



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DE VIDEOJUEGO PARA DISPOSITIVOS MÓVILES
ANDROID, CON INTERACCIÓN EN REDES SOCIALES Y ORIENTADO A
LA EDUCACIÓN VIAL DE LA POBLACIÓN



AUTOR

Juan Andrés Dávila Herrera

AÑO

2017



FACULTAD DE INGENIERÍA Y CIENCIAS AGROPECUARIAS

DESARROLLO DE VIDEOJUEGO PARA DISPOSITIVOS MÓVILES
ANDROID, CON INTERACCIÓN EN REDES SOCIALES Y ORIENTADO A LA
EDUCACIÓN VIAL DE LA POBLACIÓN.

Trabajo de Titulación presentado en conformidad con los requisitos
establecidos para optar por el título de Ingeniero en Sistemas de Computación
e Informática

Profesor Guía

Msc. Paulo Roberto Guerra Terán

Autor

Juan Andrés Dávila Herrera

Año

2017

DECLARACIÓN DEL PROFESOR GUÍA

“Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando sus conocimientos y competencias para un eficiente desarrollo del tema escogido y dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Paulo Roberto Guerra Terán
Master en Software y Sistemas
C.I: 1002856050

DECLARACIÓN DEL PROFESOR CORRECTOR

“Declaro haber revisado este trabajo, dando cumplimiento a todas las disposiciones vigentes que regulan los Trabajos de Titulación.”

Adonis Ricardo Rosales García

Master en informática aplicada

C.I: 1756883144

DECLARACIÓN DE AUTORÍA DEL ESTUDIANTE

“Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.”

Juan Andrés Dávila Herrera

C.I.: 1750654699

RESUMEN

En la actualidad el mercado de aplicaciones móviles está creciendo constantemente y son las aplicaciones interactivas o videojuegos las que tiene más acogida entre el público. El siguiente proyecto de titulación toma como base esta idea y trata acerca del desarrollo y lanzamiento del videojuego “Vial Education Runner” el cual tiene como objetivo principal la concientización de la población en temas de seguridad vial y la masificación a través de redes sociales.

Se realizaron todas las actividades que involucran este tipo de proyectos, desde el diseño y la animación hasta la programación, todo esto siguiendo una metodología para el desarrollo de software llamada SUM. El resultado final de este proyecto es una aplicación interactiva, inteligente, vistosa y divertida que se encuentre disponible para descargar al público en general.

ABSTRACT

At present, the mobile application market is constantly growing and it is the interactive applications or video games that are most welcome in the public. The following titling project is based on this idea and attempts to develop and launch the video game "Vial Education Runner" has as main objective the awareness of the population in issues of road safety and massification through social networks.

All activities involving this type of project were carried out, from the design and the animation to the programming, all that following a methodology for the development of software called SUM. The result of this project is an interactive, intelligent, colorful and fun application that is available for download for the public.

ÍNDICE

1	Introducción	1
1.1	Antecedentes.....	1
1.2	Alcance	2
1.3	Justificación	3
1.4	Objetivo General.....	4
1.5	Objetivos Específicos	4
1.6	Metodología	5
2.	Marco Teórico.....	6
3.	Situación Actual y propuesta.....	21
4.	Análisis Costo/Beneficio.....	27
5.	Desarrollo del Proyecto.....	30
5.1	Desarrollo de concepto	31
5.2	Planificación.....	32
5.3	Elaboración	36
5.3.1	Sprint 0.....	36
5.3.1.1	Instalación Unity y Visual Studio	36
5.3.1.2	Instalación de Adobe Photoshop e Illustrator	37
5.3.1.3	Compra de Licencia Android.....	38
5.3.1.5	Creación de historias de usuario en Trello	39
5.3.2	Sprint 1.....	40
5.3.2.1	Sprint Backlog	40
5.3.2.2	Historias de usuario	41
5.3.2.3	Riesgos	43
5.3.2.4	Evaluación del estado del videojuego	44
5.3.2.5	Resultados	44
5.3.2.6	Retrospectiva	47
5.3.3	Sprint 2.....	48
5.3.3.1	Sprint Backlog	48

5.3.3.2	Historias de usuario	48
5.3.3.3	Riesgos	50
5.3.3.4	Evaluación del estado del videojuego	51
5.3.3.5	Resultados	51
5.3.3.6	Retrospectiva	55
5.3.4	Sprint 3.....	56
5.3.4.1	Sprint Backlog	56
5.3.4.2	Historias de usuario	57
5.3.4.2	Riesgos	59
5.3.4.2	Evaluación del estado del videojuego	59
5.3.4.3	Resultados	60
5.3.4.4	Retrospectiva	64
5.3.5	Sprint 4.....	65
5.3.5.1	Sprint Backlog	65
5.3.5.2	Historias de usuario	66
5.3.5.4	Evaluación del estado del videojuego	68
5.3.5.5	Resultados	69
5.3.5.6	Retrospectiva	73
5.3.6	Sprint 5.....	73
5.3.6.1	Sprint Backlog	73
5.3.6.2	Historias de usuario	74
5.3.6.5	Resultados	76
5.3.6.6	Retrospectiva	79
5.4	Beta.....	80
5.4.1	Verificación del videojuego.....	80
5.4.1.1	Casos de prueba sprint 1	80
5.4.1.1	Casos de prueba sprint 2	84
5.4.1.1	Casos de prueba sprint 3	86
5.4.1.1	Casos de prueba sprint 4	89
5.4.2	Corrección del videojuego.....	92
5.5	Cierre.....	93
5.5.1.	Liberación.....	93
6.	CONCLUSIONES Y RECOMENDACIONES	95

6.1 Conclusiones.....	95
6.2 Recomendaciones.....	95
REFERENCIAS.....	97
ANEXOS.....	102

ÍNDICE DE FIGURAS

Figura 1. Porcentaje de mercado de sistemas operativos para dispositivos móviles.	8
Figura 2. Comparativa de sistemas operativos en Ecuador.	9
Figura 3. Propiedades físicas de un objeto dentro de Unity	11
Figura 4. Editor de Unity y sus principales ventanas.	13
Figura 5. Fases del diseño de personajes y escenarios.	14
Figura 6. Ejemplo de Sprite Sheet.	16
Figura 7. Editor de Sprites de Unity.	17
Figura 8. Grafos o Estados de animación en Animador de Unity	18
Figura 9. Diagrama del sistema de animación.	19
Figura 10. Señal del agente de tránsito.	21
Figura 11. Esquema del funcionamiento básico del juego	25
Figura 12. Esquema del menú	26
Figura 13. Metodología SUM.	30
Figura 14. Instalador del editor de Unity.	37
Figura 15. Adobe Photoshop para el diseño gráfico.	38
Figura 16. Compra de licencia de desarrollador para Google Play.	39
Figura 17. Trello, tareas del sprint 0.	40
Figura 18. Cuadro de tareas Kanban del Sprint 1	41
Figura 19. Evaluación del estado del videojuego tras sprint 1.	44
Figura 20. Resultados de Sprint 1. Desplazamiento del personaje	45
Figura 21. Resultados de Sprint 1. Salto del personaje	45
Figura 22. Actualización del cuadro Kanban tras sprint 1.	46
Figura 23. Burndown Primera iteración	46
Figura 24. Cuadro de tareas Kanban del Sprint 2	48
Figura 25. Resultados sprint 2. Objeto con array para trivia	52
Figura 26. Resultados sprint 2. Colisión con objeto de trivia	52
Figura 27. Resultados sprint 2. Presentación de trivia.	53
Figura 28. Resultados sprint 2. Aumento de vida.	53
Figura 29. Resultados sprint 2. Implementación del menú.	54
Figura 30. Actualización del cuadro Kanban tras sprint 2.	54
Figura 31. Burndown segunda iteración	55
Figura 32. Cuadro de tareas Kanban del Sprint 3.	57
Figura 33. Resultados Sprint 3. Acceso a información de Facebook.	60
Figura 34. Resultados Sprint 3. Inicio de sesión en Facebook.	61
Figura 35. Resultados Sprint 3. Compartir en Facebook.	61
Figura 36. Resultados Sprint 3. Compartir en Twitter.	62
Figura 37. Resultados Sprint 3. Aumento de vidas	62
Figura 38. Actualización del cuadro Kanban tras sprint 3.	63
Figura 39. Burndown tercera iteración	63

Figura 40. Cuadro de tareas Kanban del Sprint 4	66
Figura 41. Resultados Sprint 4. Material Gráfico.....	69
Figura 42. Resultados Sprint 4. Música del videojuego.....	70
Figura 43. Resultados Sprint 4. Fondo.....	70
Figura 44. Resultados Sprint 4. Animación y diseño del personaje.....	71
Figura 45. Resultados Sprint 4. Menú del videojuego	71
Figura 46. Resultados Sprint 4. Burndown Sprint 4.....	72
Figura 47. Actualización del cuadro Kanban tras sprint 4.	72
Figura 48. Actualización del tablero de historias del sprint 5.....	74
Figura 49. Resultados Sprint 5. Animación de interfaz.....	76
Figura 50. Resultados Sprint 5. Animación de obstáculos	77
Figura 51. Resultados Sprint 5. Manejo del release (beta).	77
Figura 52. Resultados Sprint 5. Manejo del release (calificación juego).	78
Figura 53. Resultados Sprint 5. Publicación de beta.....	78
Figura 54. Resultados Sprint 5. Burndown quinta iteración.....	79
Figura 55. Resultados Sprint 5. Actualización de cuadro Kanban tras sprint 5.79	
Figura 56. Caso de prueba 1.....	81
Figura 57. Caso de prueba 2.....	82
Figura 58. Caso de prueba 3 y 4.....	83
Figura 59. Caso de prueba 5-7.....	86
Figura 60. Caso de Prueba 8	87
Figura 61. Caso de prueba 9.....	88
Figura 62. Caso de prueba 10.....	89
Figura 63.Caso de prueba 12-16.....	92
Figura 64. Comentarios para la corrección del juego	93
Figura 65. Publicación de la aplicación	94

INDICE DE TABLAS

Tabla 1. Unidades vendidas y porcentaje del mercado para dispositivos móviles en 2016 y 2015.	8
Tabla 2. Número de fallecidos por accidentes de tránsito en 2016.	22
Tabla 3. Causas probables de fallecimiento en las vías.....	23
Tabla 4. Estimación de costo de la app.....	27
Tabla 5. Estimación de costo de la app (Software).	28
Tabla 6. Product Backlog	33
Tabla 7. Sprint Backlog 1	40
Tabla 8. Historia de Usuario 1: salto y desplazamiento.....	41
Tabla 9. Historia de Usuario 2: Generación de obstáculos y escenario	42
Tabla 10. Historia de usuario 3: Aumento de puntaje.....	42
Tabla 11. Historia de Usuario 4: Muerte del personaje, disminución de vidas y retención del puntaje más alto	42
Tabla 12. Puntos ganados tras Sprint 1	47
Tabla 13. Sprint 2 Backlog	48
Tabla 14. Historia de Usuario 5: Creación de menú.....	49
Tabla 15. Historia de Usuario 6: Creación de cuestionario base de preguntas	49
Tabla 16. Historia de Usuario 7: Despliegue de preguntas tras una colisión ...	49
Tabla 17. Historia de Usuario 8: Agregar vida/puntaje tras responder trivia ...	50
Tabla 18. Puntos ganados tras Sprint 2	55
Tabla 19. Sprint Backlog 3	57
Tabla 20. Historia de Usuario 9: Implementación de share por Facebook.....	57
Tabla 21. Historia de Usuario 10: Implementación de share por Twitter	58
Tabla 22. Historia de usuario 11: Dar recompensa (vida/puntaje) al usuario por compartir	59
Tabla 23. Puntos ganados tras Sprint 3	64
Tabla 24. Sprint 4 Backlog	65
Tabla 25. Historia de Usuario 12: Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos).....	66
Tabla 26. Historia de Usuario 13: Creación e implementación de Sprites (gráficos) de UI (botones, menú).....	66
Tabla 27. Historia de Usuario 14: Creación e implementación de sonidos	67
Tabla 28. Historia de Usuario 15: Animación del personaje y fondo	67
Tabla 29. Historia de Usuario 16: Animación de interfaz (botones, menú y obstáculos)	68
Tabla 30. Sprint 4 Backlog	72
Tabla 31. Sprint Backlog 5	74
Tabla 32. Historia de Usuario 17: Manejo de reléase en tienda.....	75
Tabla 33. Historia de Usuario 18: Publicación para beta.....	75
Tabla 34. Caso de prueba 1	80

Tabla 35. Caso de prueba 2.....	81
Tabla 36. Caso de prueba 3.....	82
Tabla 37. Caso de prueba 4.....	82
Tabla 38. Caso de prueba 5.....	84
Tabla 39. Caso de prueba 6.....	84
Tabla 40. Caso de prueba 7.....	85
Tabla 41. Caso de prueba 8.....	86
Tabla 42. Caso de prueba 9.....	87
Tabla 43. Caso de prueba 10.....	88
Tabla 44. Caso de prueba 11.....	89
Tabla 45. Caso de prueba 12.....	89
Tabla 46. Caso de prueba 13.....	90
Tabla 47. Caso de prueba 14.....	91
Tabla 48. Caso de prueba 15.....	91

1 Introducción

1.1 Antecedentes

Con el avance de la tecnología han surgido diversas metodologías para solucionar problemas de toda índole. Para empresas o instituciones están disponibles nuevos medios que apuntan a la expansión de su mercado o a la obtención de mayores beneficios. Destacan los videojuegos y las redes sociales, ambos medios tienen acceso a amplios mercados sobre los cuales se puede difundir una empresa o marca y se posibilita además la generación de importantes beneficios.

Para entender por qué han surgido nuevos medios de expansión del mercado, basta con revisar los avances de la tecnología de los últimos años. La industria del software ha experimentado un significativo desarrollo en las últimas décadas gracias al internet, a los dispositivos móviles y a las redes sociales. La creciente demanda de smartphones y tablets contribuye al auge de los videojuegos y las redes sociales. Actualmente el mercado de videojuegos para móviles ha crecido exponencialmente entre la población de todas las edades.

La Encuesta de Condiciones de Vida (ECV), realizada en los años 2013 y 2014 por el Instituto Nacional de Estadística y Censos (INEC p. 2) determina que el 41,5% de la población mayor de 12 años (4'995.474 usuarios) poseen una cuenta en una red social, con un promedio de uso de 7,22 horas semanales. Las redes sociales son usadas mayormente por la población entre 18 y 24 años en un 71,7%, seguidos por un 68.9% entre 12 y 17 años y un 56,3% entre 25 y 29 años. Facebook es la red social con mayor incidencia poblacional con un 40,6%, WhatsApp con un 13,7% y Twitter con un 8,5%. La población accede a redes sociales en su mayoría a través de computadoras (48,5%), seguido por el uso de teléfonos inteligentes (24,3%) y tablets (11,8%).

