

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**DISEÑO LIGERO DE VIDEOJUEGOS MULTIPLATAFORMA EN MÓVILES
CON CORONA SDK**

**PROYECTO DE TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

LÍNEA DE INVESTIGACIÓN : Ingeniería de Software

AREA PRIORIZADA PNCYT : Investigación y desarrollo para mejorar la calidad de software (cód. 04030501)

AUTOR : Bach. Jhancarlo Florencio Silva Ochoa

ASESOR : Ing. Enrique Edgardo Condor Tinoco

**ANDAHUAYLAS – APURÍMAC
PERÚ**

Diciembre, 2015

DEDICATORIA

A mis padres, por darme la vida y desde aquí les dedico mi esfuerzo, a mis abuelitos, a mis tíos, a mis hermanos, por confiar en mis capacidades y ayudarme a lograr mis sueños.

AGRADECIMIENTO

Agradezco al Ingeniero Enrique Edgardo C3ndor Tinoco, por su asesor3a durante el proceso de desarrollo del presente proyecto de tesis.

A la Ingeniera Cecilia Edith Garc3a Rivas Plata, por su asesor3a de mi proyecto de tesis cuando a3n era estudiante en la Universidad, por la confianza que deposit3 en mi persona.

A mis docentes de la universidad que en alg3n momento compart3 experiencias y retos, por demostrar excelencia acad3mica.

A mis abuelitos Emperatriz y Dalmiro, que son mis segundos padres, supieron cuidarme y guiarme por el buen camino desde muy peque1o, y ense1arme a priorizar las cosas mas importantes de la vida.

A mis t3os Dalmiro y Anny, que siempre estuvieron apoy3ndome en todo momento, exigi3ndome por ser siempre el mejor.

A mis hermanos Johan y Jessica, por brindarme los mejores deseos, respetar y apoyar a mis decisiones.

A mis amigos de la universidad, por compartir muchas experiencias juntos y siempre cumpliendo todas las responsabilidades que se nos asign3.

ÍNDICE GENERAL

	Pág.
RESUMEN	14
INTRODUCCIÓN	16
1. DATOS GENERALES	17
1.1. Título del proyecto.....	17
1.2. Autor del proyecto	17
1.3. asesor del proyecto.....	17
1.4. Línea de investigación.....	17
1.5. Área priorizada del proyecto.....	17
1.6. Institución y lugar de ejecución del proyecto	17
1.7. Duración del proyecto	17
1.8. Cronograma de ejecución del proyecto	18
1.9. Presupuesto.....	19
1.10. Financiamiento.....	20
2. PLANTEAMIENTO DEL PROBLEMA	20
2.1. Realidad problemática.....	20
2.2. Formulación del problema.....	23
2.3. Objetivos.....	23
2.3.1. Objetivo General	23
2.3.2. Objetivos Específicos	23
2.4. Justificación	24
2.5. Viabilidad de la investigación	24
2.5.1. Viabilidad Técnica	24

2.5.2. Viabilidad Económica	24
2.5.3. Viabilidad Social	25
2.6. Limitación del estudio	25
3. MARCO TEÓRICO	25
3.1.1. Framework para la programación en dispositivos móviles	25
3.1.1.1. Introducción	25
3.1.1.2. Sistemas operativos móviles	26
3.1.2. Metodologías ágiles de desarrollo	28
3.1.3. Introducción al Framework Corona SDK	30
3.1.3.1. Características	30
3.1.3.2. Lenguaje de programación de Corona SDK: Lua	31
3.1.3.3. Ciclo de vida de una aplicación móvil en Corona SDK	36
3.1.3.4. Librerías básicas usadas en videojuegos sencillos en Corona SDK	37
3.2. Estado del arte	38
4. INGENIERÍA DEL PROYECTO	39
4.1. ADMINISTRACIÓN DEL PROYECTO BASADO EN PMBOK	39
4.1.1. Plan de gestión de la integración	39
4.1.2. Plan de gestión del alcance	42
4.1.3. Plan de gestión del tiempo	42
4.1.4. Plan de gestión de los costos	43
4.1.5. Plan de gestión de la calidad	44
4.1.6. Plan de gestión de los recursos humanos	45
4.1.7. Plan de gestión de las comunicaciones	46

4.1.8. Plan de gestión de los riesgos.....	47
4.1.9. Plan de gestión de las adquisiciones.....	48
4.1.10. Plan de gestión de los interesados.....	49
4.2. METODOLOGÍA Y PLANIFICACIÓN DEL PROYECTO	51
4.2.1. Requisitos funcionales	52
4.2.2. Requisitos no funcionales	52
4.3. PUESTA EN MARCHA LA METODOLOGÍA MOBILE-D	52
4.3.1. FASE DE EXPLORACIÓN	52
4.3.1.1. Estableciendo actores.....	52
4.3.1.2. Estableciendo Usuarios.....	53
4.3.2. FASE DE INICIACIÓN	53
4.3.2.1. Software.....	53
4.3.2.2. Hardware	54
4.3.2.3. Herramientas para pruebas (Hardware)	54
4.3.2.4. Diagrama de casos de uso de la aplicación móvil	54
4.3.3. FASE DE PRODUCCIÓN	73
4.3.3.1. Diseño de la aplicación móvil	73
4.3.4. FASE DE ESTABILIZACIÓN.....	82
4.3.4.1. Implementación de la aplicación móvil	82
4.3.5. FASE DE PRUEBAS DEL SISTEMA	92
4.3.5.1. Errores encontrados en la aplicación GameDesign.....	92
4.3.5.2. Construcción de la aplicación GameDesign	93
4.3.5.3. Diseño de un videojuego multiplataforma.....	94

4.3.5.4.	Construcción del .apk para Android del videojuego diseñado	101
4.3.5.5.	Construcción del .exe para Windows desktop del videojuego diseñado ..	103
4.3.5.6.	Prueba piloto.....	104
RESULTADOS		109
CONCLUSIONES		112
RECOMENDACIONES		113
REFERENCIAS BIBLIOGRÁFICAS		114
ANEXOS		116

LISTA DE TABLAS

	Pág.
Tabla 1. Presupuesto	19
Tabla 2. Plan de gestión de la integración.....	39
Tabla 3. Plan de gestión del alcance	42
Tabla 4. Plan de gestión del tiempo	42
Tabla 5. Plan de gestión de los costos	43
Tabla 6. Plan de gestión de la calidad.....	44
Tabla 7. Plan de gestión de los recursos humanos	45
Tabla 8. Plan de gestión de las comunicaciones	46
Tabla 9. Plan de gestión de los riesgos.....	47
Tabla 10. Plan de gestión de las adquisiciones.....	48
Tabla 11. Plan de gestión de los interesados	49
Tabla 12. Información clave del proyecto	53
Tabla 13. Caso de uso: Iniciar aplicación	54
Tabla 14. Caso de uso: Selección de capítulos	55
Tabla 15. Caso de uso: seleccionar capítulo imágenes	55
Tabla 16. Caso de uso: Seleccionar capítulo texto	55
Tabla 17. Caso de uso: Seleccionar capítulo toques	56
Tabla 18. Caso de uso: Seleccionar capítulo sonido	56
Tabla 19. Caso de uso: Seleccionar capítulo animación.....	56
Tabla 20. Caso de uso: Seleccionar capítulo física.....	57
Tabla 21. Caso de uso: Seleccionar Actores 1	58
Tabla 22. Caso de uso: Seleccionar Actores 2	58
Tabla 23. Caso de uso: Seleccionar baldosa.....	59
Tabla 24. Caso de uso: Seleccionar fondo	59
Tabla 25. Caso de uso: Listar Objetos.....	62
Tabla 26. Caso de uso: Agregar física.....	62
Tabla 27. Caso de uso: Definir velocidad de animación	63
Tabla 28. Caso de uso: Definir destino 'x' de animación	63
Tabla 29. Caso de uso: Definir destino 'y' de animación	63
Tabla 30. Caso de uso: Definir rotación de animación.....	63
Tabla 31. Caso de uso: Definir movimiento aleatorio de animación.....	64
Tabla 32. Caso de uso: Definir ancho del objeto	64
Tabla 33. Caso de uso: Definir altura del objeto	64
Tabla 34. Caso de uso: Definir rotación del objeto	65
Tabla 35. Caso de uso: Definir control 1.....	65
Tabla 36. Caso de uso: Definir control 2.....	65
Tabla 37. Caso de uso: eliminar objeto	65
Tabla 38. Caso de uso: Definir tiempo creación disparo.....	66
Tabla 39. Caso de uso: Definir destino 'x' disparo	66
Tabla 40. Caso de uso: Definir destino 'y' disparo	66

Tabla 41. Caso de uso: Definir velocidad de animación disparo.....	67
Tabla 42. Caso de uso: definir estado de colisión	67
Tabla 43. Caso de uso: Definir gravedad x.....	67
Tabla 44. Caso de uso: Definir gravedad y.....	68
Tabla 45. Caso de uso: Definir vista	68
Tabla 46. Caso de uso: Definir escenas horizontales	68
Tabla 47. Caso de uso: Definir escenas verticales	69
Tabla 48. Caso de uso: Definir orientación del mapa	69
Tabla 49. Caso de uso: Definir velocidad de animación mapa	69
Tabla 50. Caso de uso: Definir audio del videojuego.....	70
Tabla 51. Caso de uso: Ejecutar videojuego	70
Tabla 52. Caso de uso: Eliminar videojuego.....	70
Tabla 53. Caso de uso: Ir al menú principal	71
Tabla 54. Caso de uso: Ayuda	71
Tabla 55. Caso de uso: Exportar videojuego.....	71
Tabla 56. Caso de uso: Seleccionar mini juego.....	72
Tabla 57. Caso de uso: Opciones	72
Tabla 58. Caso de uso: Salir	72
Tabla 59. Leyenda de la estructura de la aplicación móvil.....	78
Tabla 60. Estructura de directorios y ficheros de la aplicación móvil	78
Tabla 61. resultados pregunta 1	109
Tabla 62. resultados pregunta 2.....	110
Tabla 63. resultados pregunta 3.....	110
Tabla 64. resultados pregunta 4.....	111
Tabla 65. resultados pregunta 5.....	111
Tabla 66. Acciones: Iniciar aplicación.....	116
Tabla 67. Acciones: Selección de capítulos	117
Tabla 68. Acciones: seleccionar capitulo imágenes.....	119
Tabla 69. Acciones: Seleccionar capitulo texto.....	120
Tabla 70. Acciones: Seleccionar capitulo toques.....	121
Tabla 71. Acciones: Seleccionar capitulo sonido.....	123
Tabla 72. Acciones: Seleccionar capitulo animación	124
Tabla 73. Acciones: Seleccionar capitulo física	125
Tabla 74. Acciones: Nuevo videojuego.....	127
Tabla 75. Acciones: Abrir videojuego	128
Tabla 76. Acciones: Seleccionar Actores 1	129
Tabla 77. Acciones: Seleccionar Actores 2	130
Tabla 78. Acciones: Seleccionar baldosa	131
Tabla 79. Acciones: Seleccionar fondo.....	133
Tabla 80. Acciones: Listar Objetos	134
Tabla 81. Acciones: Agregar física	136
Tabla 82. Acciones: Definir velocidad de animación.....	138

Tabla 83. Acciones: Definir destino 'x' de animación	139
Tabla 84. Acciones: Definir destino 'y' de animación	140
Tabla 85. Acciones: Definir rotación de animación	141
Tabla 86. Acciones: Definir movimiento aleatorio de animación	142
Tabla 87. Acciones: Definir ancho del objeto.....	143
Tabla 88. Acciones: Definir altura del objeto	145
Tabla 89. Acciones: Definir rotación del objeto.....	146
Tabla 90. Acciones: Definir control 1	147
Tabla 91. Acciones: Definir control 2	148
Tabla 92. Acciones: eliminar objeto.....	149
Tabla 93. Acciones: Definir tiempo creación disparo	150
Tabla 94. Acciones: Definir destino 'x' disparo	151
Tabla 95. Acciones: Definir destino 'y' disparo	152
Tabla 96. Acciones: Definir velocidad de animación disparo	154
Tabla 97. Acciones: definir estado de colisión	155
Tabla 98. Acciones: Definir gravedad x	156
Tabla 99. Acciones: Definir gravedad y	157
Tabla 100. Acciones: Definir vista	158
Tabla 101. Acciones: Definir escenas horizontales	159
Tabla 102. Acciones: Definir escenas verticales.....	161
Tabla 103. Acciones: Definir orientación del mapa	162
Tabla 104. Acciones: Definir velocidad de animación mapa	163
Tabla 105. Acciones: Definir audio del videojuego	164
Tabla 106. Acciones: Ejecutar videojuego.....	166
Tabla 107. Acciones: Eliminar videojuego	167
Tabla 108. Acciones: Ir al menú principal	168
Tabla 109. Acciones: Ayuda.....	169
Tabla 110. Acciones: Exportar videojuego	174
Tabla 111. Acciones: Seleccionar mini juego	174
Tabla 112. Acciones: Opciones	176
Tabla 113. Acciones: Salir.....	177

LISTA DE FIGURAS

	Pág.
Figura 1. Cronograma de ejecución del proyecto.	18
Figura 2. Principales Medios de Juego.....	21
Figura 3. Frecuencia de Juego.....	22
Figura 4. Disposición de Pago.....	23
Figura 5. Código fuente del archivo main.lua.....	36
Figura 6. Código fuente de una Escena vacía.....	36
Figura 7. Ciclo de vida de Software: Mobile-D.....	51
Figura 8. Modelo conceptual de la aplicación GameDesign.....	74
Figura 9. Dispositivos de diferentes resoluciones de pantalla.....	76
Figura 10. código fuente del fichero config.lua.....	77
Figura 11. código fuente del fichero build.settings.....	77
Figura 12. definiendo iconos de la aplicación móvil.....	78
Figura 13. código fuente del fichero main.lua.....	82
Figura 14. Código fuente de la escena splash.....	83
Figura 15. Código fuente de la escena bienvenida.....	83
Figura 16. Código fuente de la escena menú principal.....	84
Figura 17. Código fuente de importación del fichero loadsave.lua.....	85
Figura 18. código fuente para guardar datos en el fichero tabla.json.....	85
Figura 19. Código fuente para importar módulos.....	86
Figura 20. Código fuente para enviar parámetros a las propiedades.....	87
Figura 21. Variables locales para recuperar información.....	89
Figura 22. Recuperación de cada propiedad de los objetos.....	90
Figura 23. Guardando información en un archivo .json.....	90
Figura 24. Eliminando el diseño de un videojuego.....	91
Figura 25. Mini juego en ejecución.....	92
Figura 26. Disponibilidad de plataforma de construcción.....	93
Figura 27. Construcción de la aplicación para Android.....	94
Figura 28. Aplicación GameDesign instalada correctamente.....	95
Figura 29. Abrir el videojuego diseñado space shooter.....	95
Figura 30. videojuego space shooter abierto.....	96
Figura 31. mensaje de alerta para ver ejecución del videojuego.....	96
Figura 32. ejecución del videojuego diseñado.....	97
Figura 33. selección del bluetooth.....	97
Figura 34. permiso para hacer uso del bluetooth.....	98
Figura 35. dispositivos disponibles para exportar videojuego diseñado.....	98
Figura 36. transferencia completa del videojuego al PC.....	99
Figura 37. recursos del videojuego exportado.....	99
Figura 38. Información importante del archivo LEEME.txt.....	100
Figura 39. Abriendo videojuego diseñado.....	100
Figura 40. Menú principal del videojuego diseñado.....	101

Figura 41. Construyendo videojuego diseñado para Android	101
Figura 42. Construcción exitosa de la aplicación para Android.....	102
Figura 43. Ejecución del videojuego diseñado para Android	102
Figura 44. Construyendo videojuego diseñado para Windows Desktop	103
Figura 45. Construcción exitosa de la aplicación para Windows desktop	103
Figura 46. Ejecución del videojuego en Windows desktop	104
Figura 47. Encuesta prueba de usabilidad	105
Figura 49. Pantalla de arranque de la aplicación móvil GameDesign	116
Figura 50. Pantalla de bienvenida	117
Figura 51. Pantalla menú principal	118
Figura 52. Pantalla capitulo imágenes.....	120
Figura 53. Pantalla capitulo texto	121
Figura 54. Pantalla capitulo toques	122
Figura 55. Pantalla capitulo sonido.....	124
Figura 56. Pantalla capitulo animación	125
Figura 57. Pantalla capitulo física	126
Figura 58. Pantalla crear videojuego	127
Figura 59. Pantalla abrir videojuego	128
Figura 60. Pantalla Actores 1	130
Figura 61. Pantalla Actores 2	131
Figura 62. Pantalla Baldosas.....	132
Figura 63. Pantalla fondos.....	133
Figura 64. Despliegue Actores 1	134
Figura 65. Despliegue Actores 2	135
Figura 66. Despliegue Baldosas.....	135
Figura 67. Despliegue física	137
Figura 68. Mensaje de alerta crear objeto	137
Figura 69. Mensaje de alerta seleccione objeto.....	137
Figura 70. Mensaje de alerta definir tipo de cuerpo	138
Figura 71. Definir velocidad de animación.....	139
Figura 72. Definir destino X de animación	140
Figura 73. Definir destino Y de animación	141
Figura 74. Definir rotación de animación	142
Figura 75. Definir movimiento aleatorio de animación	143
Figura 76. Definir ancho del objeto.....	144
Figura 77. Definir altura del objeto.....	145
Figura 78. Defina la rotación del objeto	147
Figura 79. Defina control 1 del objeto	148
Figura 80. Defina control 2 del objeto	149
Figura 81. Eliminar objetos.....	150
Figura 82. Creación bala en segundos	151
Figura 83. Defina destino X de la bala.....	152

Figura 84. Defina destino Y de la bala.....	153
Figura 85. Defina velocidad de animación bala	155
Figura 86. Defina estado al colisionar	156
Figura 87. Defina gravedad en X.....	157
Figura 88. Defina gravedad en Y.....	158
Figura 89. Defina tipo de vista	159
Figura 90. Defina cantidad de escenas horizontales	160
Figura 91. Defina cantidad de escenas verticales	162
Figura 92. Defina orientación del mapa.....	163
Figura 93. Defina velocidad de animación del mapa	164
Figura 94. Despliegue lista de audios.....	165
Figura 95. Defina audio del videojuego	165
Figura 96. Iniciar videojuego	166
Figura 97. Pantalla videojuego en ejecución	167
Figura 98. Eliminar videojuego	168
Figura 99. Volver al menú principal	169
Figura 100. Ayuda física.....	170
Figura 101. Ayuda animación.....	170
Figura 102. Ayuda objeto	171
Figura 103. Ayuda disparo	171
Figura 104. Ayuda colisiones	172
Figura 105. Ayuda gravedad	172
Figura 106. Ayuda mapa	173
Figura 107. Ayuda opciones.....	173
Figura 108. Pantalla menú mini juego	175
Figura 109. Pantalla ejecución mini juego	175
Figura 110. Menú Opciones	176
Figura 111. Mensaje de alerta salir de la aplicación	177

RESUMEN

El proyecto de tesis tiene como objetivo principal, desarrollar una aplicación móvil en Android denominado “GameDesign” para el diseño de videojuegos multiplataforma en móviles, esta aplicación permite diseñar para 2 plataformas (Android y Windows Desktop).

Existe la posibilidad de desarrollar para los (.ipa), Windows phone(.app), MAC (.mdg), el presente proyecto no considera las plataformas anteriores porque, no se cuenta con una licencia, ya que la construcción de las aplicaciones móviles para estas plataformas es restringido, necesariamente se tiene que pagar una licencia para poder distribuir la aplicación, y es muy costosa, por ello se optó desarrollar para Android ya que es gratuito y de código abierto.

La metodología empleada, es una metodología ágil, denominada Mobil-D, esta metodología es la que mejor se adecua para el proceso de desarrollo de software.

Una vez desarrollado la aplicación móvil “GameDesign”, se procederá a la instalación en un dispositivo real, y a partir de este de empezará a diseñar de manera fácil y rápida videojuegos sencillos, para luego exportarlos por medio del bluetooth a la PC, y construir el instalador del videojuego diseñado para Android (.apk) y para Windows Desktop (.exe).

La aplicación móvil, tiene un apartado de introducción, también incluye un mini juego tipo Geometry Dash, para el entretenimiento del usuario.

Como contribución se tiene, el diseño rápido y sencillo de videojuegos móviles multiplataforma, esto fue posible determinar gracias a una prueba piloto, que se hizo a un conjunto de estudiantes para probar la usabilidad de la aplicación móvil “GameDesign”.

ABSTRAC

This thesis Project main objective is to develop a mobile application for Android called "Gamedesign" design of multiplatform games on phones, this application allows design for 2 platforms (Android and Windows desktop).

There is the possibility of develop ios(.ipa), Windows Phone (.app), MAC (.mdg), this project does not consider platforms previous because, do not have a license, and that construction of the Mobile Applications for these platforms is restricted, necessarily you have to pay a License for distribute the application, and is very expensive, for it is opted to develop for Android because it is free and open source.

The methodology, is an agile methodology, called Mobil-D, This methodology is the best suitable for process of software development.

Once developed the Mobile Application "Gamedesign" shall be the installation in a real device, and from this is start design of fast and easy way simple games, then export through the bluetooth to the PC, and build the installer for Android game designed (.apk) and Windows desktop (.exe).

Mobile application, has a section introduction, also includes a mini-game type Geometry Dash, for the training user.

As a contribution you have, design fast, and simple video game mobile platform, this was possible to determine thanks to a test, which is made a set of students to test the usability of the mobile application "GameDesign".

INTRODUCCIÓN

El desarrollo de una aplicación móvil puede ser compleja, depende a la función que cumplirá, si la construcción se hizo empleando una metodología ágil, que es lo más conveniente, no habrá ningún problema posterior si se desea agregar o modificar la aplicación móvil, por ejemplo la inserción de nuevos módulos, y si también se eligió una herramienta de desarrollo multiplataforma, no existirá problemas de portabilidad.

Es por esta razón que en el presente proyecto de tesis, se definió 2 cosas muy importantes, primero que se quería lograr con el desarrollo de la aplicación móvil y para qué plataformas estaría destinado, y segundo determinar el uso de una metodología ágil de desarrollo.

Por lo tanto se tenía que tener claro la portabilidad de la aplicación móvil a desarrollar, así que se eligió el framework Corona SDK, ya que es una herramienta poderosa para construir aplicaciones móviles multiplataforma, la aplicación a desarrollar permitirá diseñar videojuegos para 2 plataformas específicamente (Android y Windows Desktop), teniendo como entregable final los instaladores (.apk) y (.exe) respectivamente.

Y por último la elección de la metodología ágil de desarrollo que se empleó fue Mobil-D , usado por grupos pequeños de desarrolladores de no más de 10 personas, bajo esta metodología permite a los desarrolladores construir aplicaciones móviles de manera rápida, es una de estas razones por el cual se eligió esta metodología.

PROYECTO DE TESIS

1. DATOS GENERALES

1.1. TÍTULO DEL PROYECTO

Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK.

1.2. AUTOR DEL PROYECTO

Nombres y apellidos : Jhancarlo Florencio Silva Ochoa
Escuela Profesional : Ingeniería de Sistemas
E-mail : jhanca_s_o2011@hotmail.com

1.3. ASESOR DEL PROYECTO

Nombres y apellidos : Enrique Edgardo Condor Tinoco
Departamento académico : Ingeniería y Tecnologías de Información
Categoría Docente : Asociado
Modalidad : Ordinario
E-mail : enricoti@gmail.com

1.4. LÍNEA DE INVESTIGACIÓN

Ingeniería de Software (cód. 0403 05)

1.5. ÁREA PRIORIZADA DEL PROYECTO

Investigación y desarrollo para mejorar la calidad de software (cód. 0403 0501)

1.6. INSTITUCIÓN Y LUGAR DE EJECUCIÓN DEL PROYECTO

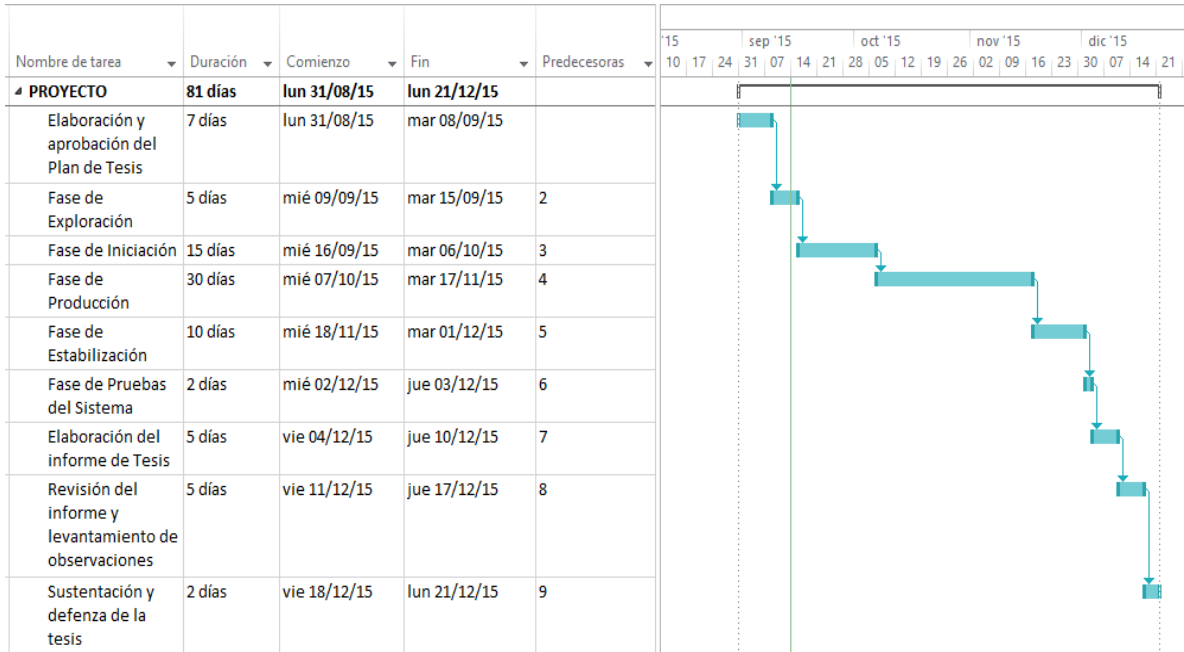
Universidad Nacional José María Arguedas, Andahuaylas ,San Jerónimo –Totoral

1.7. DURACIÓN DEL PROYECTO

4 meses

1.8. CRONOGRAMA DE EJECUCIÓN DEL PROYECTO

Figura 1. Cronograma de ejecución del proyecto.



Fuente: Elaboración propia. Utilizando Microsoft Project 2013.

1.9. PRESUPUESTO

Tabla 1. Presupuesto

ITEM	DESCRIPCIÓN	Cantidad	UNID. MEDIDA	Precio Unit	Precio parcial
1.00.00	BIENES				2680,00
1.01.00	Materiales de escritorio				2.530,00
1.01.01	Laptop Asus core i5	1	Unid	2.200,00	2.200,00
1.01.02	USB hp de 8Gb	1	Unid	30,00	30,00
1.01.03	Tablet Advance 9'	1	Unid	300,00	300,00
1.02.00	Materiales Consumibles				150,00
1.02.01	Papel Bond A4	1	ciento	8,50	8,50
1.02.02	Impresión Informe de tesis	7	Unid	20,00	140,00
1.02.03	Lapiceros	3	Unid	0,50	1,50
2.00.00	SERVICIOS				490,00
2.01.00	Servicios diversos				490,00
2.01.01	Internet	4	Mes	60,00	240,00
2.01.02	Fotocopias	500	Unid	0,10	50,00
2.01.03	Comunicaciones	400	Minutos	0,50	200,00
3.00.00	Software				-
3.01.00	Desarrollo del sistema				-
3.01.01	Corona SDK (Software Libre)	1	Unid	-	-
3.01.02	Sublime Text 3 (Software libre)	1	Unid	-	-
3.01.03	jdk-8u20-windows-i586	1	Unid	-	-
3.01.04	Lenguaje de programación LUA (Software libre)	1	Unid	-	-
4.00.00	RECURSOS HUMANOS				17.180,00
4.01.00	Asesores	1	persona	2.400,00	2.400,00
4.02.00	Desarrollo del sistema				14.780,00
4.02.01	Programador	1	persona	14.780,00	14.780,00
4.02.01.01	Fase de Exploración	30	Horas	30,00	900,00
4.02.01.02	Fase de Iniciación	60	Horas	30,00	1.800,00
4.02.01.03	Fase de Producción	192	Horas	50,00	9.600,00
4.02.01.04	Fase de Estabilización	52	Horas	30,00	1.560,00
4.02.01.05	Fase de Pruebas del Sistema	16	Horas	20,00	320,00
4.02.01.06	Elaboración de informe de tesis	20	Horas	30,00	600,00
COSTO TOTAL DE LA TESIS					20.350,00

Fuente: Elaboración propia. Utilizando Microsoft Excel 2013.

1.10. FINANCIAMIENTO

Autofinanciado	S/.20,350.00
TOTAL	S/.20,350.00

2. PLANTEAMIENTO DEL PROBLEMA

2.1. REALIDAD PROBLEMÁTICA

La programación de videojuegos es un mundo por explorar donde los desarrolladores ponen en competencia sus habilidades, destrezas y aún más la creatividad, permitiendo desarrollar nuevas metodologías de desarrollo acorde al tipo de videojuegos que se desea desarrollar. Si existe una función de los móviles que se equipare a su tarea primaria de comunicarnos, esa es la de entretenernos. En dicho campo, junto con la música, la fotografía y el video, aparecen los videojuegos.

