño prototipo del menú juegos.

11.4.2. Test

Es donde se realiza el test para evaluar la capacidad intelectual espacial, lingüística y lógico matemática; la siguiente figura muestra la forma en que se muestran los test:

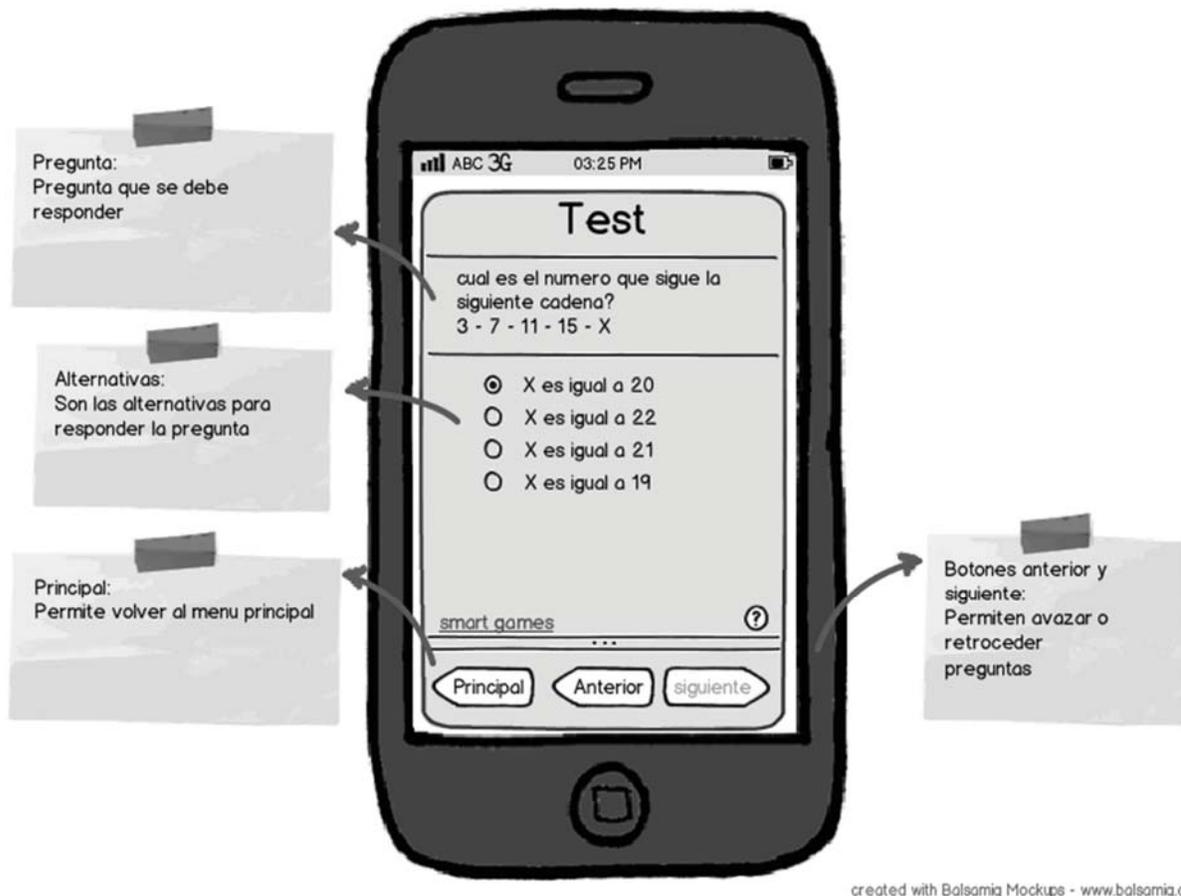


Figura 11.24 Diseño prototipo de test.

11.4.3. Menú principal

El menú principal es la primera pantalla que el usuario visualiza al ejecutar la aplicación, la siguiente figura se muestran las diferentes opciones:

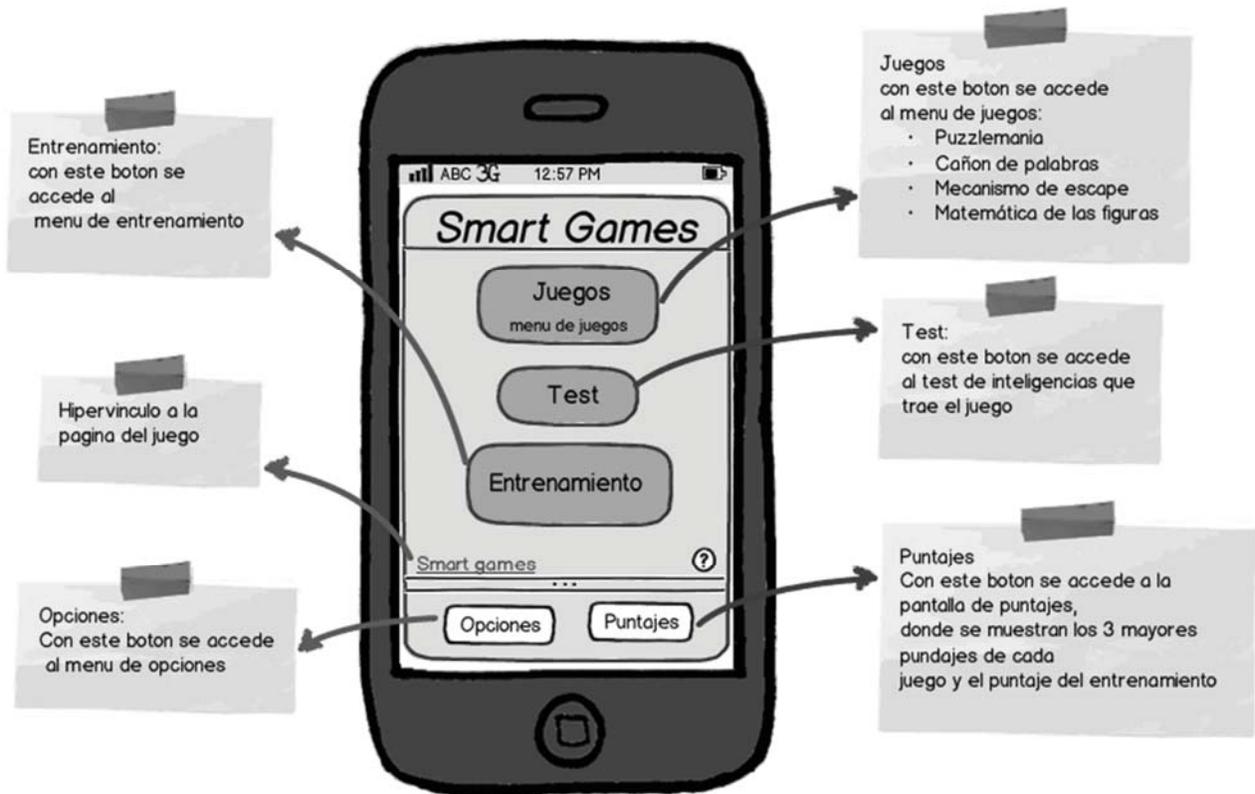


Figura 11.25 Diseño prototipo de menú principal.