La industria mundial de videojuegos es una de las que más usuarios tiene y de las más rentables, llegando a competir con la televisión o el cine, como lo indica

para la revista Forbes, Mario Valle, quien se desempeña como Director de Desarrollo de Negocios para Mercados Emergentes de Electronic Arts: “Los videojuegos son la categoría de las apps stores, tanto de Google como de Apple, que más ventas registra mes a mes. La oportunidad de emprendimiento en esta industria es muy grande, sobre todo en el desarrollo de videojuegos para dispositivos móviles”. (Valle p. 1)

Gracias al incremento de la industria de los videojuegos y los aplicativos móviles, la educación ha cambiado para interactuar más con estos medios, y un tipo de educación que no se está aprovechando es la educación vial. Como sabemos el índice de accidentes de tránsito en nuestro país es elevado, aproximadamente 30.000 accidentes han tenido lugar en nuestro país durante el año 2016 (ANT, p.1 2016). Muchos de los accidentes se han dado por el desconocimiento de las leyes o el irrespeto a las señales de tránsito. Es necesario concientizar a la ciudadanía sobre la importancia de la educación y seguridad vial, y una excelente opción de hacerlo es a través de los videojuegos, procurando que los ciudadanos puedan recrearse al mismo tiempo que aprenden y conocen acerca de temas de seguridad en las vías.

La industria del videojuego en Ecuador no está ampliamente explotada y son pocas las empresas que lucran del desarrollo de juegos. Se ha subestimado la influencia que tiene este medio de entretenimiento sobre la población y cómo el uso en conjunto con las redes sociales puede representar un negocio rentable además de agregar beneficios en torno a la educación de la población.

1.2 Alcance

El alcance de este trabajo de titulación es el de crear, diseñar y desarrollar un videojuego para dispositivos móviles con sistema operativo Android, que permita al usuario aprender más acerca de educación vial. Este videojuego se basará en evitar enemigos y obstáculos que aparezcan en pantalla conforme se acerquen al personaje del jugador de manera constante. Adicionalmente el juego incluirá

mecánicas de trivia o preguntas acerca de reglamentos, leyes o consejos de seguridad vial para que el jugador pueda aumentar su puntaje si acierta.

El usuario podrá también compartir e interactuar con sus amigos de redes sociales, a través del juego y competir en puntajes. En este caso se puede dar más oportunidades al jugador que obtenga mayores puntajes. De esta manera el juego se expande y capta la atención de más jugadores, que a su vez se concientizan y aprenden cómo manejarse en las vías.

Para el desarrollo del producto se utilizará el motor de videojuegos multiplataforma Unity y será desarrollado exclusivamente para dispositivos con sistema operativo Android, que son los más usados en Ecuador (Abad, 2015). Se empleará el lenguaje de programación C# conjuntamente con IDE Visual Studio 2015 para escribir el código necesario para la aplicación, y se usarán, además, herramientas de diseño como Adobe Photoshop y Adobe Illustrator.

Todo el proyecto es factible para su desarrollo ya que utilizará los conceptos y aprendizaje adquiridos a lo largo de la carrera en materias como: Lenguajes de Programación, Inteligencia Artificial, Introducción a la Multimedia, Ingeniería Web y Seminarios de Sistemas.

1.3 Justificación

El mercado de videojuegos no ha sido tan explotado por los desarrolladores en el país, por eso es importante incursionar a este tipo de software que es una tendencia tecnológica en todo el mundo. Junto a los videojuegos, las redes sociales pueden ser un recurso importante para la generación de valor. Gracias al internet, una idea, una propuesta, una marca o una oferta puede ser esparcida rápidamente y acogida en el mercado. Por eso es importante aprovechar este canal que nos brinda la tecnología.

El tipo de software que se plantea en este proyecto tiene una demanda creciente en el mercado. Juegos vistosos que incluyan interacción con el usuario tienden

a afirmarse a corto plazo en dispositivos móviles. “El Mercado Global De Juegos Alcanza los 108.9 Mil Millones de dólares en 2017, de los cuales las apps para móviles toman el 42%” (McDonald, 2017). Las empresas tienden a relacionarse con el cliente de manera creativa y directa a través de: campañas en redes sociales, programas con interfaces atractivas, aplicaciones con gráficos en 3D o 2D o videojuegos. Consecuentemente, las anteriores son opciones pueden ser tomadas en cuenta por los ingenieros de sistemas en un medio que siempre se está innovando.

Los contenidos dinámicos e interactivos han penetrado con mucha fuerza en la educación moderna contribuyendo a que las personas asimilen de mejor forma la información que están recibiendo y refuercen aquellos conocimientos ya adquiridos. “Las ventajas de la aplicación de multimedios radican en lo que estos materiales generan en los alumnos, como son: interés, motivación, desarrollo de la iniciativa, mayor comunicación y aprendizaje cooperativo.” (Gonzales, 2013).

Por todas estas razones se plantea realizar este proyecto que se basa en temas innovadores como las aplicaciones móviles y los videojuegos; y, que pretende demostrar la aplicabilidad de este tipo de software a un fin educativo, mientras se expande ampliamente una marca o producto mediante la interacción con redes sociales.

1.4 Objetivo General

Desarrollar un videojuego para dispositivos móviles Android con integración de dinámicas grupales que permita concientizar y educar a la población sobre seguridad vial, mediante la difusión masiva de la aplicación a través de redes sociales y la integración con mecánicas educativas.

1.5 Objetivos Específicos

- Analizar el mercado de videojuegos para móviles y determinar los temas de importancia para la educación vial, para crear una aplicación atractiva que logre concientizar a los usuarios.

- Diseñar e implementar el material requerido para realizar el videojuego (texturas, fondos, sonidos, personajes, interfaz de usuario, menús y animaciones) y desarrollar/ programar la aplicación con estos elementos.
- Implementar el módulo de trivia sobre seguridad vial y la forma en cómo las respuestas correctas benefician al jugador.
- Desarrollar la integración del juego con redes sociales. Es decir, implementar la conexión del juego a redes sociales de tal modo que se pueda complementar con el objetivo anterior y así crear un sistema de logros o beneficios cuando el usuario comparte la aplicación.

1.6 Metodología

Para el desarrollo del proyecto de titulación se ha elegido la metodología SUM. Básicamente la metodología se basa en Scrum, pero toma prestadas algunas cosas de XP. El objetivo de SUM es que se pueda desarrollar este tipo de software en el menor tiempo posible, al menor coste y con la mayor participación de cliente como sea posible.

El ciclo de vida de esta metodología es iterativo e incremental, se basa en la presentación de entregables cada cierto tiempo que agreguen valor al producto y permitan una rápida retroalimentación. “Las cinco fases secuenciales son: concepto, planificación, elaboración, beta y cierre” (Acerenza et al. 2009 p. 174). La fase de concepto es la que más difiere de los ambientes ágiles. Al ser un videojuego el producto final, es en esta etapa de concepto donde se define temas como el público objetivo, elementos de juego como: personajes, gameplay, historia, o elementos técnicos como lenguajes o herramientas para el desarrollo.

A continuación, la fase de planificación es donde se define el cronograma del proyecto con sus principales hitos y a su vez se forma el equipo que intervendrá en la fase de elaboración. Algo importante es que la planificación es flexible para adaptarse al cambio. Tenemos la fase de elaboración, que consiste en el desarrollo del juego de forma incremental e iterativa y posteriormente tenemos

la fase de Beta, la cual sirve para probar el videojuego en aspectos como gameplay, dificultad y posibles bugs. Al final, se realiza el cierre que es la etapa donde se entrega la versión final del producto y se sacan conclusiones que ayuden a futuros proyectos. Cabe recalcar que SUM también tiene en cuenta la gestión de riesgos y esta se realiza durante todo el proyecto cuando se establece la probabilidad, impacto y la forma de mitigar los riesgos.

Esta metodología define cuatro roles basados en Scrum que son: el equipo de desarrollo (Scrum Team), el cliente (Product Owner), el verificador beta y el productor interno (Scrum Master). La metodología nos sugiere que el equipo de desarrollo tiene sub roles de acuerdo al enfoque del proyecto y es aquí donde se encuentra su principal ventaja. Como se habla de un producto que abarca muchos campos es necesario que existan sub roles como artista gráfico, programador o diseñador. El rol de verificador beta no encuentra una similitud exacta con Scrum, pero es responsabilidad de él que se verifique la funcionalidad del videojuego. El estudiante hará de equipo de desarrollo, productor interno y es el mismo estudiante que, una vez avanzado el proyecto, tiene que tomar el rol de verificador beta junto al corrector. El cliente que aporte retroalimentación puede ser el profesor guía o el profesor corrector también.

SUM está pensado para el desarrollo de videojuegos en equipos pequeños y para proyectos cortos, no mayores a un año. Como se ha visto SUM tiene un ciclo de vida iterativo, este ciclo permite que se corrija cualquier detalle en el menor tiempo posible tratando de mitigar los riesgos que puedan presentarse como por ejemplo los cambios de requisitos. En la industria del videojuego se tiene una alta tasa de cambio en los requisitos, debido a la dificultad que puede presentarse al concebir un producto cuyo principal fin sea la diversión del usuario.

2. Marco Teórico

El siguiente marco teórico tiene como fin presentar todos los conceptos que intervendrán en la elaboración de la aplicación del trabajo de titulación.

2.1 Desarrollo de Aplicaciones para Dispositivos Móviles

“El desarrollo de aplicaciones móviles es el proceso en el que un software para realizar determinada tarea es desarrollado para dispositivos móviles...” (EstudioWam, s.f.). Hoy en día las aplicaciones móviles (también denominadas apps) tienen un peso significativo en el mercado de software. Como usuarios siempre queremos tener todo al alcance de nuestra mano a tan solo unos cuantos segundos. Es normal que las empresas que ofrecen soluciones tecnológicas se hayan volcado al desarrollo de aplicaciones.

Para emprender el desarrollo de una aplicación se deben tener en cuenta dos variantes: aplicaciones híbridas o aplicaciones nativas.

Para comprender la diferencia entre estas dos opciones basta con decir que las aplicaciones nativas son las hechas con el lenguaje específico de cada plataforma (Java en Android, por ejemplo) y adaptadas para correr únicamente en donde se crearon. Por otro lado, tenemos las aplicaciones híbridas que son hechas para que puedan ejecutarse en varios sistemas operativos.

Lo bueno de las aplicaciones nativas es que pueden aprovechar todo el potencial de la plataforma en la que se programan como librerías o funcionalidades de hardware, pero reducen el target de público al realizarse para una sola plataforma. Las aplicaciones híbridas en cambio sufren problemas de desempeño y de que no se puede acceder a todos los beneficios de una plataforma en específico, pero se abarata costos y tiempos a la vez que se amplía el target de consumo.

Hablando de estos dos sistemas operativos es importante que para el desarrollo de una aplicación se tenga en cuenta para cuál sistema operativo se va a desarrollar. Según la empresa Gartner (Gartner, 2016) la cual es una de las empresas más grandes en investigación y asesoramiento en tecnología de la información, Android domina ampliamente el mercado con un 86% tal como se puede ver en la tabla y en la figura 1.

Tabla 1.
Unidades vendidas y porcentaje del mercado para dispositivos móviles en 2016 y 2015.

Sistema operativo	2016 Unidades	2016 compartición del mercado (%)	2015 Unidades	2015 compartición del mercado (%)
Android	296,912.8	86.2	271,647.0	82.2
iOS	44,395.0	12.9	48,085.5	14.6
Windows	1,971.0	0.6	8,198.2	2.5
BlackBerry	400.4	0.1	1,153.2	0.3
Otros	680.6	0.2	1,229.0	0.4
Total	344,359.7	100.0	330,312.9	100.0

Adaptada de (Gartner, 2016)

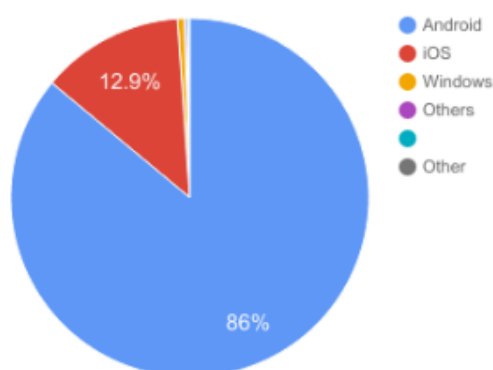


Figura 1. Porcentaje de mercado de sistemas operativos para dispositivos móviles.

Adaptada de (Gartner, 2016)

A nivel latinoamericano el panorama no es muy diferente, Android domina el mercado, en nuestro país existe una diferencia abismal de usuarios que consumen Android, tal como se ve en la figura 2, el cual lo nace a partir de los

datos que la red social Facebook compartió indicándonos cual es el sistema operativo que más se usa para ingresar a la red social:

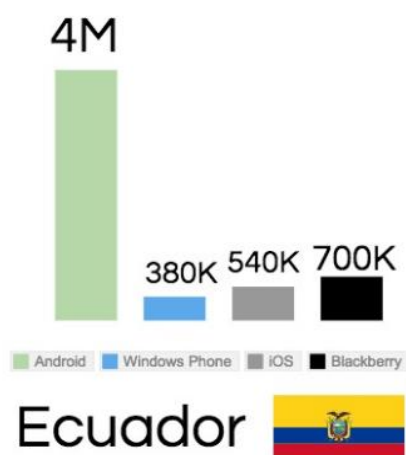


Figura 2. Comparativa de sistemas operativos en Ecuador.
Tomada de Abad, 2015

El hecho de optar por desarrollar para Android no solo se relaciona con el porcentaje de público al que se puede llegar, cuando se habla de la publicación de la aplicación, Android tiene una ventaja frente a la competencia.

Android solo necesita la compra de la licencia de desarrollador para poder publicar en Google Play, la cual tiene un costo de \$25 USD (GooglePlay Developers Console, 2017), la cual en comparación a su principal competidor Apple es cuatro veces menor (\$100 USD). Además, para desarrollar en Android no se necesita una plataforma específica. Por ejemplo, se puede desarrollar en Windows, MacOS, Linux.

Google además ofrece tres modelos de pruebas beta para que se puedan probar las aplicaciones y detectar bugs o errores antes de ser lanzadas.