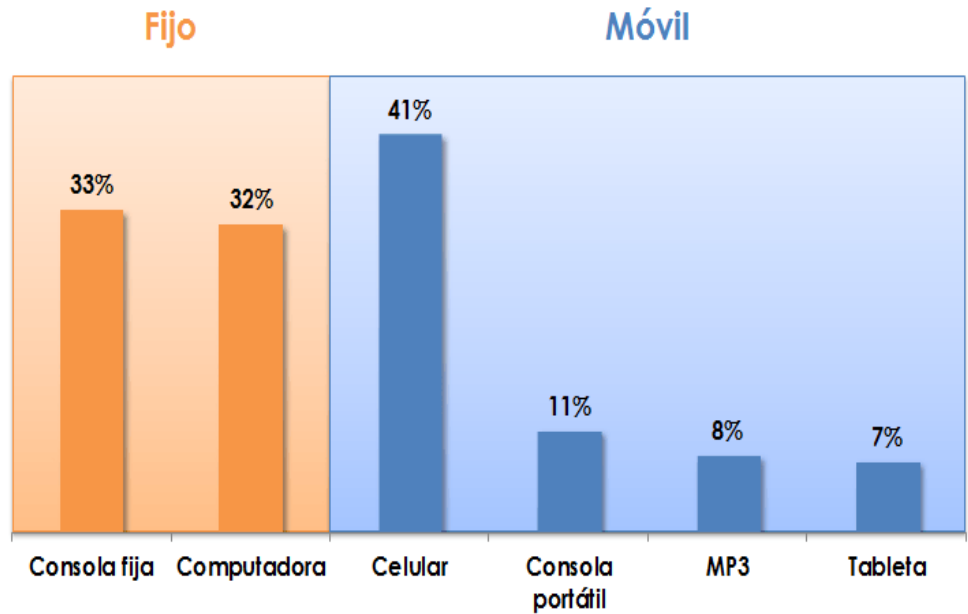
Sánchez (2013) sostiene que:

la consultora App Annie, realizó una investigación que reveló que durante el 2013, combinando las descargas de iOS y Android, superaron la cifra de ingresos para consolas portátiles. Además, compañías como Nintendo están contribuyendo a este cambio. Super Mario, Zelda o Donkey Kong estarán presentes en los móviles con juegos diseñados específicamente para esta plataforma.

Monzón (2015) sostiene que:

entre los dispositivos de acceso, se muestra un descenso en la preferencia por plataformas fijas, ya que 32% de los usuarios acostumbran jugar en consolas fijas. Por su parte, los celulares son el punto de acceso que presenta mayor preferencia, siendo usados por 41% de los video jugadores. Las tabletas si bien todavía no muestran ser un dispositivo popular en términos de acceso, se espera que tengan un elevado crecimiento, ya que muestran una preferencia de acceso similar a las consolas portátiles, dispositivos creados principalmente para el propósito de juego.

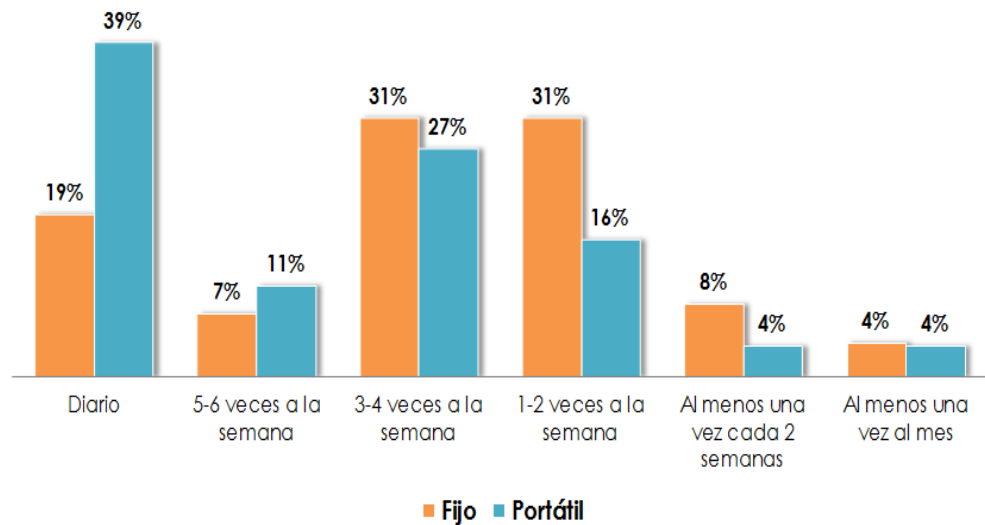
Figura 2. Principales Medios de Juego



Fuente: Elaborado por The Competitive Intelligence Unit (2013)

Con respecto a los perfiles de los video jugadores podemos distinguir que las personas que cuentan con una consola de videojuegos fija son jugadores intensivos; es decir, compran juegos regularmente, juegan a diario, usualmente dos horas o más, y hacen uso de la conexión a Internet disponible en sus dispositivos. No obstante, la posibilidad de transporte de las consolas portátiles permite una mayor intensidad de uso diario que las consolas fijas. Asimismo, se han convertido en el medio portátil de acceso al juego en línea, que permite realizar funciones adicionales, como el acceso a Internet, la descarga de juegos, entre otras características innovadoras, situación que alimenta la frecuencia de uso de estos dispositivos.

Figura 3. Frecuencia de Juego



Fuente: Elaborado por The Competitive Intelligence Unit (2013)

Conforme nos acercamos al lanzamiento de las nuevas plataformas de videojuegos, nos referimos a las consolas Xbox One de Microsoft y PlayStation 4 (PS4) del fabricante Sony, conviene hacer un breve repaso de la disposición de pago que tienen los video jugadores de consolas fijas, de manera que los fabricantes y distribuidores conozcan el equilibrio de precios necesario para la colocación de estos productos en el mercado. Al respecto, el precio de lanzamiento de la consola Xbox One se perfila en \$8,499 pesos, mientras que para el PS4 el precio final se fija en \$7,499 pesos. Por el lado de los usuarios tenemos que 7 de cada 10 están dispuestos a pagar \$5,999 o menos por consolas fijas, mientras que sólo 17% muestran disposición a erogar los recursos necesarios para adquirir las consolas referidas, es decir, apenas 2 de 10 están dispuestos a pagar \$7000 pesos o más por estas consolas. Cabe mencionar que dado el avance tecnológico y su consiguiente reducción de costos de producción, entre otros factores; es posible que los precios de oferta de estos dispositivos muestren un descenso conforme transcurren los meses después del lanzamiento, alcanzando un equilibrio de precios entre usuarios y fabricantes de consolas. Con ello, los usuarios se mostrarían más dispuestos a su adopción, incrementando a la vez su participación en el mercado de consolas fijas.

Figura 4. Disposición de Pago



Fuente: Elaborado por The Competitive Intelligence Unit (2013)

2.2. FORMULACIÓN DEL PROBLEMA

¿Cómo diseñar videojuegos multiplataforma en móviles de modo ligero desde una aplicación móvil en Android usando corona SDK?

2.3. OBJETIVOS

2.3.1. Objetivo General

Desarrollar una aplicación móvil en Android para el diseño de videojuegos multiplataforma en móviles de modo ligero usando corona SDK.

2.3.2. Objetivos Específicos

- Definir las propiedades de diseño a implementar.
- Definir el contenido teórico y los ejemplos prácticos de interacción (aplicación-usuario).
- Generar código fuente del diseño del videojuego y permitir exportar por medio del bluetooth.
- Implementar una interfaz de usuario amigable y de fácil uso de la aplicación.

2.4. JUSTIFICACIÓN

Conocer las herramientas que se emplean en el proceso de desarrollo de videojuegos multiplataforma para móviles, enfocándose principalmente en el framework Corona SDK con el cual es posible construir videojuegos multiplataforma de manera rápida ya que usa un lenguaje de programación modular denominado LUA que es de fácil sintaxis.

Además se tiene un amplio campo laboral en el mercado, como desarrolladores de videojuegos y como interesados podrían ser los estudiantes de ingeniería de sistemas de la Universidad Nacional José María Arguedas quienes pueden dedicarse profesionalmente a esta rama de la ingeniería los cuales tienen la posibilidad de formar parte de las gigantescas empresas que se dedican a este tipo de trabajo, y también formar su propia empresa de desarrollo de videojuegos.

La aplicación móvil "GameDesign", que se desarrolló en el presente proyecto, permite diseñar videojuegos de manera rápida y sencilla, lo cual se demuestra en los resultados del mismo.

2.5. VIABILIDAD DE LA INVESTIGACIÓN

2.5.1. Viabilidad Técnica

Las herramientas necesarias que se usó para implementar la aplicación móvil son las siguientes:

Hardware:

- ✓ 1 laptop Asus core i5, sistema operativo Windows 8.1, procesador 1.60 GHz
- ✓ 1 tablet Advance de 9' pulgadas, sistema operativo Android 4.4.2

Software:

- ✓ Corona SDK v2015.2777
- ✓ Sublime Text 3
- ✓ jdk-8u20-windows-i586

Cada uno de los software están disponibles en las distintas direcciones web mencionadas en las referencias bibliográficas.

2.5.2. Viabilidad Económica

El presente proyecto es viable, ya que se cuenta con los recursos económicos suficientes para la adquisición de materiales e implementos tecnológicos y profesionales capacitados en el desarrollo de software.

2.5.3. Viabilidad Social

Los estudiantes de ingeniería de sistemas de la Universidad Nacional José María Arguedas o estudiantes de otras universidades con carreras similares, tienen la posibilidad usar la aplicación "GameDesign" para construir videojuegos multiplataforma de manera rápida y sencilla .

2.6. LIMITACIÓN DEL ESTUDIO

Corona SDK permite crear aplicaciones para las siguientes plataformas, como a continuación se mencionan:

- ✓ Android
- ✓ ios
- ✓ Windows Phone
- ✓ Kindle Fire
- ✓ Mac
- ✓ Windows Desktop

El presente proyecto permite diseñar videojuegos multiplataforma para móviles de modo ligero, específicamente para (Android y Windows Desktop). Se ha limitado construir solo para estas 2 plataformas porque son gratuitas, ya que la construcción para las demás plataformas se tiene que tener una licencia de paga, y no se cuenta con recursos suficientes.

3. MARCO TEÓRICO

3.1.1. Framework para la programación en dispositivos móviles

3.1.1.1. Introducción

Se identificó la existencia de diversos framework para el desarrollo de aplicaciones móviles, cada una de estas plataformas tienen sus respectivos lenguajes de programación y herramientas de desarrollo.

Si se quiere desarrollar una aplicación móvil necesariamente se tiene que tener en cuenta para qué sistema operativo móvil se desea desarrollar, porque existen framework de desarrollo que no son multiplataforma y se necesita el conocimiento de un tipo de lenguaje de programación en específico, lo cual no sería óptimo al momento de comercializar la aplicación ya que existen sistemas operativos móviles diferentes que más adelante se detallarán sus características.

Todo desarrollador de aplicaciones móviles se cuestiona al momento de desarrollar si sería posible desarrollar aplicaciones multiplataforma y que no sea necesario saber de otros lenguajes de programación para desarrollar para una plataforma específica.

Aquí nace la necesidad de desarrollar en un framework multiplataforma, que con un solo proyecto de software sea posible construir la aplicación para los distintos sistemas operativos móviles. A continuación se lista los frameworks multiplataforma:

- Adobe Air Mobile (Adobe)
- PhoneGap (Open source)
- Appcelerator (Titanium)
- Corona SDK (Anscamobile)
- Unity 3D

3.1.1.2. Sistemas operativos móviles

A continuación se mencionan algunos sistemas operativos móviles:

Android:

Orosco (2011) sostiene que:

Android es un sistema operativo orientado a dispositivos móviles, basado en una versión modificada del núcleo Linux. Inicialmente fue desarrollado por Android Inc., una pequeña empresa, que posteriormente fue comprada por Google; en la actualidad lo desarrollan los miembros de la Open Handset Alliance (liderada por Google).

Su presentación se realizó el 5 de noviembre de 2007 junto con la fundación Open Handset Alliance, en un consorcio de numerosas compañías de hardware, software y telecomunicaciones comprometidas con la promoción de estándares abiertos para dispositivos móviles.

El android se trata de un sistema abierto, multitarea, que permite a los desarrolladores acceder a las funcionalidades principales del dispositivo mediante aplicaciones, cualquier aplicación puede ser reemplazada libremente, además desarrollarlas por terceros, a través de herramientas proporcionadas por Google, y mediante los lenguajes de programación Java y C.

iOS:

Ortega (2015) define que:

iOS es el sistema operativo que da vida a dispositivos como el iPhone, el iPad, el iPod Touch o el Apple TV. Su simplicidad y optimización son sus pilares para que millones de usuarios se decanten por iOS en lugar de escoger otras plataformas que necesitan más hardware para mover con fluidez el sistema. Cada año, Apple lanza una gran actualización de iOS que suele traer características exclusivas para los dispositivos más punteros que estén a la venta en ese momento.

BlackBerry OS:

Alegsa (2010) define que:

El BlackBerry OS es un sistema operativo para móviles desarrollado por Research In Motion (RIC) para su línea de smartphone BlackBerry.

La plataforma BlackBerry es muy conocida por su soporte nativo al email corporativo a través de MIDP, que permite activación inalámbrica completa y sincronización con Microsoft Exchange, Lotus Domin, o Novell GroupWise.

Desarrolladores ajenos a la empresa pueden escribir software empleando las clases BlackBerry API.

Windows Phone:

Orosco (2014) sostiene que:

Windows Phone es un sistema operativo móvil desarrollado por la empresa Microsoft para teléfonos inteligentes y otros dispositivos móviles. Fue lanzado al mercado el 21 de octubre de 2010 en Europa y el 8 de noviembre en Estados Unidos, con la finalidad de suplantar el conocido Windows Mobile.

Microsoft decidió realizar un cambio completo en este nuevo sistema operativo con respecto al otro, no solo se cambió el nombre, sino que se desarrolló desde cero, presentando una interfaz completamente nueva, mejor comportamiento y un mayor control sobre las plataformas de hardware que lo ejecutan, todo con el propósito de volver a ser competitivo en el mundo de los móviles.

Symbian:

Obando (2010) sostiene que:

Es un sistema operativo abierto y estándar para dispositivos de telefonía móvil. Symbian OS es actualmente un sistema operativo multitarea de 32 bits basado en ROM con una arquitectura de microkernel altamente modular que ofrece numerosas APIs (Application Programming Interfaces) para el desarrollo de aplicaciones de comunicaciones y soporta los principales estándares de la industria inalámbrica WAP, XHTML, J2ME, MIDP, MMS, Bluetooth, GPRS, CDMA, SyncML, IPv6, IPsec, etc.

Para la programación de aplicaciones se pueden utilizar distintos lenguajes: Visual Basic, Java, OPL y C++. Siendo este último el lenguaje nativo de Symbian y el que proporciona acceso a un mayor número de funcionalidades. Existen diferentes SDKs (Software Development Kit) para el desarrollo. El SDK proporciona las herramientas y la documentación necesarias para el desarrollo de aplicaciones en Symbian y un emulador del terminal móvil para PC.

Los distintos SDKs están ligados a diferentes plataformas. Cada una de estas plataformas proporciona una interfaz de usuario y un conjunto de aplicaciones del sistema para mensajería, telefonía, multimedia, agenda y otras tareas, que permite a los diferentes fabricantes personalizar sus entornos de desarrollo. Estas aplicaciones hacen uso de los motores de aplicación genéricos proporcionados por Symbian OS. Las principales plataformas existentes son UIQ, Nokia Serie 60 y Nokia Communicator.

Como podemos observar lo descrito en el punto 3.1.1.2 se definieron los conceptos de los distintos sistemas operativos móviles según algunos autores que se citan en las referencias bibliográficas.

3.1.2. Metodologías ágiles de desarrollo

Avison & Fitzgerald (2006) afirmaron que:

Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiaran a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo.

Salo & Teknillinen (2006) afirmaron que:

El uso de métodos ágiles de desarrollo de software ha recibido tanto apoyo como argumentos opuestos. El principal argumento en contra es la falta de validación. Hay también una cierta cantidad de incertidumbre para distinguir los métodos ágiles de programación ad-hoc. Sin embargo, algunos autores defienden la teoría de que los métodos ágiles proporcionan un enfoque de desarrollo organizado.

A continuación se muestran algunas metodologías ágiles para el desarrollo de aplicaciones móviles:

Extreme Programming (XP)

Beck y Zapata (2002) afirmaron que:

Se centra en las mejores prácticas para el desarrollo de software. Consta de doce prácticas: el juego de planificación, pequeñas emisiones, la metáfora, el diseño sencillo, las pruebas, la refactorización, la programación en parejas, la propiedad colectiva, integración continua, semana 40-h, los clientes en el lugar, y los estándares de codificación.

Beck y Andres (2004) afirmaron que:

La versión revisada “XP2” se compone de las siguientes prácticas “primarias”: sentarse juntos, equipo, espacio de trabajo informativo, el trabajo de energía, programación en parejas, las historias, el ciclo semanal, el ciclo trimestral, flujos de trabajo, construcción de 10 minutos, integración continua, prueba de programación y diseño incremental. También hay 11 prácticas corolario.

Scrum

Schwaber & Beedle (2002) afirmaron que:

Scrum se centra la gestión de proyectos en situaciones en las que es difícil planificar el futuro, con mecanismos de control “proceso empírico”, donde los bucles de realimentación constituyen el elemento central. El software es desarrollado por un equipo de auto-organización en incrementos (llamados “sprints”), empezando por la planificación y finalizando con un comentario. Las características que deben aplicarse en el sistema se registran en un backlog. Entonces, el dueño del producto decide qué elementos del backlog se deben desarrollar en el sprint siguiente. Los miembros del equipo coordinan su trabajo en un diario de stand-up de la reunión. Un miembro del equipo, el “Scrum Master” (equivalente al gerente del proyecto), es el encargado de resolver los problemas que impiden que el equipo trabaje eficazmente.

Rising & Janoff (2002) afirmaron que:

Scrum es una metodología que goza de gran popularidad y de la cual podemos encontrar bastantes empresas que gustan de utilizarla, entre ellas podemos encontrar empresas tan importantes como Yahoo o Google. En el año 2000 Rising y Janoff publicaron tres proyectos exitosos utilizando Scrum. (pp. 26-32).

Mobile-D

Kaleel & Harishankar (2013) afirmaron que:

Mobile-D consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. Cada una de estas fases tiene un número de etapas, tareas y prácticas asociadas.

En la primera fase, **exploración**, el equipo de desarrollo debe generar un plan y establecer las características del proyecto. Esto se realiza en tres etapas: establecimiento actores, definición del alcance y el establecimiento de proyectos. Las tareas asociadas a esta fase incluyen el establecimiento del cliente (los clientes que toman parte activa en el proceso de desarrollo), la planificación inicial del proyecto y los requisitos de recogida, y el establecimiento de procesos.

En la siguiente fase, **iniciación**, los desarrolladores preparan e identifican todos los recursos necesarios. Se preparan los planes para las siguientes fases y se establece el entorno técnico como los recursos físicos, tecnológicos y de comunicaciones (incluyendo el entrenamiento del equipo de desarrollo). Esta fase se divide en cuatro etapas: la puesta en marcha del proyecto, la planificación inicial, el día de prueba y día de salida.

En la fase de **producción** se repite la programación de tres días (planificación, trabajo, liberación) se repite iterativamente hasta implementar todas las funcionalidades. Primero se planifica la iteración de trabajo en términos de requisitos y tareas a realizar. Se preparan las pruebas de la iteración de antemano. Las tareas se llevarán a cabo durante el día de trabajo, desarrollando e integrando el código con los repositorios existentes. Durante el último día se lleva a cabo la integración del sistema (en caso de que estuvieran trabajando varios equipos de forma independiente) seguida de las pruebas de aceptación.

En la fase de **estabilización**, se llevan a cabo las últimas acciones de integración para asegurar que el sistema completo funciona correctamente. Esta será la fase más importante en los proyecto multi-equipo con diferentes subsistemas desarrollados por equipos distintos. En esta fase, los desarrolladores realizarán tareas similares a las que debían desplegar en la fase de “producción”, aunque en este caso todo el esfuerzo se dirige a la integración del sistema. Adicionalmente se puede considerar en esta fase la producción de documentación.

La última fase **pruebas del sistema**, tiene como meta la disponibilidad de una versión estable y plenamente funcional del sistema. El producto terminado e integrado se prueba con los requisitos de cliente y se eliminan todos los defectos encontrados.

3.1.3. Introducción al Framework Corona SDK

3.1.3.1. Características

De acuerdo a Corona SDK (2014) afirma que:

Para crear aplicaciones en Corona SDK se necesita instalar la aplicación Corona SDK en entorno Mac o Windows y utilizar un editor de texto para generar los archivos con el código. La versión de prueba de Corona SDK nos permite realizar pruebas con el simulador en dispositivos Android o iOS (solo en equipos Mac). Para construir aplicaciones y distribuirlas en la AppStore o en Android Market es necesario comprar una licencia.

Para crear un proyecto en Corona es necesario como mínimo una carpeta que un archivo de texto llamado "main.lua". Este archivo, es el primer archivo que lee Corona SDK por lo que si no está disponible la aplicación no puede iniciarse. Este archivo principal, a su vez, puede cargar otros archivos de código externo, o recursos de otros programas, tales como sonidos, imágenes o videos. La extensión de archivo ".lua" indica que el archivo está en lenguaje Lua, que es el que se usa para crear aplicaciones en Corona SDK.

De cara a la construcción de la aplicación, en la carpeta del proyecto existen 2 posibilidades de configuración mediante ficheros. El primero es el fichero config.lua con las dimensiones de contenido visible y el modo de escalado de la pantalla. El segundo fichero que podemos incluir es el denominado build.settings que describe las propiedades en tiempo de construcción.

3.1.3.2. Lenguaje de programación de Corona SDK: Lua

Lerusalimschy, Figueiredo y Celes (2008) mencionan que:

Lua es un lenguaje de programación de extensión diseñado para apoyar la programación general de procedimiento con instalaciones de descripción de datos. Lua también ofrece un buen soporte para la programación orientada a objetos, programación funcional y programación basada en datos. Lua está destinado a ser utilizado como un potente, ligero, lenguaje de script integrable para cualquier programa que necesita uno. Lua está implementado como una biblioteca, escrita en C limpio, el subconjunto común de estándar de C y C ++.

Como un lenguaje de extensión, Lua no tiene noción de un programa "principal": Sólo funciona embebido en un cliente anfitrión, llamado el programa de incorporación o simplemente el host. El programa de acogida puede invocar funciones para ejecutar un trozo de código Lua, puede escribir y leer variables de Lua y puede registrar funciones C para ser llamados por el código Lua. Mediante el uso de funciones de C, Lua puede ser aumentada para hacer frente a una amplia gama de diferentes dominios, creando así lenguajes de programación personalizados que comparten un marco sintáctico. La distribución de Lua incluye un programa anfitrión de muestra denominado lua que utiliza la biblioteca de Lua para ofrecer un completo, autónomo intérprete de Lua, para uso interactivo o por lotes.

Lo descrito en el punto 3.1.3.2, hace referencia al lenguaje de programación LUA, similar a javascript y python, para poder desarrollar aplicaciones bajo este lenguaje, es necesario conocer las teorías que Lerusalimschy et. al. (2008) sostiene:

Variables

Las variables son lugares donde se almacenan valores. Existen tres tipos de variables en Lua: globales, locales y campos de tabla. Antes de la primera asignación, el valor de una variable es nil.

Variables globales

Lua asume que las variables son globales, a no ser que sean declaradas explícitamente como locales. Las variables globales aun funcionan en la aplicación lo cual consume memoria, si se quiere eliminar de memoria una variable global, se le debe de asignar un valor "nil".

Variables locales

Las variables locales se definen anteponiendo la palabra local mientras que en variables globales no es necesario. Una variable local, solo puede ser visible en cualquier momento que se desee, pero si son declaradas dentro de una función, solo va ser posible acceder a esta dentro de la función.

Campos de Tabla

Los campos de tabla son elementos que componen la tabla. Los elementos de un campo de tabla son valores indexados en un array. Los campos de tabla pueden almacenar valores combinados, como string o numéricos.

Operadores aritméticos

Lua tiene los siguientes operadores matemáticos:

- +(adición)
- -(substracción)
- *(multiplicación)
- /(división)
- %(módulo)
- ^(exponenciación)
- ~ (negación)

Operadores relacionales

Lua tiene los siguientes operadores relacionales:

- ==(igualdad)
- ~= (no igualdad)
- <(menor que)
- >(mayor que)
- <=(menor o igual que)
- >=(mayor o igual que)

Devuelven un resultado TRUE o FALSE.

Operadores lógicos

Los operadores lógicos en LUA son:

- and (conjunción)
- or (disyunción)
- not (negación)

Como las estructuras de control todos los operadores lógicos consideran false y nil con falso y todo lo demás como verdadero.

Concatenación

El operador de concatenación es posible con dos puntos seguidos (..). Si ambos operandos son números entonces se convierten a strings.

Lerusalimschy et.al. (2008), define las estructuras de control y funciones en Lua de la siguiente manera:

Estructura condicional if

Esta estructura permite ejecutar una operación lógica o booleana (TRUE o FALSE), según sea el caso, que si la operación resulta TRUE se itera el primer bloque, caso contrario si resulta FALSE salta a ELSE y se ejecuta el bloque que este contiene.

Estructura repetitiva while

Bucle o lazos repetitivos que se encargan de repetir un fragmento de código mientras se cumpla cierta condición. Si no se cumple la condición se salta al siguiente bloque de instrucciones y permitir la ejecución continua del programa.

Estructura for

A diferencia de los otros, este bucle usa una variable como índice, el cual tendrá un valor inicial que aumentara progresivamente una cantidad (step es el incremento o decremento, sino se lo coloca, por default step será 1).

Al igual que los otros, realizara una comprobación, si la variable no ha alcanzado un valor final continuara con la ejecución del boque de código, cuando la variable haya alcanzado o sobrepasado el valor final, este bucle se detendrá.

Funciones

Las funciones en Lua tienen entrada (entre paréntesis), la función realiza algunas tareas, y los resultados se pueden devolver.

Las funciones pueden ser variables, por lo que una tabla puede guardarlas como propiedades. Esto permite usos muy flexibles de una mesa. Se puede utilizar para agrupar lógicamente una familia de funciones en una tabla.

Lerusalimschy et. al. (2008), define los eventos de pantalla, animación, física y gestión de memoria de la siguiente manera:

En Corona. Están acostumbrados a desencadenar respuestas en su aplicación como un toque en la pantalla, la detección de un evento del sistema específico, la realización de un temporizador, colisión de dos cuerpos físicos, etc. Los eventos son detectados gracias a la (`addEventListener`), método de objeto que dirige Corona para realizar un seguimiento (escuchar) el evento asociado a una función de detector.

Runtime Eventos

Eventos en tiempo de ejecución se envían al oyente Runtime global. Estos eventos no están dirigidos a cualquier objeto particular; sino que se transmiten a todos los oyentes interesados. Eventos en tiempo de ejecución comunes incluyen los siguientes:

EnterFrame: Ocurren en el intervalo de fotogramas por segundo de la solicitud (30 o 60).

System: Se envían cuando se producen determinados eventos externos, por ejemplo, cuando el dispositivo tiene que suspender o salir de la aplicación.

Orientation: Se producen cuando la orientación del dispositivo cambia de vertical a horizontal o viceversa.

Eventos Locales

Eventos locales se envían a un solo oyente y no se transmiten a la ejecución global. Eventos locales comunes incluyen los siguientes:

Hit: Se producen cuando el usuario toca la pantalla del dispositivo.

Collision: Se producen cuando dos objetos de física colisionan.

Timer: Se producen cuando un corredor temporizador completa su duración.

Audio: Pueden ser enviados cuando un audio de archivos termina de reproducirse.

Animación y movimiento

La biblioteca de transición proporciona funciones y métodos para la transición / objetos de visualización tween o grupos de visualización durante un periodo específico de tiempo. Las operaciones de biblioteca incluyen:

- Posibilidad de pausar, reanudar o cancelar una animación (o todas las animaciones).

- Capacidad para etiquetar varias animaciones por su nombre y luego hacer pausa, reanudar o cancelar todas las animaciones que comparten la misma etiqueta.
- Posibilidad de pausar, reanudar o cancelar todas las transiciones de forma amplia-objeto. Por ejemplo, si ha aplicado un movimiento independiente y transición de fundido para un objeto esto le permite controlar tanto las transiciones a la vez.

Motor físico

Física se utilizan comúnmente para aplicaciones que implican una simulación de objetos que se mueven, chocan, e interactúan bajo diversas fuerzas físicas como la gravedad. Corona hace que sea muy fácil de agregar la física para sus aplicaciones, incluso si usted nunca ha trabajado con un motor antes. Mientras que el motor subyacente se construye alrededor de la Box2D popular, hemos tomado un enfoque de diseño diferente que elimina gran parte de la codificación que tradicionalmente se requiere.

Para trabajar con el motor de física Corona, comienza con objetos de visualización. Corona trata a las propiedades físicas del cuerpo como una extensión de sus objetos gráficos: cualquier objeto de visualización estándar, incluyendo imágenes, objetos vectoriales o sprites animados puede ser "hecho físico" y van a interactuar automáticamente con otros objetos en la simulación.

Gestión de la memoria

Los dispositivos móviles han limitado la memoria disponible para su uso, por lo que debe tenerse cuidado para asegurar que el consumo de memoria total de su aplicación se mantiene al mínimo.

Lua realiza la gestión de memoria automática. Esto significa que usted no tiene que preocuparse acerca de la asignación de memoria para los objetos nuevos. Tampoco hace falta memoria para liberar explícitamente cuando los objetos ya no son necesarios. Lua gestiona la memoria automáticamente ejecutando un recolector de basura de vez en cuando para recoger todos los objetos "muertos" (objetos que ya no son accesibles desde Lua). Toda la memoria utilizada por Lua está sujeta a la gestión automática. Sin embargo, le corresponde a usted para decirle Lua lo que debe considerar basura. Por ejemplo, cualquier cosa almacenada en una variable global no se considera basura, incluso si la aplicación no utiliza nunca de nuevo. Del mismo modo, cualquier cosa almacenada en una tabla o matriz no será considerada basura incluso una variable u objeto que fue declarado inicialmente en el ámbito local. En ambos casos, le corresponde a usted para asignar nil a estas

posiciones. Esto asegura que su correspondiente memoria no se bloquea y puede ser liberado por el recolector de basura.