11.4.4. Menú opciones

La siguiente figura muestra la forma en la que se muestra el menú de opciones por pantalla:

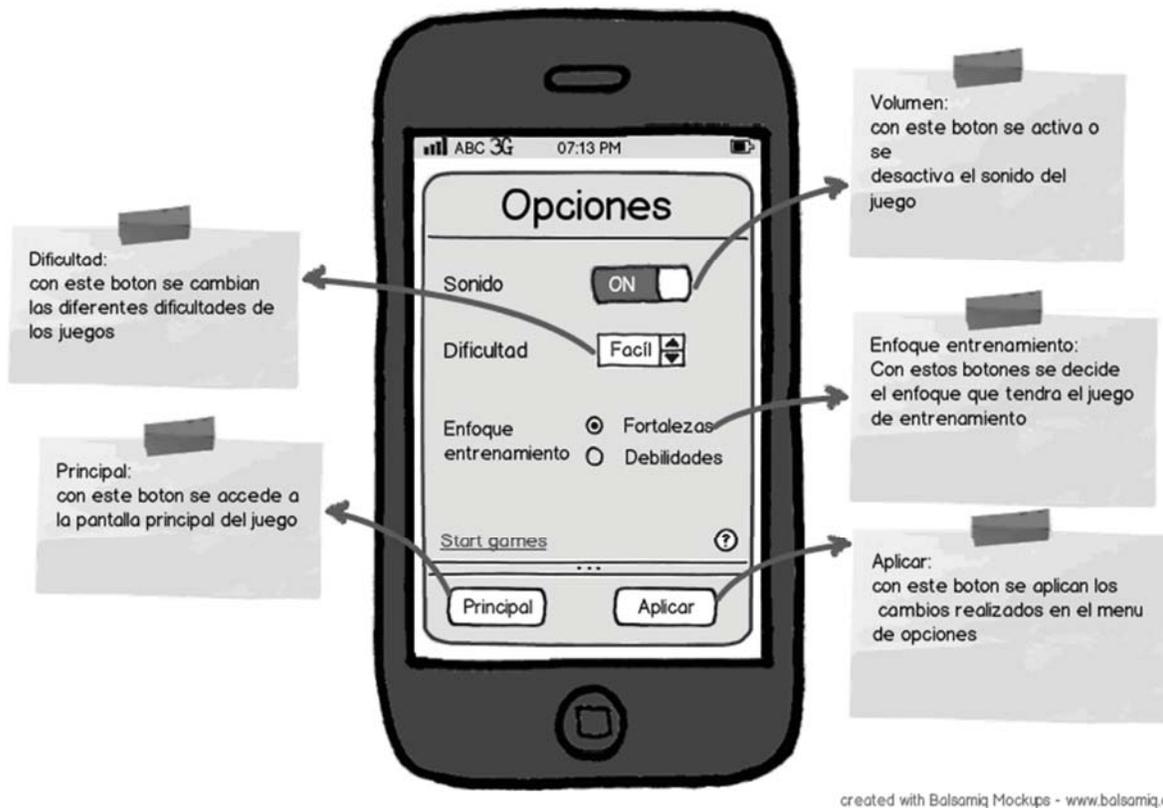


Figura 11.26 Diseño prototipo menú opciones.

11.4.5. Puntajes

La siguiente figura muestra la forma en que se conforma la pantalla de puntajes:

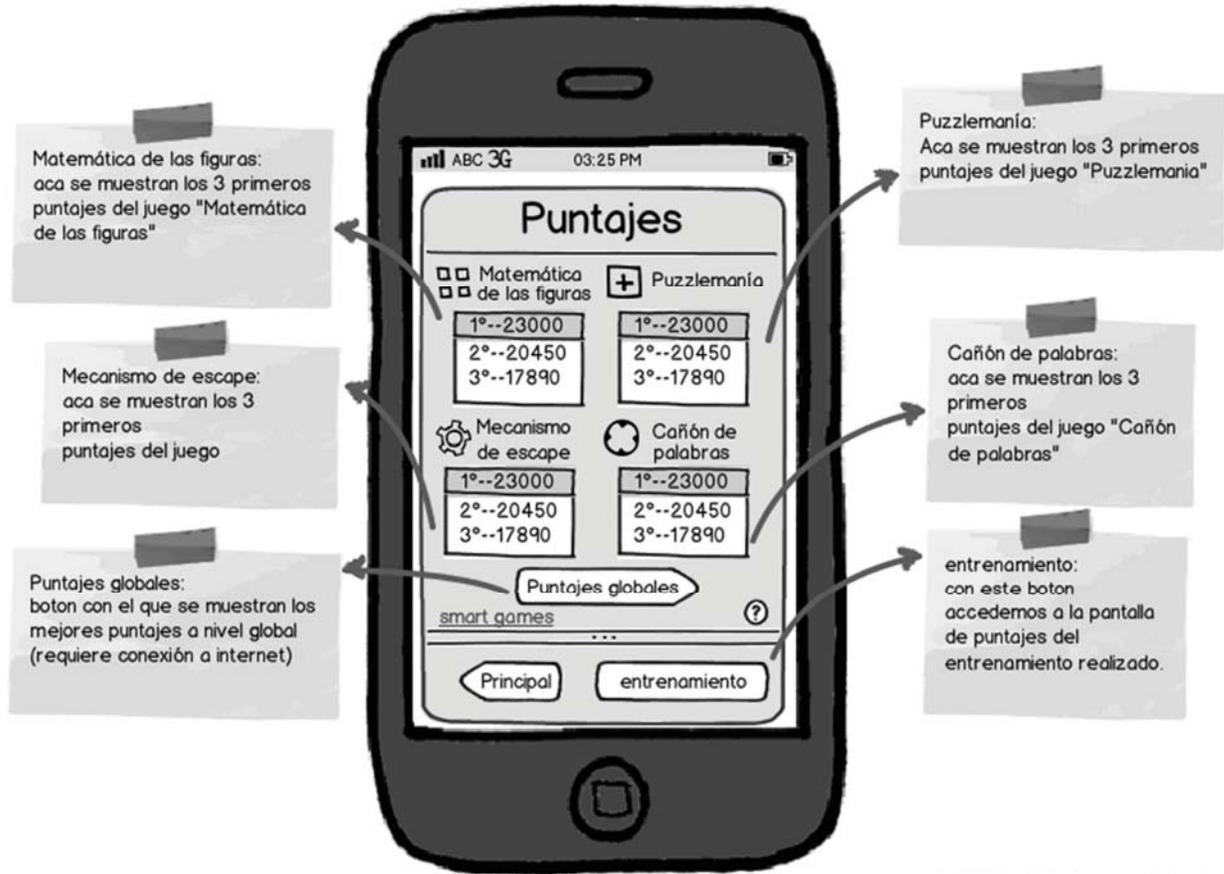


Figura 11.27 Diseño prototipo de puntajes.