- **Beta Testing Cerrado:** El desarrollador provee acceso a la aplicación a ciertas personas o testers a través de su cuenta de correo Gmail.
- **Beta Testing Abierto:** Este método permite compartir la aplicación a través de un enlace que se puede compartir de cualquier manera a las personas que deseemos.

- **Beta Testing Cerrado con comunidades de Google Plus:** Parecida a la beta testing cerrado solo que se maneja testers de una comunidad en la red social Google Plus.

Es importante mencionar que para obtener retroalimentación de las aplicaciones los desarrolladores se apoyan mucho en la información que les proporcionan las tiendas donde las publican. Tanto la AppStore de Apple y Google Play ofrecen datos interesantes como comentarios de los clientes, número de descargas o hasta la ubicación geográfica de donde se descargó la aplicación.

A pesar de que el framework Unity permite desarrollar las aplicaciones tanto para Android e iOS el costo se eleva en este último debido a la necesidad de tener una máquina con macOS y una licencia de desarrollador que cuesta \$100 USD (precio oficial 2017). Por estos motivos para el desarrollo del proyecto se planea lanzar la aplicación para dispositivos móviles Android, tanto por el gran target de público que abarca como el bajo costo de desarrollo.

2.2 Desarrollo de Videojuegos con Unity

Unity es un motor de desarrollo para juegos y contenidos interactivos que fue creado por Unity Technologies en el 2005 y que ha tenido una gran acogida en los últimos años debido que permite la compilación para diferentes plataformas, entre las empresas más conocidas que usan Unity podemos nombrar por ejemplo a Electronic Arts, Disney o Cartoon Network.

2.2.1 Motor gráfico y codificación

Un motor gráfico es el software que permite la creación de un videojuego ya que renderiza las imágenes en pantalla, simula la física e iluminación, y gestiona las animaciones, la memoria, y las conexiones. (Diez, 2012)

La iluminación global, la corrección de color, el soporte de sombras en tiempo real, la compatibilidad con varios softwares de modelado 3D son unas de las

tantas características que hacen que los desarrolladores puedan usar el motor Unity tanto para juegos independientes como para juegos AAA de gran escala.

La física es un elemento importante para el desarrollo de mundos o entornos donde ocurre un videojuego. Unity utiliza el motor de físicas nVidia PhysX el cual es uno de los más utilizados en el mercado, juegos famosos como World of Warcraft o Titanfall hacen uso del mismo (Nvidia, 2017). Entre sus bondades está la implementación de física en cuerpos sólidos o blandos, Personajes de geometrías complejas o hasta explosiones. Esto quiere decir que se tienen en cuenta temas como la gravedad o la masa para darle un determinado comportamiento a los objetos. Como se observa en la figura 3, a los objetos en Unity se les puede otorgar este tipo de características.

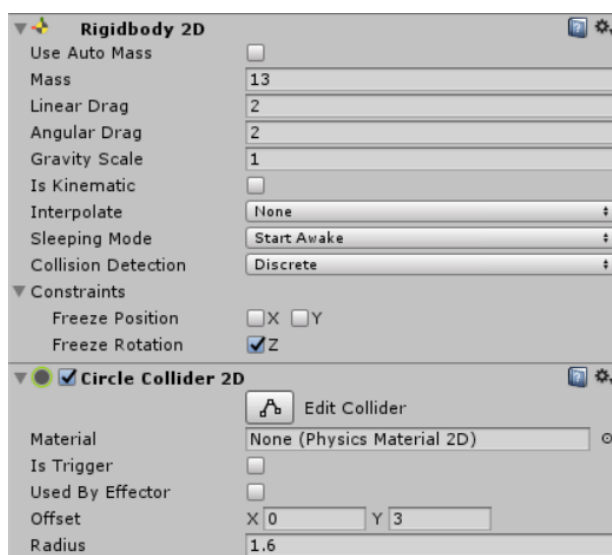


Figura 3. Propiedades físicas de un objeto dentro de Unity

Para poder programar el videojuego Unity ofrece compatibilidad con dos de los lenguajes más usados a nivel mundial: C# y JavaScript. Por otro lado, la integración nativa de Visual Studio está disponible desde la versión de Unity 5.2. y proporciona una experiencia de depuración completa de Unity dentro de Visual Studio que permite por ejemplo correr el videojuego y depurar el código en Visual Studio.

Por último: “Unity cuenta con posibilidades de red para nuestros juegos: sincronización de estados, llamadas a procedimientos remotos, comunicación en tiempo real, conectividad con sistemas como ODBC, integración web...” (Dorado, Julián p. 1)

2.2.2 Editor Unity

El editor de Unity es la herramienta principal con la que se creará el videojuego, este consta de varias secciones o ventanas que permite al desarrollador trabajar de manera más simple y ordenada. Como podemos ver en la figura 4 las siguientes ventanas son:

- **Ventana del proyecto (Project Window):** Muestra los assets (elementos del juego, como audio, texturas, gráficos) que están disponibles.
- **La vista de escena (Scene View):** Esta es la vista donde se trabaja el juego, esta permite navegar visualmente y editar la escena del videojuego
- **La ventana de Jerarquía (Hierarchy):** Según la documentación de Unity “La ventana de jerarquía es una representación de texto jerárquico de cada objeto en la escena. Cada elemento en la escena tiene una entrada en la jerarquía, por lo que las dos ventanas están inherentemente vinculadas. La jerarquía revela la estructura de cómo los objetos están agrupados el uno al otro.” (Unity docs., 2017)
- **La ventana del Inspector (Inspector):** Posibilita la edición y visualización de las propiedades de un objeto seleccionado.
- **Barra de herramientas (Toolbar):** Posee las características más importantes para el desarrollo del juego como la reproducción o la pausa, manejo de cuenta en Unity, manipulación de la escena y objetos.



Figura 4. Editor de Unity y sus principales ventanas.
Adaptada de Unity docs, 2017

2.2.3 Tienda de Unity (Asset Store)

La compartición de contenido es una de las novedades que ofrece Unity, y es que permite a los usuarios compartir ya sea gratuitamente o de forma pagada todo tipo de material para que pueda ser usada por otros desarrolladores, desde música o imágenes hasta personajes, todo tipo de Assets tienen la posibilidad de compartirse o comercializarse.

Para esto existe la tienda de Unity o Asset Store la cual está acoplada con el editor y permite no solo la compra y descarga de contenido para utilizarlo en el juego, sino que facilita la importación del contenido al proyecto en el que se esté trabajando.

2.2.4 Plataformas objetivo de Unity

El poder de Unity radica en que permite desarrollar videojuegos para muchas plataformas diferentes, entre las cuales se encuentran:

- Windows
- iOS

- Android
- Mac
- Linux
- PlayStation 4
- Xbox One
- Oculus Rift
- Nintendo Switch

2.3 Diseño de Personajes/escenarios y Animación

El diseño es un tema de mucha importancia al momento de realizar un videojuego, al ser un medio de entretenimiento, este debe ser vistoso y agradable al público, no se puede centrar únicamente en cumplir con las funcionalidades previstas.

2.3.1 Diseño de personajes y escenarios

El diseño de los personajes o enemigos de un videojuego debe basarse en un proceso que evoluciona con el tiempo y avanza desechando ideas antiguas por nuevas y mejores. Por lo general este proceso lo realizan expertos en el tema como diseñadores gráficos. El diseño de personajes puede dividirse en tres procesos fundamentales (véase figura 5): concepto, elaboración y afinación (Adams, 2014, p. 45).

El concepto es la fase donde se proponen las ideas que luego se mantendrán como objetivos durante el proyecto, la elaboración es la fase donde se detalla el concepto, se agregan los detalles de diseño y se realiza pruebas, y la afinación es cuando se hacen pequeños ajustes o cambios.



Figura 5. Fases del diseño de personajes y escenarios.
Adaptada de Adams, 2014, p. 45

2.3.1.1 Fase de Concepto

El primer paso para el diseño es crear una idea o bosquejar lo que se quiere llevar a cabo, para ello se puede utilizar dibujos a mano, pero ahora existe la tecnología para dibujar directamente en un archivo electrónico. Programas como Adobe Illustrator o Photoshop son ideales para este tipo de trabajo ya que además de todas las herramientas y funcionalidades que ofrecen, se integran completamente con lápices digitales los cuales permiten dibujar con mayor precisión como si se tratará de un papel.

Muchas veces el proceso de concepto requiere que intervengan dos o más personas para poder llegar a un consenso acerca de cómo debe ser un determinado personaje o enemigo. Además, debe establecerse primero un contexto. El personaje debe tener un trasfondo, u objetivo o una historia que permita concebirlo de acuerdo al contexto que lo rodea.

2.3.1.2 Fase de elaboración

a) Modelado

El siguiente paso se aplica tanto a videojuegos 2D y 3D. Si se trabaja en un entorno tridimensional es aquí donde se pasa el dibujo a modelado 3D. El modelado se puede realizar con programas como Maya los cuales permiten generar cuerpos a través de cientos de polígonos. Con el pasar de los años los modelos tienden a tener más y más polígonos, es así como personajes de juegos de antaño tienen una menor cantidad de polígonos. Según Juan Rubio, escritor de la revista Vandal, juegos como Crash Bandicoot para PlayStation lanzado en 1996 contaba con 512 polígonos, mientras que el personaje principal de la serie Uncharted para PlayStation 3 cuenta con alrededor de 30.000 polígonos.

Cuando hablamos de videojuegos en 2D el proceso se simplifica ya que solo es necesario dibujar todo el material necesario de un personaje. Para esto se realizan dibujos donde se encuentren todos los frames o imágenes que se

utilizaran para la animación del personaje. Como no se trata de un modelo 3D es necesario contar con varios ángulos y posiciones para que el dibujo pueda cobrar vida. A la unión de todos los dibujos de un escenario o personaje en un solo archivo se le llama Sprite Sheet, por lo general esto se realiza con el fin de ahorrar recursos para la compilación del juego. La figura 6 es un ejemplo de Sprite Sheet de un personaje.



Figura 6. Ejemplo de Sprite Sheet.
Tomado de codeandweb, 2016

Para trabajar con los Sprite Sheet, Unity posee el Sprite Editor (figura 7) el cual es una poderosa herramienta que permite extraer elementos de una imagen compuesta. Este permite al usuario cortar o seleccionar los recuadros interiores dentro de una imagen para crear dentro de un proyecto varios sprites individuales con los cuales se pueden trabajar o crear animaciones.

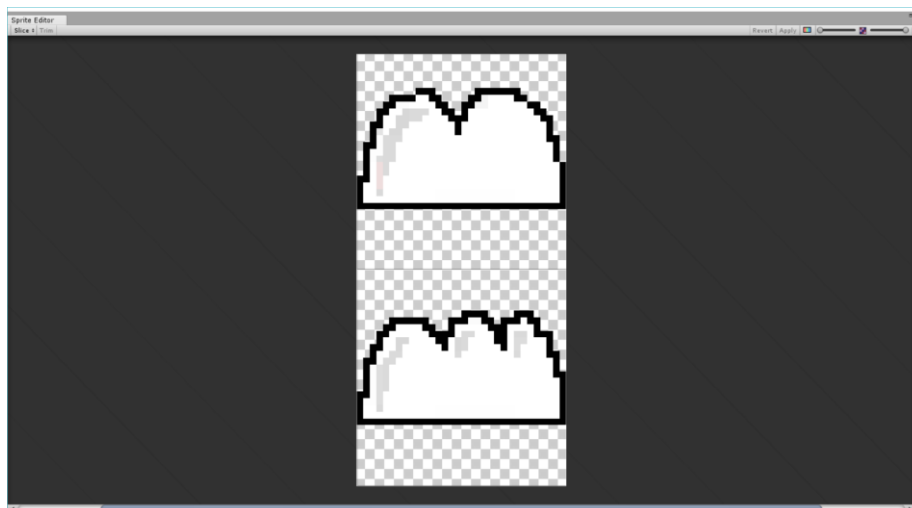


Figura 7. Editor de Sprites de Unity

b) Pintado

El pintado se realiza una vez se haya dado forma al personaje, para gráficos en 2D solo basta con dar el color el momento de realizar los Sprite Sheet, pero para gráficos 3D se utilizan texturas o imágenes 2D que se sobre imponen sobre los polígonos. Esta técnica “se conoce como desplegado (unwrapping), se puede hacer automáticamente, pero si se quiere un mejor resultado es más interesante que el artista distribuye según la importancia.” (Eurogamer, 2016).

c) Animación

El proceso de animación comienza por la creación de un esqueleto para los personajes. A este proceso se le llama Rigging, de esta forma el diseñador y el programador son los encargados de determinar cómo los polígonos o las partes de un Sprite Sheet se mueven de acuerdo al esqueleto del personaje.

Cuando ya se tiene un esqueleto para los personajes se procede a crear un grafo de animaciones el cual permite agregarle cierta lógica al estado de los personajes y su reacción ante un determinado evento (figura 8). Podemos por ejemplo crear un grafo donde se indique el estado o posición base del personaje, por ejemplo, el estar quieto, y crear varias ramas de estados que nacen de esta posición base, como por ejemplo correr o saltar, y que se activen mediante triggers que se llaman en el código.

Es posible animar un elemento de manera más sencilla sin utilizar rigging, esto se consigue presentando varios frames o imágenes de un mismo elemento (una persona que con cada frame eleva su mano, por ejemplo) en modo secuencial a una determinada velocidad de tal manera que parezca que se está moviendo.

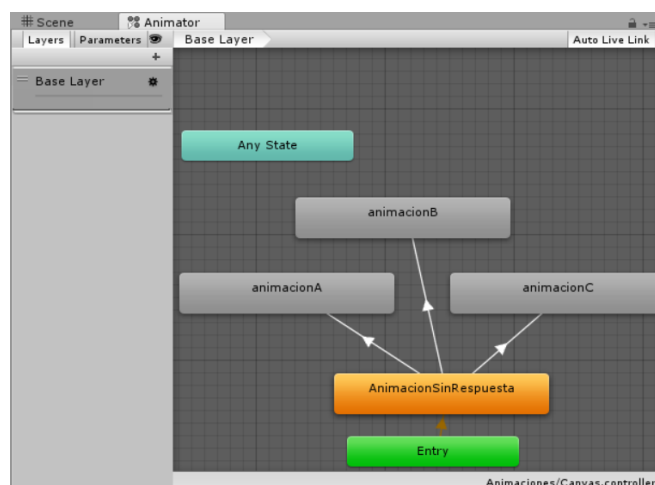


Figura 8. Grafos o Estados de animación en Animador de Unity

2.3.2 Animación en Unity

Unity tiene entre todas sus características un sistema de animación llamado Mecanim el cual permite animar fácilmente objetos, propiedades y personajes. Además, facilita la visualización de clips de animación gracias a un reproductor que maneja Triggers, características de capas y la creación y manejo de grafos.