3.1.3.3. Ciclo de vida de una aplicación móvil en Corona SDK

Toda aplicación en Corona SDK, inicia con el archivo main.lua, este archivo debe de incluir el siguiente código fuente, que se visualiza en la siguiente imagen:

Figura 5. Código fuente del archivo main.lua

```
local composer=require "composer"
composer.gotoScene( "scene1","fade",1000 )
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

La librería composer contiene un método gotoScene(), este método está encargado del manejo de escenas, como primer parámetro se debe poner el nombre del archivo sin la extensión .lua, el segundo parámetro es el tipo de animación para ir a la “escene1”, y por último el tercer parámetro es el tiempo que se demora la aplicación para pasar a la “escene1”, este fichero debe de contener las siguientes líneas de código que se muestran en la siguiente imagen:

Figura 6. Código fuente de una Escena vacía

```
local composer = require 'composer'
local scene    = composer.newScene()
-----
function create( )
end
function show( )
end
function hide( )
end
function destroy( )
end
-----
scene:addEventListener( 'create',scene)
scene:addEventListener( 'show',scene)
scene:addEventListener( 'hide',scene)
scene:addEventListener( 'destroy',scene)
return scene
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Como se puede apreciar en la imagen anterior existen 4 funciones, estas funciones dan vida a una escena en corona SDK y es aquí donde ya se puede empezar a programar cualquier, a continuación se detalla la descripción de cada una de las funciones mencionadas:

create (): Esta función se ejecuta cada vez que ingresamos a una nueva escena, aquí podemos llamar a las distintas funciones que tendrá la aplicación, por ejemplo: crear imágenes, crear personajes, iniciar la física del videojuego, cargar sonidos, etc.

show (): Esta función se ejecuta después de la función anterior, aquí se pueden agregar eventos de toques de pantalla y también timers que tendrá el videojuego.

hide (): Esta función se ejecuta una sola vez, cuando se pasa a una nueva escena que tendrá la aplicación, aquí se puede eliminar los elementos creados en la función create (), también parar todas las animaciones y timers declarados como locales.

destroy (): Esta función destruye la escena actual, se ejecuta al momento de ser creado la siguiente escena de la aplicación.

3.1.3.4. Librerías básicas usadas en videojuegos sencillos en Corona SDK

A continuación se detalla algunas librerías necesarias para crear un videojuego:

Composer: Librería usada para el manejo de escenas, ejemplo: gotoScene(), purgeScene(), removeScene(), loadScene().

Widget: Librería usada para crear newButton(), newPickerWheel(), newProgressView(), newSlider(), newSpinner(), newTabBar().

Physics: Librería usada para agregar física a un objeto, y sea afectado por la gravedad, también para poder detectar colisiones entre objetos.

Audio: Librería usada para el manejo del audio de la aplicación

Display: Librería para obtener propiedades de la aplicación y también crear objetos geométricos y textos.

Math: Librería usada para hacer cálculos matemáticos avanzados que son usados en el desarrollo de videojuegos.

Timer: Librería usada para crear y controlar tiempos (bucles, loop), en la aplicación móvil.

Transition: Librería usada para crear animaciones de transición de objetos o cuerpos físicos.

3.2. ESTADO DEL ARTE

Tras numerosas investigaciones que se realizó acerca de las distintas metodologías ágiles para el desarrollo de aplicaciones móviles, los desarrolladores determinaron diversas metodologías de desarrollo de acuerdo al tipo de aplicaciones móviles que desean desarrollar, a continuación se rescatan algunas definiciones, ventajas y desventajas de las metodologías ágiles más usadas en la actualidad por los desarrolladores.

Ahonen, Barret y Golding (2002) afirman que:

La metodología para el desarrollo de aplicaciones para móviles se fundamenta en la experiencia de investigaciones previas en aplicaciones móviles, la evaluación del potencial de éxito para servicios de tercera generación denominada 6 M, la ingeniería de software educativo con modelado orientado por objetos (ISE-OO), y principalmente en los valores de las metodologías ágiles. La 6 M's debe su nombre a los seis atributos que se miden para evaluar el éxito del servicio: Movement (Movimiento), Moment (Momento), Me (Yo), Multi-user (Multiusuario), Money (Dinero) y Machines (Máquinas).

Lo que se concluye del párrafo anterior es que esta metodología nueva, permite desarrollar de una manera más ambiciosa una aplicación móvil que no solo se vea el producto final sino las necesidades que posteriormente se podrían presentar, o sea que la aplicación que se desarrolle evolucione si es que en un futuro existe una tendencia, cualquier aplicación móvil que se desarrolle bajo la metodología 6 M se vería fortalecida ya que se adaptaría a los cambios que podrían presentarse.

Astels (2003) afirma que:

Desarrollo orientado a las pruebas, condiciona la mentalidad de los desarrolladores guiándolos a través del desarrollo y enfocándose en la calidad del producto final, es un estilo de desarrollo donde se mantiene un juego de pruebas del programador exhaustivo, ninguna parte del código pasa a producción a no ser que pase sus pruebas asociadas, se escriben primero las pruebas y estas determinan el código que se necesita escribir y debido a su radical planteamiento a la hora de escribir código, cambia drásticamente la mentalidad de cualquier equipo de desarrollo, generalmente agilizando los resultados y aumentando la calidad del sistema.

Queda en evidencia lo que sostiene D. Astels, que el desarrollo orientado a pruebas es importante para asegurar que nuestra aplicación móvil sea de calidad, y realizando pruebas constantes para poder identificar las deficiencias que aún tiene nuestra aplicación y así de esta manera agilizar su desarrollo.

4. INGENIERÍA DEL PROYECTO

4.1. ADMINISTRACIÓN DEL PROYECTO BASADO EN PMBOK

4.1.1. Plan de gestión de la integración

Tabla 2. *Plan de gestión de la integración*

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK		
PATROCINADOR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto		
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	Fecha	2015/08/31
REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	Fecha	2015/09/07
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	Fecha	2015/09/08
REVISIÓN (Correlativo)	DESCRIPCIÓN (REALIZADA POR) (Motivo de la revisión y entre paréntesis quien la realizó)	FECHA (de la revisión)	
01	Preparación de Acta de Constitución (Jhancarlo Florencio Silva Ochoa)	2015/09/07	
BREVE DESCRIPCIÓN DEL PRODUCTO O SERVICIO DEL PROYECTO (Características, funcionalidades, soporte entre otros)			
El presente proyecto tuvo como finalidad que la aplicación "GameDesign" permita diseñar videojuegos de manera rápida y sencilla para dispositivos móviles en plataforma Android y Windows Desktop, permitiendo al usuario iniciarse en el maravilloso mundo de creación de videojuegos.			
ALINEAMIENTO DEL PROYECTO			
1. OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN		2. PROPÓSITO DEL PROYECTO (Beneficios que tendrá la	

(A qué objetivo estratégico se alinea el proyecto)	organización una vez que el producto del proyecto esté operativo o sea entregado)
1.1 Desarrollar talleres dentro y fuera de la Universidad en temas de desarrollo de aplicaciones móviles.	2.1 Impulsar la creación de videojuegos 2D para dispositivos móviles. 2.2 Promover la investigación del desarrollo de videojuegos 2D para dispositivos móviles y aplicarlos en los estudios universitarios de los estudiantes de ingeniería de sistemas.
3. OBJETIVOS DEL PROYECTO (Principalmente en términos de costo, tiempo, alcance, calidad)	
3.1. Desarrollar una aplicación móvil en Android para el diseño de videojuegos multiplataforma en móviles de modo ligero, esta aplicación podrá permitir diseñar para 2 plataformas (Android y Windows Desktop). 3.2. Finalizar el proyecto en un máximo de 4 meses a partir de la fecha del Acta de Constitución del Proyecto.	
4. FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO (Componentes o características que deben cumplirse en el proyecto para considerarlo exitoso)	
4.1. Proceso de adquisición de equipamiento y software dentro de los plazos establecidos y las características requeridas. 4.2. Diseño de software acorde a los requerimientos de la parte interesada (stakeholder).	
5. REQUERIMIENTOS DE ALTO NIVEL (Principales condiciones y/o capacidades que debe cumplir el producto o servicio y la Gestión del Proyecto)	
5.1. La aplicación móvil se construyo bajo la metodología Mobil-D. 5.2. La aplicación móvil fue instalado en una tablet de 9 pulgadas y en un Smart pone ambos con sistema operativo móvil Android	
EXTENSIÓN Y ALCANCE DEL PROYECTO	
6. FASES DEL PROYECTO (Agrupamiento lógico de actividades relacionadas que usualmente culminan elaborando un entregable principal. Cada Fase se ejecutará como un proyecto. Al fin de fase se puede tomar la decisión de continuar o no con las siguientes fases)	7. PRINCIPALES ENTREGABLES (Un único y verificable producto, resultado o capacidad de realizar un servicio que debe ser elaborado para completar un proceso, una fase o un proyecto)
Fase I – Gestión del proyecto	Acta de constitución del proyecto. Identificación de los Stakeholder.

Fase II – Diseño	Definición de casos de uso de la aplicación móvil Definición de las interfaces que tendrá la aplicación móvil.
Fase III – Requerimiento de Adquisiciones	Adquisición de laptop, Tablet, Smartphone Adquisición de entorno de desarrollo (Corona SDK), sublime Text 3
Fase IV – Implementación	Módulos de la aplicación móvil (Mini juego, manual de usuario)
Fase V – Arranque del Sistema	Pruebas de la aplicación móvil.
Fase VI – Termino de Obra	Aplicación móvil completamente construida.
8. INTERESADOS CLAVE (Persona u organización que está activamente involucrado en el proyecto o cuyos intereses pueden ser afectados positiva o negativamente por la ejecución del proyecto o por el producto que elabora)	
<p>8.1 Estudiantes de ingeniería de sistemas de la Universidad Nacional José María o estudiantes de otras universidades con carreras iguales o similares.</p> <p>8.2 Analista de sistemas</p> <p>8.3 Programador</p>	
9. RIESGOS (Evento o condición incierta que, si ocurriese, tiene un efecto positivo o negativo sobre los objetivos del proyecto)	
<p>9.1 Demora en la adquisición de las herramientas físicas y lógicas necesarias.</p> <p>9.2 No contar con los recursos económicos suficientes.</p> <p>9.3 Retraso en la programación de la aplicación móvil.</p>	
10. HITOS PRINCIPALES DEL PROYECTO (Un evento significativo para el proyecto)	
<p>10.1 Aprobación del acta de constitución del proyecto</p> <p>10.2 Alcanzar todos los requerimientos funcionales que tendrá la aplicación móvil</p> <p>10.3 Pruebas y correcciones de la aplicación móvil</p>	
11. PRESUPUESTO DEL PROYECTO (La estimación aprobada para el proyecto o cualquier otro componente de la estructura de desglose de trabajo, u otra actividad del cronograma)	
El costo del proyecto será asumido en un 100% por el Jefe del Proyecto.	

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.2. Plan de gestión del alcance

Tabla 3. Plan de gestión del alcance

Nombre del proyecto:	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK
Preparado por:	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto
Fecha:	2015/09/10
1. Describir cómo será administrado el alcance del Proyecto: Lo primero a determinar fue, a quienes está dirigido el proyecto, a partir de allí se definió que el alcance del proyecto sería que la aplicación móvil será usada por los estudiantes de ingeniería de sistemas.	
2. Evaluar la estabilidad del alcance del proyecto (cómo manejar los cambios, la frecuencia e impacto de los mismos): Los cambios del proyecto deben ser evaluados y aprobados. El Jefe de Proyecto debe cuantificar el impacto y proveer alternativas de solución. Los cambios solicitados serán revisados en las reuniones semanales, se indica el estado de los mismos en la reunión siguiente a la que fue solicitado.	
3. ¿Cómo los cambios al alcance, serán identificados y clasificados? El Jefe del Proyecto, revisará la solicitud de cambios del alcance y hará una evaluación del mismo. El podrá requerir del solicitante información adicional.	
4. Describir cómo los cambios del alcance serán integrados al proyecto: Si el impacto del cambio no modifica la línea base del proyecto será aprobado por el Jefe del Proyecto, y se actualizará las líneas base y todos los planes del proyecto.	
5. Comentarios adicionales:	

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.3. Plan de gestión del tiempo

Tabla 4. Plan de gestión del tiempo

Nombre del proyecto:		Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK
Preparado por:		Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto
Fecha:		2015/09/15
Persona(s) autorizada(s) a solicitar cambio en cronograma:		
Nombre	Cargo	Ubicación
Jhancarlo Florencio Silva Ochoa	Jefe del Proyecto	Universidad Nacional José María Arguedas

Persona(s) que aprueba(n) requerimiento de cambio de cronograma:		
Nombre	Cargo	Ubicación
Jhancarlo Florencio Silva Ochoa	Jefe del Proyecto	Universidad Nacional José María Arguedas
Razones aceptables para cambios en cronograma del Proyecto (por ejemplo, retrasos debido a entrega de materiales o disponibilidad de personal; clima; adelantar el cumplimiento debido a término de fase o proceso, etc.): <ul style="list-style-type: none"> - Solicitud de cambio de alcance por parte del usuario. - Accidentes de trabajo. - Mal establecimiento de la secuencia de actividades. 		
Describir cómo calcular y reportar el impacto en el proyecto por el cambio en cronograma (tiempo, costo, calidad, etc.): <ul style="list-style-type: none"> - Indicar en el informe a la persona responsable del mismo y fecha de ocurrencia del problema. - Descripción del problema indicando el grado de urgencia. - Impacto del mismo sobre el proyecto (Costo, Calidad, tiempo y alcance). - Descripción de las alternativas de solución detallando el impacto en las diferentes áreas (costo, calidad, tiempo y alcance). 		
Describir cómo los cambios al cronograma serán administrados: La administración del Alcance se efectuará de la siguiente manera: <ul style="list-style-type: none"> - Designación de Responsabilidades : <ul style="list-style-type: none"> o Planificación (Jefe de Proyecto). o Ejecución (Jefe de Proyecto). o Seguimiento y Control (Jefe de Proyecto). 		

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.4. Plan de gestión de los costos

Tabla 5. Plan de gestión de los costos

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK		
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/15
REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/17
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/19
Persona(s) autorizada(s) a solicitar cambios en el costo:			
Nombre	Cargo	Ubicación	
Jhancarlo	Jefe del Proyecto	Universidad Nacional José	

Florencio Silva Ochoa		María Arguedas
Persona(s) que aprueba(n) requerimientos de cambios en costo contractual:		
Nombre	Cargo	Ubicación
Jhancarlo Florencio Silva Ochoa	Jefe del Proyecto	Universidad Nacional José María Arguedas
Persona(s) que aprueba(n) requerimiento de cambio de costo interno ofrecido:		
Jefe del Proyecto		
Razones aceptables para cambios en el Costo del Proyecto (por ejemplo: Aprobación de cambios en el alcance, incremento de costos en los materiales, etc.):		
<ul style="list-style-type: none"> - Adecuación en el Alcance del Proyecto (cambios) - Ampliaciones en el Alcance del Proyecto 		
Describir como calcular e informar el impacto en el proyecto por el cambio en el costo (tiempo, calidad, etc.):		
<ul style="list-style-type: none"> - Persona que solicita el cambio. - Descripción de las características de la situación que requiere una solicitud de cambio de costos. - Impacto del mismo sobre el proyecto (Costo, Calidad, tiempo y alcance). - Descripción de las alternativas de solución detallando el impacto en las diferentes áreas (costo, calidad, tiempo y alcance). - Recomendación en la selección de la alternativa de solución (Propuesta). 		

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.5. Plan de gestión de la calidad

Tabla 6. Plan de gestión de la calidad

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK		
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/25
REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/27
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/29

GESTIÓN DE CALIDAD DEL PROYECTO
(Descripción de cómo se van a aplicar los procesos de gestión de calidad del proyecto. Herramientas a emplear, normativas, reglamentos, responsables, áreas de aplicación, etc.)
PLANIFICAR LA CALIDAD
(Explicar y sustentar cómo se ha elaborado el presente plan de gestión de calidad)
El Plan de gestión de calidad describe cómo el equipo de dirección del proyecto implementará la política de calidad de la organización ejecutante. Es un componente o un plan subsidiario del plan para la dirección del proyecto. El plan de gestión de calidad proporciona entradas al plan general para la dirección del proyecto y aborda el control de calidad, el aseguramiento de la calidad y métodos de mejora continua de los procesos del proyecto. (PMBOK, 2008).
REALIZAR ASEGURAMIENTO DE CALIDAD
(Explicar cómo se va a realizar el proceso de aseguramiento de calidad)
Aseguramiento de Calidad. Es responsable el Jefe de Proyecto ejecutar el Aseguramiento de Calidad durante todo el Proyecto, revisa el Planeamiento de los procesos del proyecto contra lo ejecutado, plantea acciones preventivas o correctivas según sean necesario. Se informa semanalmente al Equipo del Proyecto.
REALIZAR CONTROL DE CALIDAD
(Explicar cómo se va a realizar el proceso de control de calidad)
Control de Calidad. El Jefe de Proyecto de la ejecución del Control de Calidad. Se revisan los entregables de los proyectos conforme se vayan presentando, se emiten las observaciones o conformidades en la reunión semanal de calidad. Se definen dos procedimientos para el Control de Calidad:
- Revisión de Contenidos y
- Revisión de Forma.
Control de Calidad del Contenido: Se revisan la calidad de los entregables conjuntamente con el equipo de expertos.
Control de Calidad de Redacción y Formato: Se revisan la redacción de los documentos entregables, las observaciones que se puedan formular.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.6. Plan de gestión de los recursos humanos

Tabla 7. *Plan de gestión de los recursos humanos*

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK		
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/02

REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/04	
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/06	
ROL EN EL PROYECTO	Jefe del proyecto			
FUNCIÓN/RESPONSABILIDAD PRINCIPAL (Descripción de lo que se espera de éste rol)				
<ul style="list-style-type: none"> - El Jefe del Proyecto es responsable de la ejecución y entrega del proyecto. - Es el responsable de la definición del Personal que integra el equipo de proyecto de su organización y gestión. - Controla la información de gastos del proyecto. 				
COMPETENCIAS REQUERIDAS / RESPONSABILIDADES (Conocimientos, habilidades, actitudes)				
DISPONIBILIDAD (Posibilidad de contar con el recurso humano en el tiempo requerido)				
Descripción del puesto a requerir	Externo interno	Fecha ingreso	Fecha retiro	Tiempo requerido
Jefe del proyecto	interno	2015/08/25	2015/12/22	4 meses
INCORPORACIÓN AL PROYECTO (Cómo se incorpora el personal al proyecto)				
Evento disparador (Hecho que identifica la necesidad de hacer gestiones sobre el personal)		Sincronización prevista (Acciones requeridas para gestionar al personal)		
Renuncia		Inmediato con la designación de reemplazo por el Jefe del Proyecto		
Enfermedad		Inmediato con la designación de reemplazo por el Jefe del Proyecto		

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.7. Plan de gestión de las comunicaciones

Tabla 8. *Plan de gestión de las comunicaciones*

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK			
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/07	
REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/09/08	
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/09	
GESTIÓN DE COMUNICACIONES DEL PROYECTO				

(descripción de cómo se van a aplicar los procesos de gestión de comunicaciones del proyecto. herramientas a emplear, responsables, áreas de aplicación, etc.)

PLANIFICACIÓN DE LA COMUNICACIÓN

(explicar y sustentar cómo se ha elaborado el presente plan de gestión de comunicación)

Determina las necesidades e informaciones y comunicación de los interesados: quién necesita qué información, para cuando la necesita, cómo le será suministrada y por quién. El Jefe del Proyecto debe considerar la cantidad de canales necesarios desde un principio con el fin de que la información fluya y llegue a todos los involucrados.

Los requisitos de comunicación incluyen:

- Organigramas.
- Relaciones de responsabilidades de la organización.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.8. Plan de gestión de los riesgos

Tabla 9. *Plan de gestión de los riesgos*

Nombre del Proyecto:	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK
Preparado por:	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto
Fecha:	2015/10/10
Descripción de la metodología de gestión del riesgo a ser usada:	
Alcances	
- La identificación, priorización y seguimiento de riesgos más críticos será realizado por el Jefe de Proyecto.	
Herramientas	
- Tormenta de ideas.	
Fuentes de Datos	
- La identificación de todos los riesgos será por parte de todos los integrantes del proyecto según experiencia y juicio de especialistas.	
Roles y responsabilidades:	
- Jefe de Proyecto: Responsable de identificación, priorización y seguimiento de riesgos, proponer acciones para afrontar los riesgos identificados.	
- Analista de sistemas: Responsable de determinar la funcionalidad que tendrá la aplicación móvil.	
- Programador: Responsable de la programación e integración de la aplicación móvil.	

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.9. Plan de gestión de las adquisiciones

Recursos para la adquisición

(Personal involucrado en la gestión de las adquisiciones)

Para este proyecto los responsables de realizar las labores de compra y contratación son:

- El Jefe del Proyecto en la parte de subcontratos (analista, programador).
- El Jefe del Proyecto en la parte de adquisición de equipos físicos, y herramientas de desarrollo.

En el presente proyecto el Jefe de Proyecto, analista y programador fue responsable de una sola persona.

Productos y servicios contratados

(Describir los productos o servicios que el proyecto contrato indicando las restricciones, supuestos y límites que tendrán.)

- Servicio de Programación de la aplicación móvil .
- Servicio de proveedor para compra de equipos físicos.

Procedimientos para la gestión de adquisiciones

Describir los procedimientos que se usarán en el proyecto y que procedimientos usó el comprador.

Tabla 10. Plan de gestión de las adquisiciones

Nº	PROCEDIMIENTO EN EL PROYECTO PARA SOLICITUD DE COMPRA	PROCEDIMIENTO EN EL ÁREA LOGÍSTICA PARA REALIZAR ORDEN DE COMPRA
1	Este documento describe el producto pedido, cantidad en unidades a pedir. Luego de evaluar el comprador de la empresa a los proveedores regresa una plantilla para tomar la decisión técnica y económica del proveedor que mejor se ajuste a los requerimientos del proyecto.	Se elaboro la orden de compra correspondiente. Este documento describe claramente la fecha, nombre completo del proveedor, persona de contacto si es posible, número de teléfono, descripción completa del producto a pedir, cantidad en unidades a pedir, precio y monto total.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.1.10. Plan de gestión de los interesados

Tabla 11. *Plan de gestión de los interesados*

PROYECTO	Diseño ligero de videojuegos multiplataforma en móviles con Corona SDK		
PREPARADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/10
REVISADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/11
APROBADO POR	Jhancarlo Florencio Silva Ochoa-Jefe del Proyecto	fecha	2015/10/12

Nombres y apellidos	organización	cargo	Información de contacto	Requerimientos sobre el producto	influencia					Influencia sobre	Tipo de interés
					I	P	E	S	C		
Jhancarlo Florencio Silva Ochoa	UNAJMA	Jefe del Proyecto	jhanca_s_o2011@hotmail.com	Encargado de brindar la buena viabilidad y disponibilidad de recursos que se necesita en el proyecto.	F	F	F	F	F	Presente en todas las fases.	Control del Proceso.

Jhancarlo Florencio Silva Ochoa	UNAJMA	Analista de Sistemas	jhanca_s_o2011@hotmail.com	Encargado en diseñar la infraestructura de la aplicación, teniendo en cuenta los requisitos funcionales, alcance y limitaciones que la aplicación debe tener.		F	F		F	Fase de planificación, ejecución, cierre	Planificar la infraestructura de la aplicación, y dar solución a los problemas que se podrían presentar.
Jhancarlo Florencio Silva Ochoa	UNAJMA	Programador de Sistemas	jhanca_s_o2011@hotmail.com	Encargado de la implementación y pruebas de la aplicación donde la infraestructura de esta, que es diseñada anteriormente por el analista de sistemas.		F	F		F	Fase de planificación, ejecución, cierre	Cumplir con la programación de la aplicación móvil en el cronograma establecido.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Influencia I: Inicio; P: Planificación; E: Ejecución; S: Supervisión y Control; C: Cierre F: Favorable; C: Contraria / A: Alta; R: Regular; B: Baja.

4.2. METODOLOGÍA Y PLANIFICACIÓN DEL PROYECTO

La metodología de desarrollo que se empleó en el presente proyecto fue Mobile-D, el cual consta de cinco fases: exploración, iniciación, producción, estabilización y prueba del sistema. La descripción breve de cada uno de las fases se referencia en las metodologías de desarrollo de aplicaciones móviles en el presente documento, así que no se entrará en detalles.

En la siguiente figura se muestra el ciclo de vida del software con la metodología Mobile-D:

Figura 7. Ciclo de vida de Software: Mobile-D



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Previamente al desarrollo de software con esta metodología, se realizó la definición del objetivo del proyecto, la elección de la plataforma de desarrollo y el esbozo de los requisitos de la aplicación.

El desarrollo de la aplicación didáctica está disponible para que se pueda instalar en dispositivos móviles con sistema operativo Android, en los Smartphone y Tablets, las pruebas y seguimiento de la aplicación se hizo en una Tablet de 9 pulgadas con sistema Android y en un Smartphone también con sistema Android.

A continuación se detalla los requisitos funcionales y no funcionales que tiene la aplicación:

4.2.1. Requisitos funcionales

- Aplicación multi – resolución para Smartphone y Tablets.
- La estructura de la aplicación debe construirse para una posible actualización futura.

4.2.2. Requisitos no funcionales

- El Framework que se empleó fue Corona SDK.
- El usuario debe ser capaz de interactuar con la aplicación intuitivamente y sin ninguna dificultad.
- Aplicación disponible para dispositivos móviles con sistema operativo Android y Windows Desktop.

4.3. PUESTA EN MARCHA LA METODOLOGÍA MOBILE-D

4.3.1. Fase de Exploración

Busca establecer un plan de proyecto , identificar a los clientes e interesados del proyecto. Además, deben identificarse claramente cuál será el nombre comercial del proyecto, objetivo general, tipo de aplicación y la plataforma para la cual se desarrolló el proyecto.

4.3.1.1. Estableciendo actores

Identificar los actores y el equipo de desarrollo para el presente proyecto, es indispensable, a continuación se lista la relación de los actores que participaron de forma activa durante el desarrollo de software:

- **Director del proyecto:** encargado de brindar la buena viabilidad y disponibilidad de recursos que se necesita en el proyecto.
- **Analista de sistemas:** encargado en diseñar la infraestructura de la aplicación, teniendo en cuenta los requisitos funcionales, alcance y limitaciones que la aplicación debe tener.
- **Programador:** encargado de la implementación y pruebas de la aplicación donde la infraestructura de esta, es diseñada anteriormente por el analista de sistemas.

El presente proyecto de tesis fue desarrollado por una sola persona (tesista), en el periodo establecido, cumpliendo éste las tres responsabilidades mencionadas anteriormente.

4.3.1.2. Estableciendo Usuarios

Identificar los beneficiarios, que serán las personas que harán uso de la aplicación móvil, a continuación se define:

- **Usuarios directos:** Estudiantes de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas.
- **Usuarios indirectos:** Estudiantes y docentes de Ingeniería de Sistemas o carreras afines de otras Universidades Nacionales o Privadas.

Información clave del proyecto

A continuación se determinó el nombre comercial, objetivo general, tipo de aplicación y la plataforma para el cual se desarrolló el proyecto.

Tabla 12. Información clave del proyecto

INFORMACIÓN CLAVE DEL PROYECTO	
NOMBRE COMERCIAL	GameDesign
OBJETIVO GENERAL	diseñar videojuegos multiplataforma (Android y Windows Desktop)
TIPO DE APLICACIÓN	Educativo-instructivo
PLATAFORMA	Android y Windows Desktop

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.3.2. Fase de Iniciación

Definir las herramientas (de desarrollo, físicas, para pruebas), diagramas de caso de uso. A continuación, se detalla las distintas herramientas y recursos necesarios que se usó para el desarrollo de la aplicación móvil:

4.3.2.1. Software

- Corona SDK v2015.2777
- Sublime Text 3
- jdk-8u20-windows-i586

4.3.2.2. Hardware

- 1 laptop Asus core i5, sistema operativo Windows 8.1, procesador 1.60 GHz

4.3.2.3. Herramientas para pruebas (Hardware)

- 1 tablet Advance de 9' pulgadas, sistema operativo Android 4.4.2
- 1 Smartphone con sistema operativo Android 4.4.2
- 1 laptop Asus core i5, sistema operativo Windows 8.1

4.3.2.4. Diagrama de casos de uso de la aplicación móvil

- **Escena de carga:** La escena tiene un logotipo de la Universidad Nacional José María Arguedas. En esta escena no hay acciones que realizar, de manera automática la escena de carga da paso a la escena de bienvenida con un tiempo de 2 segundos, tiempo que se dio en la programación de la aplicación.
- **Escena de Bienvenida:** En esta escena se visualiza el título de la aplicación, "DISEÑA TU VIDEOJUEGO", para poder paso a la escena del menú principal de debe de presionar cualquier parte de la escena.

Tabla 13. Caso de uso: Iniciar aplicación

CASO DE USO:	Iniciar aplicación
ACTORES:	Usuario
PROPÓSITO:	Dar paso al menú principal de la aplicación móvil
RESUMEN:	El usuario permite a la aplicación dar paso al menú principal para poder interactuar en los diferentes capítulos.
TIPO:	Primario
PRE-CONDICIONES:	Debe de haber finalizado la carga de la escena Splash.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- **Escena del Menú principal:** Se visualiza las distintas opciones que tiene la aplicación móvil. Desde esta escena es posible acceder a cualquier opción de la aplicación.