11.4.6. Entrenamiento

La siguiente figura muestra como se conforma la pantalla de entrenamiento:



Figura 11.28 Diseño prototipo de entrenamiento.

12. Diagrama de componentes

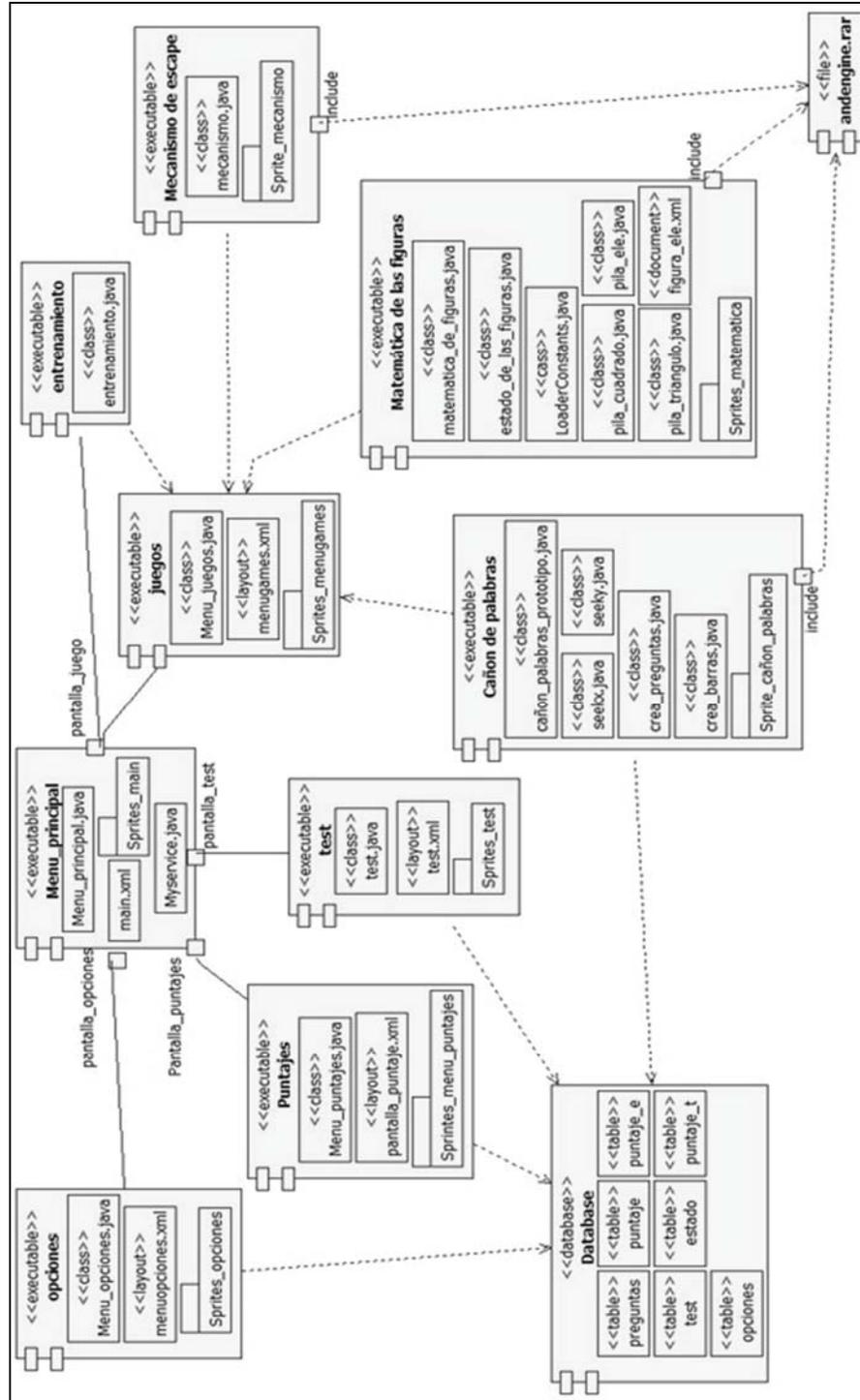


Figura 12.1 Diagrama de componente.

En el diagrama de superior se muestra la relación entre los diferentes componentes del sistema, la mayoría de estos compuestos por una clase (o clases) en java, algún “layout” en xml (a excepción de los juegos) además de un paquete de imágenes llamadas “sprites” que son las imágenes que se utilizan al momento de crear la interface de cada componente.

Se muestra la relación que hay entre la base de datos y los componentes que la utilizan.

Se puede apreciar las conexiones entre los diferentes componentes mediante las llamadas que se están en las líneas que conectan cada uno

13. Prototipo de la aplicación

En la aplicación se implementaron las siguientes funcionalidades:

13.1. Menú principal y menú juegos.

Es el menú donde se tienen las principales funcionalidades de la aplicación, la siguiente imagen muestra el menú principal.

En el menú de juegos se puede seleccionar diferentes tipos de juegos, tales como mecanismo de escape o cañón de palabras, etc.



Figura 13.1 Prototipo de menú principal y menú juegos.

13.2. Matemática de figuras

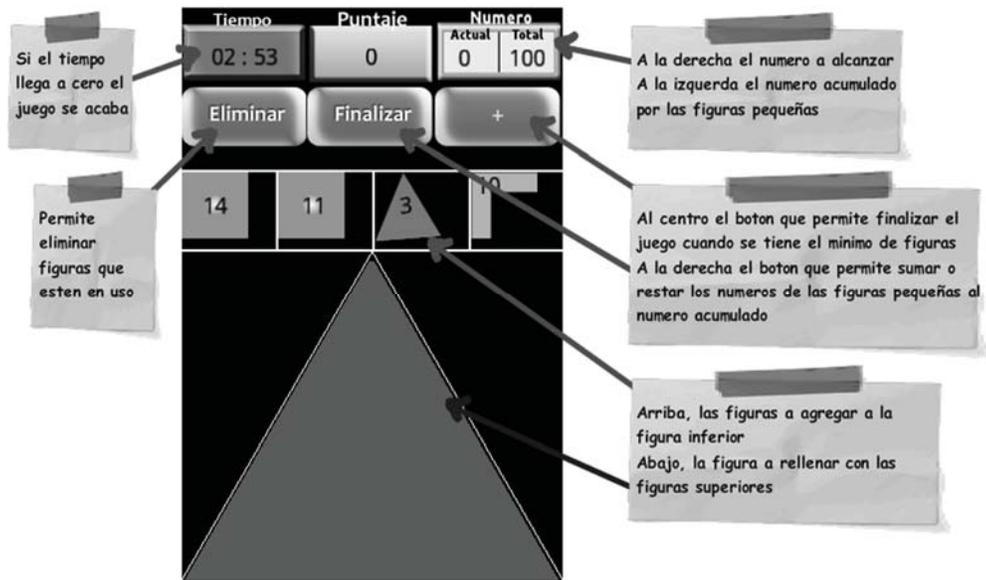


Figura 13.2 Prototipo matemática de figuras.