- **Flujo de la animación en Unity**

Todo el flujo de animación de Unity se basa en clips de animaciones (Animation Clips) los cuales “contienen información acerca cómo ciertos objetos deberían cambiar su posición, rotación, u otras propiedades en el tiempo. Cada clip puede ser pensado como una sola grabación lineal.” (Unity Technologies, 2017).

Estos clips pueden proceder de fuentes externas como procesos de captura de movimientos o herramientas externas como Maya (Software de animación 3D). Una vez se tengan los clips, estos se organizan a modo de grafos en el controlador de animación (Animation Controller) de Unity el cual trabaja como

una “state machine” o máquina de estados ya que es aquí donde se indica que clip debe estar reproduciéndose, cuándo y cómo deben cambiar los clips.

En el caso de personajes humanoides Unity permite hacer un acoplamiento de las animaciones desde cualquier fuente, esto quiere decir que se pueden aplicar el mismo conjunto de animaciones a varios personajes humanoides. Estas características son potenciadas por el sistema Avatar de Unity, donde se manejan los personajes humanoides de manera con un formato integrado.

Por último, todas estas piezas se adjuntan a un objeto (GameObject en Unity) mediante un componente de Animación (Animator Component) o componente de animación. Tal como podemos ver en la figura 9, se describe el la interacción y las partes del sistema de animación explicados. En el punto 1 tenemos los clips de animación, en el 2 tenemos los mismos clips arreglados a manera de estados en el controlador de Animación, en el 3 el esqueleto o Rigging mapeado con el sistema de Avatar y en el 4 tenemos ya el objeto o GameObject que se utiliza en el juego con su componente de Animación anidado el controlador y el avatar.

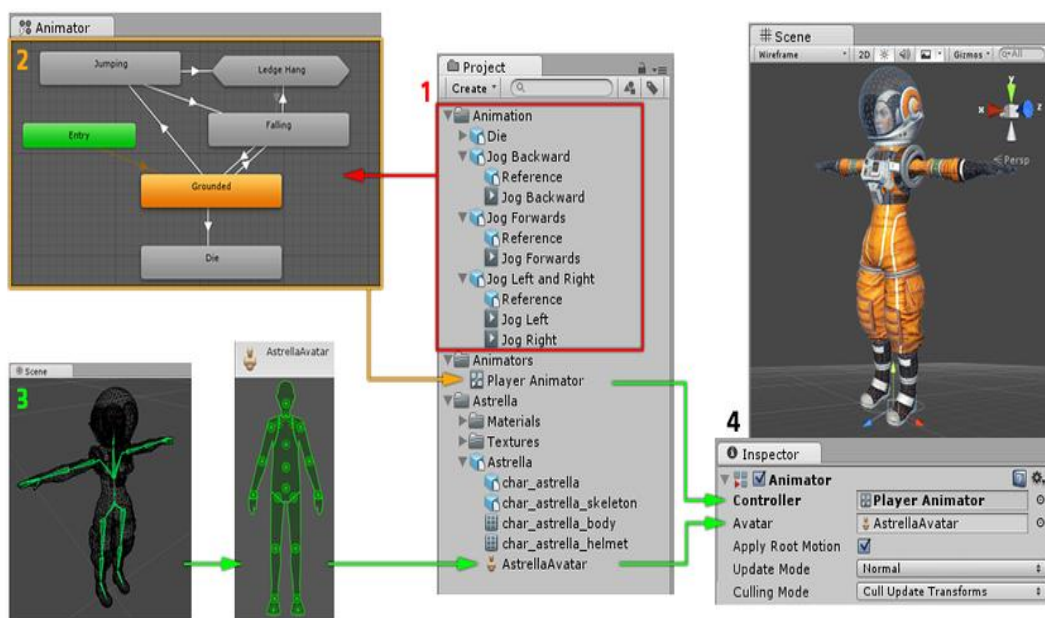


Figura 9. Diagrama del sistema de animación. Tomado de Unity Technologies, 2017

2.4 Educación vial

La educación vial es otro punto importante del proyecto de titulación ya que esta es considerada como el objetivo o propósito de la aplicación. La educación vial son todos los conocimientos o conjunto de reglas que las personas deben de tener en cuenta para transitar en las calles, ya se a pie o en un vehículo. (Ecured, 2017)

El objetivo principal de la educación vial es el de educar a la población para la correcta circulación en las vías, ya sea en condición de peatón o de conductor (de cualquier clase de vehículo) con el fin de que este actúe de manera responsable y pueda prevenir accidentes.

En el sistema de transito intervienen tres factores: el factor humano, el vial y el de máquina. El factor humano está compuesto por los conductores y los peatones. El factor vial son todos los agentes externos como vías, el clima o la señalización. El factor máquina son los vehículos que circulan en las calles. (Aneta, 2013).

Para lograr evitar la mayor cantidad de accidentes es importante que los usuarios conozcan acerca de las señales de tránsito, ya que son estas las que ayudan a prevenir peligros, dan información, controlan el tránsito y aumentan la seguridad.

Según una de las principales escuelas de conducción del país, las señales se dividen de la siguiente manera (Aneta, 2013):

- **Señales verticales:**

- **Regulatorias:** Son aquellas que informan de restricciones para el uso de las vías, son rectangulares con letras negras y símbolos rojos.
- **Preventivas:** Son las que avisan a los conductores sobre posibles peligros para la seguridad y por ende se debe reducir la velocidad. Por lo general son de fondo amarillo y símbolos negros.

- **Informativas:** Indican la dirección o distancia sobre servicios o puntos de interés a lo largo de una ruta.
- **Trabajos en la vía o especiales:** Como su nombre lo dicen indican cuando se realizan trabajo en la vía o cuando existen desvíos, inundaciones, accidentes, derrumbes u otros peligros.
- **Señales horizontales**
 - **Longitudinales:** Se refiere a todas las líneas que están pintadas a lo largo de la vía como las líneas de división del sentido de circulación, división de carriles, de borde y de viraje.
 - **Transversales:** Son la línea de pare, ceda el pase y los cruces peatonales.
- **Semáforos:** Se instalan en intersecciones de mucha afluencia y son para brindar orden y seguridad a los flujos de tráfico.
- **Señales de agentes de tránsito:** Son la señales manuales y luminosas de los agentes de tránsito. Como ejemplo tenemos en la figura 10 la señal de un agente equivalente a la luz amarilla del semáforo.



Figura 10. Señal del agente de tránsito.

Tomado de ANT, 2017

3. Situación Actual y propuesta

Son los accidentes de tránsito una de las principales causas de muerte o que generan miles de personas lesionadas cada año en nuestro país. De ahí nace la

importancia de educar a la gente acerca del tema para poder crear una sociedad que valore no solo su vida, sino que respeta la de los demás. Según datos de la Agencia Nacional de tránsito, en el año 2016 fallecieron 1967 personas en accidentes de tránsito, tal como podemos ver en tabla 2.

Tabla 2.

Número de fallecidos por accidentes de tránsito en 2016.

Provincias	Total, a diciembre de 2016	Porcentaje (%)
Azuay	61	3,10
Bolívar	33	1,68
Cañar	47	2,39
Carchi	34	1,73
Chimborazo	77	3,91
Cotopaxi	58	2,95
El oro	84	4,27
Esmeraldas	48	2,44
Galápagos	-	-
Guayas	458	23,28
Imbabura	78	3,97
Loja	47	2,39
Los ríos	135	6,86
Manabí	141	7,17

Morona Santiago	36	1,83
Napo	13	0,66
Orellana	20	1,02
Pastaza	11	0,56
Pichincha	331	16,83
Santa Elena	43	2,19
Santo domingo de los Tsáchilas	68	3,46
Sucumbíos	40	2,03
Tungurahua	93	4,73
Zamora Chinchipe	11	0,56
Total	1967	100

Adaptado de ANT, 2017.

Las causas de estos accidentes son varias, pero la mayoría se dieron por causas que podrían haber sido evitadas con una correcta difusión de la educación vial. La educación vial trata temas como los límites de velocidad, señales de tránsito, y hasta recomendaciones antes cualquier adversidad (clima, por ejemplo). Si analizamos la tabla 3 podemos observar que alrededor del 70% de muertes son por causas que corresponden con los temas que justamente se imparten en este tipo de educación.

Tabla 3.
Causas probables de fallecimiento en las vías.

Causas Probables	Total, a diciembre de 2016	Porcentaje (%)
Conducir desatento a las condiciones de tránsito (celular, pantallas de video, comida, maquillaje o cualquier otro elemento distractor)	736	37,72
Conducir vehículo superando los límites máximos de velocidad	259	13,17
No ceder el derecho de vía o preferencia de paso al peatón	148	7,52
No transitar por las aceras o zonas de seguridad destinadas para el efecto	128	6,51
No respetar las señales reglamentarias de tránsito (pare, ceda el paso, luz roja del semáforo, por ejemplo)	107	5,44
Otros	589	29,64
Total	1967	100

Adaptado de ANT, 2017

El videojuego que se propone como producto final del proyecto de finalización es un juego que será desarrollado para el sistema operativo Android para toda clase de dispositivos móviles que tenga como fin el educar a las personas en temas de seguridad vial. El jugador interactúa a través de la pantalla táctil de los dispositivos y está pensado para jugadores de cualquier índole. Tanto jugadores

casuales (aquellos que no dedican más de unos minutos a los videojuegos) o jugadores aficionados (aquellos que dedican horas a este pasatiempo) pueden disfrutar del juego.

El juego se trata de un Endless Runner, que en español significa “corredor infinito”. Este tipo de videojuegos se ha vuelto muy popular en los últimos años con exponentes como Temple Run o Mario Run, se trata de un “género donde el jugador debe avanzar de manera irremediable en una misma dirección, generalmente escapando de algún enemigo o peligro, y cuyo objetivo es avanzar lo máximo posible antes de morir” (Gamerdic, 2014)

Con esta premisa se quiere realizar un juego en donde el jugador comience con tres vidas o tres ocasiones en las que pueda topar algún obstáculo, y cuyo puntaje se obtenga mientras más tiempo se mantenga el jugador con vida evadiendo obstáculos ya que el puntaje aumentará con el pasar de los segundos. (figura 11) Ante todo, esto se incluirá preguntas de opción múltiple con temas de seguridad vial que pasarán momentáneamente el juego y le darán al jugador la oportunidad de ganar una vida al jugador o un aumento de puntaje.

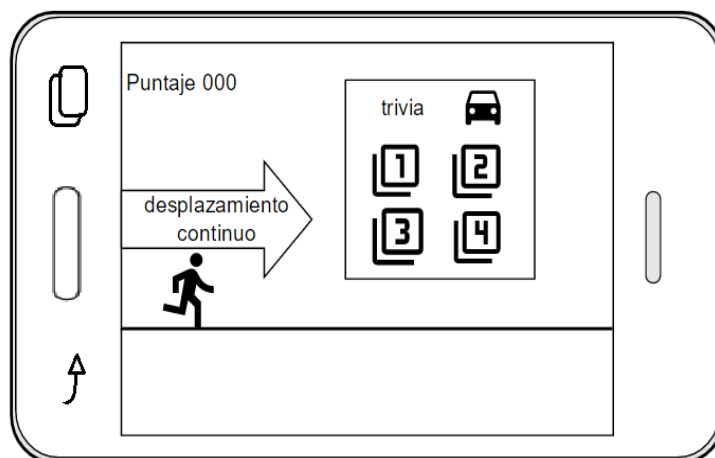


Figura 11. Esquema del funcionamiento básico del juego

Las preguntas de opción múltiple sobre seguridad vial serán seleccionadas de varias fuentes, pero la fuente principal será el banco de preguntas que ofrece la Agencia Nacional de tránsito para aprobar la prueba de licencia B y F (la prueba

más común para obtener la licencia, la de conductor no profesional). Este tipo de preguntas son las más convencionales y las que presentan temas básicos sobre cómo manejarse en las calles.

El proyecto también contempla un modelo de acoplamiento con redes sociales, el cual permite al jugador compartir el juego y retar a sus amigos específicamente en las redes sociales de Facebook y Twitter. En el caso de Facebook el juego accederá a la cuenta asociada con el dispositivo móvil y publicará en su nombre y con su lista de amigos podrá desplegar los contactos a los que el usuario quisiera invitar a descargar la app. En cambio, para Twitter el juego abrirá la aplicación nativa y publicará un tweet con el puntaje del jugador y un enlace para que se pueda descargar el juego desde Google Play. El juego ofrecerá un menú principal que tenga cuatro opciones (figura 12): jugar, compartir en Facebook, compartir en Twitter y salir.

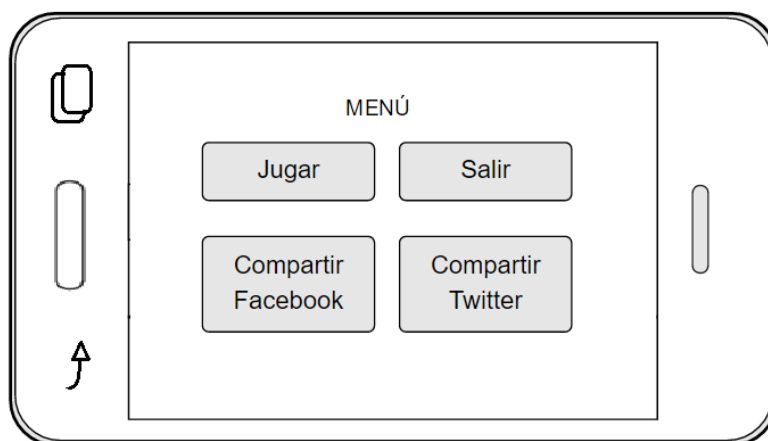


Figura 12. Esquema del menú

Al compartir la aplicación con sus amigos o contactos el jugador obtendrá un bonus de vidas para que pueda conseguir puntajes aún más altos. De esta manera se incita a que el jugador siga jugando al mismo tiempo que aprende sobre seguridad vial y que comparta la aplicación para que más gente la descargue.

4. Análisis Costo/Beneficio

El análisis costo beneficio no toma en cuenta un retorno de inversión económico. La app está planteada para que sea gratuita y cualquier persona con un dispositivo Android pueda bajarla de la tienda de Google llamada Google Play. El beneficio de la realización del proyecto se centra en la fomentación de la educación vial en las personas.