Tabla 14. Caso de uso: Selección de capítulos

CASO DE USO:	Selección de capítulos
ACTORES:	Usuario
PROPÓSITO:	Navegar por los distintos capítulos de la aplicación móvil
RESUMEN:	El usuario podrá seleccionar cualquier capítulo del menú principal
TIPO:	Primario
PRE-CONDICIONES:	Caso de uso inmediato después de haber tocado la pantalla para ingresar al menú principal

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- **Escena Capitulo diapositivas:** Se visualiza uno de los 6 primeros capítulos seleccionables de la opción Introducción, esta escena contiene los botones que permiten avanzar de diapositiva o regresar al menú principal para poder elegir otro capítulo que se quiere visualizar.

Tabla 15. Caso de uso: seleccionar capitulo imágenes

CASO DE USO:	Seleccionar capitulo Imágenes
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de imágenes en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a imágenes.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 16. Caso de uso: Seleccionar capitulo texto

CASO DE USO:	Seleccionar capitulo Texto
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de textos en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a textos.

TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 17. Caso de uso: Seleccionar capítulo toques

CASO DE USO:	Seleccionar capítulo Toques
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de toques de escena en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a eventos de toque de escena.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 18. Caso de uso: Seleccionar capítulo sonido

CASO DE USO:	Seleccionar capítulo Sonido
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de sonidos en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a sonidos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 19. Caso de uso: Seleccionar capítulo animación

CASO DE USO:	Seleccionar capítulo Animación
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de animaciones en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a animaciones.

TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 20. Caso de uso: Seleccionar capítulo física

CASO DE USO:	Seleccionar capítulo Física
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de la física en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a animaciones.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 21. Caso de uso: Nuevo videojuego

CASO DE USO:	Nuevo videojuego
ACTORES:	Usuario
PROPÓSITO:	Crear un nuevo videojuego
RESUMEN:	Opción para poder crear un nuevo videojuego
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 22. Caso de uso: Abrir videojuego

CASO DE USO:	Abrir videojuego
ACTORES:	Usuario
PROPÓSITO:	Poder abrir un videojuego
RESUMEN:	Editar y seguir diseñando un videojuego creado anteriormente.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Escena de selección de Objetos(Actores y fondo) :** Esta escena contiene 4 botones en la parte superior los cuales al ser pulsados llaman a otras escenas respectivamente donde se encuentran los objetos que el usuario seleccione los cuales estarán disponibles en la escena principal de diseño, una vez seleccionado los objetos, el usuario debe de pulsar el botón ‘visualizar’ que se encontrará en la parte inferior para dar pase a la escena principal de diseño del videojuego. A continuación se detalla la secuencia de pasos :
 Antes de ingresar a la escena principal de diseño, se debe de seleccionar objetos los cuales se usará en la escena de diseño, son 4 categorías que se consideró:
Actores 1: En esta sección contiene a los personajes que estarán en el videojuego.
Actores 2: En esta sección contiene a los personajes de otra categoría que estarán en el videojuego.
Baldosas: Esta sección contiene los obstáculos que podría tener el videojuego.
Fondo: Aquí se puede seleccionar el fondo de pantalla que tendrá el videojuego.

Tabla 21. Caso de uso: Seleccionar Actores 1

CASO DE USO:	Seleccionar Actores 1
ACTORES:	Usuario
PROPÓSITO:	Seleccionar uno o más objetos
RESUMEN:	Los objetos seleccionados podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 22. Caso de uso: Seleccionar Actores 2

CASO DE USO:	Seleccionar Actores 2
ACTORES:	Usuario
PROPÓSITO:	Seleccionar uno o más objetos
RESUMEN:	Los objetos seleccionados podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario

PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
-------------------------	--

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 23. Caso de uso: Seleccionar baldosa

CASO DE USO:	Seleccionar baldosa
ACTORES:	Usuario
PROPÓSITO:	Seleccionar una o más baldosas
RESUMEN:	Las baldosas seleccionadas podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 24. Caso de uso: Seleccionar fondo

CASO DE USO:	Seleccionar fondo
ACTORES:	Usuario
PROPÓSITO:	Seleccionar imagen de fondo
RESUMEN:	El fondo seleccionado se podrá visualizar en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- **Escena principal de diseño:** En esta escena se puede diseñar videojuegos a gusto del usuario, como flappi birt, racer, space shooter, en esta escena se cuenta con 4 apartados, se detalla a continuación cada uno de ellos:
 - ✓ **Objetos:** lista desplegable (Actores 1, Actores 2, Baldosas), los cuales al ser pulsados se activan para poder crear estos objetos en la escena del videojuego.
 - ✓ **Propiedades:** lista desplegable, se consideró las siguientes propiedades para poder diseñar el videojuego:
 - ✓ **Física:** las propiedades de física son las siguientes:

- **Cuerpo dinámico:** el objeto va ser afectado por la gravedad.
- **Cuerpo estático:** el objeto permanece quieto pero si detecta colisiones.
- ✓ **Animación:** las propiedades de animación son las siguientes:
 - **Speed:** valor numérico para definir la velocidad de movimiento del objeto.
 - **Destino X:** valor numérico destino en el eje x.
 - **Destino Y:** valor numérico destino en el eje y.
 - **Rotación:** se puede definir el tipo de rotación(Horario o anti horario) durante la animación.
 - **Aleatorio:** punto aleatorio durante la animación en el eje 'x' o 'y'.
- ✓ **Objeto:** las propiedades de objeto son las siguientes:
 - **Ancho:** valor numérico para modificar el ancho del objeto.
 - **Altura:** valor numérico para modificar la altura del objeto.
 - **Rotación:** valor numérico para modificar la rotación del objeto.
 - **Control 1:** si se desea controlar al objeto en el eje 'x' o 'y'.
 - **Control 2:** si se desea controlar al objeto en cualquier punto de la escena (eje 'x' e 'y'), o sino darle un impulso en el eje 'y'.
 - **Eliminar:** si se desea eliminar un objeto de la escena.
- ✓ **Disparo:** las propiedades de disparo son las siguientes:
 - **Tiempo creación:** valor numérico para crear el disparo(bala), la creación de las balas será infinito.
 - **Destino X:** valor numérico destino en el eje x.
 - **Destino Y:** valor numérico destino en el eje y.
 - **Speed:** valor numérico que define la velocidad de animación del disparo.

- ✓ **Colisiones:** las propiedades de colisiones son las siguientes:
 - **Eliminar objeto:** si el objeto detecta colisión, este se elimina de la escena.
 - **Mantener objeto:** si el objeto detecta colisión, este permanece en la escena.

- ✓ **Gravedad:** las propiedades de gravedad son las siguientes:
 - **Gravedad X:** valor numérico para definir la gravedad que tendrá en videojuego en el eje x.
 - **Gravedad Y:** valor numérico para definir la gravedad que tendrá en videojuego en el eje y.
 - **DrawMode:** vista de los objetos creados, vista normal(vista igual al momento de crear los objetos), vista hybrid (vista avanzada, se podrá visualizar como los objetos son cuerpos geométricos).

- ✓ **Mapa:** las propiedades de mapa son las siguientes:
 - **Escenas Horizontal:** valor numérico para definir la cantidad de escenas horizontales que tendrá en videojuego.
 - **Escenas vertical:** valor numérico para definir la cantidad de escenas verticales que tendrá el videojuego.
 - **Orientación:** para definir la orientación de la animación del mapa si es horizontalmente o verticalmente.
 - **Speed:** valor numérico para definir la velocidad de animación del mapa.
 - **Audio:** para definir si el videojuego tendrá una música de fondo.

- ✓ **Opciones:** las propiedades de opciones son las siguientes:
 - **Ejecutar:** para visualizar el videojuego en ejecución, con las propiedades asignadas, es aquí cuando se guarda y genera el código fuente del videojuego.

- **Eliminar:** si se desea eliminar el videojuego actual(ojo no existe un backup).
- **Menú Principal:** si se desea retornar al menú principal de la aplicación móvil.
- ✓ **Ayuda:** se lista todas las propiedades y al pulsar cualquiera de ellos se visualiza información adicional, el cual es de ayuda acerca de la propiedad(el uso).
- ✓ **Escena diseño:** por defecto se tiene una única escena en la cual es aquí donde se crearan cada uno de los objetos que el usuario desee y también la asignación de las distintas propiedades mencionadas anteriormente.
- ✓ **Flechas direccionales:** para poder desplazarse a cualquier escena adicional que se haya definido (escenas horizontales o escenas verticales).
- ✓ **Escena Ejecución:** en esta escena se visualizara el videojuego en ejecución.

Tabla 25. Caso de uso: Listar Objetos

CASO DE USO:	Listar Objetos
ACTORES:	Usuario
PROPÓSITO:	Visualizar lista de Objetos
RESUMEN:	El usuario ya podrá crear objetos en la escena del diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado Actores 1, Actores 2 , Baldosas.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 26. Caso de uso: Agregar física

CASO DE USO:	Agregar física
ACTORES:	Usuario
PROPÓSITO:	Agregar física a los objetos
RESUMEN:	El usuario ya podrá agregar física a los objetos de la escena.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado algún objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 27. Caso de uso: Definir velocidad de animación

CASO DE USO:	Definir velocidad de animación
ACTORES:	Usuario
PROPÓSITO:	Definir la velocidad de animación
RESUMEN:	Posibilidad de definir la velocidad de animación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 28. Caso de uso: Definir destino 'x' de animación

CASO DE USO:	Definir destino 'x' de animación
ACTORES:	Usuario
PROPÓSITO:	Definir el destino 'x' de animación
RESUMEN:	Posibilidad de definir el destino 'x' de animación del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 29. Caso de uso: Definir destino 'y' de animación

CASO DE USO:	Definir destino 'y' de animación
ACTORES:	Usuario
PROPÓSITO:	Definir el destino 'y' de animación
RESUMEN:	Posibilidad de definir el destino 'y' de animación del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 30. Caso de uso: Definir rotación de animación

CASO DE USO:	Definir rotación de animación
ACTORES:	Usuario
PROPÓSITO:	Definir rotación de animación
RESUMEN:	Posibilidad de definir el sentido de rotación durante la animación.

TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 31. Caso de uso: Definir movimiento aleatorio de animación

CASO DE USO:	Definir movimiento aleatorio de animación
ACTORES:	Usuario
PROPÓSITO:	Definir movimiento aleatorio de animación del objeto.
RESUMEN:	Posibilidad de definir un movimiento aleatorio durante la animación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 32. Caso de uso: Definir ancho del objeto

CASO DE USO:	Definir ancho del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir el ancho del objeto
RESUMEN:	Posibilidad de definir el ancho del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 33. Caso de uso: Definir altura del objeto

CASO DE USO:	Definir altura del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir la altura del objeto
RESUMEN:	Posibilidad de definir la altura del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 34. *Caso de uso: Definir rotación del objeto*

CASO DE USO:	Definir rotación del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir la rotación del objeto
RESUMEN:	Posibilidad de definir la rotación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 35. *Caso de uso: Definir control 1*

CASO DE USO:	Definir control 1
ACTORES:	Usuario
PROPÓSITO:	Definir control 1 del objeto.
RESUMEN:	Posibilidad de definir el control del objeto en la pantalla.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 36. *Caso de uso: Definir control 2*

CASO DE USO:	Definir control 2
ACTORES:	Usuario
PROPÓSITO:	Definir control 2 del objeto.
RESUMEN:	Posibilidad de definir el control del objeto en la pantalla.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 37. *Caso de uso: eliminar objeto*

CASO DE USO:	Eliminar objeto
ACTORES:	Usuario
PROPÓSITO:	Eliminar objetos
RESUMEN:	Posibilidad de eliminar los objetos creados en la escena

TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 38. *Caso de uso: Definir tiempo creación disparo*

CASO DE USO:	Definir tiempo creación disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir el tiempo de creación
RESUMEN:	Posibilidad de definir cada cuanto tiempo se crea el disparo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 39. *Caso de uso: Definir destino 'x' disparo*

CASO DE USO:	Definir destino 'x' disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir destino 'x' disparo.
RESUMEN:	Posibilidad de definir el destino 'x' del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 40. *Caso de uso: Definir destino 'y' disparo*

CASO DE USO:	Definir destino 'y' disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir destino 'y' disparo.
RESUMEN:	Posibilidad de definir el destino 'y' del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 41. Caso de uso: Definir velocidad de animación disparo

CASO DE USO:	Definir velocidad de animación disparo
ACTORES:	Usuario
PROPÓSITO:	Definir velocidad de animación del disparo
RESUMEN:	Posibilidad de definir la velocidad de animación del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 42. Caso de uso: definir estado de colisión

CASO DE USO:	Definir estado de colisión
ACTORES:	Usuario
PROPÓSITO:	Definir el estado de colisión
RESUMEN:	Posibilidad de definir el estado de colisión del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 43. Caso de uso: Definir gravedad x

CASO DE USO:	Definir Gravedad x
ACTORES:	Usuario
PROPÓSITO:	Definir la gravedad en el eje x que tendrá la escena
RESUMEN:	Posibilidad de definir la gravedad en el eje x, puede variar de un valor positivo o negativo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la gravedad en el eje indicado.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 44. Caso de uso: Definir gravedad y

CASO DE USO:	Definir Gravedad y
ACTORES:	Usuario
PROPÓSITO:	Definir la gravedad en el eje 'y' que tendrá la escena
RESUMEN:	Posibilidad de definir la gravedad en el eje y, puede variar de un valor positivo o negativo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la gravedad en el eje indicado.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 45. Caso de uso: Definir vista

CASO DE USO:	Definir vista
ACTORES:	Usuario
PROPÓSITO:	Ver con detalle las colisiones
RESUMEN:	Los objetos creados en pantalla se podrán visualizar de distinta manera si se elige modo 'hybrid'.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la vista.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 46. Caso de uso: Definir escenas horizontales

CASO DE USO:	Definir escenas horizontales
ACTORES:	Usuario
PROPÓSITO:	Definir la cantidad de escenas horizontales.
RESUMEN:	Posibilidad de definir la cantidad de escenas horizontales que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 47. Caso de uso: Definir escenas verticales

CASO DE USO:	Definir escenas verticales
ACTORES:	Usuario
PROPÓSITO:	Definir la cantidad de escenas verticales.
RESUMEN:	Posibilidad de definir la cantidad de escenas verticales que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 48. Caso de uso: Definir orientación del mapa

CASO DE USO:	Definir orientación del mapa
ACTORES:	Usuario
PROPÓSITO:	Definir la orientación de animación del mapa.
RESUMEN:	Definir la orientación de la animación que tendrá el mapa del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado mas de una escena(horizontales o verticales)

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 49. Caso de uso: Definir velocidad de animación mapa

CASO DE USO:	Definir velocidad de animación mapa
ACTORES:	Usuario
PROPÓSITO:	Definir la velocidad de animación del mapa
RESUMEN:	Posibilidad de definir la velocidad de animación del mapa del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 50. Caso de uso: Definir audio del videojuego

CASO DE USO:	Definir audio del videojuego
ACTORES:	Usuario
PROPÓSITO:	Definir el audio del videojuego
RESUMEN:	Definir el audio que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 51. Caso de uso: Ejecutar videojuego

CASO DE USO:	Ejecutar videojuego
ACTORES:	Usuario
PROPÓSITO:	Ver la ejecución del videojuego diseñado.
RESUMEN:	Una vez creado los objetos en la escena y haber guardado sus respectivas propiedades y definir también propiedades de la escena, ahora se puede visualizar la ejecución del diseño del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder ejecutar el videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 52. Caso de uso: Eliminar videojuego

CASO DE USO:	Eliminar videojuego
ACTORES:	Usuario
PROPÓSITO:	Eliminar el videojuego diseñado.
RESUMEN:	Eliminar el videojuego que actualmente se esta diseñando.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un diseño de un videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 53. Caso de uso: Ir al menú principal

CASO DE USO:	Ir al menú principal
ACTORES:	Usuario
PROPÓSITO:	Ir al menú principal de la aplicación móvil
RESUMEN:	Ir al menú principal de la aplicación móvil, saliendo de la escena de diseño del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un diseño de un videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 54. Caso de uso: Ayuda

CASO DE USO:	Ayuda
ACTORES:	Usuario
PROPÓSITO:	Información disponible del uso de las propiedades
RESUMEN:	Información disponible del uso de cada uno de las propiedades que se le asigna a los objetos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción de ayuda.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 55. Caso de uso: Exportar videojuego

CASO DE USO:	Exportar videojuego
ACTORES:	Usuario
PROPÓSITO:	Exportar el código del videojuego
RESUMEN:	Exportar el videojuego por medio del bluetooth, para la construcción del .apk(Android) y .exe(Windows Desktop)
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos el diseño de un videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- **Escena Mini Juego:** En esta escena se podrá observar un videojuego tipo geometry dash .Este juego está totalmente funcional.

Tabla 56. Caso de uso: Seleccionar mini juego

CASO DE USO:	Seleccionar Mini Juego
ACTORES:	Usuario
PROPÓSITO:	Poder jugar un videojuego tipo Geometry Dash.
RESUMEN:	Poder jugar un videojuego tipo Geometry Dash, este videojuego esta totalmente funcional.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Mini Juego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- **Escena opciones:** En esta escena, se podrá visualizar información referente al desarrollador de la aplicación móvil.

Tabla 57. Caso de uso: Opciones

CASO DE USO:	Opciones
ACTORES:	Usuario
PROPÓSITO:	Créditos al desarrollador de la aplicación móvil
RESUMEN:	Información del desarrollador de la aplicación móvil.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el menú Opciones.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 58. Caso de uso: Salir

CASO DE USO:	Salir
ACTORES:	Usuario
PROPÓSITO:	Salir de la aplicación móvil.
RESUMEN:	Salir de la aplicación móvil.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el menú Salir.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.3.3. Fase de Producción

Es en esta fase donde se empieza el diseño de la aplicación móvil según a los casos de uso que se realizó en la etapa de inicialización, donde se determinó los requerimientos logrando definir la funcionalidad que la aplicación tiene.

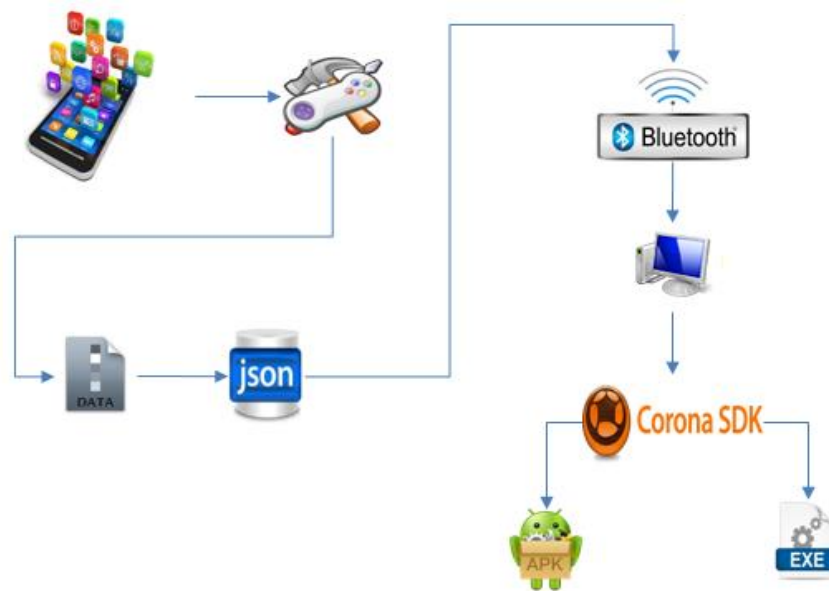
4.3.3.1. Diseño de la aplicación móvil

Tomando en base al análisis realizado en la fase de iniciación en el cual se determino los casos de uso de la aplicación móvil. A continuación se determina la estructura de la aplicación móvil:

Introducción: En este apartado se muestra 6 capítulos prácticos de interacción, como contenido de introducción al desarrollo de videojuegos para dispositivos móviles.

Diseño: Este apartado es la parte mas importante de la aplicación móvil, es aquí donde el usuario podrá diseñar, guardar, ejecutar, eliminar y exportar un videojuego por medio del bluetooth, para luego construir el instalador del videojuego diseñado para 2 plataformas: Android(.apk), y Windows Desktop(.exe). El proceso de construcción multiplataforma se ara directamente con el Framework Corona SDK, al momento de la vista preliminar del diseño del videojuego es aquí donde internamente se genera un fichero .json, este fichero contiene la información de diseño , que será leído por un fichero .lua el cual es el encargado de mostrar la vista del videojuego diseñado . A continuación se muestra una imagen para el mejor entendimiento del proceso de diseño y construcción de videojuegos multiplataforma:

Figura 8. Modelo conceptual de la aplicación GameDesign



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Mini juego: La aplicación móvil tiene un videojuego tipo Geometry dash como ejemplo, totalmente funcional.

Opciones: Descripción y créditos al desarrollador de la aplicación móvil.

Diseño de la aplicación móvil

Para el diseño de la aplicación móvil, se consideró lo siguiente:

- ✓ **Librerías y módulos**
- ✓ **Adaptación dinámica del contenido**
- ✓ **Definición del icono de la aplicación**
- ✓ **Estructura de la aplicación móvil**

A continuación se detalla cada uno de estos:

Librerías

A continuación se detalla cada una de las librerías que se uso para la implementación de la aplicación móvil:

- ✓ **audio.*:** para el manejo del audio de la aplicación móvil.
- ✓ **composer.*:** para el manejo de escenas de la aplicación móvil.
- ✓ **display.*:** para crear imágenes, textos, cuerpos geométricos.
- ✓ **easing.*:** para el tipo de animación que se puede aplicar a un objeto.
- ✓ **io.*:** para eliminar un fichero de la aplicación móvil.
- ✓ **json.*:** para la lectura y escritura de información almacenada internamente en la aplicación móvil.
- ✓ **math.*:** para cálculos matemáticos avanzados.
- ✓ **native.*:** para crear campos de texto.
- ✓ **os.*:** para salir de la aplicación móvil.
- ✓ **Package.*:** para la disposición de POO entre otros ficheros .lua
- ✓ **Physics.*:** para agregar física a un cuerpo creado en la escena.
- ✓ **Table.*:** para crear atributos de los objetos
- ✓ **Timer.*:** para el manejo de tiempos(loops).
- ✓ **Transition.*:** para la animación de objetos.
- ✓ **Widget.*:** para crear botones nativos.

Módulos

A continuación se detalla cada una de los módulos que se uso para la implementación de la aplicación móvil:

- ✓ **loadsave.lua:** para la lectura y escritura de ficheros .json
- ✓ **gameUI.lua:** para eventos touch al momento del diseño del videojuego.
- ✓ **Dusk.lua:** para el manejo de mapas (usado en el mini juego de ejemplo)
- ✓ **movieclip.lua:** para la animación de explosión(usado al momento de las colisiones).

Adaptación dinámica del contenido

Figura 9. Dispositivos de diferentes resoluciones de pantalla



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Para comenzar es necesario decidir el tamaño del contenido con independencia de las dimensiones de la pantalla. Este método proporciona el sistema de coordenadas para el código de aplicaciones realizadas con Corona SDK , que será independiente de la cantidad real de píxeles en la pantalla del dispositivo.

Para esto se define el tamaño que queremos al que se adapte nuestra aplicación , el tamaño de contenido en el archivo de la aplicación llamado **config.lua** . el código no tiene que conocer el tamaño real del dispositivo.

Config.lua: hay varios modos:

- ✓ **letterbox:** es el modo más común ya que no se recorta el contenido aunque por el contrario puede mostrar zonas negras
- ✓ **ZoomEven:** en este modo el contenido se escala para llenar completamente la pantalla, aunque puede que alguna parte de la pantalla se vea recortado los bordes
- ✓ **ZoomStretch:** el contenido se escala para completar totalmente la anchura y altura de la pantalla.
- ✓ **Sin parámetro scale:** esto desactiva el modo de escalado de contenidos, y produce el mismo resultado que no tener un archivo config.lua.

Figura 10. código fuente del fichero config.lua

```
application = {
  content = {
    width = 320,
    height = 480,
    scale = "letterBox",
    fps = 30,
  },
}
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Build.settings: en este fichero se configura la orientación que tendrá la aplicación móvil siendo posible 4 orientaciones:

- ✓ **Portrait:** orientación de forma vertical
- ✓ **portraitUpsideDown:** orientación de forma vertical invertida.
- ✓ **landscapeLeft:** orientación de horizontal
- ✓ **landscapeRight:** orientación de forma horizontal invertida.

La aplicación 'GameDesign', se configuró de forma horizontal, por la misma necesidad de comodidad al momento del diseño de videojuego:

Figura 11. código fuente del fichero build.settings

```
settings = {
  orientation = {
    default = "landscapeLeft",
    supported = { "landscapeLeft", "landscapeRight" }
  },
  window = { defaultMode = "fullscreen" , } ,
}
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Definición del icono de la aplicación

También en este fichero build.settings, se determina la declaración del icono de la aplicación móvil asociado con extensión (.png) icono para dispositivos con sistema operativo Android, mientras que el icono (.ico) es el icono para el sistema operativo windows desktop.

Figura 12. definiendo iconos de la aplicación móvil

```
iphone = {
  plist =
  {
    CFBundleIconFiles = {
      "Icon-xxxhdpi.png",
      "Icon-xxhdpi.png",
      "Icon-xhdpi.png",
      "Icon-hdpi.png",
      "Icon-mdpi.png",
      "Icon-ldpi.png",
      "Icon-win32.ico",
    },
  }
}
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Estructura de la aplicación móvil

A continuación se muestra la estructura de los directorios y ficheros que tiene la aplicación móvil ‘GameDesign’:

Tabla 59. Leyenda de la estructura de la aplicación móvil

COLOR DE RELLERO	SIGNIFICADO
	Directorio
	Ficheros.lua
	Ficheros.json
	Fichero.settings
	Descripción

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Estructura de directorios y ficheros de la aplicación móvil

Tabla 60. Estructura de directorios y ficheros de la aplicación móvil

NIVEL 0	NIVEL 1	NIVEL 2	NIVEL 3	NIVEL 4	DESCRIPCIÓN	
GameDesign	code	capitulos	cap1.lua		Diapositiva de Imágenes.	
			cap2.lua		Diapositiva de Textos.	
			cap3.lua		Diapositiva de Toques	
			cap4.lua		Diapositiva de Sonido.	
			cap5.lua		Diapositiva de Animación.	
			cap6.lua		Diapositiva de Física.	
		diseño	Audio			Directorio que contiene 10 audios disponibles.
			biblioteca		Actores1.lua	Interfaz para seleccionar Actores 1.

				Actores2.lua	Interfaz para seleccionar Actores 2.
				Baldosas.lua	Interfaz para seleccionar Baldosas.
				Fondos.lua	Interfaz para seleccionar un fondo.
			conductas	Animacion.json	Lista de propiedades de animación.
				Audio.json	Lista de nombres de los audios.
				Ayuda.json	Lista de propiedades
				Colisiones.json	Lista de propiedades de colisiones.
				Disparo.json	Lista de propiedades de disparo.
				Física.json	Lista de propiedades de física.
				Gravedad.json	Lista de propiedades de gravedad
				Mapa.json	Lista de propiedades del Mapa.
				Objeto.json	Lista de propiedades de objeto.
				Opciones.json	Lista de opciones del videojuego.
			funciones	Animacion.lua	Definir la animación de los objetos creados.
				Ayuda. lua	Ayuda acerca del uso de cada una de las propiedades.
				Colisiones. lua	Definir que objetos serán colisionados.
				Disparo. lua	Definir un objeto para que pueda disparar.
				Física. lua	Definir la física de los objetos creados.
				Gravedad. lua	Definir la gravedad 'x' e 'y' del videojuego.
				Mapa.lua	Definir la cantidad de escenas horizontales o verticales que tendrá el videojuego.
				Objeto. lua	Definir algunas propiedades básicas de los objetos creados.
				Opciones. lua	Posibilidad de ejecutar o eliminar el videojuego diseñado, también si se desea retornar al menú principal.
			generado	JuegoGenerado.lua	Escena principal donde se visualiza la ejecución del juego diseñado.
				archivos	Directorio que contiene un proyecto Corona SDK por defecto, listo

					para que pueda leer el videojuego diseñado, también se encuentra un archivo LEEME.txt el cual tiene información acerca del proceso de construcción del videojuego final (.apk, .exe).
			juego	Principal.lua	Escena principal donde se diseña el videojuego.
			gameUI.lua		Permite mover los objetos por pantalla y evitar crear objetos uno sobre otro.
			loadsave.lua		Permite la lectura y escritura de ficheros .json
		intro	inicio.lua		Pantalla donde existe la primera interacción entre aplicación y usuario.
			menu.lua		Pantalla del menú principal de la aplicación móvil.
			splash.lua		Primera pantalla de la aplicación móvil que se visualiza.
		pantallas	pantalla1.lua		Pantalla de Introducción a las diferentes diapositivas.
			pantalla2.lua		Pantalla que da pase a la interfaz para crear o abrir el diseño de un videojuego.
			pantalla3.lua		Pantalla que da pase al mini juego.
			pantalla4.lua		Pantalla de créditos del desarrollador de la aplicación.
			Pantalla5.lua		Pantalla que da pase a la selección de los diferentes objetos para poder diseñar.
	Dusk				Directorio que contiene ficheros .lua encargados del manejo del mapa del minijuego.

	img	dibujos		Directorio que contiene las imágenes usadas en el tutorial de diapositivas.	
		diseño	Actores1		Directorio que contiene las imágenes de los actores1
			Actores2		Directorio que contiene las imágenes de los actores2
			baldosas		Directorio que contiene las imágenes de las baldosas.
			fondos		Directorio que contiene las imágenes de fondo.
			balas		Directorio que contiene las imágenes de disparo.
			Explosión		Directorio que contiene las imágenes de explosión.
		fondo		Directorio que contiene las imágenes usadas en el tutorial de diapositivas, también los títulos de las distintas interfaces.	
		iconos		Directorio que contiene los iconos de los widgets usados en la aplicación móvil.	
	opciones		Directorio que contiene las imágenes de las opciones.		
	Geometry Dash	code	img		Directorio que contiene las imágenes del videojuego Geometry Dash.
			niveles	nivel1.lua	Escena central del videojuego Geometry dash.
			pantallas	pantallaMenu.lua	Menú principal del mini juego.
			sonido		Sonido del mini juego
		mapas	Level1.json		Fichero .json donde se encuentra el diseño del mapa del mini juego.
sonido		Directorio que contiene el audio usado en el tutorial de diapositivas (sonidos).			
build.settings		Fichero de construcción de la aplicación móvil.			
config.lua		Fichero de configuración de la aplicación móvil.			

	movieclip.lua	Fichero encargado de la animación de explosión que se puede tener.
	main.lua	Fichero de arranque de la aplicación móvil.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

4.3.4. Fase de Estabilización

En esta fase se desarrolla la implementación de la aplicación, se integra los módulos con la programación del código el cual le da funcionalidad a las interfaces de usuario.

En esta fase, se asegura que el proyecto (como un todo) funcione correctamente, realizando las correcciones necesarias y generando la documentación del proyecto.

4.3.4.1. Implementación de la aplicación móvil

Toda aplicación en Corona SDK, inicia desde un fichero llamado main.lua, así que la implementación de la aplicación GameDesign inicia por este fichero:

Figura 13. código fuente del fichero main.lua