13.3. Mecanismo de escape

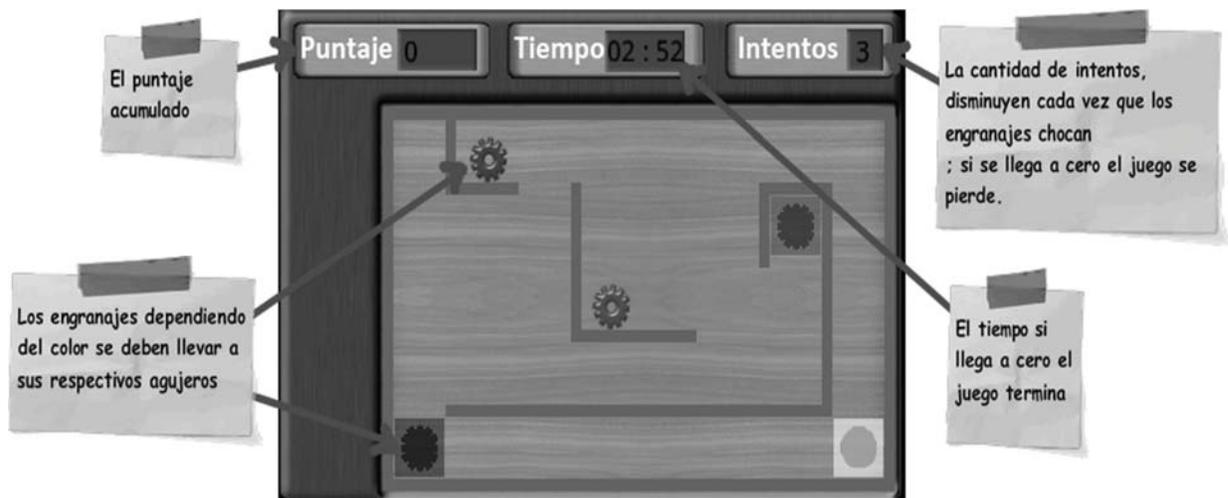


Figura 13.3 Prototipo mecanismo de escape.

13.4. Cañón de palabras

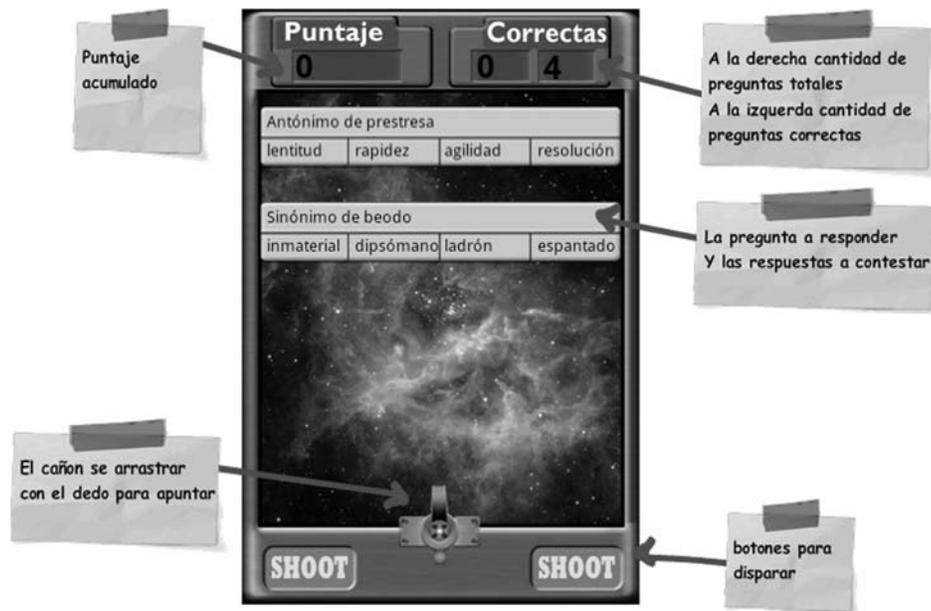


Figura 13.4 Prototipo cañón de palabras.

13.5. Menú opciones



Figura 13.5 Prototipo menú opciones.

13.6. Test

El prototipo de test, utiliza la base de datos local para tomar las preguntas y las respuestas que son mostrarlas en la pantalla del “Smartphone”.

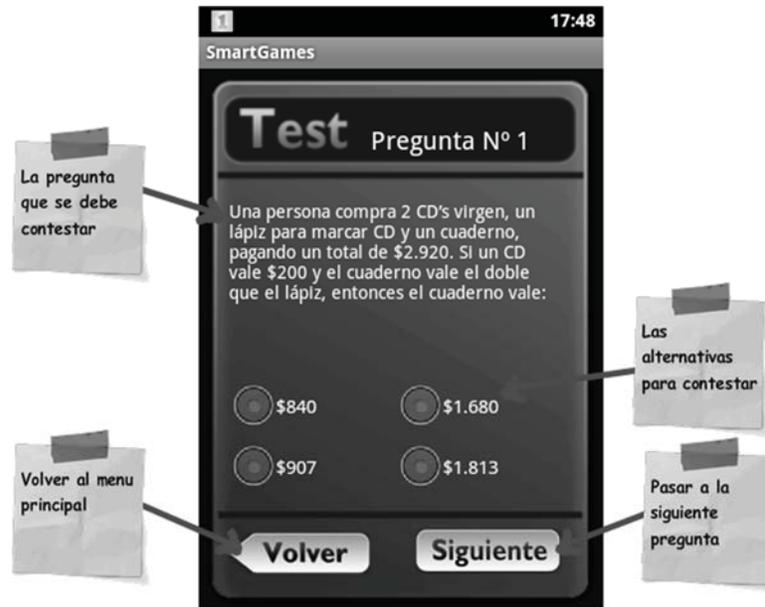


Figura 13.6 Prototipo test.