Idealmente en el campo laboral se puede calcular los costos de este tipo de aplicación haciendo un desglose de los elementos necesarios para poder terminar la app, en este caso, en un tiempo estimado de 3 meses. Como se puede ver en la tabla 4 el desarrollo de este tipo de apps requiere de uno o varios desarrolladores acompañados de diseñadores, todo esto dependiendo de la envergadura del proyecto y de la metodología de trabajo. Como se está hablando de una aplicación con gráficos poco complejos en 2D el precio no aumentará, pero cabe mencionar que entre más complicado el juego y mejores los gráficos, el costo es obviamente mayor.

Tabla 4.
Estimación de costo de la app.

Persona	Función	Costo (USD)
Desarrollador (equipo de desarrollo)	Elaboración y programación del juego	Precio hora= 5.00* 80 horas mensuales x 3 meses x 5 hora Total= 1200.00 USD
Diseñador (equipo de desarrollo)	Diseño de personajes, texturas y música.	Precio hora= 5.00* 80 horas mensuales x 3 meses x 5 hora Total= 1200.00 USD
Productor Interno	Dirección, soporte	Precio hora= 5.00*

(Scrum Master)	y coordinación con cliente	80 horas mensuales x 3 meses x 5 hora Total= 1200.00 USD
Total		3600.00 USD

Nota: Se toma en cuenta el sueldo de un programador junior \$800 aproximadamente, se propone un plan a medio tiempo con 20 horas semanales/ 80 horas mensuales. *El precio hora se obtiene de la división de \$800 mensuales entre 160 horas laborales.

Este estimado solo toma en cuenta los sueldos de las personas que tienen que estar involucrados en este tipo de proyectos, pero se debe incluir otros tipos de gastos como el uso de licencias, equipos, infraestructura para obtener un precio real más aproximado.

El producto podría costar aproximadamente la cantidad que se muestra en la tabla 5 si se ofreciera a una empresa o persona, en el caso de proyecto de titulación no se tienen en cuenta estos rubros ya que se asume que el estudiante trabajará en todos los roles y no se estima el costo de equipos o de internet, por ejemplo.

Tabla 5.
Estimación de costo de la app (Software).

Detalle	Costo (USD)
2 licencia Unity Pro (125 USD)	750.00*
1 licencia de Google Developer	25.00
3 equipos (laptops)	2400.00
2 licencia Adobe (Photoshop e Illustrator)	180.00*
1 Tablet Android para testing	400.00
1 Tablet gráfica Wacom	200.00
Servicios Básicos e internet	150.00

Total	4105.00
Total + pagos a personal	7705.00

Nota*: los precios se encuentran multiplicados por los 3 meses que toma el proyecto. Los precios de las licencias son del año 2017

En un escenario empresarial las aplicaciones buscan por lo general captar gran parte del público con aplicaciones gratuitas, pero generan ganancia ya sea a través de compras dentro de la aplicación o con la incorporación de publicidad. De aquí surge el término “freemium” que consiste en “brindar un servicio gratuito a una gran base de usuarios y un servicio premium a unos pocos.” (Gutiérrez, 2014) De esta forma se ofrecen compras o beneficios al jugador que mejora su cuenta a premium.

Como el objetivo es disminuir la cantidad de accidentes y siniestros que ocurren día a día, se puede hacer una comparación con las pérdidas que ocurren cuando ocurren los mismos. “El costo se ve representando en mermas económicas, de infraestructura, gastos médicos y la pérdida productiva de la población afectada” (Diario La Hora, 2011). A pesar de que en nuestro país no existan datos exactos de lo que cuesta realmente este tipo de accidentes, según los datos de la agencia nacional de tránsito “cada muerte le cuesta al país USD 240 000 aproximadamente. En ese monto se incluyen los gastos derivados del siniestro y una estimación económica de lo que dejaría de aportar al Ecuador, en cuanto a su productividad.” (Revista Líderes, 2012) Esto quiere decir que las ciudades pierden agentes productivos con cada persona fallecida y eso sin tomar en cuenta el presupuesto que se necesita para cubrir los daños causados. En la ciudad de Cuenca se destinaba alrededor de 250.000.00 USD en el año 2012 para el plan de seguridad vial que se encarga entre otras cosas de la reparación de los bienes afectados en la urbe durante un accidente. (Revista Líderes, 2012.).

Si tomamos el costo del desarrollo de la aplicación este no sobrepasaría los 8000.00 USD, comparado con los miles de dólares que le cuesta el país cada accidente de tránsito o con los miles de dólares que pierden las víctimas

involucradas, se puede decir que el costo de desarrollo es ínfimo en relación a los beneficios que se obtienen. Con esto dicho, podemos ahora plantear la hipótesis de que lanzando la aplicación al mercado y obteniendo un número considerable de descargas se podría llegar a evitar accidentes mediante una correcta concienciación. El beneficio intrínseco de la aplicación como se dijo es el bienestar social. “La seguridad vial es fundamental en la formación y educación de los ciudadanos, para garantizar el respeto y preservación de la propia vida y la de los demás en las vías públicas.” (Pac,2011) con la difusión de las normas o estándares acerca de la seguridad vial se quiere concientizar y sobre todo educar a la población, con esto no solo se logra que se eduquen las personas, sino que la concientización pueda evitar que se produzcan siniestros o inclusive muertes.

5. Desarrollo del Proyecto

Para el desarrollo del proyecto se ha decidido utilizar la metodología Sum, como se menciona en el apartado de Metodología. Como podemos ver en la figura 13 Sum divide en 5 fases el desarrollo del proyecto.



Figura 13. Metodología SUM.
Tomada de (Acerenza et al., 2009, p. 7)

El ciclo de vida se basa en la presentación de entregables cada cierto tiempo que agreguen valor y permitan una rápida retroalimentación. La fase de concepto es donde se define temas como el público objetivo, elementos de juego como: personajes, gameplay, o elementos técnicos como lenguajes o herramientas para el desarrollo. En la fase de planificación se define el cronograma del proyecto con sus principales hitos. La fase de elaboración consiste en el desarrollo del juego de forma incremental e iterativa y posteriormente tenemos la fase de Beta, la cual sirve para corregir aspectos como gameplay, dificultad y posibles bugs. Por último, el cierre es la etapa donde se entrega la versión final del aplicativo. La gestión de riesgos y esta se realiza durante todo el proyecto cuando se establece la probabilidad, impacto y la forma de mitigar los riesgos.

5.1 Desarrollo de concepto

El principal objetivo de esta parte del desarrollo es definir las bases de todo el proyecto. Es importante que todos los involucrados en el proyecto estén presentes y aporten al desarrollo del concepto, al tratarse de un videojuego se necesita varios puntos de opinión (desarrolladores, artistas, cliente) Sum nos dice que se tienen que definir aspectos técnicos de negocio y técnicos.

- **Aspecto técnico**

Como se ha mencionado anteriormente el juego será desarrollado para dispositivos móviles Android, por la facilidad para desarrollar y la gran cantidad de público que usa este sistema operativo. Por otro lado, se usará el framework de Unity el cual es un poderoso motor de desarrollo para juegos y contenidos interactivos.

Para la programación de los scripts se utilizará el lenguaje C# y el IDE Visual Studio 2015. Se guardará un repositorio del juego en GitHub y para el diseño gráfico se utilizará los productos de Adobe: Illustrator y Photoshop debido a la facilidad de manejar gráficos e imágenes, necesarias para un videojuego en 2D.

- **Aspecto de negocio**

La app se piensa desarrollar para un público que puede ir desde los trece años en adelante ya que presenta temas básicos de seguridad vial los cuales se empiezan a tratar desde la adolescencia. Se trata de una aplicación gratis para los usuarios en la tienda de Google y cuyo beneficio se refleja en la concientización y educación de la gente.

- **Elementos de juego**

Se decidió por elaborar un juego Endless Runner, en el cual un personaje se desplaza por la pantalla constantemente mientras evade obstáculos. El jugador comenzará con tres vidas o tres ocasiones en las que pueda topar algún obstáculo, el puntaje aumenta mientras más tiempo se mantenga el jugador con vida evadiendo obstáculos ya que el puntaje aumentará con el pasar de los segundos.

Se incluirá la opción de compartir la aplicación a través de Twitter y Facebook para incentivar a los usuarios a competir en puntajes y que se distribuya la aplicación (Ver sección 3 Situación actual y propuesta).

5.2 Planificación.

La planificación es una fase muy importante donde se define el cronograma del proyecto junto con sus principales hitos. Como se trata de un videojuego la planificación debe ser flexible para adaptarse al cambio.

Se pueden realizar varias iteraciones como sean necesarias hasta acabar con el desarrollo del producto, pero se debe definir la cantidad de las iteraciones y la duración de cada una de ellas. Para lograr esto se puede acceder a las historias de usuario que son las que constituyen el backlog de nuestro Producto. Backlog en la metodología SUM no es más que una lista de objetivos y requisitos que recompila y prioriza las expectativas del cliente. (ProyectoAgiles, s.f.) (Ver tabla 6). Cada historia de usuario se califica con un riesgo inexistente, bajo, medio o alto, de acuerdo a la posibilidad de no cumplir con los plazos establecidos, a la

dificultad técnica de implementación, o a que el producto final difiera de la necesidad real del cliente.

Tabla 6.
Product Backlog

Numero	Historia de usuario	Sprint	Esfuerzo	Riesgo
1	Salto y desplazamiento del jugador	1	3	alto
2	Generación de obstáculos y escenario	1	3	alto
3	Aumento de puntaje	1	3	alto
4	Muerte del personaje y retención del puntaje más alto	1	3	alto
5	Creación de menú	2	1	medio
6	Creación de cuestionario base de preguntas	2	5	medio
7	Despliegue de preguntas al existir una colisión	2	5	medio
8	Agregar vida/puntaje tras responder trivía	2	1	medio
9	implementación de share por Facebook	3	8	medio
10	implementación de share por Twitter	3	3	medio
11	Dar recompensa (vida/puntaje) al usuario por compartir	3	1	medio
12	Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)	4	3	ninguno
13	Creación e implementación de Sprites (gráficos) de UI (botones, menú)	4	3	ninguno

14	Creación e implementación de sonidos	4	1	ninguno
15	Animación del personaje y fondo	4	3	ninguno
16	Animación de interfaz (botones, menú y obstáculos)	4-5	3	ninguno
17	Manejo de reléase en tienda	5	5	ninguno
18	Publicación para beta	5	3	medio
TOTAL			57	

Si tomamos la definición anterior de que en nuestro Sprint 0 se va a producir todo lo necesario para empezar a trabajar debemos poner énfasis en la preparación de nuestro ambiente de desarrollo.

- **Estimación de Historias de Usuario**

La estimación se realizó utilizando la escala de Fibonacci que se usa en “Planning Poker” y a partir de la experiencia del estudiante (Juicio experto). Planning Poker no es más que una mecánica cuyo objetivo es “obtener una medida de tamaño relativo de todas las historias respecto a sí mismas”. (Allue X. 2015). De esta manera, se definió que para este caso de estudio se iba utilizar una escala de 1,3,5 y 8 donde 1 representa la historia con menor esfuerzo y 8 la de mayor esfuerzo, de existir una historia que tenga un valor mayor a 8 esta tiene que ser separada en tareas más pequeñas. Como no existe un equipo de trabajo con el que comparar o debatir las calificaciones de cada historia se basó en el juicio del estudiante. Los story points lo único que señalan es el tamaño relativo que tiene cada historia del backlog en relación a las demás. Es así como se tomó el product backlog y se escogió la funcionalidad que suponga menor esfuerzo para el equipo de trabajo de acuerdo a su tiempo y experiencia. En este caso la implementación de sonidos o la creación de un menú fueron las historias con más baja calificación, posteriormente se realiza repasa cada historia del backlog y se le asigna una puntuación.

- **Duración de las Iteraciones**

Debido a la característica de este proyecto donde se dispone de 12 semanas para la elaboración del mismo, se dispuso que cada iteración tuviera una duración de dos semanas ya que como se tiene que cumplir con 5 iteraciones es importante que estas se cumplan en el tiempo estimado. De acuerdo a la naturaleza del proyecto y como Sum nos indica este tipo de proyectos requieren muchos cambios y retroalimentación y por eso no se realizan Sprints más largos para tener tiempo suficiente ante cualquier imprevisto.

- **Priorización de Historias y Estimación de Velocidad**

La priorización se da de acuerdo a la importancia del cliente. En este caso se prioriza las historias de acuerdo al nivel de importancia o valor que agreguen al negocio es así que por ejemplo la funcionalidad básica del juego se asigna a la primera iteración mientras que la iteración de diseño y animaciones se puso en último lugar. Otra forma de determinar la prioridad de cada historia es clasificándolas de la siguiente manera:

- Imperativas: aquí por ejemplo tenemos las historias 1 a 8 (salto, desplazamiento, menú y trivía) ya que son las que se necesitan tener obligatoriamente para que funcione el producto.
- Importantes: aquí irían las historias 9 a 11 (interacción en redes sociales) debido a que son importantes porque agregan valor, pero no son urgentes.
- Cosméticas o prescindibles: Aquí se ubican el resto de historias ya que son aquellas que pueden ser pospuestas ante cualquier urgencia y aportan poco valor.

La estimación de velocidad se podría realizar analizando valores históricos de anteriores proyectos similares, esto es beneficios ya que permite un trabajo continuo ya establecido, otra forma de estimar la velocidad es calcular la velocidad tras 2 o 3 iteraciones de tal manera planificar de mejor manera el resto

de Sprints. Como este no es el caso, ya que no existe proyectos previos y es un proyecto corto en el que no se puede esperar a un par de Sprint, se decidió usar la planificación basada en compromiso. En este método y mediante la priorización de historias realizado en el punto anterior, los miembros del equipo escogen tantos objetivos como creen que son capaces de completar en una iteración. De tal forma que la velocidad se verifica viendo que los puntos de historia comprometidos en una iteración sean congruentes con las siguientes. En esta metodología todo miembro del equipo se compromete y se evita estimaciones demasiado optimistas o pesimistas.

5.3 Elaboración

En esta fase se describe las 5 iteraciones planificadas del proyecto. Se incluye la gestión de riesgos, la cual es una fase que debe estar presente a lo largo de todo el proyecto. Según la metodología, Sum necesita que se liberen productos funcionales tras cada iteración de tal manera que se incluye una fase de beta por cada sprint.

5.3.1 Sprint 0

El sprint 0 es el que se realiza antes de desarrollar la aplicación en sí, “el Sprint 0 produce todo lo necesario para poder comenzar a trabajar con un enfoque ágil” (Petit. A, 2017). Es importante acotar que en el sprint 0 por lo general se realiza reuniones y se prepara al equipo de desarrollo para que se establezca roles y una división general del trabajo, este paso se puede omitir en este caso que el estudiante representa todo el equipo de trabajo.