```
local composer=require "composer"
composer.gotoScene( "code.intro.splash","fade",1000 )
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Con las líneas de código de la imagen anterior es posible ir a la primera escena de la aplicación móvil:

Figura 14. Código fuente de la escena splash

```
local composer = require "composer"
local scene =composer.newScene()
function scene:create()
    grupo=self.view
    fondo()
end
function fondo( )
    img=display.newImage("img/fondo/splash.png")
    img.x,img.y=240,160
    grupo:insert( img )
end
function tiempo()
    composer.gotoScene( "code.intro.inicio","fade",1000 )
end
timer.performWithDelay( 3000,tiempo,1)
function scene:show()
end
function scene:hide()
end
function scene:destroy()
end
scene.addEventListener( "create", scene )
scene.addEventListener( "show", scene )
scene.addEventListener( "hide", scene )
scene.addEventListener( "destroy", scene )
return scene
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Una vez cargado la primera interfaz, este da pase a la escena de bienvenida en 3 segundos:

Figura 15. Código fuente de la escena bienvenida

```
function irScene( )
    composer.gotoScene( "code.intro.menu","fade",1000)
end
Runtime:addEventListener( "touch", irScene )
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Como se puede observar se agrego un evento al **Runtime**(pantalla del dispositivo), de tipo 'touch', que al tocar la pantalla, se ejecuta la función

`irScene()`, el cual da pase al menú principal de la aplicación móvil 'GameDesign':

Figura 16. Código fuente de la escena menú principal

```
function irScene(event)
  modo=event.target.id
  for i=1,4 do
    if i==modo then
      composer.gotoScene( "code.pantallas.pantalla"..i,"fade",1000 )
    end
  end
  if modo==5 then
    alerta = native.showAlert( "Mensaje", "¿Desea salir de GameDesign?", { "No", "Si" },accion)
  end
end

function accion(event)
  if event.index == 2 then
    os.exit()
  end
end
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

A partir de esta escena ya se puede navegar por los distintos capítulos que tiene la aplicación móvil.

Implementación del menú Introducción

En este menú se cuenta con 6 capítulos asociados a 6 ficheros .lua:

- ✓ **cap1.lua:** referente al manejo de imágenes
- ✓ **cap2.lua:** referente al manejo de textos
- ✓ **cap3.lua:** referente al manejo de toques.
- ✓ **cap4.lua:** referente al manejo de sonido.
- ✓ **cap5.lua:** referente al manejo de animación.
- ✓ **cap6.lua:** referente al manejo de la física.

En cada uno de los ficheros mencionados anteriormente se creo 4 botones y un simulador:

- ✓ **botón siguiente:** encargado de mostrar la línea de código fuente siguiente , la descripción y la simulación.
- ✓ **botón anterior:** encargado de mostrar la línea de código fuente anterior, la descripción y la simulación.
- ✓ **botón automático:** encargado de mostrar en código fuente , la descripción y la simulación de forma automática, ya que se implemento que cada línea de código fuente se visualice cada segundo.

- ✓ **botón atrás:** para salir del capítulo actual y dar pase a la selección de los demás capítulos.
- ✓ **simulador:** imagen en la cual se visualizará cada acción que se debe de ejecutar según el código fuente asignado.

Implementación del menú diseñar

Antes de diseñar un videojuego se debe crear un nuevo proyecto, asignándole un nombre y seguidamente seleccionar los objetos necesarios para diseñar, solo de esta manera se podrá ingresar a la interfaz principal de diseño del videojuego. A continuación se describe los ficheros .lua asociados al pre-diseño de un videojuego:

Ficheros dependientes

Los siguientes ficheros dependen del fichero **loadsave.lua**:

- ✓ **Actores1.lua**
- ✓ **Actores2.lua**
- ✓ **Baldosas.lua**
- ✓ **Fondos.lua**

Los ficheros mencionados anteriormente contienen las siguientes líneas de código, donde todos los objetos de seleccionados se guardaran en el fichero **tabla.json**, que en el simulador se puede observar en file→Show Project Sandbox→ Documents:

- ✓ Para poder cargar los datos si es que existen:

Figura 17. Código fuente de importación del fichero loadsave.lua

```
local clase =require "code.diseño.loadsave"
datos = clase.cargarTabla("tabla.json", system.DocumentsDirectory)
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

- ✓ Para guardar nuevos objetos seleccionados:

Figura 18. código fuente para guardar datos en el fichero tabla.json

```
local t = {
    nombreActor1= IDActor1,--volvemos a grabar actores1
    nombreActor2=IDActor2,--volvemos a grabar actores2
    nombreBaldosa=IDBaldosa,--volvemos a grabar baldosas
    fondo=IDfondo,--grabamos fondo
}
--guardamos array 't'en el fichero tabla.json
clase.guardarTabla(t, "tabla.json", system.DocumentsDirectory)
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Una vez seleccionado los objetos se puede pasar a la escena principal de diseño el cual está asociado al fichero principal.lua:

El fichero Principal.lua, depende de los siguientes ficheros:

- ✓ **gameUI.lua**
- ✓ **loadsave.lua**
- ✓ **Animacion.lua**
- ✓ **Colisiones.lua**
- ✓ **Objeto.lua**
- ✓ **Disparo.lua**
- ✓ **Fisica.lua**
- ✓ **Gravedad.lua**
- ✓ **Mapa.lua**
- ✓ **Opciones.lua**
- ✓ **Ayuda.lua**

Figura 19. Código fuente para importar módulos

```
--importamos modulo gameUI para hacer uso de sus metodos
local gameUI      = require "code.diseño.gameUI"
--importamos modulo loadsave para hacer uso de sus metodos
local clase       = require "code.diseño.loadsave"
--importamos todos los modulos que almacenan las propiedades de diseño
local animacion   = require "code.diseño.funciones.Animacion"
local colisiones  = require "code.diseño.funciones.Colisiones"
local elemento    = require "code.diseño.funciones.Objeto"
local disparo     = require "code.diseño.funciones.Disparo"
local fisica      = require "code.diseño.funciones.Fisica"
local gravedad    = require "code.diseño.funciones.Gravedad"
local mapa        = require "code.diseño.funciones.Mapa"
local opciones    = require "code.diseño.funciones.Opciones"
local ayuda       = require "code.diseño.funciones.Ayuda"
-----
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Envío de parámetros de los objetos a cada una de las propiedades:

Figura 20. Código fuente para enviar parámetros a las propiedades

```
if evento=='Fisica' then fisica.datos(fila.id,objeto,fondo,escenario,dragBody) end
if evento=='Colisiones'then colisiones.datos(fila.id,objeto,fondo,escenario,dragBody) end
if evento=='Objeto' then elemento.datos(fila.id,objeto,fondo,escenario,dragBody) end
if evento=='Gravedad' then gravedad.datos(fila.id,objeto,fondo,escenario,dragBody,escena) end
if evento=='Animacion' then animacion.datos(fila.id,objeto,fondo,escenario,dragBody) end
if evento=='Disparo' then disparo.datos(fila.id,objeto,fondo,escenario,dragBody) end
if evento=='Mapa' then mapa.datos(fila.id,objeto,fondo,escenario,dragBody,mapapro,grupo300) end
if evento=='Opciones' then opciones.datos(fila.id,objeto,fondo,escenario,dragBody,escena,proyectoID,mapapro) end
if evento=='Ayuda' then ayuda.datos(fila.id,objeto,fondo,escenario,dragBody) end
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

A continuación se detallara como fue la implementación de cada una de las propiedades, tener en cuenta lo siguiente:

- ✓ Todo objeto creado en la pantalla será manipulado e identificado por la variable **elemento**, al cual se le asignará cada una de las propiedades.
- ✓ **elemento[i]**, viene a ser un objeto 'i' creado en la pantalla de diseño.
- ✓ La manipulación de la escena del mapa será identificado con la variable **proScene**.
- ✓ La manipulación de las propiedades del mapa del videojuego será identificado con la variable **mapapro**.

Implementación de la propiedad Física

Asociado al fichero Física.lua, para la asignación de propiedades de física se determinó lo siguiente:

- ✓ **elemento[i].cuerpo:** posibilidad de asignarle el tipo de cuerpo 'dinámico o estático'.

Implementación de la propiedad Animación

Asociado al fichero Animacion.lua, para la asignación de propiedades de animación se determinó lo siguiente:

- ✓ **elemento[i]. animacionVelocidad:** atributo para la velocidad de animación del objeto.
- ✓ **elemento[i]. animacionDestinoX:** atributo para asignar el punto destino en el eje X de animación del objeto.

- ✓ **elemento[i]. animacionDestinoY:** atributo para asignar el punto destino en el eje Y de animación del objeto.
- ✓ **elemento[i]. animacionRotacion:** atributo para asignar si el objeto va tener una rotación durante la animación, se definió el tipo de rotación como Horario o anti horario.
- ✓ **elemento[i]. animacionAleatorio:** atributo para desplazar al objeto en el eje X e Y durante la animación.

Implementación de la propiedad Objeto

Asociado al fichero Objeto.lua, para la asignación de propiedades de objeto se determinó lo siguiente:

- ✓ **elemento[i]. ancho:** atributo para asignar en ancho del objeto
- ✓ **elemento[i]. alto:** atributo para asignar la altura del objeto
- ✓ **elemento[i]. rotacion:** atributo para asignar una rotación al objeto.
- ✓ **elemento[i]. control:** atributo para controlar al objeto en cualquier eje 'x' o 'y', 'x e y', o darle un impulso.
- ✓ **elemento[i]. visible:** atributo para mostrar u ocultar el objeto.

Implementación de la propiedad Disparo

Asociado al fichero Disparo.lua, para la asignación de propiedades de disparo se determinó lo siguiente:

- ✓ **elemento[i]. disparoTiempoCreacion:** atributo para asignar cada cuantos segundos se creara el disparo.
- ✓ **elemento[i]. disparoTiempoAnimacion:** atributo para asignar la velocidad de animación del disparo.
- ✓ **elemento[i]. disparoDestinoX:** atributo para asignar el punto destino en el eje 'x' del disparo.
- ✓ **elemento[i]. disparoDestinoY:** atributo para asignar el punto destino en el eje 'y' del disparo.

Implementación de la propiedad Colisiones

Asociado al fichero Colisiones.lua, para la asignación de propiedades de colisiones se determinó lo siguiente:

- ✓ **elemento[i]. colisionActivo:** atributo para asignar si el objeto al colisionar se va a eliminar o mantener en la escena.

Implementación de la propiedad Gravedad

Asociado al fichero Gravedad.lua, para la asignación de propiedades de gravedad se determinó lo siguiente:

- ✓ **proScene .vista:** atributo para asignar el modo de vista (normal o hybrid).

- ✓ **proScene . gravedadx:** atributo para asignar el valor de la gravedad del videojuego en el eje 'x'.
- ✓ **proScene . gravedady:** atributo para asignar el valor de la gravedad del videojuego en el eje 'y'.

Implementación de la propiedad Mapa

Asociado al fichero Mapa.lua, para la asignación de propiedades de mapa se determinó lo siguiente:

- ✓ **mapapro. horizontal:** atributo para determinar la cantidad de escenas horizontales que tendrá el videojuego.
- ✓ **mapapro. vertical:** atributo para determinar la cantidad de escenas verticales que tendrá el videojuego.
- ✓ **mapapro. animacion:** atributo para determinar la orientación de la animación del mapa, puede que la animación sea en forma horizontal o vertical.
- ✓ **mapapro. speed:** atributo para determinar la velocidad de la animación del mapa.
- ✓ **mapapro. audio:** atributo para determinar un fondo de música que tendrá el videojuego.

Implementación de la propiedad Opciones

Asociado al fichero Opciones.lua, se considero las siguientes opciones:

- ✓ **ejecutar:** es aquí donde las propiedades asignadas a cada objeto en la escena diseño se guardan en un fichero .json y a su vez se preparan estas propiedades para ser enviado a la escena de ejecución del videojuego.

Figura 21. Variables locales para recuperar información.

```

local nombre={}
local ancho={}
local alto={}
local rotacion={}
local control={}
local posX ={}
local posY ={}
local creado ={}
local cuerpo ={}
local animacionVelocidad={}
local animacionDestinoX={}
local animacionDestinoY={}
local animacionRotacion={}
local animacionAleatorio={}
local disparoTiempoCreacion={}
local disparoTiempoAnimacion={}
local disparoDestinoX={}
local disparoDestinoY={}
local colisionActivo ={}
local objetoVisible={}

```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 22. Recuperación de cada propiedad de los objetos

```
for i=1, #elemento do
    nombre[i]=elemento[i].nombre
    ancho[i]=elemento[i].ancho
    alto[i]=elemento[i].alto
    rotacion[i]=elemento[i].rotacion
    control[i]=elemento[i].control
    posX[i]=elemento[i].posx
    posY[i]=elemento[i].posy
    creado[i]=elemento[i].creado
    cuerpo[i]=elemento[i].cuerpo
    animacionVelocidad[i]=elemento[i].animacionVelocidad
    animacionDestinoX[i]=elemento[i].animacionDestinoX
    animacionDestinoY[i]=elemento[i].animacionDestinoY
    animacionRotacion[i]=elemento[i].animacionRotacion
    animacionAleatorio[i]=elemento[i].animacionAleatorio
    disparoTiempoCreacion[i]=elemento[i].disparoTiempoCreacion
    disparoTiempoAnimacion[i]=elemento[i].disparoTiempoAnimacion
    disparoDestinoX[i]=elemento[i].disparoDestinoX
    disparoDestinoY[i]=elemento[i].disparoDestinoY
    colisionActivo[i]=elemento[i].colisionActivo
    objetoVisible[i]=elemento[i].visible
end
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 23. Guardando información en un archivo .json

```
-----ALMACENAMOS DATOS DEL OBJETO CON CADA UNA DE SUS PROPIEDADES
local t = {
    fisica=escena,
    mapapro=mapa,
    nombre=nombre,
    ancho=ancho,
    alto=alto,
    rotacion=rotacion,
    control=control,
    posX=posX,
    posY=posY,
    creado=creado,
    cuerpo=cuerpo,
    animacionVelocidad=animacionVelocidad,
    animacionDestinoX=animacionDestinoX,
    animacionDestinoY=animacionDestinoY,
    animacionRotacion=animacionRotacion,
    animacionAleatorio=animacionAleatorio,
    disparoTiempoCreacion=disparoTiempoCreacion,
    disparoTiempoAnimacion=disparoTiempoAnimacion,
    disparoDestinoX=disparoDestinoX,
    disparoDestinoY=disparoDestinoY,
    colisionActivo=colisionActivo,
    objetoVisible=objetoVisible,
    fondo=fondoNombre,
}
clase.guardarTabla(t, "..Proyecto..".json",system.DocumentsDirectory)
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Al momento de ejecutar se preparan todas las propiedades asignadas a todos los objetos, para ser almacenadas en un fichero **.json**, el nombre de este fichero será el nombre que se puso al momento de crear el videojuego.

- ✓ **eliminar:** también se puede eliminar el diseño de un videojuego, como se muestra a continuación:

Figura 24. Eliminando el diseño de un videojuego

```
for i=1,#datos.proyectos do
  if proyectoID==datos.proyectos[i] then
    local temporal = system.pathForFile("../proyectoID../.json", system.DocumentsDirectory )
    local eliminar= os.remove(temporal)
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

La variable local **temporal**, es la ruta del fichero **.json** del proyecto que se desea eliminar, una vez determinado la ruta se procede a eliminar con la librería **os.remove(temporal)**.

- ✓ **menú principal:** muestra un mensaje de alerta para volver al menú principal de la aplicación móvil.

Implementación de la propiedad Ayuda

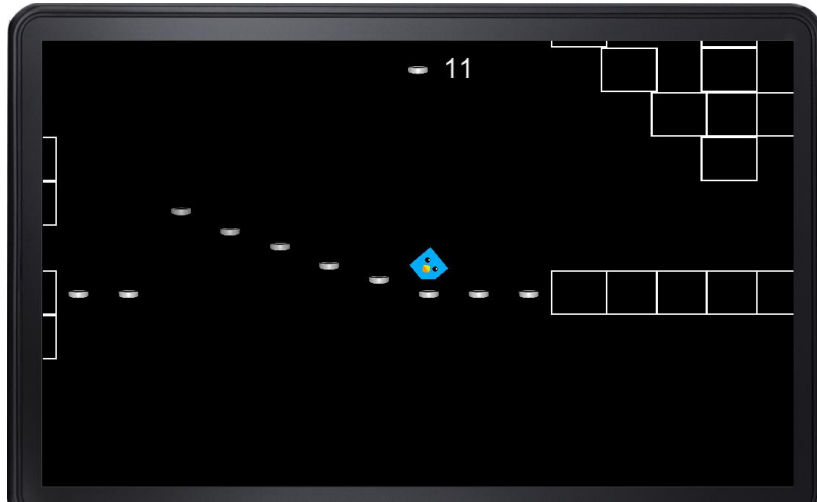
Asociado al fichero **Ayuda.lua**, aquí se muestra mensajes de ayuda de cada una de la propiedades que tiene la aplicación móvil.

Implementación del menú mini juego

La aplicación **GameDesign**, incluye un minijuego tipo **GeometryDash**, totalmente funcional, a continuación se muestra su implementación.

- ✓ **pantallaMenu.lua:** fichero del menú principal del minijuego en el cual se crearon dos opciones(jugar y salir).
- ✓ **nivel1.lua:** fichero principal del desarrollo del mini juego.
- ✓ **level1.json:** fichero que contiene el diseño del mapa del mini juego, realizado con **tilde Map**.

Figura 25. Mini juego en ejecución



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

4.3.5. Fase de Pruebas del Sistema

La etapa de pruebas busca entregar una versión completamente estable y funcional de la aplicación móvil , se comparó y probó la aplicación móvil contra los requisitos del usuario y se corrigieron todos los defectos encontrados.

Esta es la última fase que se aplica en la metodología Mobile-D, que es el conjunto de pruebas de la aplicación e identificar algunos errores y repararlos. Una vez aprobado y aceptado la aplicación según los requisitos del usuario y sistema que debe cumplir, es aquí cuando ya tendríamos el entregable final eliminando todos los defectos encontrados.

4.3.5.1. Errores encontrados en la aplicación GameDesign

Antes de definir que la aplicación móvil GameDesign a sido culminada, se hizo una pruebas para determinar la existencia de errores y repararlos. Por ello se ha elegido la forma tradicional de hacerlo el cual es ir probando todas las opciones posibles que admite cada escena de la aplicación. Dado que hay escenas que son similares el trabajo se simplifica y es posible reducir bastante el tiempo de verificación.

Tras someter a prueba la aplicación se detectaron fallos de funcionamiento, en este caso el fallo procede de las imágenes que se muestra en la pantalla, particularmente en los botones. Tras diseñar los botones y seguidamente se probó la funcionalidad de la aplicación en un dispositivo real , el error existía en el dispositivo real y no en el simulador lo cual fue muy trabajoso encontrar

cierto error. Cuando se eliminó el objeto el error desaparecía, por este problema se eliminó el objeto, y se logró corregir el error.

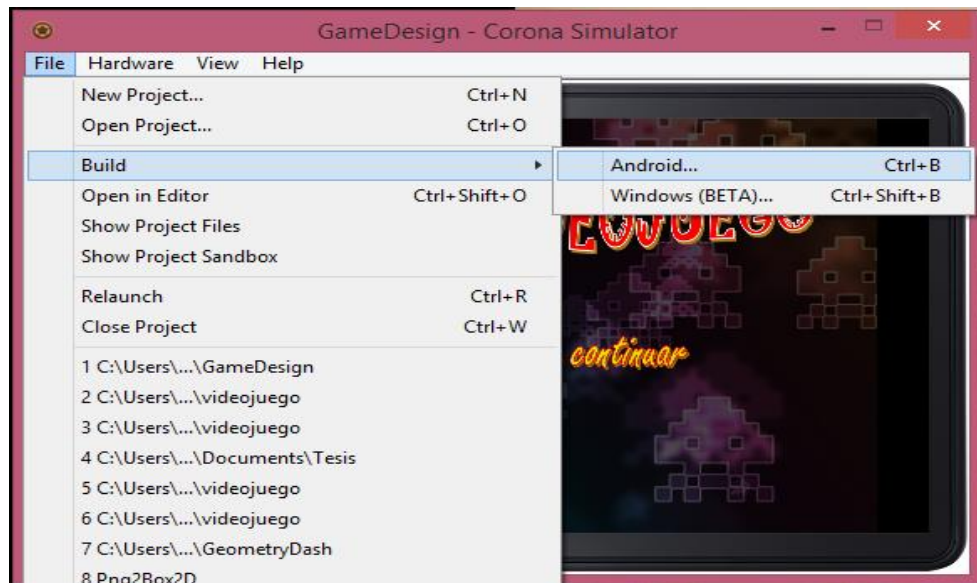
Otros fallos comunes son las rutas de las imágenes. Suelen estar equivocadas y simplemente basta con corregirlas o colocar las imágenes en la ruta correcta.

Otro error común al momento de desarrollar el videojuego es la de no pausar los **timers**, la física o los eventos de toque de pantalla (**'tap'**, **'touch'**), ya que en el simulador no se notaba el error, pero al probarlo en un dispositivo real si se encontró, se reviso cada fichero .lua y en el método de **scene:hide()**, es aquí donde los eventos, tiempos, física, etc. Se debe de cancelar y ponerlos a **'nil'** para que en una nueva escena no se encuentre estos errores.

4.3.5.2. Construcción de la aplicación GameDesign

Para la construcción de la aplicación GameDesign.apk es necesario que se tenga una cuenta como desarrollador en la pagina oficial de Corona Labs, la construcción de la aplicación es online, ya que el código fuente es enviado a los servidores de Corona Labs, y una vez revisado y descartado los errores encontrados, nos retorna el paquete de Android .apk, mientras que la construcción de la aplicación para Windows Desktop por suerte es de modo offline:

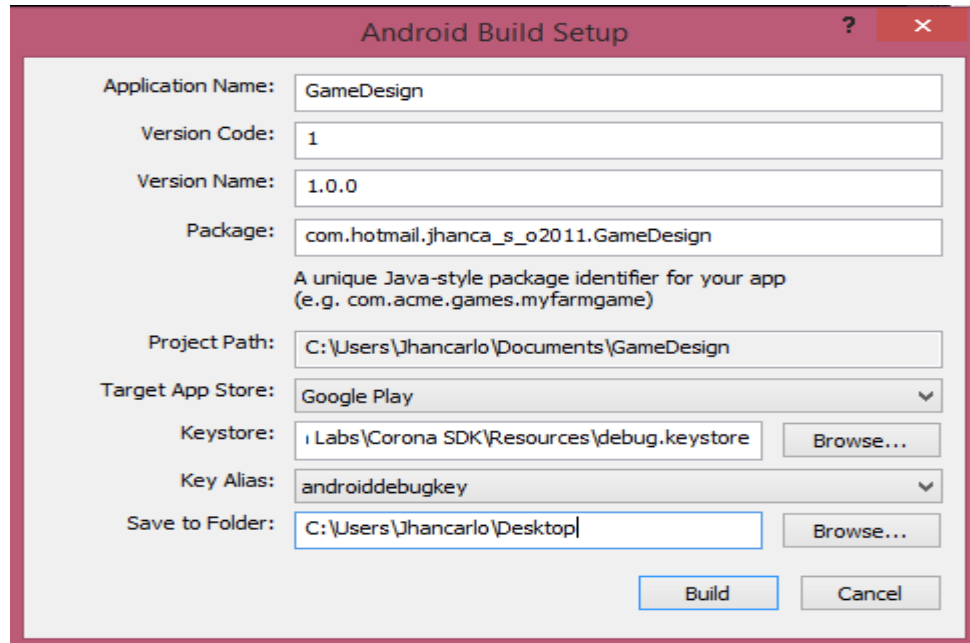
Figura 26. Disponibilidad de plataforma de construcción



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Como se puede visualizar en la imagen anterior, Corona SDK permite la construcción de la aplicación para Android(.apk) de modo online y para Windows Desktop(.exe) de modo offline.

Figura 27. Construcción de la aplicación para Android



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

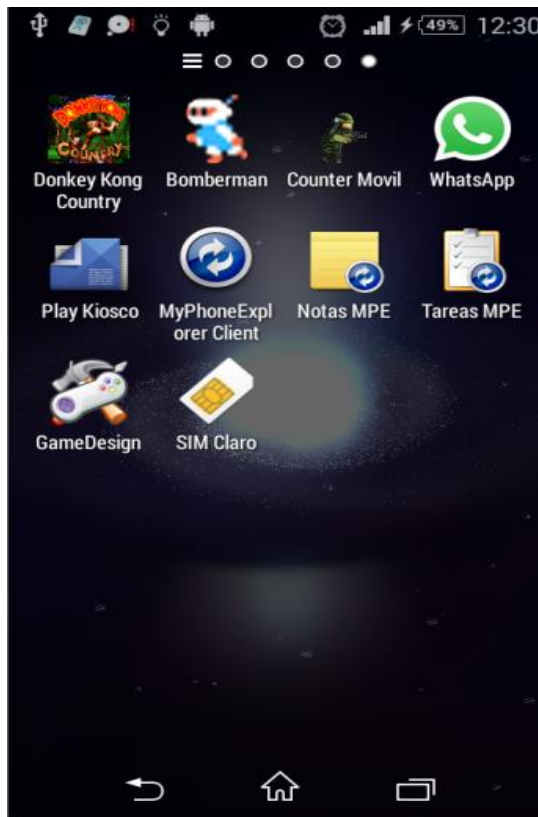
En la imagen anterior se tiene un formulario donde se debe de llenar todos los campos, como el nombre de la aplicación, la versión, y la ruta donde se guardara el (**GameDesign.apk**).

4.3.5.3. Diseño de un videojuego multiplataforma

Una vez construido la aplicación GameDesign.apk procedemos a instalarlo en un dispositivo real, para diseñar un videojuego sencillo y luego construir el .apk y el .exe.

A continuación se muestra la aplicación ya instalada en un dispositivo real con sistema operativo Android 4.4:

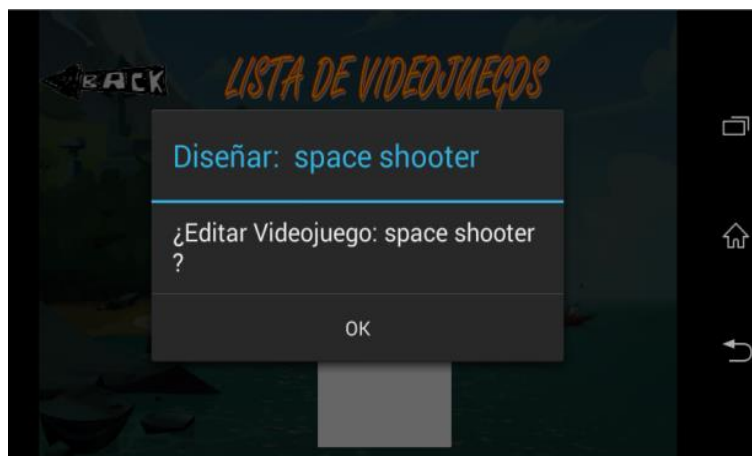
Figura 28. Aplicación GameDesign instalada correctamente



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Una vez instalado la aplicación GameDesign, se diseñó un videojuego sencillo de nombre **space shooter**.