Como se muestra en la imagen cada pregunta tiene 4 posibles respuestas, además el usuario puede elegir no responde respuestas y pasar a la siguiente pregunta.

13.7. Puntaje



Figura 13.7 Menú de puntaje y puntaje de test.

14. Plan de Pruebas

14.1. Propósito

Este plan trata de cumplir los siguientes objetivos:

- Definir el tipo de pruebas a realizarse
- Identificar los casos de pruebas más relevantes
- Identificar las entradas para los casos de prueba
- Identificar las salidas que debe tener la aplicación.
- Identifica los recursos necesarios al realizar las pruebas

14.2. Entorno

El plan de pruebas definido para este proyecto, consta de una aplicación desarrollada para “Smartphone’s” que soporten el sistema operativo Android versión 2.2; la cual está destinada a apoyar a estudiantes que necesiten estimar o apoyar su capacidad intelectual para entrar a la universidad a través de 3 juegos y un test.

La aplicación hace uso del acelerómetro del “Smartphone” en los juegos, bases de datos tanto locales como externas (en un servidor).

14.3. Alcance

Se realizan las siguientes pruebas de interfaz sobre la aplicación para comprobar que la aplicación es consistente con los requisitos.

14.4. Visión general

La siguiente tabla muestra la documentación utilizada para realizar el plan de prueba:

Tabla 14.1 Documentos utilizados para realizar el plan de prueba.

Documento	Creado	Revisado
Especificación de requisitos	Creado	Si
Especificación de casos de uso	Creado	Si
Especificación de diseño	Creado	Si
Manual de usuario	No	No
Plan de trabajo	Creado	Si

14.5. Sistema utilizado para las pruebas

Tabla 14.2 Describe los componentes utilizados en el sistema en el que se realizan las pruebas

Recursos del sistema	
Recursos	Nombre/Tipo
Lenovo Z470	Intel Core i3-2310M 2.1GHz/RAM: 3GB dual channel/Disco duro: 320GB/ Windows 7 Home Premium
Samsung Galaxy Ace S5830	Procesador: 800 MHz ARM 11, Adreno 200 GPU, Qualcomm MSM7227 chipset. Memoria: interna 158 MB almacenamiento, externa microSD hasta 32GB. Sistema operativo: Android 2.3.6.

14.6. Pruebas a realizar

14.6.1. Pruebas de interfaz

De los anteriores documentos se rescata la siguiente lista que identifica las pruebas unitarias que se realizaron:

- Pruebas sobre los juegos:
 - Colisión de elementos.
 - Colisión entre bala de cañón y respuesta (caso cañón de palabras).
 - Colisión entre figuras
 - Colisión entre mecanismo
 - Acción sobre menú pausa, menú victoria y derrota.
 - Al entrar al menú.
 - Al superar un nivel.
 - Al perder en un nivel.
 - Acción de elementos del menú.
 - Condición de victoria y derrota.
 - Guardado de niveles, fases y puntajes.

- Sobre el test:
 - Verificación de respuesta correcta.
 - Verificación de puntaje.

- Sobre opciones:
 - Que las opciones seleccionadas se aplique correctamente.
 - Las opciones seleccionadas perduren luego de cerrar el juego.

14.7. Casos de prueba

La tabla de casos de prueba tiene el siguiente formato: lugar de la aplicación donde se aplica el caso de prueba, la acción de entrada, la salida esperada y la salida especial.

Tabla 14.3 Casos de prueba.

Lugar de la aplicación	Acción de entrada	Salida esperada	Salida especial
Cañón de palabras	El tocar el botón para disparar	El proyectil salga disparado y colisione con una respuesta (dificultades normal, fácil y difícil).	El proyectil salga en dirección diagonal y salga por los costados de la pantalla sin tocar ninguna respuesta (dificultad difícil).
Cañón de palabras	Colisión de proyectil con respuesta.	El proyectil colisiona con la respuesta ocurre una de los siguientes eventos: Si la respuesta es correcta aumenta el puntaje Si la respuesta es incorrecta disminuye el margen de aparición de palabras.	-
Matemática de figuras	Mover el “Smartphone” (uso de acelerómetro) o arrastrar una figura por la pantalla	La figura no se superpone tanto con muros como con otras figuras.	-

Tabla 14.4 Casos de prueba de interfaz.

Lugar de la aplicación	Acción de entrada	Salida esperada	Salida especial
Mecanismo de escape	Mover el “Smartphone” (uso de acelerómetro).	Si al mover las piezas con el acelerómetro: En caso de que sean mecanismos colisionan y luego se eliminan, dependiendo de la cantidad de intentos el juego se reanuda o termina. En caso de que sean esferas se eliminan al colisionar, el juego termina. Las figuras no se superponen con muros.	-
En cualquier juego	Presionar el botón menú, volver, inicio o de apagado/encendido.	Al presionar el botón menú del “Smartphone” se espera que el juego quede pausado, se muestre un recuadro con las opciones, las opciones son distintas para los juegos normales como los de entrenamiento.	-
En cualquier juego	Se cumple con la condición de victoria o derrota.	Se muestra un menú con las opciones al superar un nivel o perder.	-
En cualquier juego	En el menú al presionar principal.	Concluye el juego y vuelve al menú principal.	-
En cualquier juego	En el menú al presionar resumir o al estar en el menú y presionar la tecla menú del “Smartphone” o la tecla volver.	El juego se reanuda.	-
En cualquier juego normal.	En el menú al presionar el botón fase.	Concluye el juego y vuelve a la selección de fases.	-
En cualquier juego	En el menú al presionar el botón principal.	Concluye el juego y vuelve al menú principal.	-
Cañón de palabras	Condición de victoria: logra contestar la cantidad de preguntas exigidas para superar el nivel.	Se despliega el menú de victoria.	-
Cañón de palabras	Condición de derrota: Alguna pregunta colisiona con el cañón.	Se despliega el menú de derrota.	-

Tabla 14.5 Casos de prueba de interfaz.