5.3.1.1 Instalación Unity y Visual Studio

La principal herramienta de desarrollo del proyecto, la instalación de la misma es vital para comenzar a programar y desarrollar. Es importante que se disponga de un ordenador con las características necesarias para poder probar y depurar el juego. Según la página oficial los requisitos mínimos son los siguientes (Unity Technologies, 2017)

- OS: Windows 7 SP1+, 8, 10; Mac OS X 10.8+.
- Windows XP y Vista no son compatibles; y las versiones de servidor de Windows & OS X no se han probado.
- GPU: Tarjeta gráfica con DX9 (modelo de shader 3.0) o DX11 con capacidades de funciones de nivel 9.3.

Unity pedirá crear una cuenta para ello se utilizó la misma cuenta que se va asociar a la tienda de Google. La instalación es sumamente sencilla ya que solo se necesita seguir los pasos en pantalla, y lo más importante de esto es que el Editor de Unity instala todos los componentes necesarios para poder crear juegos en Android como el Java JDK y el Android SDK y a su vez permite instalar la versión gratis de Visual Studio (Community) en su última versión, como lo muestra la figura 14.

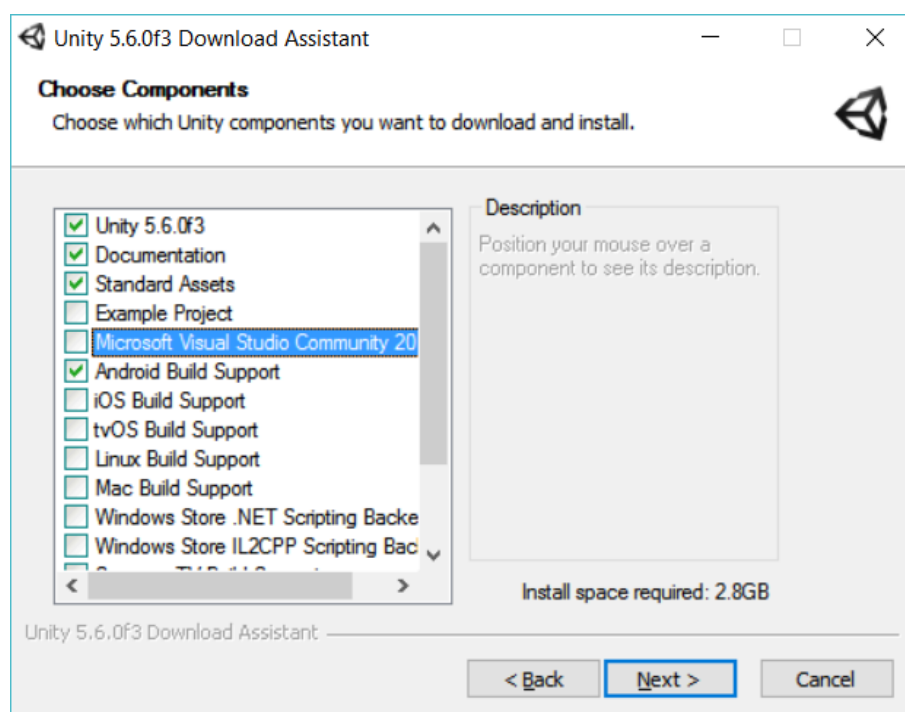


Figura 14. Instalador del editor de Unity.

5.3.1.2 Instalación de Adobe Photoshop e Illustrator

Estos programas son necesarios para la creación del material gráfico del juego. Ambos programas permiten la edición de imágenes, pero Photoshop permite una

mayor modificación a gráficos y fotografías mientras que Illustrator permite crear dibujos desde cero con mayor precisión. Ambos programas tienen compatibilidad con tabletas digitalizadoras gráficas que permiten dibujar a mano dentro de las mismas (figura 15).



Figura 15. Adobe Photoshop para el diseño gráfico

5.3.1.3 Compra de Licencia Android

Esta licencia es necesaria para poder publicar aplicaciones en la tienda de Google Play. La licencia no necesita renovación y en 2017 tiene un precio de 25.00 USD. Se necesita tarjeta de crédito para poder adquirirla y el proceso de obtención de la misma trata simplemente de crear una cuenta de Google y comprar la licencia tras aceptar los términos y condiciones (figura 16)

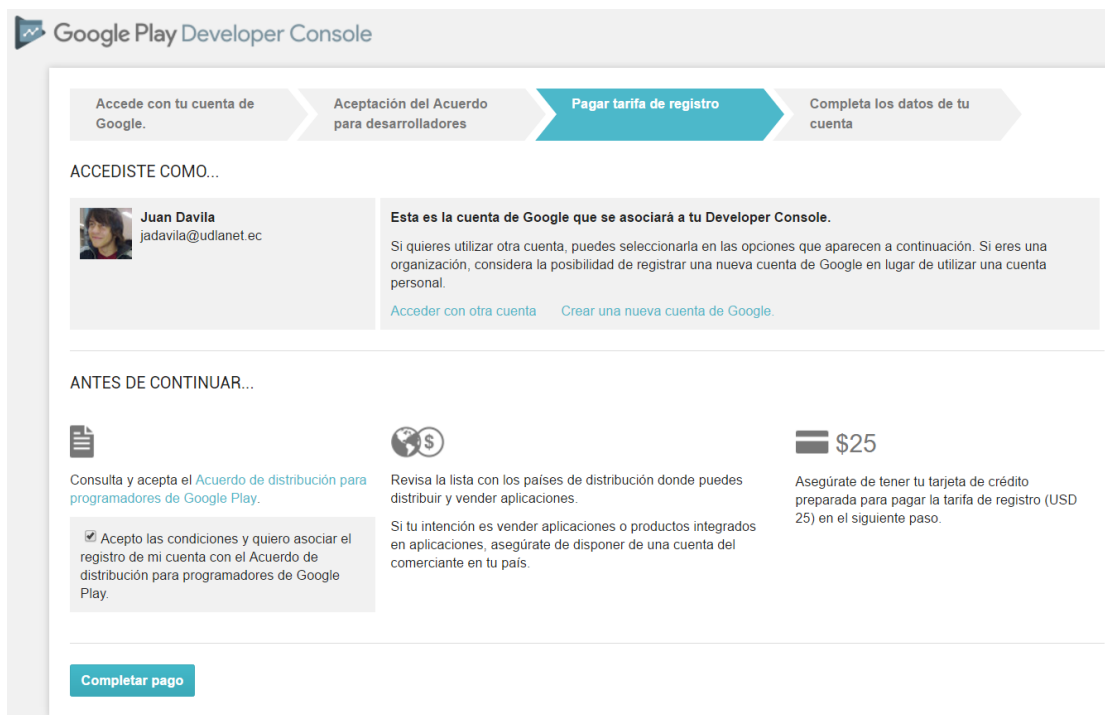


Figura 16. Compra de licencia de desarrollador para Google Play.

5.3.1.5 Creación de historias de usuario en Trello

Trello es una herramienta web que permite la creación de un tablero de trabajo que permita ordenar y editar las tareas a realizarse, se complementa mucho con la metodología sum y los Sprints a desarrollar con sus respectivas historias de usuario. Como se puede observar en la figura 17, se creó el tablero del Sprint 0, en él se puede ver tres columnas principales que son las que normalmente se maneja en este tipo de tableros: tareas pendientes, en ejecución y finalizadas. (“To Do, Doing, Done”).

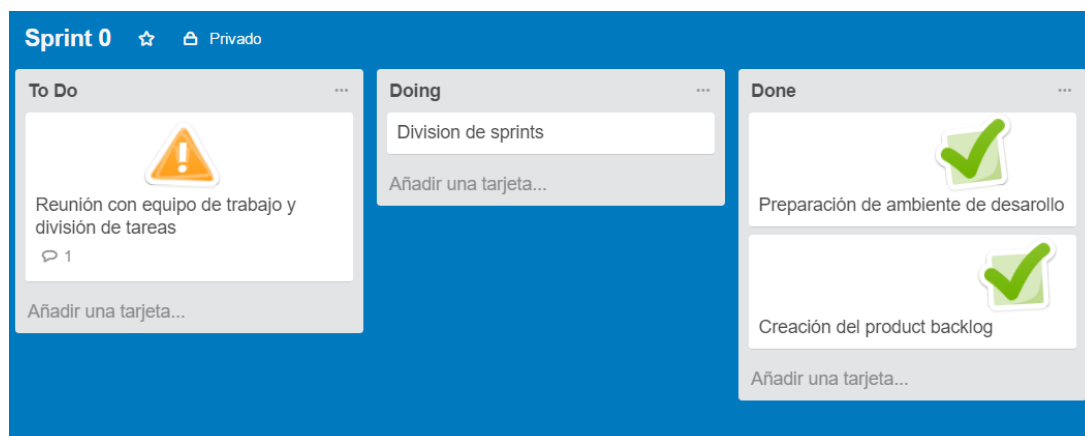


Figura 17. Trello, tareas del sprint 0.

5.3.2 Sprint 1

5.3.2.1 Sprint Backlog

El sprint backlog corresponde a todas las historias de usuario que se desarrollaran en un determinado sprint. (Tabla 7). Los puntos ganados representan los puntos de esfuerzo que se obtuvieron al cumplir con las tareas tras acabar el sprint.

Tabla 7.
Sprint Backlog 1

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
1	Salto y desplazamiento del jugador	1	3	0
2	Generación de obstáculos y escenario	1	3	0
3	Aumento de puntaje	1	3	0
4	Muerte del personaje y retención del puntaje más alto	1	3	0

En este caso el primer sprint corresponde a la funcionalidad básica del juego.

El sprint puede representarse en un tablero como tareas por realizar. (figura 18)

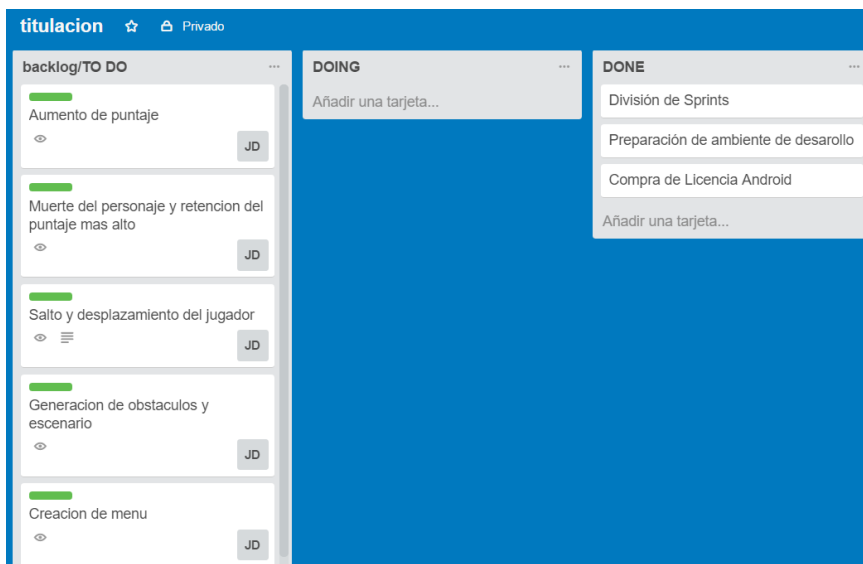


Figura 18. Cuadro de tareas Kanban del Sprint 1

5.3.2.2 Historias de usuario

Tabla 8.

Historia de Usuario 1: salto y desplazamiento

Número	1	Usuario	Jugador
Nombre de la Historia	Salto y desplazamiento del jugador		
Prioridad	Alta	Riesgo	Alto
Puntos estimados	3	Iteración	1
Descripción	Yo como jugador quiero que el personaje corra y salte por el escenario para poder evadir enemigos y activar trivias.		
Validación	El personaje salta máximo dos veces cuando el jugador presiona la pantalla El personaje se mueve hacia la derecha constantemente mientras la cámara lo sigue.		

Tabla 9.

Historia de Usuario 2: Generación de obstáculos y escenario

Número	2	Usuario	Jugador
Nombre de la Historia	Generación de obstáculos y escenario		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	3	Iteración	1
Descripción	Como jugador quiero ver que aparezcan obstáculos conforme el personaje avanza por el escenario para poder evadirlos con saltos.		
Validación	<ol style="list-style-type: none"> 1. Se generan un suelo donde el personaje puede desplazarse constantemente a hacia la derecha. 2. Se genera obstáculos en el escenario conforme avanza el tiempo. 		

Tabla 10.

Historia de usuario 3: Aumento de puntaje

Número	3	Usuario	Jugador
Nombre de la Historia	Aumento de puntaje		
Prioridad	Alta	Riesgo	Alto
Puntos estimados	3	Iteración	1
Descripción	Quiero como jugador quiero que conforme el personaje avance por el escenario su puntaje aumente paulatinamente para poder tener una meta u objetivo.		
Validación	Existe un texto en la parte superior de la pantalla que indica el puntaje del jugador, el mismo que aumenta de 1 en 1 conforme pasan los segundos.		

Tabla 11.

Historia de Usuario 4: Muerte del personaje, disminución de vidas y retención del puntaje más alto

Número	4	Usuario	Jugador
---------------	---	----------------	---------

Nombre de la Historia	Muerte del personaje y retención del puntaje más alto		
Prioridad	Alta	Riesgo	Alto
Puntos estimados	3	Iteración	1
Descripción	Como jugador quiero que la partida termine cuando haya chocado tres veces con los obstáculos para que el juego tenga un final y me rete a seguir jugando		
Validación	<p>El personaje solo puede colisionar con los obstáculos de acuerdo al número de vidas que tenga, en este caso inicialmente 3.</p> <ol style="list-style-type: none"> 1. Una vez perdidas todas las vidas se para el juego finalizando la partida. 2. Se muestra un menú con el puntaje máximo obtenido en todas las partidas jugadas. 		

5.3.2.3 Riesgos

Al tratarse del primer sprint es la parte que más riesgo presenta ya que al tratarse del manejo de un framework nuevo como Unity se puede realizar el desarrollo de manera más experimental. Como se desarrolla la base del proyecto y la funcionalidad básica del mismo es importante que se tenga en cuenta el orden y estándares para poder llevar un mejor trabajo.

En resumen, se identificaron los siguientes posibles problemas:

- Falta de experiencia del equipo de trabajo
- Desconocimiento de las características de Unity y de estándares del mismo
- Posible atraso en la liberación

Para poder mitigarlos se trata de recurrir a la documentación que ofrece Unity Technologies y de realizar un trabajo ordenado y probar cada historia de usuario de manera óptima para poder tener una buena base durante todo el proyecto.