Figura 29. Abrir el videojuego diseñado space shooter



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 30. videojuego space shooter abierto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 31. mensaje de alerta para ver ejecución del videojuego



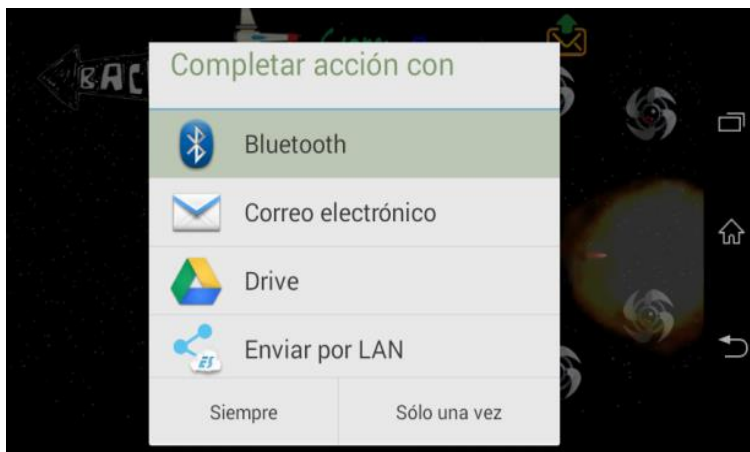
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 32. ejecución del videojuego diseñado



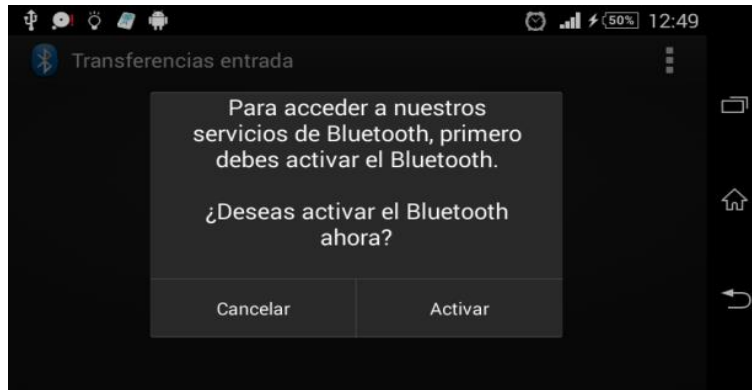
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 33. selección del bluetooth



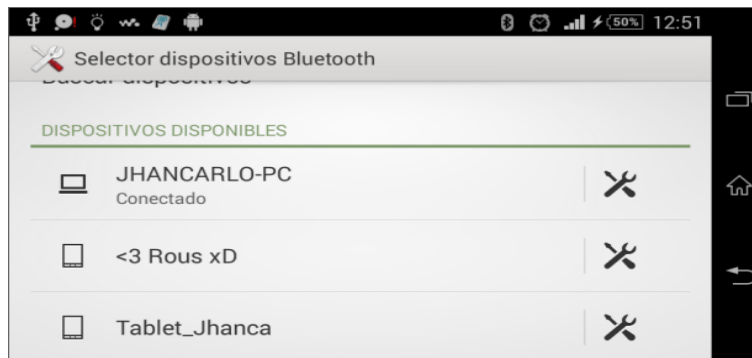
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 34. permiso para hacer uso del bluetooth



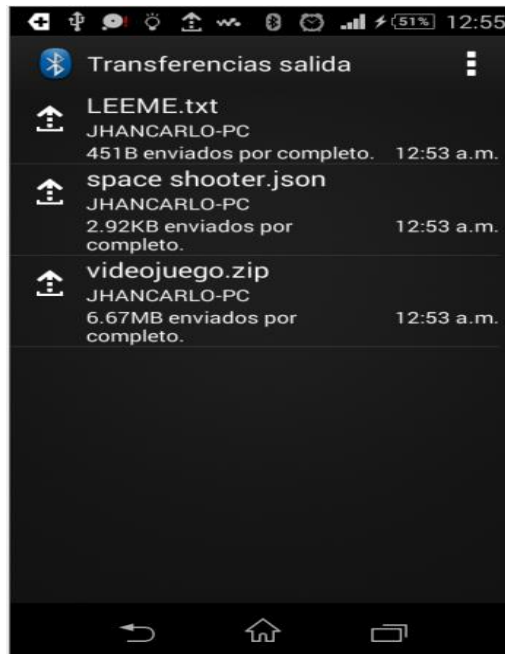
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 35. dispositivos disponibles para exportar videojuego diseñado



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 36. transferencia completa del videojuego al PC



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

A continuación se muestra los recursos del videojuego exportado al pc:

Figura 37. recursos del videojuego exportado



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Para la construcción del **.apk** y **.exe** se debe de realizar lo siguiente:

- ✓ Abrir el archivo LEEME.txt:

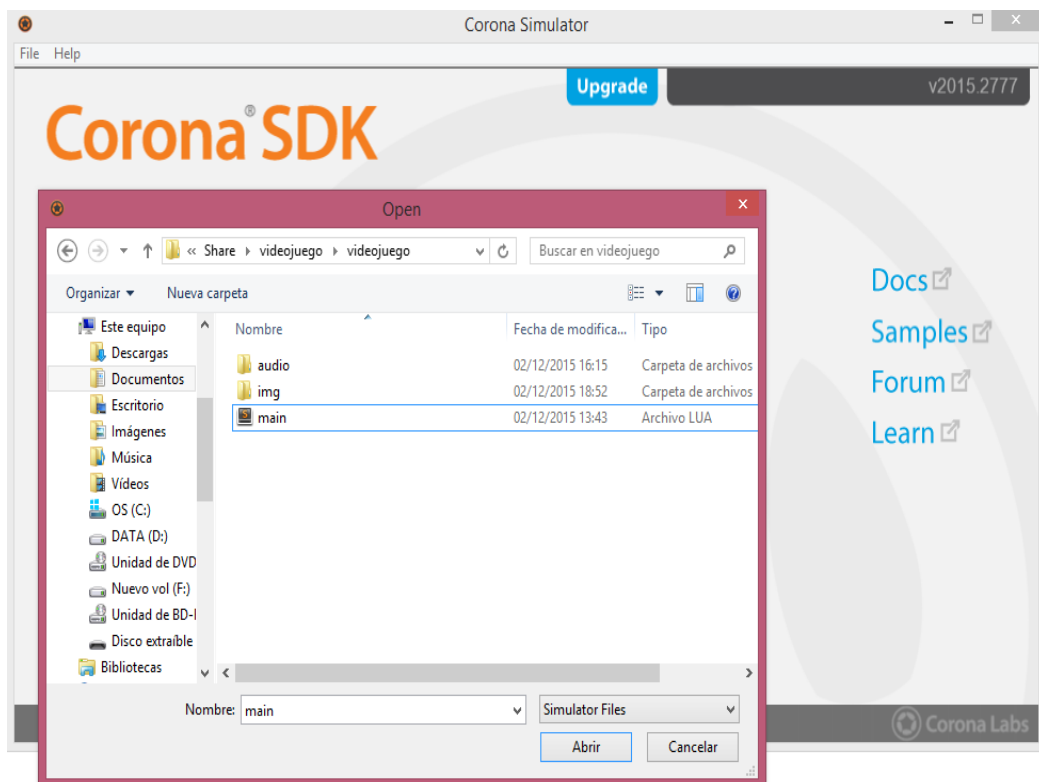
Figura 38. Información importante del archivo LEEME.txt

```
***HOLA BIENVENIDO A GAMEDESIGN***  
-Antes de la contruccion del .apk y .exe considere lo siguiente:  
*****  
-descomprimir el archivo videojuego.zip  
-copiar y pegar el fichero .json en la carpeta raiz del videojuego  
-cambiar el nombre del fichero **.json por GameDesign.json  
-Abrir el videojuego con Corona SDK.  
-Y listo, contruya su videojuego para android y Windows Desktop.  
*****
```

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Una vez terminado lo anterior, abrimos el el archivo **main** del videojuego diseñado:

Figura 39. Abriendo videojuego diseñado



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

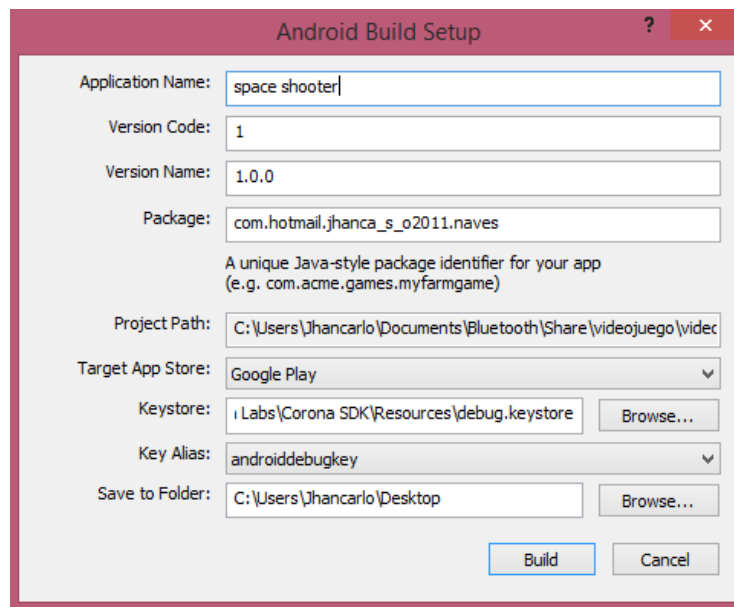
Figura 40. Menú principal del videojuego diseñado



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

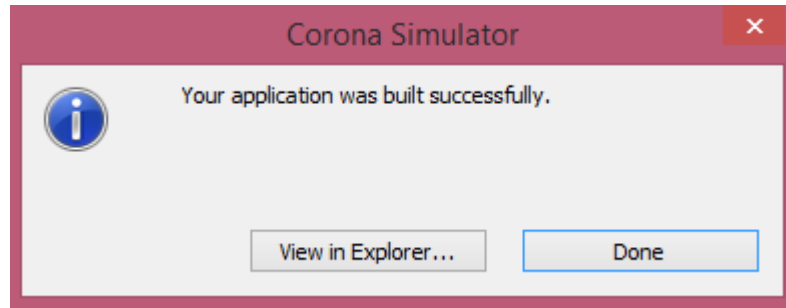
4.3.5.4. Construcción del .apk para Android del videojuego diseñado

Figura 41. Construyendo videojuego diseñado para Android



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 42. Construcción exitosa de la aplicación para Android



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

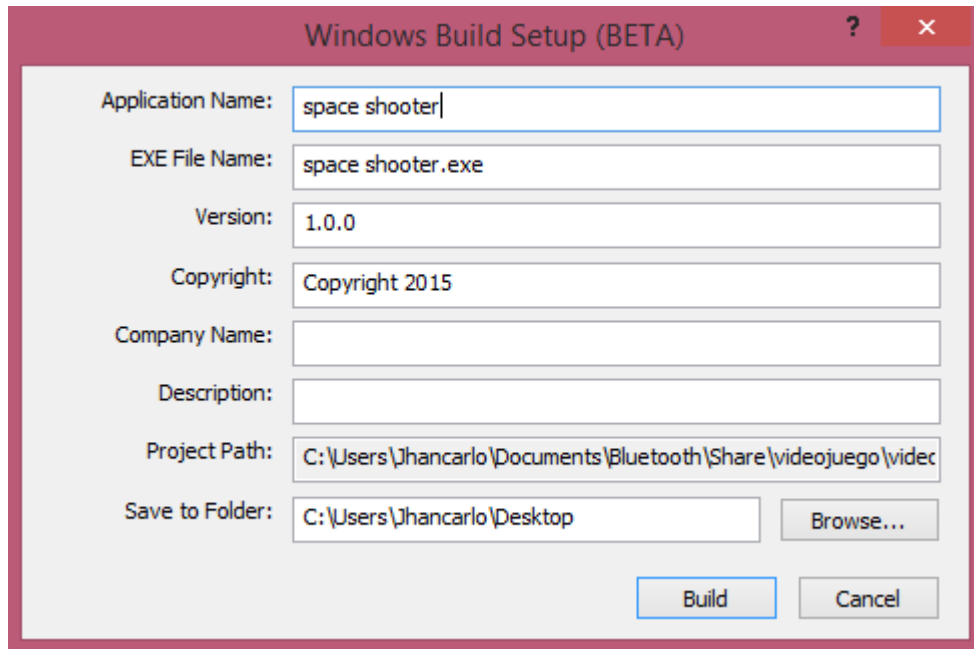
Figura 43. Ejecución del videojuego diseñado para Android



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

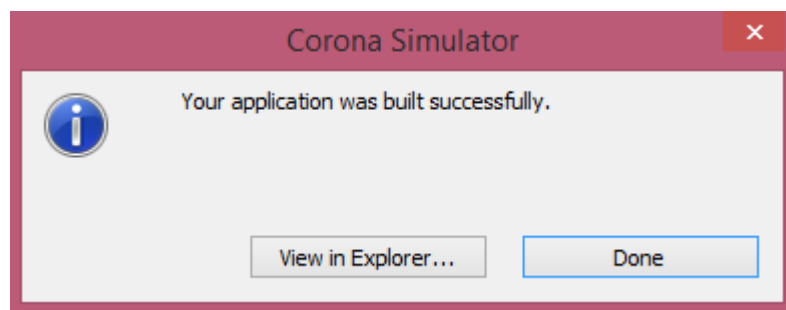
4.3.5.5. Construcción del .exe para Windows desktop del videojuego diseñado

Figura 44. Construyendo videojuego diseñado para Windows Desktop



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 45. Construcción exitosa de la aplicación para Windows desktop



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 46. Ejecución del videojuego en Windows desktop



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

4.3.5.6. Prueba piloto

La prueba piloto es una prueba rápida, que tiene como finalidad la usabilidad de una aplicación.

Se hizo una prueba a 10 estudiantes de Ingeniería de sistemas de la Universidad Nacional José María Arguedas.

A cada estudiante se le dio 3 elementos, para realizar las pruebas de usabilidad de la aplicación móvil “GameDesign”:

- ✓ Una Laptop con el Corona SDK instalado.
- ✓ Una Tablet Advance 9' pulgadas con sistema operativo Android 4.4.
- ✓ Instrucciones para crear el videojuego “Space Shooter”.

Una vez culminado la prueba, se pasó a realizar una encuesta de prueba de usabilidad de la aplicación móvil, a cada estudiante. Como a continuación se observa en la figura:

Figura 47. Encuesta prueba de usabilidad

Encuesta Prueba de Usabilidad: Aplicación móvil "GameDesign"

Las siguientes preguntas deberá responderlas teniendo en cuenta la siguiente escala de valoración:

Muy Difícil	Difícil	Mas o Menos	Fácil	Muy Fácil
1	2	3	4	5

Pregunta nº 1:

- ¿Cómo fue para usted navegar por las distintas interfaces de la aplicación? ____

Pregunta nº 2:

- ¿Cómo fue para usted diseñar el videojuego "Space Shooter"? ____

Pregunta nº 3:

- ¿Cómo fue para usted construir la aplicación para Android "Space Shooter.apk"? ____

Pregunta nº 4:

- ¿Cómo fue para usted construir la aplicación para Windows "Space Shooter.exe"? ____

Pregunta nº 5:

- En general , ¿Cómo considera el uso de la aplicación móvil? ____

¿Tiene alguna sugerencia para mejorar la aplicación móvil "GameDesign"?

Si ____ no ____

¿Cuál?

Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

A continuación se muestra las instrucciones para diseñar el videojuego "Space Shooter"

INSTRUCCIONES PARA DISEÑAR EL VIDEOJUEGO “SPACE SHOOTER”

1. En la Tablet que se le proporcionó, busque la aplicación “**GameDesign**”, el icono se muestra a continuación:



2. Luego de encontrarlo selecciónelo.
3. En el menú principal seleccione la opción “**Diseñar**”.
4. Seleccione “**Nuevo**”, ingrese el nombre “**Space Shooter**”, en el campo de texto, luego pulse el aspa.
5. Presione sobre “**Actores 2**”, ahora busque y seleccione el “**actor 7**”, saldrá un mensaje de alerta, presione el botón “**si**”, para añadir el actor, haga lo mismo para el “**actor 8**”.
6. Presione sobre “**Fondos**”, ahora busque y seleccione el “**fondo 5**”, saldrá un mensaje de alerta, presione el botón “**si**”, para añadir el fondo.
7. Presione sobre el botón “**visualizar**”, que se encuentra en la parte inferior.
8. En objetos, seleccione “**ACTORES2**”, luego presione sobre el “**actor 7**”, pulse el escenario para que se cree el actor en la escena.
9. Ahora pulse sobre el actor creado, diríjase a propiedades y seleccione “**OBJETO**”, luego presione sobre “**Control1**”, saldrá un mensaje de alerta, pulse sobre el botón “**eje x**”, luego presione sobre el botón “**back**”, de la parte inferior.
10. Vuelva a seleccionar el actor creado, diríjase a propiedades y seleccione “**DISPARO**”.
11. Presione sobre “**tiempo creación**”, en la caja de texto ingrese el valor de 0.5, y luego presione sobre el aspa. Ahora presione sobre “**Destino Y**”, e ingrese el valor de -100, presione sobre el aspa. Y por ultimo presione sobre “**Speed**”, e ingrese el valor de 5, presione sobre el aspa, luego presione sobre el botón “**back**”, de la parte inferior.
12. Ahora diríjase a propiedades y seleccione “**GRAVEDAD**”, luego presione sobre “**Gravedad X**”, e ingrese el valor de cero, presione sobre el aspa, después presione sobre “**Gravedad Y**”, también ingrese el valor de cero, presione sobre el aspa, luego presione sobre el botón “**back**”, de la parte inferior.
13. Ahora diríjase a propiedades y seleccione “**MAPA**”, luego presione sobre “**Escenas Vertical**”, ingrese el valor de 10, presione el aspa, después

- presione sobre "**Orientación**", saldrá un mensaje de alerta, presione sobre el botón "**vertical**", ahora presione sobre "**Speed**", ingrese el valor de 2, presione el aspa, y por ultimo presione sobre "**Audio**", seleccione "**Audio 2**", saldrá un mensaje de alerta, presione sobre el botón "**si**", luego presione sobre el botón "**back**", de la parte inferior.
14. Ahora si observa, debajo de los objetos existen 4 flechas (**arriba, abajo, izquierda y derecha**), son para desplazarnos a lo largo del mapa, que en el paso 13 se ingreso un valor de 10, referente a 10 escenas verticales.
 15. Presione la flecha "**arriba**", observara que se desplazo una escena hacia arriba, seleccione el "**actor 8**" y pulse el escenario para que se cree en la escena, este actor ahora será el enemigo, ahora pulse sobre el enemigo creado, diríjase a propiedades y seleccione "**Animación**", luego presione sobre "**Speed**", ingrese el valor de 1, presione el aspa, luego presione sobre "**Destino Y**", ingrese el valor de 400, presione el aspa, luego presione sobre "**Rotación**", saldrá un mensaje de alerta, presione el botón "**Horario**", y por ultimo presione sobre "**Aleatorio**", saldrá un mensaje de alerta, presione sobre el botón "**eje x**", luego presione sobre el botón "**back**", de la parte inferior.
 16. Continuamos sobre esta escena, seleccione el enemigo y diríjase a propiedades y seleccione "**COLISIONES**", ahora presione sobre "**Opciones**", saldrá un mensaje de alerta, presione el botón "**Eliminar Objeto**", luego presione sobre el botón "**back**", de la parte inferior, ahora volveremos a la primera escena, para ello presione la fecha "**abajo**".
 17. Seleccione el actor que creo por primera vez, diríjase a propiedades y seleccione "**FÍSICA**", ahora presione "**Cuerpo**", saldrá un mensaje de alerta, presione el botón "**Dinamico**", luego presione sobre el botón "**back**", de la parte inferior.
 18. Nuevamente seleccione el actor ,diríjase a propiedades y seleccione "**COLISIONES**", ahora presione sobre "**Opciones**", saldrá un mensaje de alerta, presione el botón "**Eliminar Objeto**", luego presione sobre el botón "**back**", de la parte inferior.
 19. Ahora ya se termino con todo el diseño, solo queda ejecutar y exportar el videojuego, para ello diríjase a propiedades y seleccione "**OPCIONES**", ahora presione sobre "**Ejecutar**", saldrá un mensaje de alerta, presione el botón "**si**".
 20. Observara que el videojuego esta funcionando tal como se diseño, en esta escena existen 2 opciones, en la parte superior izquierda el botón "**back**", para seguir diseñando el videojuego, y en la parte superior derecha un "**correo**", ahora para exportar el videojuego presione el **correo**.

21. Como podrá observar se lista opciones por el cual se desea exportar el código fuente, seleccione la opción “**Bluetooth**”, saldrá un mensaje de alerta para activar el bluetooth, presione el botón “**Activar**”, se lista los dispositivos que tiene activo el **bluetooth**.
22. Ahora en la laptop JHANCARLO-PC, diríjase a la siguiente ruta: **C:\Users\Jhancarlo\Documents\Bluetooth\Share**, una vez aquí diríjase a la tablet y seleccione **JHANCARLO-PC**, para enviar el código fuente del videojuego, a la ruta indicada en la laptop, espere que termine de enviarse por completo el código fuente, puede ver el progreso de envío en la tablet .
23. Una vez exportado por completo los recursos del videojuego, se tiene lo siguiente:
 - **LEEME.txt**
 - **Space Shooter.json**
 - **videojuego.zip**
24. Abra y lea con detenimiento el archivo **LEEME.txt**, haga lo que se indica.
25. Y listo ya tiene su videojuego multiplataforma:
 - Space Shooter.apk para Android.
 - Space Shooter.exe para Windows Desktop.

RESULTADOS

La prueba de usabilidad de la aplicación móvil “**GameDesign**” fue realizada durante los días 8,9,10 de diciembre, la muestra estuvo conformada por estudiantes mujeres y hombres entre 18 y 23 años, los cuales tenían conocimiento en el uso de dispositivos móviles, al iniciar la prueba solo se les entrego la tablet, la laptop y las instrucciones, en ningún momento se les proporcionó capacitación sobre la aplicación móvil en general la duración de la prueba tuvo un promedio de 20,5 min, luego se les realizó la encuesta la cual proporcionó los siguientes resultados:

- Para los encuestados, navegar por las distintas interfaces fue muy fácil:

Tabla 61. resultados pregunta 1

Estudiantes	Muy Dificil	Difícil	Mas o Menos	Fácil	Muy Fácil
1				x	
2					x
3					x
4					x
5				x	
6					x
7					x
8				x	
9					x
10					x

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Fácil 30 %
- Muy fácil 70 %

- El diseño del videojuego “**Space Shooter**”, fueron percibidas por los usuarios como fáciles:

Tabla 62. resultados pregunta 2

Estudiantes	Muy Difícil	Difícil	Mas o Menos	Fácil	Muy Fácil
1			x		
2				x	
3					x
4			x		
5				x	
6					x
7				x	
8				x	
9			x		
10				x	

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Mas o menos 30 %
- Fácil 50%
- Muy fácil 20 %

- Construir la aplicación “**Space Shooter.apk**” para Android, fue muy fácil:

Tabla 63. resultados pregunta 3

Estudiantes	Muy Difícil	Difícil	Mas o Menos	Fácil	Muy Fácil
1					x
2					x
3					x
4				x	
5					x
6					x
7				x	
8				x	
9					x
10					x

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Fácil 30 %
- Muy fácil 70 %

- Construir la aplicación “**Space Shooter.exe**” para Windows Desktop, fue muy fácil.

Tabla 64. resultados pregunta 4

Estudiantes	Muy Difícil	Difícil	Mas o Menos	Fácil	Muy Fácil
1				x	
2					x
3					x
4					x
5					x
6					x
7					x
8				x	
9					x
10					x

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Fácil 20 %
- Muy fácil 80 %

- Los usuarios en general consideran el uso de la aplicación, como muy fácil.

Tabla 65. resultados pregunta 5

Estudiantes	Muy Difícil	Difícil	Mas o Menos	Fácil	Muy Fácil
1				x	
2					x
3				x	
4					x
5			x		
6					x
7					x
8				x	
9					x
10					x

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

- Mas o menos 10 %
- Fácil 30 %
- Muy fácil 60 %

CONCLUSIONES

- a) Luego de las múltiples etapas superadas durante el proceso de desarrollo de la aplicación móvil “GameDesign”, se concluye que los objetivos han sido cumplidos a cabalidad.
- b) La interfaz amigable y fácil uso de la aplicación móvil, se cumplió .
- c) El diseño y construcción de videojuegos multiplataforma alcanzados satisfactoriamente.
- d) En el desarrollo de aplicaciones móviles, es muy importante la elaboración de diagramas de caso de uso, los cuales permiten visualizar de distintas perspectivas el proyecto, así de esta manera tener una mejor comprensión de la aplicación que se pretende desarrollar.
- e) La utilización de la metodología Mobile-D, permite agilizar y organizar el trabajo, de tal manera para que en futuras modificaciones se tenga una excelente base de documentación.
- f) El uso del framework Corona SDK, permitió que la aplicación “GameDesign” sea multiplataforma, este framework es muy potente, esta mas destinado al desarrollo de videojuegos, porque tiene un motor avanzado de física y animaciones.
- g) El uso de JSON , fue fundamental en el desarrollo de la aplicación móvil, ya que gracias a la lectura y escritura de estos ficheros es posible guardar, cargar y eliminar datos de cada uno de los videojuegos que se pretende diseñar.

RECOMENDACIONES

Si en un futuro se pretende mejorar la aplicación móvil “**GameDesign**”, se recomienda descargar la última versión del framework Corona SDK, porque las versiones de este framework, están disponibles semanalmente.

Si existe alguna librería que no funcione, es porque las últimas versiones del framework actualizaron sus librerías, y todo este proceso de migración está disponible en la página oficial de Corona Labs, por ejemplo, en antiguas versiones del Corona SDK para el manejo de escenas se usaba el famoso “**storyboard**”, ahora con la versión actual con el que se desarrolló la aplicación móvil del presente proyecto, referente a la versión Corona SDK V2015.2777, ya no se usa “**storyboard**”, sino “**composer**”, que es la que actualmente se encarga del manejo de escenas, en Corona SDK.

Si se pretende desarrollar videojuegos en este framework, existe la posibilidad de desarrollar para **ios**(.ipa), **Windows phone**(.app), **MAC** (.mdg), el presente proyecto no considera las plataformas anteriores porque, no se cuenta con una licencia, ya que la construcción de las aplicaciones móviles para estas plataformas es restringido, necesariamente se tiene que pagar una licencia para poder distribuir la aplicación, y es muy costosa, por ello se optó desarrollar para Android ya que es gratuito y de código abierto.

REFERENCIAS BIBLIOGRÁFICAS

- Abrahamsson, P., & Hanhineva, A. (2004). *Mobile-D: An Agile Approach for Mobile Application Development*.
- Carlos José M.(2015), *Tendencias de Consumo de Videojuegos*. Recuperado de: <http://www.altadensidad.com/?p=50894>
- Corona SDK (2014). *Language and API Reference*. Disponible en: <http://www.coronalabs.com>
- Cristhian Obando. (2010, 25 de Noviembre). *Symbian OS*. Recuperado de: <http://symbianparamoviles.blogspot.com/2010/11/sistema-operativo-symbian.html>
- David Orosco. (2014,16 de Diciembre). *Definición de Windows Phone*. Recuperado de: <http://conceptodefinicion.de/windows-phone/>
- David Orosco (2011, 01 de Junio). *Definición de Android*. Recuperado de: <http://conceptodefinicion.de/android/>
- D. E. Avison., & G. Fitzgerald. (2006). *Information system development. Maidenhead: McGraw-Hill Education*.
- K. Beck y J. Zapata Martínez. (2002). *Una Explicación de la programación extrema*.
- K. Beck., & C. Andres. (2004). *Extreme programming*.
- K. Schwaber., & M. Beedle. (2002). *Agile software development with Scrum*.
- Leandro Alegsa. (2010,19 de Noviembre). *Definición de BlackBerry OS (sistema operativo)*. Recuperado de: <http://www.alegsa.com.ar/Dic/blackberry%20os.php>
- L. Rising., & N. S. Janoff. (2000). *The Scrum software development process for small teams, vol. 17, pp. 26-32*.
- Leandro Ortega. *IOS*. Recuperado de: <http://www.actualidadiphone.com/category/ios/>
- Marcos Sánchez(2013), *App Annie*, Recuperado de: www.hobbyconsolas.com
- Ministerio de la Protección Social. (2014). 3.1 *Los Estándares De Calidad*. Recuperado de: <http://mps1.minproteccionsocial.gov.co/evtmedica/linea%207/3.1estandares.html>
- O. Salo y Valtion teknillinen tutkimuskeskus. (2006). *Enabling software process improvement in agile software development teams and organisations*.

Project Management Institute. (2014). *¿Qué es un estándar?*. Recuperado de:
<http://americalatina.pmi.org/latam/pmbokguideandstandards/whatisastandar.aspx>

Roberto Lerusalimschy, Luis Henrique de Figueiredo y Waldemar Celes. (2015). *Lua 5.3 Reference Manual*. Más información en: <http://www.Lua.org>

S. B. Kaleel., & S. Harishankar. (2013). *Applying Agile Methodology in Mobile Software Engineering: Android Application Development and its Challenge*.

Corona SDK, disponible en : <http://developer.coronalabs.com/downloads/windows>

Sublime Text 3 disponible en <http://www.sublimetext.com/3>

jdk-8u20-windows-i586 disponible en:
<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

ANEXOS

Deberá presentarse información adicional que sea relevante (mapas, cuadros, graficas, figuras, vistas fotográficas, estadísticas, etc.) que establezca una mejor comprensión de la investigación.