Lugar de la aplicación	Acción de entrada	Salida esperada	Salida especial
Matemática de figuras	Condición de victoria: Se pulsa el botón para terminar y se tiene un número y figura similar al exigido, dentro del límite de tiempo.	Se despliega el menú de victoria.	-
Matemática de figuras	Condición de derrota: Se pulsa el botón para terminar y no se tiene un número y figura similar al exigido, o se acaba el tiempo.	Se despliega el menú de derrota.	-
Mecanismo de escape	Condición de victoria: La esfera blanca llega a su lugar de destino.	Se despliega el menú de victoria.	-
Mecanismo de escape	Condición de derrota: Se pierdan los tres intentos, la esfera blanca toque la esfera negra.	Se despliega el menú de derrota.	-
Puntajes	Superar el nivel de algún juego o responder el test completo.	Se muestren los mejores 3 puntajes de test y de cada juego normal o de entrenamiento.	-
Fase o niveles de cada juego.	Luego de superar un nivel.	Que se agregue una nueva fase o nivel dependiendo del nivel superado, que el cambio persista aun cuando se cierre la aplicación.	-
Test	Pantalla de recuento de puntaje.	La cantidad de respuestas correctas en cada inteligencia, rango en el que se encuentra la persona y una recomendación.	-
Opciones	Al cambiar una opción.	Que el cambio se aplique a toda la aplicación, el cambio persista aun cuando se cierre la aplicación.	-

14.8. Resultado de las pruebas

La siguiente tabla muestra los resultados que arrojó la aplicación al realizar los casos de prueba:

Tabla 14.6 Resultado de casos de prueba parte 1.

Lugar de la aplicación	Acción de entrada	Salida del programa	Causa	Solución
Cañón de palabras	El tocar el botón para disparar	En ocasiones el programa falla cuando la bala sale de la pantalla y no colisiona con ninguna respuesta.	La pila de proyectiles está vacía.	Indicar que la pila no tiene elementos.
Cañón de palabras	Colisión de proyectil con respuesta.	Un error con la base de datos produce que, a veces no se detecte la respuesta correcta. Si se dispara un proyectil entre dos respuestas y una de ellas es la respuesta correcta, se selecciona la incorrecta.	No se cierra correctamente la base de datos.	Cerrar la base de datos y comprobar si esta efectivamente cerrada.
Matemática de figuras	Mover el “Smartphone” (uso de acelerómetro) o arrastrar una figura por la pantalla	La figura en forma de ele o el hexágono se acoplan las líneas que forman los vértices de la figura a completar.	La figura no tiene correctamente implementada las colisiones.	Ajustar el FixtureDef para que detecte correctamente las colisiones.
Mecanismo de escape	Mover el “Smartphone” (uso de acelerómetro).	Se cumple correctamente con ello. Tiene un problema de uso de memoria que se da en Smartphones de Samsung un ejemplo de ello es el Samsung Galaxy Ace.	-	-
En cualquier juego	Presionar el botón menú, volver, inicio o de apagado/encendido.	Tal como se explica en la salida esperada, realiza las acciones correctamente, tanto en los juegos normales como en los de entrenamiento.	-	-
En cualquier juego	Se cumple con la condición de victoria o derrota.	Es como se explica en la salida esperada.	-	-
En cualquier juego	En el menú al presionar principal.	Se cumple con la salida esperada.	-	-
En cualquier juego	En el menú al presionar resumir o al estar en el menú y presionar la tecla menú del “Smartphone” o la tecla volver.	Se cumple con la salida esperada.	-	-

Tabla 14.7 Resultados de casos de prueba parte 2.

Lugar de la aplicación	Acción de entrada	Salida del programa	Causa	Solución
En cualquier juego normal.	En el menú al presionar el botón fase.	Se cumple con la salida esperada.	-	-
En cualquier juego	En el menú de derrota al presionar el botón de reiniciar.	Se cumple con la salida esperada.	-	-
Cañón de palabras	Condición de victoria: logra contestar la cantidad de preguntas exigidas para superar el nivel.	Se cumple con la salida esperada	-	-
Cañón de palabras	Condición de derrota: Alguna pregunta colisiona con el cañón.	Se cumple con la salida esperada.	-	-
Matemática de figuras	Condición de victoria: Se pulsa el botón para terminar y se tiene un número y figura similar al exigido, dentro del límite de tiempo.	Se cumple con la salida esperada	-	-
Matemática de figuras	Condición de derrota: Se pulsa el botón para terminar y no se tiene un número y figura similar al exigido, o se acaba el tiempo.	Se cumple con la salida esperada.	-	-
Mecanismo de escape	Condición de victoria: La esfera blanca llega a su lugar de destino.	Se cumple con la salida esperada.	-	-

Tabla 14.8 Resultados de los casos de prueba parte 3.

Lugar de la aplicación	Acción de entrada	Salida del programa	Causa	Solución
Mecanismo de escape	Condición de derrota: Se pierdan los tres intentos, la esfera blanca toque la esfera negra. Se acabe el tiempo.	Se pierde en caso de perder los tres intentos y chocar la esfera negra con la blanca. El juego continua aunque el tiempo llegue a 00:00, el tiempo se detiene en 59:59.	Mal definida la condición de término.	Crear una condición que indique que si termina el juego se pierda.
Puntajes	Superar el nivel de algún juego o responder el test completo.	Se cumple con la salida esperada	-	-
Fase o niveles de cada juego.	Luego de superar un nivel.	Se cumple con la salida esperada.	-	-
Test	Pantalla de recuento de puntaje.	Se cumple con la salida esperada; rara vez ocurren errores con la base de datos.	No se cierra correctamente la base de datos.	Cerrar la base de datos y comprobar si esta efectivamente cerrada.
Opciones	Al cambiar una opción.	Se cumple con la salida esperada.	-	-

15. Conclusiones y trabajos futuros.

A lo largo del desarrollo de este proyecto se profundiza en diferentes áreas de investigación, diferentes áreas de aplicación y se aprendió mucho sobre varias áreas que no poseen mucha exploración en la informática actual, tales como la inteligencia, la educación, los juegos serios, entre otros.

El uso de las inteligencias múltiples por sí solo no es una herramienta que permita predecir si un alumno será capaz de tener un rendimiento adecuado para la ingresar a la universidad; solo con el apoyo de preguntas PSU relacionadas con esa inteligencia, se alcanza el objetivo de predicción tanto de éxito como de rendimiento académico en la universidad.

Gracias a los dos estudios mencionados en este informe se pudo asentar firmemente una base teórica y a la misma vez aplicada, que puede sustentar el proyecto como tal. Por una parte el test MIDAS y el test realizado a los alumnos de primer año fue consecuente el uno con el otro, por lo cual se obtuvieron valiosos datos relacionados a si seguir o no con el camino de las inteligencias múltiples. Por otra parte las bases que nos dio el estudio “Estudio validez predictiva de la PSU sobre el éxito académico” para poder realizar las preguntas de la aplicación fueron de mucha ayuda, ya que se pudo tener una guía y un modelo a seguir.

Se aprendió mucho sobre el ámbito de los dispositivos móviles, ya sea tanto por la parte de la programación, como por la funcionalidad que tienen este tipo de aplicaciones; ya que estas afectan directamente al usuario y pueden llegar a ser muy selectivas, como el caso de este proyecto. Gracias a AndEngine se abrió una puerta muy grande al desarrollo de aplicaciones que requieran movimientos, gravedad, interacción entre objetos, entre otros factores.