5.3.2.4 Evaluación del estado del videojuego

Como lo indica la metodología es necesario liberar un entregable al final de cada iteración. En esta evaluación se espera encontrar fallas o errores para corregirse a futuro. (figura 19)

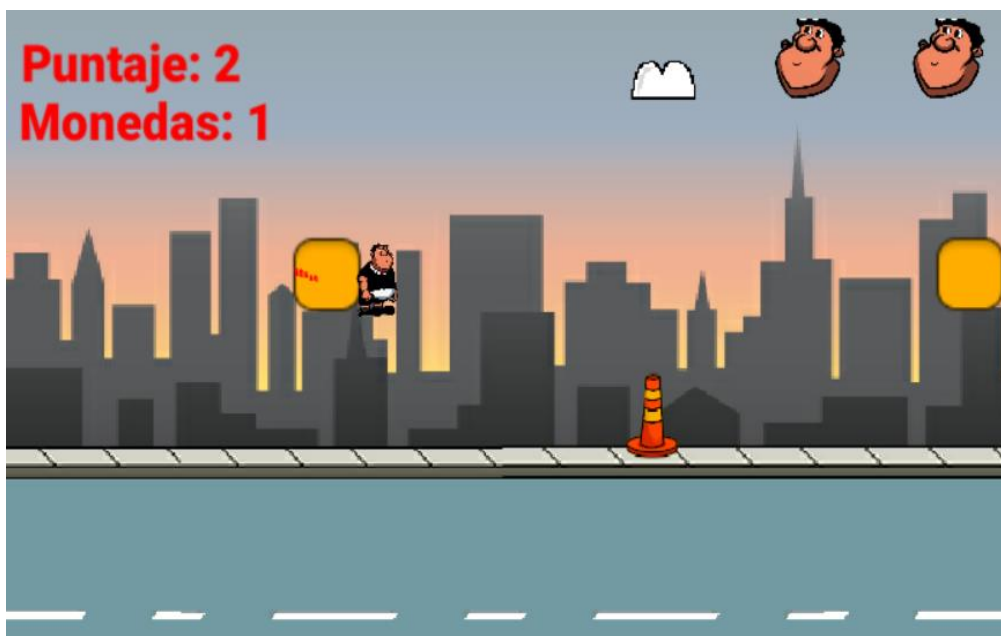


Figura 19. Evaluación del estado del videojuego tras sprint 1

Tras la presentación de la beta se denotaron como era de esperarse pequeñas fallas como la ubicación del personaje respecto a la cámara y es el espacio de visión para evadir los obstáculos y sobre todo se evidenció la falta de velocidad al personaje, lo que hace que el jugador pueda evadir fácilmente los obstáculos resultando en una experiencia poco entretenida.

5.3.2.5 Resultados

Conforme a lo esperado, el motor de física de Unity ayuda mucho a los desarrolladores a controlar el comportamiento de los cuerpos para poder hacer que salten o se desplacen. Las cuatro historias de usuario se pudieron realizar en el plazo establecido. Como se puede ver en la figura 20, existe un personaje que se desplaza en un escenario y pueda saltar.



Figura 20. Resultados de Sprint 1. Desplazamiento del personaje

En la figura 21, por otro lado, podemos ver que aumenta el puntaje con el tiempo y el jugador posee tres vidas u oportunidades.



Figura 21. Resultados de Sprint 1. Salto del personaje

Como todas las tareas se pudieron cumplir a tiempo el cuadro de trabajo con las tareas se actualiza y se pasan las tareas correspondientes a la sección de finalizado (Done) (Figura 22).

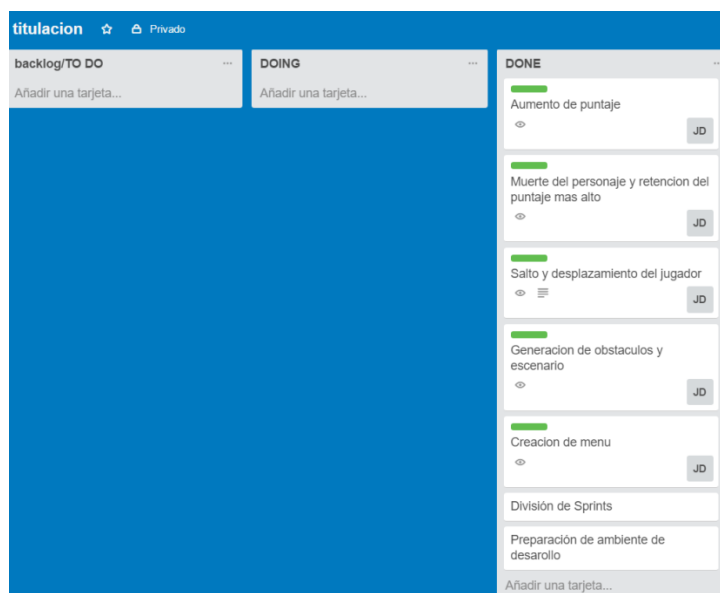


Figura 22. Actualización del cuadro Kanban tras sprint 1.

En la figura 23 se presenta el cuadro de “Burndown Chart” o diagrama de quemado con el cual se ve el avance y cumplimiento de las historias de usuario en el tiempo. El objetivo es que se pueda observar el correcto avance de proyecto (el resultado esperado es que se hayan cumplido todas las historias al final del tiempo estimado)

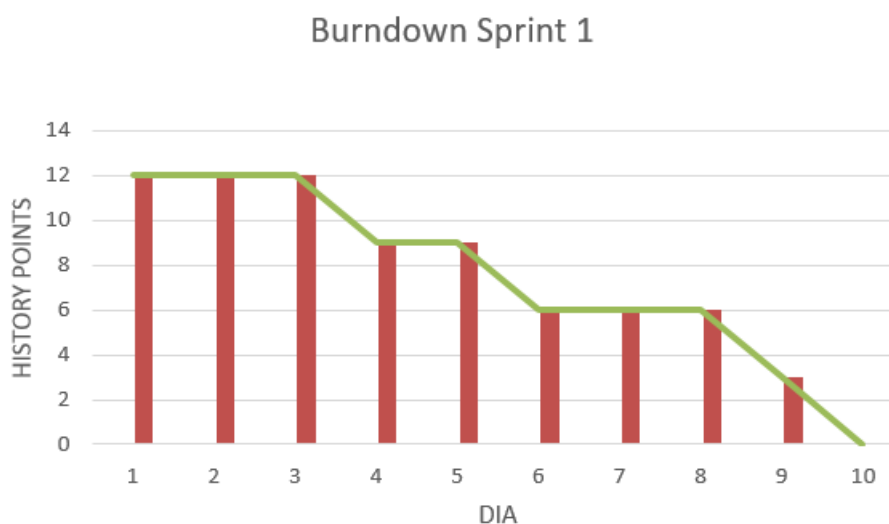


Figura 23. Burndown Primera iteración

Se actualizan los puntos ganados tras la finalización del primer sprint. (Obsérvese Tabla 12)

Tabla 12.
Puntos ganados tras Sprint 1

Núm.	Historia de usuario	Sprint	Esfuerzo	Ganados
1	Salto y desplazamiento del jugador	1	3	3
2	Generación de obstáculos y escenario	1	3	3
3	Aumento de puntaje	1	3	3
4	Muerte del personaje y retención del puntaje más alto	1	3	3

- **Velocidad de desarrollo**

La velocidad de desarrollo de la primera iteración corresponde a la suma de los puntos de historia por cada historia de usuario terminada en este caso 12 puntos

5.3.2.6 Retrospectiva

Este sprint posiblemente fue el más difícil en su realización debido a que es el primer acercamiento del equipo de desarrollo a la tecnología de Unity y por ende el tiempo de elaboración fue el más grande, se espera que este tiempo sea menor conforme se gana experiencia con el editor y con la codificación.

- **Qué cosas han funcionado bien.**

El uso de la herramienta de Unity, ha facilitado mucho el tiempo de desarrollo

- **Qué se puede mejorar.**

No se debe menospreciar los cambios en el aspecto gráfico y el tiempo que ello conlleva, no se debe perder tiempo en incluir cosas que no están dentro de las historias de usuario. La velocidad de desplazamiento del personaje.

- **Qué se ha aprendido.**

El uso de Rigidbody en Unity, o cuerpo rígido no es más que un atributo que se puede agregar a los objetos para que tengan propiedades físicas. La codificación básica para manejar los cuerpos y como asociarlos en conjunto.

- **Cuáles son los problemas que podrían impedir progresar adecuadamente.**

Ninguno

5.3.3 Sprint 2

El sprint 2 se centra en el módulo de trivia de seguridad vial. Por eso se incluyen todas las historias de usuario que correspondan con esta temática. La duración al igual que el resto de Sprint es de dos semanas y se abarca las historias de usuario de la tabla 13

5.3.3.1 Sprint Backlog

Tabla 13.
Sprint 2 Backlog

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
5	Creación de menú	2	1	0
6	Creación de cuestionario base de preguntas	2	5	0
7	Despliegue de preguntas al existir una colisión	2	3	0
8	Agregar vida/puntaje tras responder trivia	2	1	0

Se ponen las historias de usuario de este sprint en el tablero de Trello, figura 24.

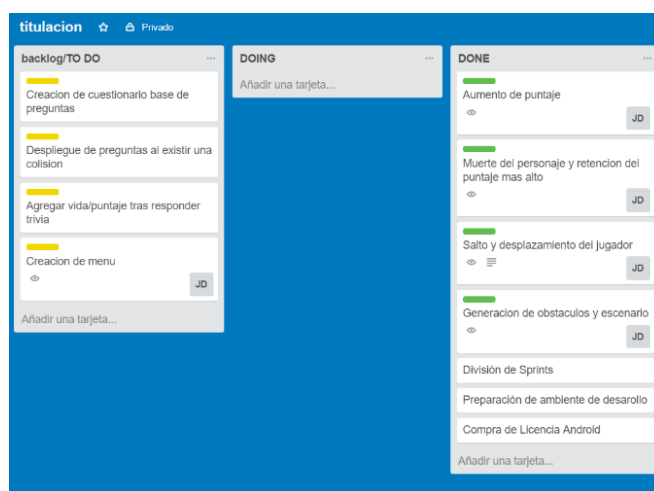


Figura 24. Cuadro de tareas Kanban del Sprint 2

1. Las tareas con etiqueta amarilla corresponden al sprint 2.

5.3.3.2 Historias de usuario

Tabla 14.

Historia de Usuario 5: Creación de menú

Número	5	Usuario	Jugador
Nombre de la Historia	Creación de menú		
Prioridad	Alta	Riesgo	Bajo
Puntos estimados	1	Iteración	2
Descripción	Como jugador quiero ver un menú principal para poder elegir que hacer dentro de la aplicación.		
Validación	1. Aparece un menú con la opción de salir, jugar, compartir en Facebook y compartir en Twitter. 2. Aparece un menú después de morir que permite volver a jugar y volver al menú principal		

Tabla 15.

Historia de Usuario 6: Creación de cuestionario base de preguntas

Número	6	Usuario	Jugador
Nombre de la Historia	Creación de cuestionario base de preguntas		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	5	Iteración	2
Descripción	Como jugador quiero que exista un cuestionario con preguntas adecuadas para la educación vial para poder aprender correctamente acerca del tema.		
Validación	Existe una lista de una clase que tenga como atributos, un id, una pregunta y tres opciones de respuesta. La lista debe estar llena con ítems y puede editarse fácilmente desde Unity		

Tabla 16.

Historia de Usuario 7: Despliegue de preguntas tras una colisión

Número	7	Usuario	Jugador
Nombre de la Historia	Despliegue de preguntas al existir una colisión		
Prioridad	Alta	Riesgo	Bajo
Puntos estimados	5	Iteración	2

Descripción	Como jugador quiero que aparezca un recuadro con una pregunta y tres posibles respuestas sobre temas de seguridad vial para poder elegir la opción correcta.
Validación	<ol style="list-style-type: none"> 1. Al topar el obstáculo aparece un recuadro en pantalla con una pregunta, una imagen, un contador y tres opciones. 2. El contador indica que se tiene 10 segundos para responder

Tabla 17.

Historia de Usuario 8: Agregar vida/puntaje tras responder trivía

Número	8	Usuario	Jugador
Nombre de la Historia	Agregar vida/puntaje tras responder trivía		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	1	Iteración	2
Descripción	Como jugador quiero que el personaje tenga una vida más o tener más puntaje si contesto correctamente a la trivía para poder mejorar mi récord.		
Validación	<ol style="list-style-type: none"> 1. Si se eligió la opción incorrecta se desactiva el panel y el juego continua. 2. Si se eligió la respuesta correcta se continua con el juego, pero se otorga una vida extra o 10 puntos <p>Si pasan los 10 segundos se desactiva el panel y se continua con el juego</p>		

5.3.3.3 Riesgos

El riesgo de esta iteración es medio ya que a diferencia de la iteración anterior aquí ya se trabaja con una base, sin embargo, se identificó los siguientes riesgos:

- Creación de un cuestionario uniforme y que responda a las necesidades del producto.
- Identificación de la mejor manera para mostrar en pantalla la trivía.

Para el primero problema se decidió utilizar como base el banco de preguntas para obtener la licencia de conducir en nuestro país, se estableció como requisito

que la pregunta sea fácil de entender y que no sea demasiado larga ya que este tipo de preguntas alejan o aburren al jugador.

El segundo riesgo fue mucho más difícil de mitigar ya que se debe elegir la forma que se presenta al jugador el módulo de trivia, en un principio se optó por usar una pantalla o escena que solo presente la trivia al jugador, pero se terminó optando por incluir la trivia en medio juego mientras el personaje colisiona con un determinado objeto, esto con el propósito de hacer más dinámico el juego.

5.3.3.4 Evaluación del estado del videojuego

De igual forma se encontraron algunas falencias como por ejemplo el color de los paneles de pregunta no siempre permite una rápida lectura lo que dificultaba la elección de una respuesta en el tiempo límite.

Además, se evidencio la necesidad de revisar el banco de preguntas para asegurarse que todas las imágenes correspondan al texto que se presenta al usuario, esto debido a que se reportó la aparición de una pregunta con una imagen que no correspondía.

5.3.3.5 Resultados

El sprint se completó de acuerdo a lo estimado, el personaje puede saltar y colisionar o topar un objeto que tenga un símbolo de pregunta. Se pensó en utilizar una base de datos con el cuestionario de seguridad vial, pero se utilizó un objeto que tenga un array de clase Pregunta. (véase figura 25).