MANUAL DE USUARIO

A continuación se detalla el funcionamiento de la aplicación móvil GameDesign:

Figura 48. Pantalla de arranque de la aplicación móvil GameDesign



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 66. Acciones: Iniciar aplicación

CASO DE USO:	Iniciar aplicación
ACTORES:	Usuario
PROPÓSITO:	Dar paso al menú principal de la aplicación móvil
RESUMEN:	El usuario permite a la aplicación dar paso al menú principal para poder interactuar en los diferentes capítulos.
TIPO:	Primario

PRE-CONDICIONES:	Debe de haber finalizado la carga de la escena Splash.
ACCIONES	
ACTOR	APLICACIÓN
1. El usuario debe de tocar la pantalla del dispositivo.	2. La aplicación da paso al menú principal.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 49. Pantalla de bienvenida



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 67. Acciones: Selección de capítulos

CASO DE USO:	Selección de capítulos
ACTORES:	Usuario
PROPÓSITO:	Navegar por los distintos capítulos de la aplicación móvil
RESUMEN:	El usuario podrá seleccionar cualquier capítulo del menú principal
TIPO:	Primario
PRE-CONDICIONES:	Caso de uso inmediato después de haber tocado la pantalla para ingresar al menú principal

ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa el capítulo introducción.	2) La aplicación muestra los seis capítulos de introducción.
3) El usuario pulsa el capítulo diseñar.	4) La aplicación muestra la escena donde se crea el nombre del videojuego a diseñar.
5) El usuario pulsa el capítulo mini juego.	6) La aplicación muestra un videojuego completamente funcional.
7) El usuario pulsa el capítulo opciones.	8) La aplicación muestra la escena en la cual se puede ver los créditos del programador.
9) El usuario pulsa el capítulo salir.	10) Muestra un mensaje de dialogo para confirmar si se desea salir de la aplicación.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 50. Pantalla menú principal



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 68. Acciones: seleccionar capítulo imágenes

CASO DE USO:	Seleccionar capítulo Imágenes
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de imágenes en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a imágenes.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	
ACTOR	APLICACIÓN
<p>1) El usuario pulsa la opción de Imágenes.</p> <p>3) El usuario pulsa la opción de “siguiente”.</p> <p>5) El usuario pulsa la opción de “anterior”.</p> <p>7) El usuario pulsa la opción “automático”.</p> <p>9) El usuario pulsa la opción “atrás”.</p>	<p>2) La aplicación muestra la escena de imágenes.</p> <p>4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.</p> <p>6) Se muestra las líneas de código mostradas anteriormente menos la actual, también la simulación va sincronizada.</p> <p>8) La aplicación muestra automáticamente el código fuente, la descripción y el simulador.</p> <p>10) La aplicación vuelve a la escena de introducción.</p>

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 51. Pantalla capítulo imágenes



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 69. Acciones: Seleccionar capítulo texto

CASO DE USO:	Seleccionar capítulo Texto
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de textos en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a textos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la opción de texto.	2) La aplicación muestra la escena de Texto.
3) El usuario pulsa la opción de "siguiente".	4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.

5) El usuario pulsa la opción de “anterior”.	6) Se muestra las líneas de código mostradas anteriormente menos la actual, también la simulación va sincronizada.
7) El usuario pulsa la opción “automático”.	8) La aplicación muestra automáticamente el código fuente, la descripción y el simulador.
9) El usuario pulsa la opción “atrás”.	10) La aplicación vuelve a la escena de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 52. Pantalla capitulo texto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 70. Acciones: *Seleccionar capitulo toques*

CASO DE USO:	Seleccionar capitulo Toques
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de toques de escena en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a eventos de toque de escena.
TIPO:	Primario

PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	
ACTOR	APLICACIÓN
<p>1) El usuario pulsa la opción de Toques.</p> <p>3) El usuario pulsa la opción de “siguiente”.</p> <p>5) El usuario pulsa la opción de “anterior”.</p> <p>7) El usuario pulsa la opción “automático”.</p> <p>9) El usuario pulsa la opción “atrás”.</p>	<p>2) La aplicación muestra la escena de Toques.</p> <p>4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.</p> <p>6) Se muestra las líneas de código mostradas anteriormente menos la actual, también la simulación va sincronizada.</p> <p>8) La aplicación reproduce automáticamente el código fuente, la descripción y el simulador.</p> <p>10) La aplicación vuelve a la escena de introducción.</p>

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 53. Pantalla capitulo toques



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 71. Acciones: Seleccionar capítulo sonido

CASO DE USO:	Seleccionar capítulo Sonido
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de sonidos en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a sonidos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la opción de Sonido.	2) La aplicación muestra la escena de Sonido.
3) El usuario pulsa la opción de "siguiente".	4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.
5) El usuario pulsa la opción de "anterior".	6) Se muestra las líneas de código mostradas anteriormente menos la actual, también la simulación va sincronizada.
7) El usuario pulsa la opción "automático".	8) La aplicación reproduce automáticamente el código fuente, la descripción y el simulador.
9) El usuario pulsa la opción "atrás".	10) La aplicación vuelve a la escena de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 54. Pantalla capitulo sonido



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 72. Acciones: Seleccionar capitulo animación

CASO DE USO:	Seleccionar capitulo Animación
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de animaciones en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a animaciones.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la opción de Animación.	2) La aplicación muestra la escena de Animación.
3) El usuario pulsa la opción de "siguiente".	4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.
5) El usuario pulsa la opción de "anterior".	6) Se muestra las líneas de código mostradas anteriormente menos la

7) El usuario pulsa la opción "automático".	actual, también la simulación va sincronizada.
9) El usuario pulsa la opción "atrás".	8) La aplicación reproduce automáticamente el código fuente, la descripción y el simulador. 10) La aplicación vuelve a la escena de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 55. Pantalla capítulo animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 73. Acciones: *Seleccionar capítulo física*

CASO DE USO:	Seleccionar capítulo Física
ACTORES:	Usuario
PROPÓSITO:	Ver contenido del manejo de la física en Corona SDK
RESUMEN:	La aplicación muestra la simulación del código fuente respecto a animaciones.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de introducción.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la opción de Física.	2) La aplicación muestra la escena de Física.
3) El usuario pulsa la opción de "siguiente".	4) Se muestra una línea de código, y el parte superior de la escena, la descripción de lo que hace el código, y también se visualiza la ejecución en el simulador.
5) El usuario pulsa la opción de "anterior".	6) Se muestra las líneas de código mostradas anteriormente menos la actual, también la simulación va sincronizada.
7) El usuario pulsa la opción "automático".	8) La aplicación reproduce automáticamente el código fuente, la descripción y el simulador.
9) El usuario pulsa la opción "atrás".	10) La aplicación vuelve a la escena de introducción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 56. Pantalla capítulo física



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 74. Acciones: Nuevo videojuego

CASO DE USO:	Nuevo videojuego
ACTORES:	Usuario
PROPÓSITO:	Crear un nuevo videojuego
RESUMEN:	Opción para poder crear un nuevo videojuego
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario pulsa la opción de Nuevo.	2) La aplicación muestra la escena de nuevo videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 57. Pantalla crear videojuego



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 75. Acciones: Abrir videojuego

CASO DE USO:	Abrir videojuego
ACTORES:	Usuario
PROPÓSITO:	Poder abrir un videojuego
RESUMEN:	Editar y seguir diseñando un videojuego creado anteriormente.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario pulsa la opción de Abrir.	2)La aplicación muestra la escena donde se lista los videojuegos diseñados.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 58. Pantalla abrir videojuego



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 76. Acciones: Seleccionar Actores 1

CASO DE USO:	Seleccionar Actores 1
ACTORES:	Usuario
PROPÓSITO:	Seleccionar uno o más objetos
RESUMEN:	Los objetos seleccionados podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario selecciona el tabview 'Actores 1. 3) El usuario selecciona el(los) objeto(s). 5) El usuario pulsa el botón 'sí'. 7) El usuario pulsa el botón 'no'". 9) El usuario selecciona un objeto ya añadido. 11) El usuario pulsa botón 'sí'. 13) El usuario pulsa botón 'no'	2) La aplicación muestra el conjunto de objetos disponibles. 4) La aplicación muestra un mensaje de alerta para poder confirmar la selección del objeto. 6) La aplicación aplica opacidad al objeto seleccionado. 8) La aplicación cierra el mensaje de alerta y cancela la selección. 10) La aplicación muestra un mensaje de alerta para poder confirmar si se desea quitar el objeto. 12) La aplicación quita la selección del objeto. 14) La aplicación mantiene la selección del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 59. Pantalla Actores 1



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 77. Acciones: Seleccionar Actores 2

CASO DE USO:	Seleccionar Actores 2
ACTORES:	Usuario
PROPÓSITO:	Seleccionar uno o más objetos
RESUMEN:	Los objetos seleccionados podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario selecciona el tabview 'Actores 2. 3) El usuario selecciona el(los) objeto(s). 5) El usuario pulsa el botón 'si'. 7) El usuario pulsa el botón 'no'".	2) La aplicación muestra el conjunto de objetos disponibles. 4) La aplicación muestra un mensaje de alerta para poder confirmar la selección del objeto. 6) La aplicación aplica opacidad al objeto seleccionado. 8) La aplicación cierra el mensaje de alerta y cancela la selección.

9) El usuario selecciona un objeto ya añadido.	10) La aplicación muestra un mensaje de alerta para poder confirmar si se desea quitar el objeto.
11) El usuario pulsa botón 'sí'.	12) La aplicación quita la selección del objeto.
13) El usuario pulsa botón 'no'	14) La aplicación mantiene la selección del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 60. Pantalla Actores 2



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 78. Acciones: Seleccionar baldosa

CASO DE USO:	Seleccionar baldosa
ACTORES:	Usuario
PROPÓSITO:	Seleccionar una o más baldosas
RESUMEN:	Las baldosas seleccionadas podrán estar disponibles en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	

ACTOR	APLICACIÓN
1)El usuario selecciona el tabview 'Baldosas'.	2)La aplicación muestra el conjunto de baldosas disponibles.
3) El usuario selecciona la(s) baldosa(s).	4) La aplicación muestra un mensaje de alerta para poder confirmar la selección de la baldosa.
5) El usuario pulsa el botón 'si'.	6) La aplicación aplica opacidad la baldosa seleccionada.
7) El usuario pulsa el botón 'no'".	8) La aplicación cierra el mensaje de alerta y cancela la selección.
9) El usuario selecciona una baldosa ya añadida.	10) La aplicación muestra un mensaje de alerta para poder confirmar si se desea quitar la baldosa.
11) El usuario pulsa botón 'si'.	12) La aplicación quita la selección de la baldosa.
13) El usuario pulsa botón 'no'	14) La aplicación mantiene la selección de la baldosa.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 61. Pantalla Baldosas



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 79. Acciones: Seleccionar fondo

CASO DE USO:	Seleccionar fondo
ACTORES:	Usuario
PROPÓSITO:	Seleccionar imagen de fondo
RESUMEN:	El fondo seleccionado se podrá visualizar en la interfaz de diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Diseñar.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario selecciona el tabview 'Fondos. 3) El usuario selecciona una imagen de fondo. 5) El usuario pulsa el botón 'si'. 7) El usuario pulsa el botón 'no'".	2) La aplicación muestra el conjunto de fondos disponibles. 4) La aplicación muestra un mensaje de alerta para poder confirmar la selección del fondo. 6) La aplicación guarda la imagen de fondo. 8) La aplicación tendrá una imagen por defecto si el usuario no selecciona ninguno.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 62. Pantalla fondos



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 80. Acciones: Listar Objetos

CASO DE USO:	Listar Objetos
ACTORES:	Usuario
PROPÓSITO:	Visualizar lista de Objetos
RESUMEN:	El usuario ya podrá crear objetos en la escena del diseño.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado Actores 1, Actores 2 , Baldosas.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la opción 'ACTORES 1, ACTORES 2 o BALDOSAS. 3) El usuario navega por el scroll. 5) El usuario pulsa el botón 'back'.	2) La aplicación muestra el conjunto de objetos disponibles. 4) La aplicación renderiza la lista objetos. 6) La aplicación vuelve al estado inicial para poder seleccionar otra opción.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 63. Despliegue Actores 1



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 64. Despliegue Actores 2



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 65. Despliegue Baldosas



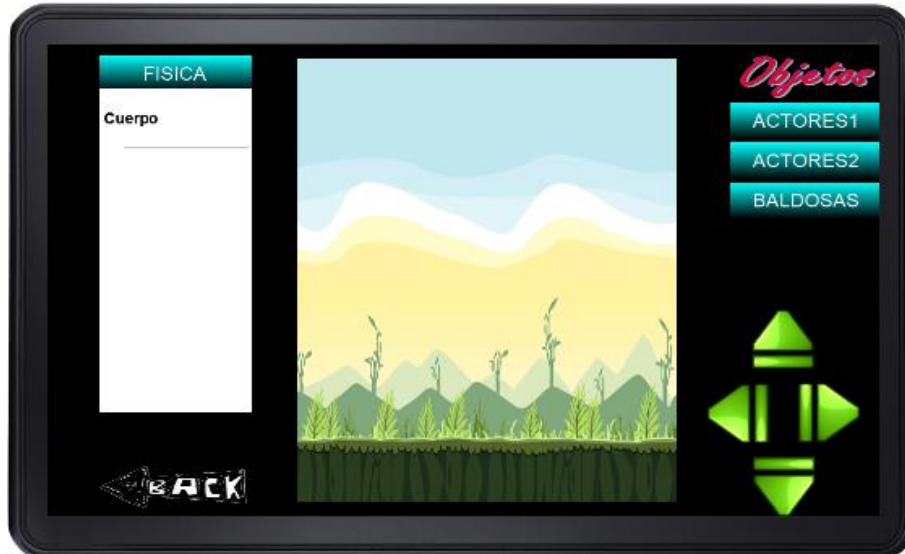
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 81. Acciones: Agregar física

CASO DE USO:	Agregar física
ACTORES:	Usuario
PROPÓSITO:	Agregar física a los objetos
RESUMEN:	El usuario ya podrá agregar física a los objetos de la escena.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado algún objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
<p>1) El usuario pulsa la propiedad de 'FÍSICA'.</p> <p>3) El usuario pulsa sobre un objeto creado en la escena.</p> <p>5) El usuario pulsa la opción 'cuerpo'.</p> <p>7) El usuario pulsa el botón 'dinámico'.</p> <p>9) El usuario pulsa el botón 'estático'</p>	<p>2) La aplicación muestra el conjunto de propiedades de física.</p> <p>4) La aplicación pone opaco al objeto seleccionado.</p> <p>6) La aplicación verifica si existe por lo menos un objeto en la escena, sino existe muestra un mensaje de alerta. Si existe un objeto creado, la aplicación verifica si el objeto esta seleccionado, sino muestra otro mensaje de alerta. Al estar creado y seleccionado el objeto, la aplicación muestra un mensaje de alerta.</p> <p>8) La aplicación define que el tipo de cuerpo del objeto será 'dinámico'.</p> <p>10) La aplicación define que el tipo de cuerpo del objeto será 'estático'.</p>

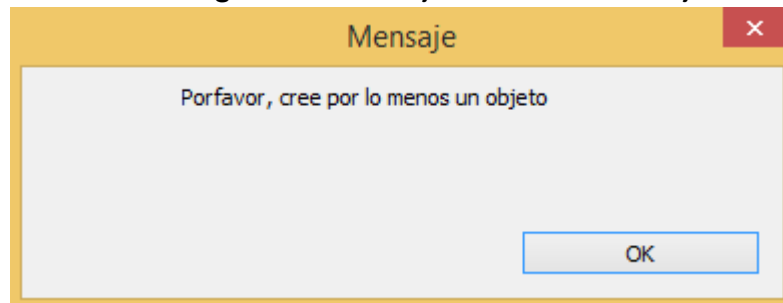
Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 66. Despliegue física



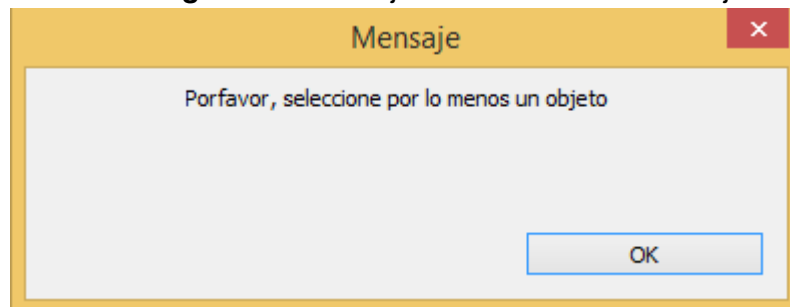
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 67. Mensaje de alerta crear objeto



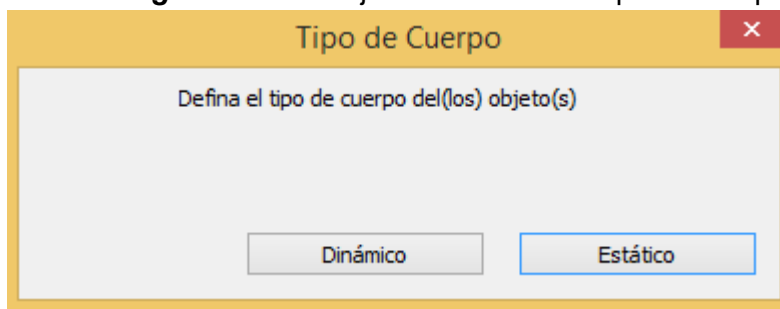
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 68. Mensaje de alerta seleccione objeto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 69. Mensaje de alerta definir tipo de cuerpo



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 82. Acciones: Definir velocidad de animación

CASO DE USO:	Definir velocidad de animación
ACTORES:	Usuario
PROPÓSITO:	Definir la velocidad de animación
RESUMEN:	Posibilidad de definir la velocidad de animación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'ANIMACIÓN.	2) La aplicación muestra el conjunto de propiedades de animación.
3) El usuario selecciona la opción 'Speed'.	4) La aplicación muestra un textfield para poder ingresar un valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa.	8) La aplicación guarda la propiedad de velocidad de animación.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 70. Definir velocidad de animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

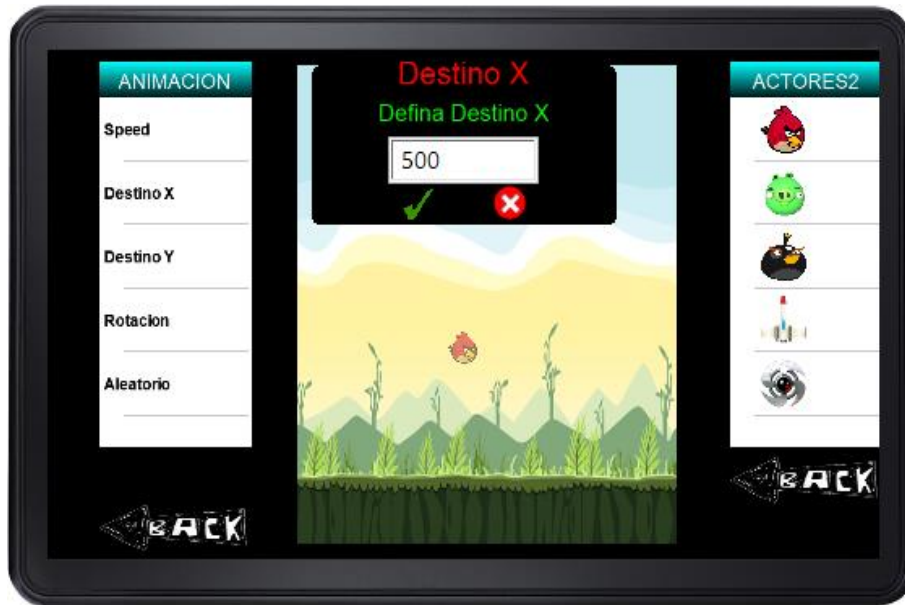
Tabla 83. Acciones: Definir destino 'x' de animación

CASO DE USO:	Definir destino 'x' de animación
ACTORES:	Usuario
PROPÓSITO:	Definir el destino 'x' de animación
RESUMEN:	Posibilidad de definir el destino 'x' de animación del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'ANIMACIÓN.	2) La aplicación muestra el conjunto de propiedades de animación.
3) El usuario selecciona la opción 'Destino X'.	4) La aplicación muestra un textfield para poder ingresar el valor numerico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa.	

9) El usuario pulsa la imagen 'equis'.	8) La aplicación guarda la propiedad del destino X de animación del objeto. 10) La aplicación oculta el textfield sin guardar el valor numérico.
--	---

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 71. Definir destino X de animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

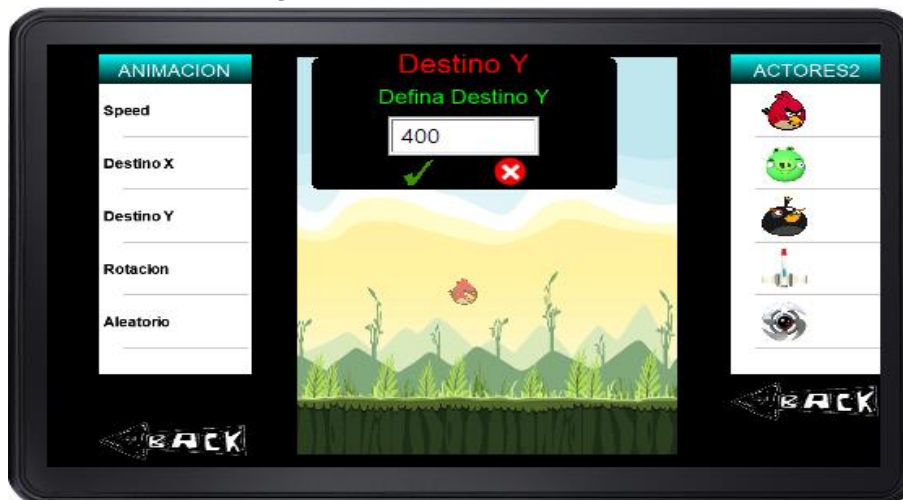
Tabla 84. Acciones: Definir destino 'y' de animación

CASO DE USO:	Definir destino 'y' de animación
ACTORES:	Usuario
PROPÓSITO:	Definir el destino 'y' de animación
RESUMEN:	Posibilidad de definir el destino 'y' de animación del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'ANIMACIÓN.	2) La aplicación muestra el conjunto de propiedades de animación.

3) El usuario selecciona la opción 'Destino Y'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad del destino Y de animación del objeto.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 72. Definir destino Y de animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

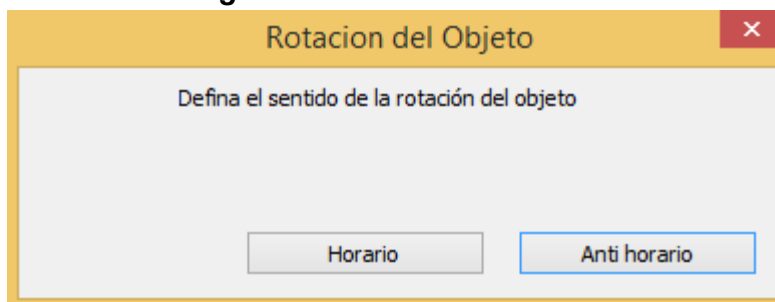
Tabla 85. Acciones: Definir rotación de animación

CASO DE USO:	Definir rotación de animación
ACTORES:	Usuario
PROPÓSITO:	Definir rotación de animación
RESUMEN:	Posibilidad de definir el sentido de rotación durante la animación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'ANIMACIÓN.	2) La aplicación muestra el conjunto de propiedades de animación.
3) El usuario selecciona la opción 'Rotación'.	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'Horario'.	6) La aplicación guarda la propiedad de rotación en sentido horario del objeto.
7) El usuario selecciona el botón 'anti horario'.	8) La aplicación guarda la propiedad de rotación en sentido anti horario del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 73. Definir rotación de animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

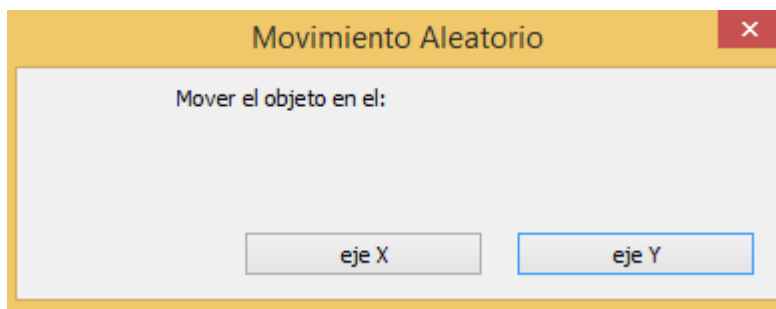
Tabla 86. Acciones: Definir movimiento aleatorio de animación

CASO DE USO:	Definir movimiento aleatorio de animación
ACTORES:	Usuario
PROPÓSITO:	Definir movimiento aleatorio de animación del objeto.
RESUMEN:	Posibilidad de definir un movimiento aleatorio durante la animación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'ANIMACIÓN.	2) La aplicación muestra el conjunto de propiedades de animación.
3) El usuario selecciona la opción 'Aleatorio.	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'eje x'.	6) La aplicación guarda la propiedad de movimiento aleatorio en el eje 'x' del objeto.
7) El usuario selecciona el botón 'eje y'.	8) La aplicación guarda la propiedad de de movimiento aleatorio en el eje 'y' del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 74. Definir movimiento aleatorio de animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 87. Acciones: Definir ancho del objeto

CASO DE USO:	Definir ancho del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir el ancho del objeto
RESUMEN:	Posibilidad de definir el ancho del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO'.	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Ancho'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad del ancho del objeto.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 75. Definir ancho del objeto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 88. Acciones: Definir altura del objeto

CASO DE USO:	Definir altura del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir la altura del objeto
RESUMEN:	Posibilidad de definir la altura del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO.'	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Altura.'	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa.'	8) La aplicación guarda la propiedad de la altura del objeto.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Figura 76. Definir altura del objeto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 89. Acciones: Definir rotación del objeto

CASO DE USO:	Definir rotación del objeto
ACTORES:	Usuario
PROPÓSITO:	Definir la rotación del objeto
RESUMEN:	Posibilidad de definir la rotación del objeto.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO.	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Rotación.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa.	8) La aplicación guarda la propiedad de rotación del objeto.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 77. Defina la rotación del objeto



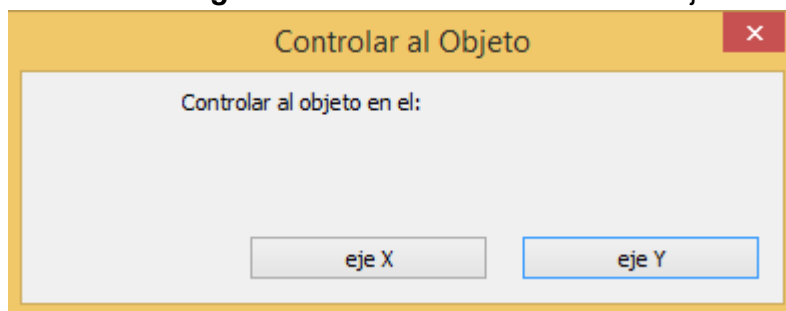
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 90. Acciones: Definir control 1

CASO DE USO:	Definir control 1
ACTORES:	Usuario
PROPÓSITO:	Definir control 1 del objeto.
RESUMEN:	Posibilidad de definir el control del objeto en la pantalla.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO.'	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Control 1.'	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'eje x'.	6) La aplicación guarda la propiedad de control en el eje 'x' del objeto.
7) El usuario selecciona el botón 'eje y'.	8) La aplicación guarda la propiedad de control en el eje 'y' del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 78. Defina control 1 del objeto



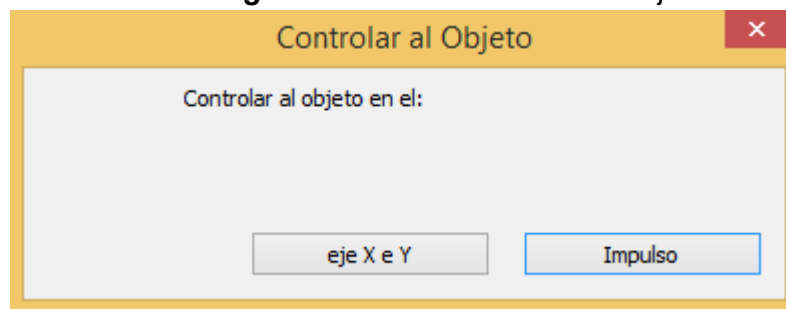
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 91. Acciones: Definir control 2

CASO DE USO:	Definir control 2
ACTORES:	Usuario
PROPÓSITO:	Definir control 2 del objeto.
RESUMEN:	Posibilidad de definir el control del objeto en la pantalla.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO.'	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Control 2.'	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'eje xy'.	6) La aplicación guarda la propiedad de control en el eje 'xy' del objeto.
7) El usuario selecciona el botón 'impulso'.	8) La aplicación guarda la propiedad de impulso del objeto.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 79. Defina control 2 del objeto



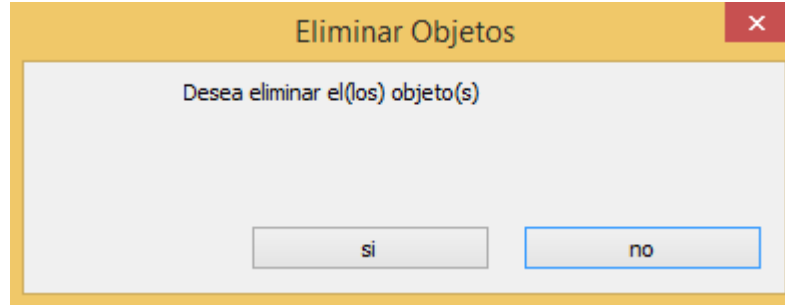
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 92. Acciones: eliminar objeto

CASO DE USO:	Eliminar objeto
ACTORES:	Usuario
PROPÓSITO:	Eliminar objetos
RESUMEN:	Posibilidad de eliminar los objetos creados en la escena
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'OBJETO.'	2) La aplicación muestra el conjunto de propiedades del objeto.
3) El usuario selecciona la opción 'Eliminar'.	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'sí'.	6) La aplicación elimina el objeto de la escena.
7) El usuario selecciona el botón 'no'.	8) La aplicación mantiene el objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 80. Eliminar objetos



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 93. Acciones: Definir tiempo creación disparo

CASO DE USO:	Definir tiempo creación disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir el tiempo de creación
RESUMEN:	Posibilidad de definir cada cuanto tiempo se crea el disparo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'DISPARO.'	2) La aplicación muestra el conjunto de propiedades de disparo.
3) El usuario selecciona la opción 'Tiempo creación'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de tiempo creación del disparo.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 81. Creación bala en segundos