Por otro lado, el trabajo realizado para poder montar la aplicación sobre el sistema Android fue de mucha utilidad, ya que se aprendió todo lo básico para poder programar y crear herramientas para este tipo de ambiente.

El análisis y la investigación realizados en este proyecto tomaron tanto tiempo como la implementación del código.

Se tuvieron que tomar decisiones claves al momento de enfrentar el proyecto, tales como remover un juego de los planteados originalmente, acotar algunas propuestas que parecían

muy buenas en el papel, pero al momento de tratar de implementarlas se volvían muy difíciles de llevarlas a cabo.

En general, todo el trabajo realizado durante la concepción de este proyecto fue de mucha utilidad para los integrantes del grupo, dando mucha experiencia, tanto en lo personal como en lo profesional.

Como punto final cabe destacar que lo presentado es solo el prototipo de una aplicación final y mucho más grande, ya que el proyecto solo fue acotado para poder terminar en el tiempo estimado.

Como trabajo futuro se estima el desarrollar lo siguiente: implementar en un servidor, una página web y una base de datos destinada a mostrar y almacenar los mejores puntajes. Tanto de los juegos como de la aplicación.

Con el fin de identificar el puntaje de cada jugador se desea desarrollar un sistema de gestión de usuarios, implementado en la aplicación, en la página web y en la base de datos,

Además se tenía contemplado crear un juego tipo “rpg”, en el cual cada usuario tiene un personaje, el cual depende de los puntajes del test arrojados por la aplicación, con el fin de mejorar la competitividad de los usuarios en la aplicación.

Este proyecto como tal es aplicable a muchas otras áreas dentro de la universidad, puede abarcar varios sectores como las comunicaciones, la educación, las ciencias, las artes, etc. Puede ser escalable a más tipos de procesos universitarios o puede ser modificada para acotar otras aristas. Teniendo en cuenta todas estas premisas, el proyecto presentado tiene una gran posibilidad de ser continuado fuera de la universidad y tal vez pueda ser un escalón para una aplicación de más envergadura.

16. Bibliografía

- [1] **Howard Gardner.** *Estructuras de la mente. Teoría de las inteligencias múltiples.* 1994.
- [2] **Jorge Manzi Astudillo, David Bravo Urrutia, Guido del Pino Manresa, Graciela Donoso Retamales, Manuel MartínezMartínez, Raúl Pizarro Sánchez.** *Estudio acerca de la Validez Predictiva de los Factores de Selección a las Universidades del Consejo de Rectores, admisiones 2003 a 2006.* [En línea] 01 de 03 del 2013. [Citado el 22 de 09 del 2012]. http://www.cta-PSU.cl/docs/Estudio_de_Validez_Predictiva_2003-2006.pdf.
- [3] **Branton Shearer.** *The MIDAS: A professional manual.* 2007.
- [4] **Felicia Patrick.** *Video juego en el aula.* [En línea] 01 de 03 del 2013. [Citado el 22 de 09 del 2012]. http://games.eun.org/upload/GIS_HANDBOOK_ES.pdf.
- [5] **Google Inc.** *Información OS Android.* [En línea] 01 de 03 del 2013. [Citado el 22 de 09 del 2012]. <http://www.Android.com/developers/>.
- [6] **Jesús Valverde Berrocoso.** *Taxonomía de juegos serios.* [En línea] 01 de 03 del 2013. [Citado el 22 de 09 del 2012]. <http://nodoeducativo.blogspot.com/2009/03/una-taxonomia-de-juegos-serios-serious.html>.
- [7] **Daiana Yamila Rigo, Danilo Donolo.** *Una medida de las inteligencias múltiples en contextos universitarios.* Revista Electrónica de Desarrollo de Competencias (REDEC) - N° 6 - Vol. 2 – 2010 Universidad de Talca. [En línea] 01 de 03 del 2013. [Citado el 22 de 09 del 2012]. <http://www.educandus.cl/ojs/index.php/fcompetencias/article/viewFile/75/79>
- [8] **Ángel Alexis Reyes Elgueta, María Daniela Torres Pavez.** *La PSU y otros factores de rendimiento y éxito académico universitario: El caso de la Pontificia Universidad Católica de Valparaíso.* Tesis de título. Escuela de Ingeniería Comercial. Pontificia Universidad Católica de Valparaíso. 2009.

Anexo

A:Test de inteligencias



TEST DE INTELIGENCIAS MULTIPLES (H. GARDNER)

Parte 1 Test para identificar preferencias

Instrucciones:

Lee cuidadosamente cada una de las afirmaciones siguientes.

a.- Si crees que refleja una característica tuya y te parece que la afirmación es verdadera, escribe "V".

b.- Si crees que no refleja una característica tuya y te parece que la afirmación es falsa, escribe una "F".

c.- Si estás dudoso porque a veces es verdadera y a veces falsa no escribas nada y déjala en blanco.

- 1.- _____ Prefiero hacer un mapa que explicarle a alguien como tiene que llegar a un lugar determinado.
- 2.- _____ Puedo sumar o multiplicar mentalmente con mucha rapidez
- 3.- _____ Me gusta trabajar con calculadora y computadoras
- 4.- _____ ¿Disfruto de una buena charla, prédica o sermón?
- 5.- _____ Siempre distingo el Norte del Sur, esté donde esté.
- 6.- _____ Siempre entiendo los gráficos que vienen en las instrucciones de equipos o Instrumentos.
- 7.- _____ Me gusta resolver puzzles y entretenerme con juegos electrónicos.
- 8.- _____ Me enoja cuando escucho una discusión o una afirmación que me parece ilógica o absurda.
- 9.- _____ A menudo puedo captar relaciones entre números con mayor rapidez y facilidad que algunos de mis compañeros.
- 10.- _____ Me gusta construir modelos, maquetas o hacer esculturas.
- 11.- _____ Soy bueno para encontrar el significado preciso de las palabras.
- 12.- _____ Puedo mirar un objeto de una manera y con la misma facilidad verlo dado vuelta o al revés.
- 13.- _____ Me gusta trabajar con números y figuras.
- 14.- _____ Solamente con mirar las formas de las construcciones y estructuras me siento a gusto.
- 15.- _____ Me gusta escribir en las redes sociales.

Parte 2 Test inteligencias múltiples.

Marque la alternativa correcta, existe solo una alternativa correcta.