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

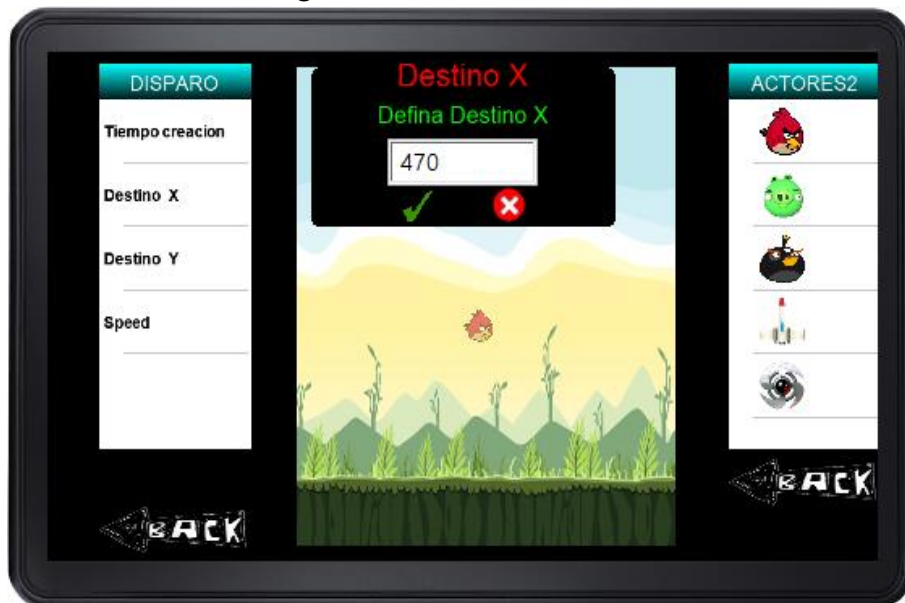
Tabla 94. Acciones: Definir destino 'x' disparo

CASO DE USO:	Definir destino 'x' disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir destino 'x' disparo.
RESUMEN:	Posibilidad de definir el destino 'x' del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'DISPARO'.	2) La aplicación muestra el conjunto de propiedades de disparo.
3) El usuario selecciona la opción 'Destino X'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	

7) El usuario pulsa la imagen 'aspa'.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
9) El usuario pulsa la imagen 'equis'.	8) La aplicación guarda la propiedad de destino 'x' del disparo. 10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 82. Defina destino X de la bala



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

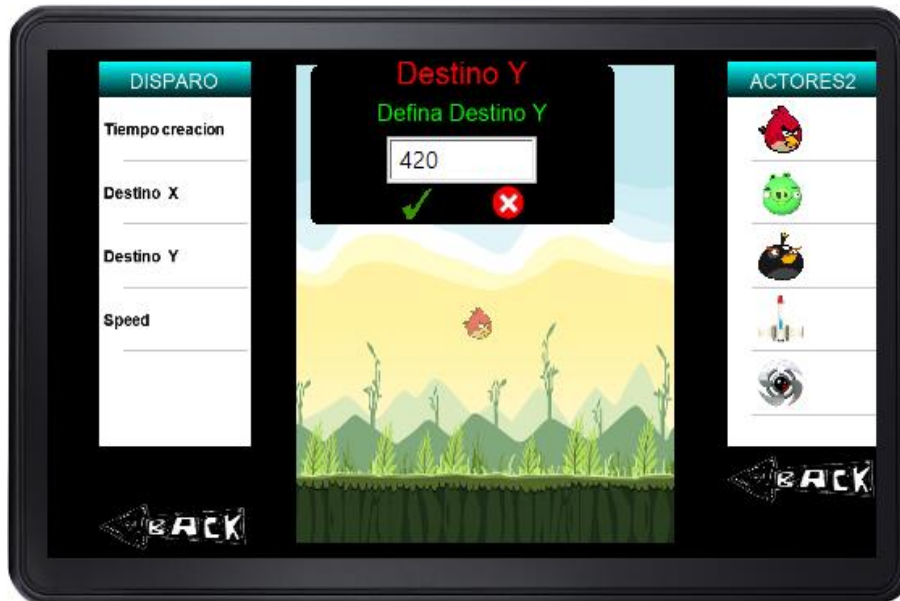
Tabla 95. Acciones: Definir destino 'y' disparo

CASO DE USO:	Definir destino 'y' disparo.
ACTORES:	Usuario
PROPÓSITO:	Definir destino 'y' disparo.
RESUMEN:	Posibilidad de definir el destino 'y' del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'DISPARO'.	2) La aplicación muestra el conjunto de propiedades de disparo.
3) El usuario selecciona la opción 'Destino Y'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de destino 'y' del disparo.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 83. Defina destino Y de la bala



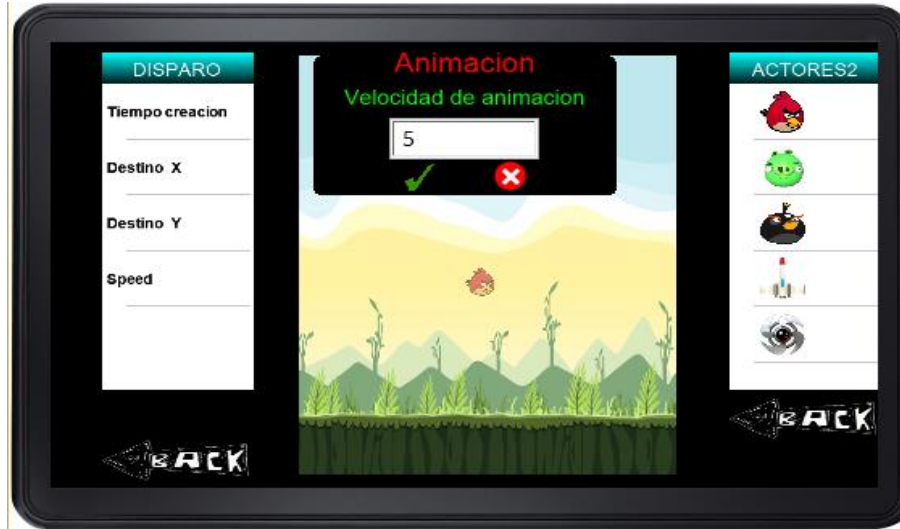
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 96. Acciones: Definir velocidad de animación disparo

CASO DE USO:	Definir velocidad de animación disparo
ACTORES:	Usuario
PROPÓSITO:	Definir velocidad de animación del disparo
RESUMEN:	Posibilidad de definir la velocidad de animación del disparo al momento de la creación.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'DISPARO.	2) La aplicación muestra el conjunto de propiedades de disparo.
3) El usuario selecciona la opción 'Speed'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa el valor numérico.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de velocidad de animación del disparo.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 84. Defina velocidad de animación bala



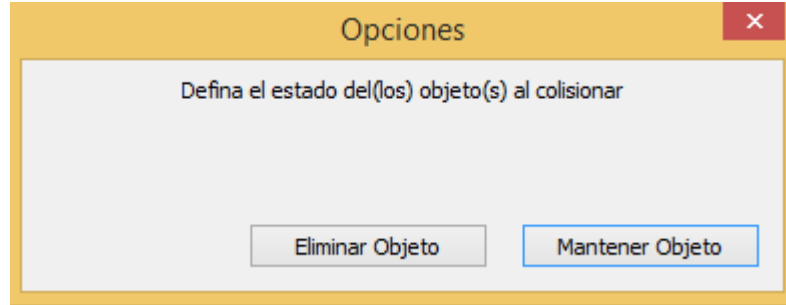
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 97. Acciones: definir estado de colisión

CASO DE USO:	Definir estado de colisión
ACTORES:	Usuario
PROPÓSITO:	Definir el estado de colisión
RESUMEN:	Posibilidad de definir el estado de colisión del objeto
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto en la escena.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'COLISIONES.	2) La aplicación muestra el conjunto de propiedades de colisiones.
3) El usuario selecciona 'Opciones.	4) La aplicación muestra un mensaje de alerta.
5) El usuario selecciona el botón 'Eliminar objeto'.	6) La aplicación elimina el objeto al colisionar con otro.
7) El usuario selecciona el botón 'Mantener objeto'.	8) La aplicación mantiene el objeto en la escena.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 85. Defina estado al colisionar



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 98. Acciones: Definir gravedad x

CASO DE USO:	Definir Gravedad x
ACTORES:	Usuario
PROPÓSITO:	Definir la gravedad en el eje x que tendrá la escena
RESUMEN:	Posibilidad de definir la gravedad en el eje x, puede variar de un valor positivo o negativo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la gravedad en el eje indicado.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario pulsa la propiedad 'GRAVEDAD'.	2) La aplicación muestra el conjunto de propiedades de gravedad.
3) El usuario selecciona la opción 'Gravedad X'.	4) La aplicación muestra un textfield para poder ingresar el valor de la gravedad.
5) El usuario ingresa el valor de la gravedad.	6) La aplicación activa el focus del textfield para poder ingresar el valor numérico.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de gravedad de la escena.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 86. Defina gravedad en X



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 99. Acciones: Definir gravedad y

CASO DE USO:	Definir Gravedad y
ACTORES:	Usuario
PROPÓSITO:	Definir la gravedad en el eje 'y' que tendrá la escena
RESUMEN:	Posibilidad de definir la gravedad en el eje y, puede variar de un valor positivo o negativo.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la gravedad en el eje indicado.
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario pulsa la propiedad 'GRAVEDAD'.	2) La aplicación muestra el conjunto de propiedades de gravedad.
3) El usuario selecciona la opción 'Gravedad Y'.	4) La aplicación muestra un textfield para poder ingresar el valor de la gravedad.
5) El usuario ingresa el valor de la gravedad.	6) La aplicación activa el focus del textfield para poder ingresar el valor.

7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de gravedad de la escena.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 87. Defina gravedad en Y



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

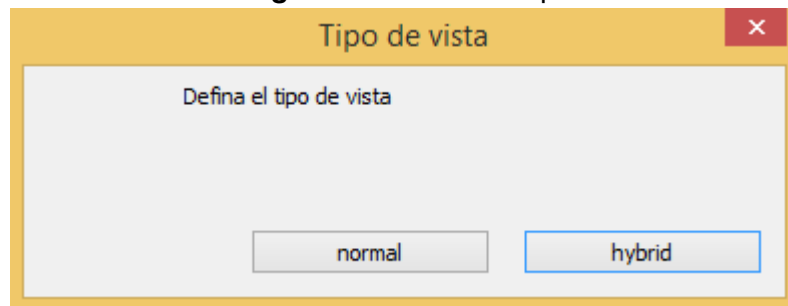
Tabla 100. Acciones: Definir vista

CASO DE USO:	Definir vista
ACTORES:	Usuario
PROPÓSITO:	Ver con detalle las colisiones de objetos geométricos
RESUMEN:	Los objetos creados en pantalla se podrán visualizar de distinta manera si se elige modo 'hybrid'.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder visualizar el efecto de la vista.
ACCIONES	

ACTOR	APLICACIÓN
1)El usuario pulsa el evento 'GRAVEDAD'.	2) La aplicación muestra el conjunto de eventos de gravedad.
3) El usuario selecciona la opción 'DrawMode'.	4) La aplicación muestra un mensaje de alerta con las dos opciones de vista.
5) El usuario presiona el botón 'normal'.	6) La aplicación guarda la vista en modo 'normal'.
7) El usuario presiona el botón 'hybrid'.	8) La aplicación guarda la vista en modo 'hybrid'.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 88. Defina tipo de vista



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 101. Acciones: Definir escenas horizontales

CASO DE USO:	Definir escenas horizontales
ACTORES:	Usuario
PROPÓSITO:	Definir la cantidad de escenas horizontales.
RESUMEN:	Posibilidad de definir la cantidad de escenas horizontales que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'MAPA'.	2) La aplicación muestra el conjunto de propiedades del mapa.
3) El usuario selecciona la opción 'Escenas Horizontales'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa un valor numérico	6) La aplicación activa el focus del textfield para poder ingresar el valor.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de cantidad de escenas horizontales del videojuego.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 89. Defina cantidad de escenas horizontales



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 102. Acciones: Definir escenas verticales

CASO DE USO:	Definir escenas verticales
ACTORES:	Usuario
PROPÓSITO:	Definir la cantidad de escenas verticales.
RESUMEN:	Posibilidad de definir la cantidad de escenas verticales que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'MAPA'.	2) La aplicación muestra el conjunto de propiedades del mapa.
3) El usuario selecciona la opción 'Escenas Vertical'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa un valor numérico	6) La aplicación activa el focus del textfield para poder ingresar el valor.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de cantidad de escenas verticales del videojuego.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 90. Defina cantidad de escenas verticales



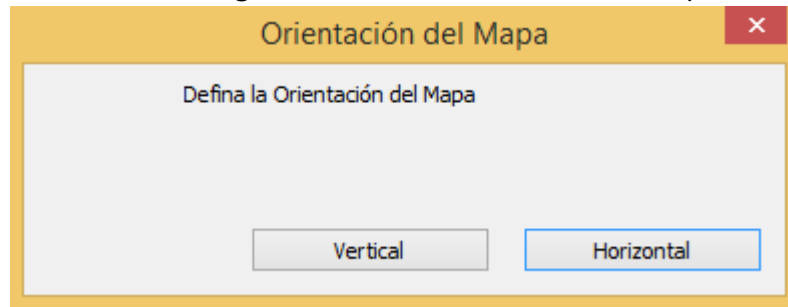
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 103. Acciones: Definir orientación del mapa

CASO DE USO:	Definir orientación del mapa
ACTORES:	Usuario
PROPÓSITO:	Definir la orientación de animación del mapa.
RESUMEN:	Definir la orientación de la animación que tendrá el mapa del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado mas de una escena(horizontales o verticales)
ACCIONES	
ACTOR	APLICACIÓN
1)El usuario pulsa la propiedad 'MAPA'.	2) La aplicación muestra el conjunto de propiedades del mapa.
3) El usuario selecciona la opción 'Orientación.	4) La aplicación muestra un mensaje de alerta con las dos opciones de orientación.
5) El usuario presiona el botón 'vertical.	6) La aplicación guarda la orientación vertical de la animación del mapa
7) El usuario presiona el botón 'horizontal.	8) La aplicación guarda la orientación horizontal de la animación del mapa

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 91. Defina orientación del mapa



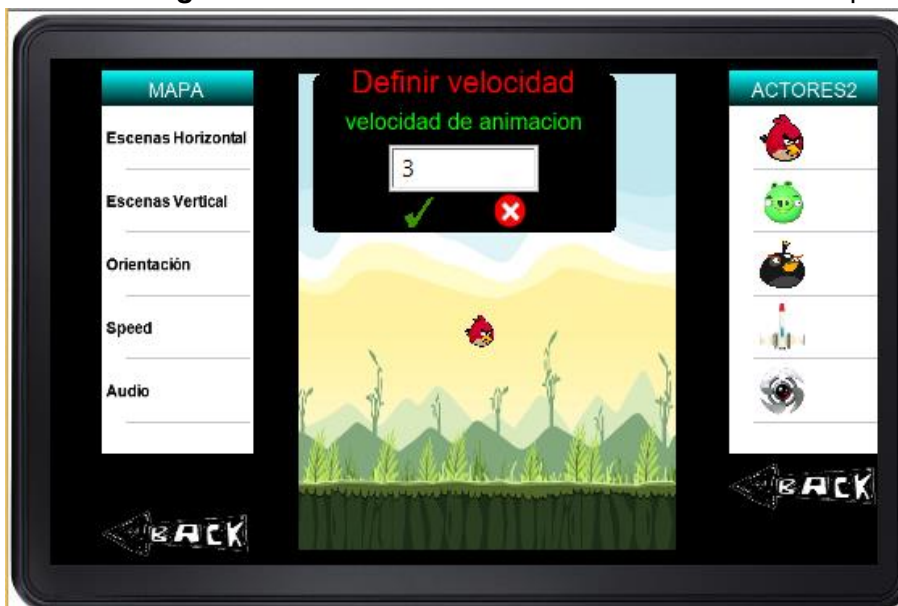
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 104. Acciones: Definir velocidad de animación mapa

CASO DE USO:	Definir velocidad de animación mapa
ACTORES:	Usuario
PROPÓSITO:	Definir la velocidad de animación del mapa
RESUMEN:	Posibilidad de definir la velocidad de animación del mapa del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'MAPA'.	2) La aplicación muestra el conjunto de propiedades del mapa.
3) El usuario selecciona la opción 'Speed'.	4) La aplicación muestra un textfield para poder ingresar el valor numérico.
5) El usuario ingresa un valor numérico	6) La aplicación activa el focus del textfield para poder ingresar el valor.
7) El usuario pulsa la imagen 'aspa'.	8) La aplicación guarda la propiedad de velocidad de animación del mapa del videojuego.
9) El usuario pulsa la imagen 'equis'.	10) La aplicación oculta el textfield sin guardar el valor numérico.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 92. Defina velocidad de animación del mapa



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

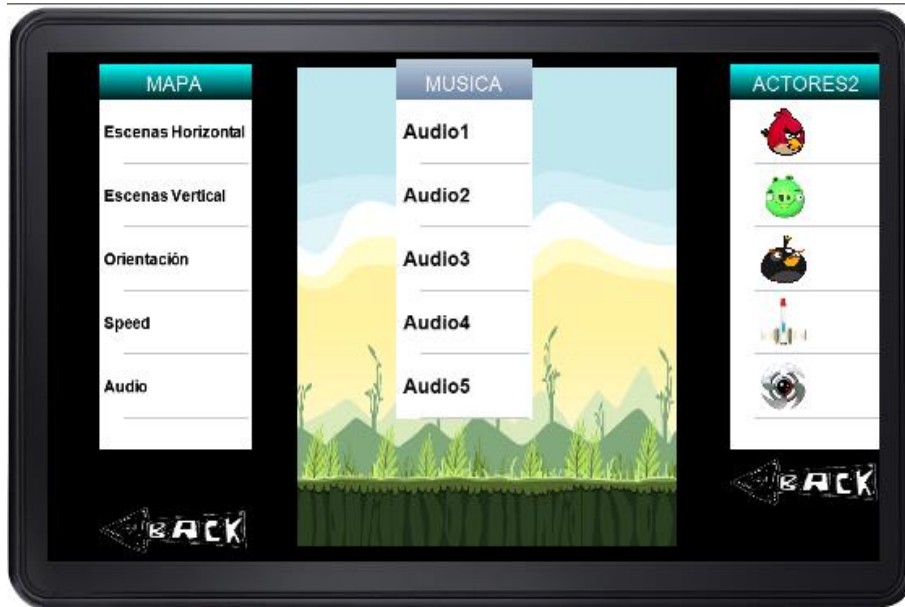
Tabla 105. Acciones: Definir audio del videojuego

CASO DE USO:	Definir audio del videojuego
ACTORES:	Usuario
PROPÓSITO:	Definir el audio del videojuego
RESUMEN:	Definir el audio que tendrá el videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la propiedad de MAPA.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad 'MAPA'.	2) La aplicación muestra el conjunto de propiedades del mapa.
3) El usuario selecciona la opción 'Audio'.	4) La aplicación muestra una lista desplegable con 10 audios disponibles.
5) El usuario selecciona cualquier audio de la lista.	6) La aplicación muestra un mensaje de alerta.
7) El usuario presiona el botón 'si'.	8) La aplicación guarda el audio seleccionado como música de fondo del videojuego.

9) El usuario presiona el botón 'no'.	10) la aplicación cancela la asignación del audio y no considera ninguno.
---------------------------------------	---

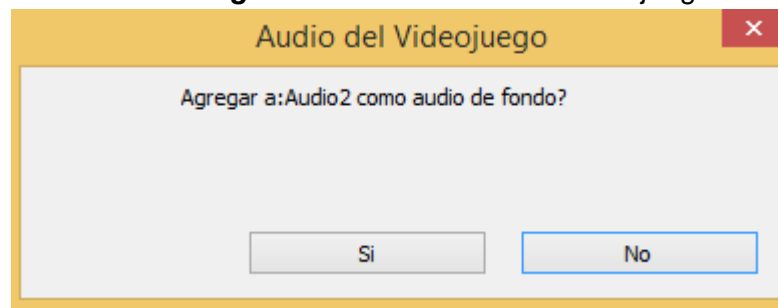
Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 93. Despliegue lista de audios



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 94. Defina audio del videojuego



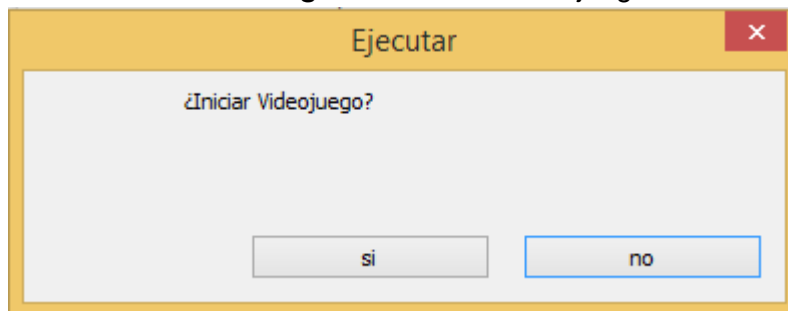
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 106. Acciones: Ejecutar videojuego

CASO DE USO:	Ejecutar videojuego
ACTORES:	Usuario
PROPÓSITO:	Ver la ejecución del videojuego diseñado.
RESUMEN:	Una vez creado los objetos en la escena y haber guardado sus respectivas propiedades y definir también propiedades de la escena, ahora se puede visualizar la ejecución del diseño del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un objeto de la escena, para poder ejecutar el videojuego.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'OPCIONES. 3) El usuario pulsa la opción de 'Ejecutar'.	2) La aplicación muestra el conjunto de propiedades de opciones . 4. La aplicación guarda el diseño del videojuego, y da pase a la escena de visualización de la ejecución del videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 95. Iniciar videojuego



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 96. Pantalla videojuego en ejecución



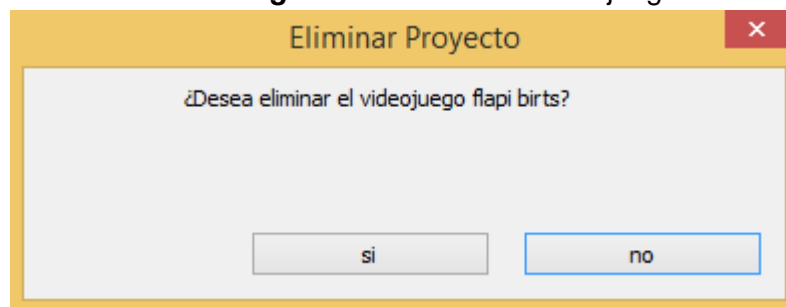
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 107. Acciones: Eliminar videojuego

CASO DE USO:	Eliminar videojuego
ACTORES:	Usuario
PROPÓSITO:	Eliminar el videojuego diseñado.
RESUMEN:	Eliminar el videojuego que actualmente se esta diseñando.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un diseño de un videojuego.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'OPCIONES.	2) La aplicación muestra el conjunto de propiedades de opciones .
3) El usuario pulsa la opción de 'Eliminar'.	4) La aplicación elimina el videojuego actualmente diseñado.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 97. Eliminar videojuego



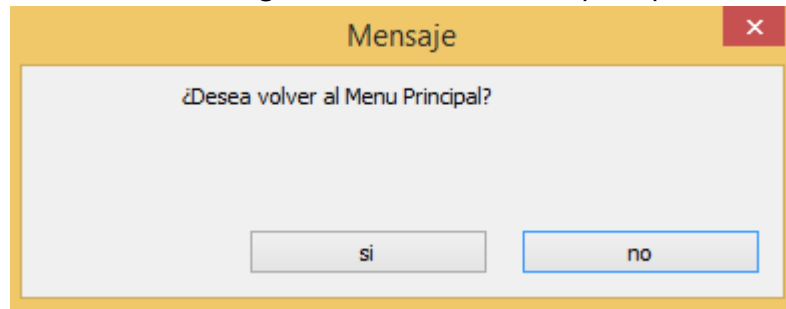
Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 108. Acciones: Ir al menú principal

CASO DE USO:	Ir al menú principal
ACTORES:	Usuario
PROPÓSITO:	Ir al menú principal de la aplicación móvil
RESUMEN:	Ir al menú principal de la aplicación móvil, saliendo de la escena de diseño del videojuego.
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos un diseño de un videojuego.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la propiedad de 'OPCIONES.	2) La aplicación muestra el conjunto de propiedades de opciones .
3) El usuario pulsa la opción de 'Menú principal'.	4) La aplicación da pase a la escena del menú principal de la aplicación móvil.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 98. Volver al menú principal



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 109. Acciones: Ayuda

CASO DE USO:	Ayuda
ACTORES:	Usuario
PROPÓSITO:	Información disponible del uso de las propiedades
RESUMEN:	Información disponible del uso de cada uno de las propiedades que se le asigna a los objetos.
TIPO:	Primario
PRE-CONDICIONES:	Haber seleccionado la opción de ayuda.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la opción de 'AYUDA'. 3) El usuario pulsa la cualquier elemento de la lista.	2) La aplicación muestra una lista de cada una de las propiedades. 4) La aplicación muestra la información de ayuda de la propiedad seleccionada.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 99. Ayuda física



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 100. Ayuda animación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 101. Ayuda objeto



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 102. Ayuda disparo



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 103. Ayuda colisiones



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 104. Ayuda gravedad



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 105. Ayuda mapa



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 106. Ayuda opciones



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 110. Acciones: Exportar videojuego

CASO DE USO:	Exportar videojuego
ACTORES:	Usuario
PROPÓSITO:	Exportar el código del videojuego
RESUMEN:	Exportar el videojuego por medio del bluetooth, para la construcción del .apk(Android) y .exe(Windows Desktop)
TIPO:	Primario
PRE-CONDICIONES:	Haber creado por lo menos el diseño de un videojuego.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario pulsa la imagen del correo	2) La aplicación muestra un mensaje de alerta en la que se indica si que quiere activar el bluetooth.
3) El usuario activa el bluetooth del dispositivo.	4) La aplicación muestra una lista de los dispositivos disponibles a los cuales se quiere enviar el código del videojuego.
5) El usuario presiona sobre un dispositivo.	6) La aplicación empieza a enviar el código fuente generado del videojuego.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Tabla 111. Acciones: Seleccionar mini juego

CASO DE USO:	Seleccionar Mini Juego
ACTORES:	Usuario
PROPÓSITO:	Poder jugar un videojuego tipo Geometry Dash.
RESUMEN:	Poder jugar un videojuego tipo Geometry Dash, este videojuego esta totalmente funcional.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el capítulo de Mini Juego.
ACCIONES	

ACTOR	APLICACIÓN
1) El usuario selecciona la opción mini juego.	2) La aplicación muestra el videojuego y el usuario podrá jugar.

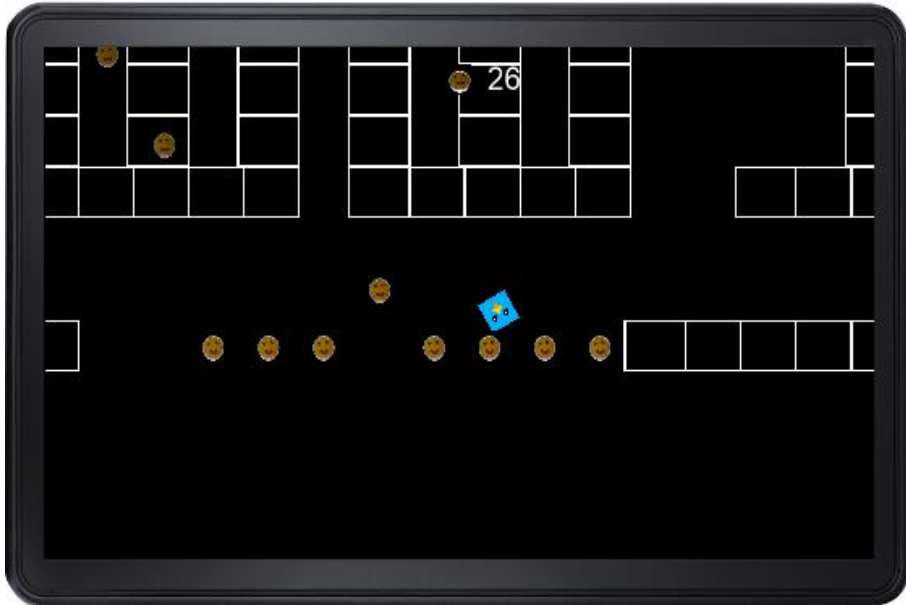
Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 107. Pantalla menú mini juego



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Figura 108. Pantalla ejecución mini juego



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 112. Acciones: Opciones

CASO DE USO:	Opciones
ACTORES:	Usuario
PROPÓSITO:	Créditos al desarrollador de la aplicación móvil
RESUMEN:	Información del desarrollador de la aplicación móvil.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el menú Opciones.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario selecciona el menú 'Opciones'.	2) La aplicación muestra información del desarrollador de la aplicación móvil.
3) El usuario presiona la pantalla.	4) La aplicación cierra la escena actual y da pase al menú principal.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 109. Menú Opciones



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.

Tabla 113. Acciones: Salir

CASO DE USO:	Salir
ACTORES:	Usuario
PROPÓSITO:	Salir de la aplicación móvil.
RESUMEN:	Salir de la aplicación móvil.
TIPO:	Secundario
PRE-CONDICIONES:	Haber seleccionado el menú Salir.
ACCIONES	
ACTOR	APLICACIÓN
1) El usuario selecciona el menú 'Salir. 3) El usuario presiona el botón 'sí'. 5) El usuario presiona el botón 'no'	2) La aplicación muestra un mensaje de alerta. 4) La aplicación móvil se cierra. 6) La aplicación cierra el mensaje de alerta, y cancela el cierre de la aplicación móvil.

Fuente: Elaboración propia. Utilizando tablas de Microsoft Office 2013

Figura 110. Mensaje de alerta salir de la aplicación



Fuente: Elaboración propia. Utilizando Recortes de Windows 8.1.