Inteligencia lógico-matemática

- 1.- ¿Cómo continua la siguiente serie de números? 5- 6- 8- 11- 15- ...
a) 20 b) 17 c) 19 d) 22 e) 25
- 2.- Si yo tengo 9 manzanas, guardo 2 y me como todas las demás excepto por 3 ¿Cuántas manzanas me quedan?
a) 4 b) 5 c) 3 d) 2 e) 15
- 3.- Lea las siguientes frases y deduzca una solución:
Un hombre fue apuñalado y murió.
Camila fue vista dejando la escena del crimen poco tiempo después de la hora del homicidio.
Fue encontrada sangre en la ropa de Camila que pertenecía a la víctima.
Fueron encontradas huellas dactilares de Camila en el arma homicida.
Del crimen no se encontraron más testigos.
ADN de Camila fue encontrado en el cuerpo de la víctima.

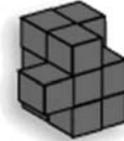
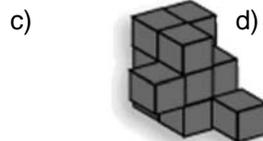
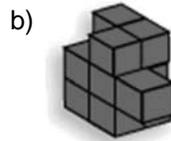
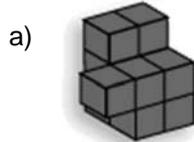
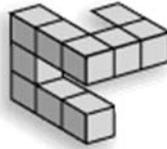
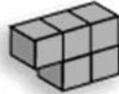
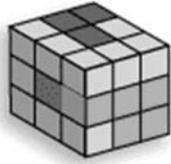
¿Se puede asegurar que Camila es la homicida?
a) Si b) No
4. Tres amigos van a almorzar a un restaurant, luego de comer deben pagar \$2500, entre los 3 entregan \$3000, el cual recibe la camarera, ¿Cuál es la propina que recibe la camarera del vuelto si se necesita entregar vuelto igual, mayor a cero a cada uno de los amigos?
a) 500 b) 600 c) 400 d) 100 e) 200
- 5.- Juan es más alto que Víctor, Jorge es más alto que Juan que es más alto que Víctor y más corto que Jorge. ¿Quién es el mas pequeño entre Juan, Jorge y Víctor?
a) Ellos son del mismo tamaño b) Víctor c) Juan d) Jorge
- 6.- Si la suma de las edades de A, B y C es de 60 años, y A es la mitad de la edad de C, mientras que B es la edad de C menos la edad de A: ¿Qué edades son, respectivamente, A, B, C?
a) 10, 30, 20 b) 20, 0, 40 c) 10, 10, 40 d) 15, 15, 30
- 7.- Lleno a Villavieja, me cruce con 7 viejas, cada una con 7 sacos, cada saco con siete ovejas. ¿Cuántas viejas y ovejas iban a Villavieja?
a) 49 b) 343 c) 7 d) 21 e) 0

Inteligencia espacial

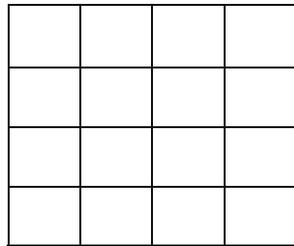
1.- Si Marcos está parado a mi derecha y yo estoy parado a su derecha, él está orientado hacia:

- a) la misma dirección que yo
- b) la dirección opuesta
- c) mi derecha
- d) mi izquierda

2.- Indique la parte que falta para armar el cubo:



3.- Indique cuantos cuadrados hay en la figura

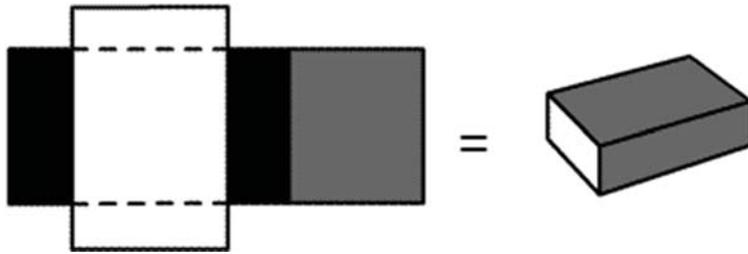


- a) 27 b) 17 c) 16 d) 25 e) 30

4.- La esfera de este reloj debe cortarse en seis partes de forma cualquiera, de modo que la suma de los números que haya en cada parte sea la misma. Trace sobre el reloj las partes a cortar

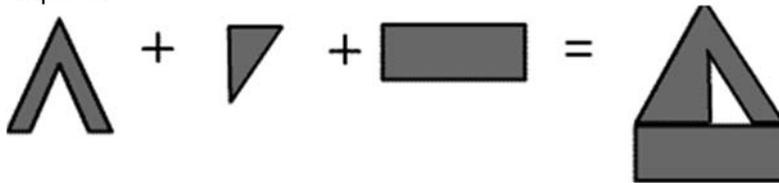


5.- Indique si:



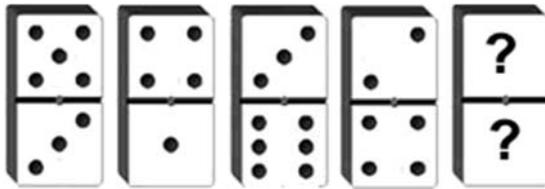
- a) Verdadero b) Falso

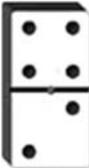
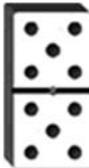
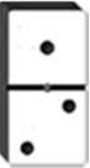
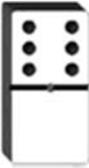
6.- Indique si:



- a) Verdadero b) Falso

7.- ¿Qué dado falta para completar la serie?



- a)  b)  c)  d) 

Inteligencia Lingüística

- 1.- Identifique la palabra que no pertenece al grupo:
a) Zopenco b) Illetrado c) Erudito d) Bodoque
- 2.- Identifique la palabra que no pertenece al grupo:
a) Gritar b) Vociferar c) bramar d) cuchichear
- 3.- Señale la alternativa que complete los espacios correctamente de la siguiente oración:
"Santiago, así como en Valparaíso _____ en la plaza _____ estos lugares en los que _____ incidentes.
a) Está – fue –hubieron
b) Están – tuvo –hubieron
c) Están- fueron –hubo
d) Quedan – fueron ha habido
e) Queda – hubo – habría
- 4.- En el grupo que sigue, indique las dos palabras que, por su significado, no reflejan el mismo concepto:
a) Pegamento y clavo b) serrucho y sierra
b) Clavo y tornillo d) cuerda y cola
- 5.- _____ es a ABDE como EFHI es a _____
a) 1234 – 4567 b) 1245 – 1245 c) 1245 – 5689 d) 1267 – 7889
- 6.- _____ es a palmera como uva es a _____
a) dátil – vid b) arbusto – vid c) árbol – vino d) arbusto – dulce
- 7.- "Dicho de una persona: A quien no se infunde fácilmente terror, o a quien nada intimida."
a) Impertérrito b) Atónito c) Impertinencia d) Impaciencia.