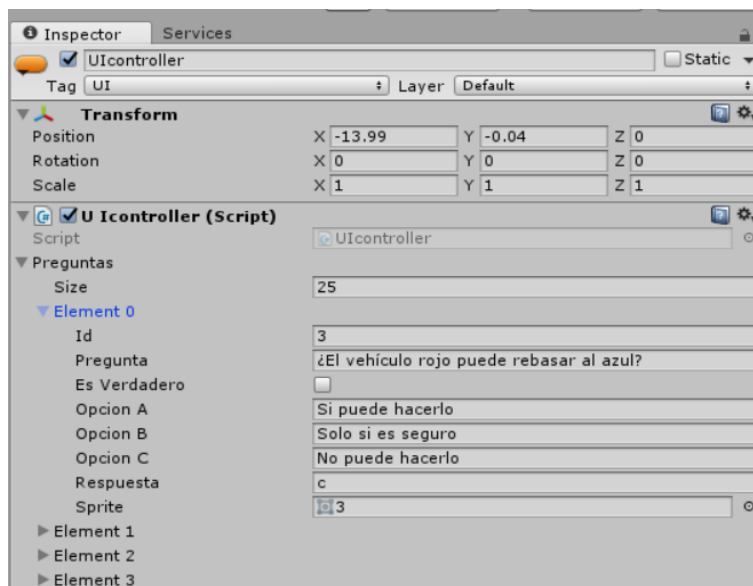


Figura 25. Resultados sprint 2. Objeto con array para trivia

Cuando se dé dicha colisión el juego automáticamente se pausará y aparecerá una ventana con una pregunta al azar sobre seguridad vial. Esta pregunta al azar será seleccionada de una lista de objetos de la clase Pregunta (figura 26).



Figura 26. Resultados sprint 2. Colisión con objeto de trivia

El usuario tiene tres opciones o botones que puede presionar, cada botón tiene una posible respuesta. (figura 27)

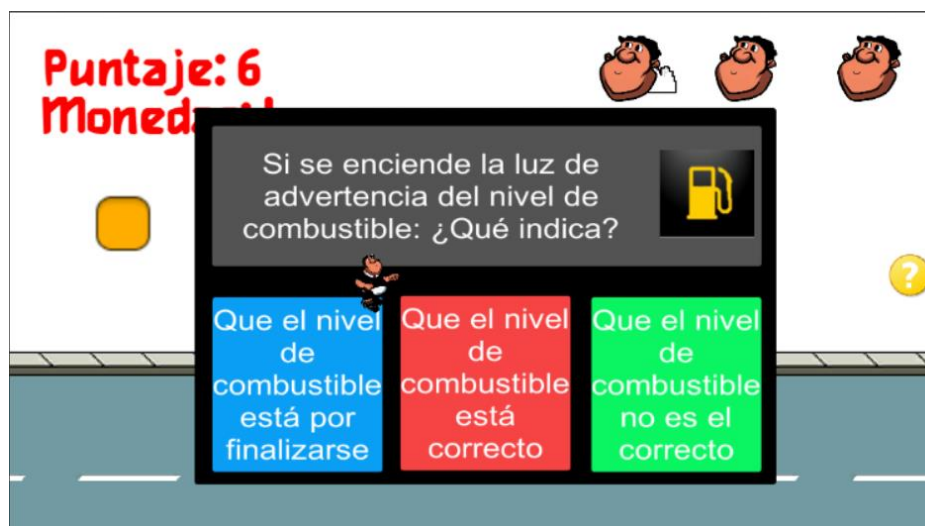


Figura 27. Resultados sprint 2. Presentación de trivia.

Si la respuesta es la correcta el usuario es recompensado con una vida o con 10 puntos al total de su puntaje actual, caso contrario el juego seguirá como si nada hubiese ocurrido. (Figura 28)



Figura 28. Resultados sprint 2. Aumento de vida.

Ahora al iniciar la aplicación se mostrará primero un menú de acuerdo al esquema indicado previamente en la figura 11 (Véase figura 29).



Figura 29. Resultados sprint 2. Implementación del menú
1. Diseño está sujeto a cambios

Como todas las tareas se pudieron cumplir a tiempo el cuadro de trabajo con las tareas se actualiza y se pasan las tareas correspondientes a la sección de finalizado (Done) (Figura 30).

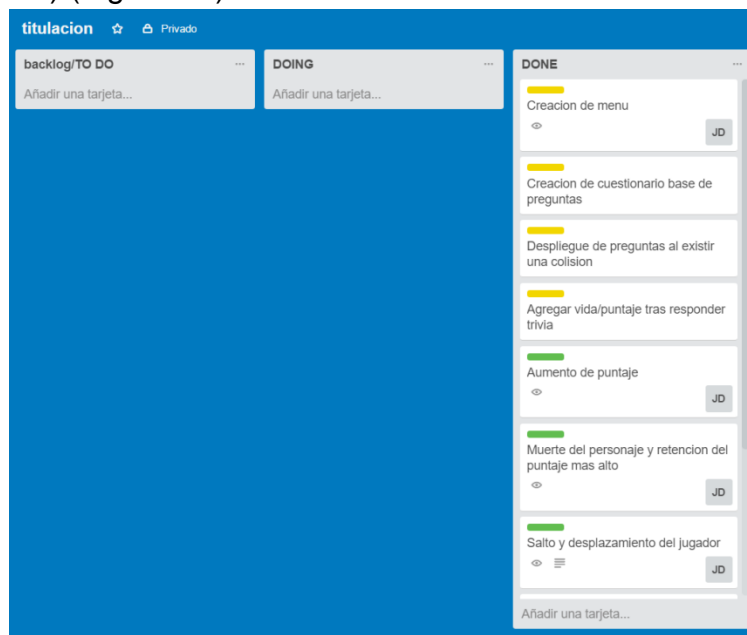


Figura 30. Actualización del cuadro Kanban tras sprint 2.

En la figura 31 se presenta el Burndown del segundo sprint, por el momento como se cumplieron todas las historias de usuario, la evolución coincide con el resultado esperado. Hay que notar que en los días 7 y 8 se volvió a incrementar los history points pendientes debido a la aparición de una tarea relacionada despliegue de preguntas, el cual era la inclusión de un temporizador que permita mostrar el panel de preguntas solo durante 10 segundos.

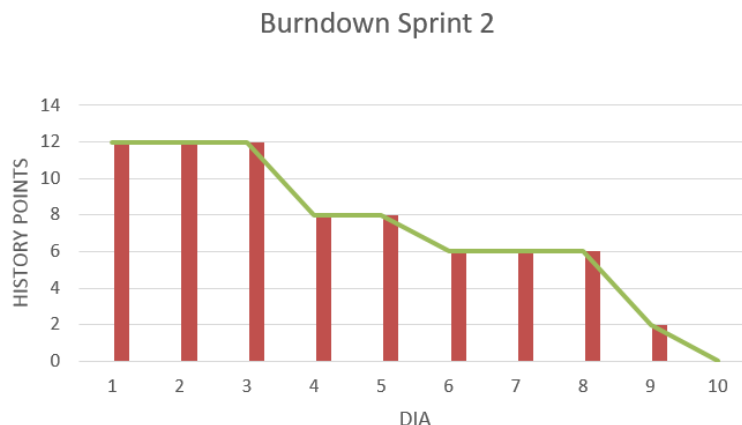


Figura 31. Burndown segunda iteración

Se actualizan los puntos ganados tras la finalización del primer sprint.
(Obsérvese Tabla 18)

Tabla 18.

Puntos ganados tras Sprint 2

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
5	Creación de menú	2	1	1
6	Creación de cuestionario base de preguntas	2	5	5
7	Despliegue de preguntas al existir una colisión	2	3	5
8	Agregar vida/puntaje tras responder trivía	2	1	1

- **Velocidad de desarrollo**

La velocidad de desarrollo de la primera iteración corresponde a la suma de los puntos de historia por cada historia de usuario terminada en este caso 10 puntos. En este punto podemos decir que la velocidad de desarrollo es de 11 ya que hacemos el promedio con el anterior sprint.

5.3.3.6 Retrospectiva

A pesar de que este sprint se podía presentar más difícil porque se trata de la creación de una mecánica propia de la aplicación la cual tiene que ser programada desde cero y no existe guías para basarse, el sprint terminó de

acuerdo a lo establecido. Se evidenció la cantidad de recursos en línea con los que se puede aprender para la creación de diversos proyectos.

- **Qué cosas han funcionado bien.**

El uso de un GameObject que anide a su vez una lista con objetos de una clase, de esta forma se ahorró el tiempo de desarrollo y se simplifico la creación de una base de preguntas.

- **Qué se puede mejorar.**

Se plantea incluir en el siguiente sprint un contador que solo muestre la trivia durante diez segundos, se ha decidido que no resulta conveniente dejar que el jugador tenga todo el tiempo para responder.

- **Qué se ha aprendido.**

La forma de agrupas objetos para la interfaz gráfica mediante el uso de Canvas en Unity, además como pausar una aplicación mediante la interrupción del paso del tiempo (en Unity se asume que el tiempo transcurre normalmente).

- **Cuáles son los problemas que podrían impedir progresar adecuadamente.**

Las preguntas pueden ser difíciles para contestar lo que desaliente al jugador. El cambio de escena, al momento de ir al menú ocupa recursos de memoria del dispositivo y podría ralentizar la aplicación.

5.3.4 Sprint 3

5.3.4.1 Sprint Backlog

El tercer sprint trata del módulo de interacción con redes sociales. El proyecto plantea que la aplicación se propague en el mercado a través de las redes sociales. Los usuarios serán capaces de compartir los puntajes obtenidos mediante Twitter y Facebook. Se motiva al jugador a que comparta el juego de tal manera que pueda obtener mejores puntajes después de compartir el juego sus contactos (Tabla 19).

Tabla 19.
Sprint Backlog 3

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
9	Implementación de share por Facebook	3	8	0
10	Implementación de share por Twitter	3	3	0
11	Dar recompensa (vida/puntaje) al usuario por compartir	3	1	0

Las tres tareas de este sprint tienen que ubicarse en el tablero de Trello para poder seguir el avance de las mismas (figura 32).

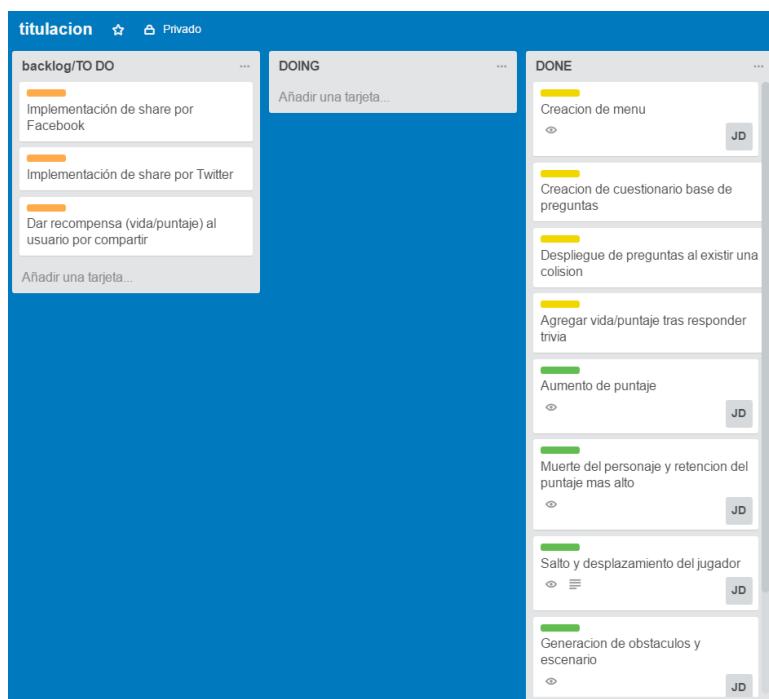


Figura 32. Cuadro de tareas Kanban del Sprint 3.
1. Las tareas con etiqueta naranja corresponden al sprint 3

5.3.4.2 Historias de usuario

Tabla 20.
Historia de Usuario 9: Implementación de share por Facebook

Número	9	Usuario	Jugador

Nombre de la Historia	Implementación de share por Facebook		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	8	Iteración	3
Descripción	Como jugador quiero poder invitar a la gente a jugar el juego mediante Facebook para poder compartir mi experiencia con mis contactos.		
Validación	<ol style="list-style-type: none"> 1. Existe en menú con el logo de Facebook en el menú. 2. Se habilitan dos botones nuevos, uno para compartir y otro para invitar. 3. Si el usuario comparte el juego aparecer en su muro de Facebook. <p>Si el usuario invita a sus amigos estos reciben una invitación con el link de la aplicación para descárgala en su Android.</p>		

Tabla 21.

Historia de Usuario 10: Implementación de share por Twitter

Número	10	Usuario	Jugador
Nombre de la Historia	Implementación de share por Twitter		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	3	Iteración	3
Descripción	Como jugador quiero poder invitar a la gente a jugar el juego mediante Twitter para poder compartir mi experiencia con mis contactos.		
Validación	<ol style="list-style-type: none"> 1. Existe un botón con el logo de Twitter en el menú. 2. Se abre una ventana con texto predeterminado indicando el puntaje obtenido y el enlace a la tienda para descargar la app. 		

Tabla 22.

Historia de usuario 11: Dar recompensa (vida/puntaje) al usuario por compartir

Número	11	Usuario	Jugador
Nombre de la Historia	Dar recompensa (vida/puntaje) al usuario por compartir		
Prioridad	Alta	Riesgo	Medio
Puntos estimados	1	Iteración	3
Descripción	Como jugador quiero tener más vidas tras compartir la aplicación para poder obtener mejores puntajes.		
Validación	<ol style="list-style-type: none"> 1. Una vez compartido el juego el usuario inicia una partida nueva con una vida más si compartió por Facebook, Twitter o si invito amigos. 2. Si hizo las tres acciones se le agregara 10 puntos y 2 vidas. 		

5.3.4.2 Riesgos

Esta etapa presento mucha incertidumbre que podría convertirse en lentitud al desarrollar o en el no cumplimiento de los objetivos. Los principales riesgos de esta iteración son:

- Incorporación de un SDK desconocido
- Acoplamiento de funcionalidades nuevas en diferente entorno, hasta el momento se trabajó en C# en Unity, pero se agregan nuevas funcionalidades que corresponden al SDK de Facebook.

5.3.4.2 Evaluación del estado del videojuego

Tras probar el entregable al final de la iteración se puede decir que el juego cumple con todos los requisitos ya que se cumplieron todas las historias de usuario. A nivel de errores se pudo detectar que algunas veces la cancelación de un tweet en Twitter hace que se permanezca en la aplicación de Twitter, pero eso es normal ya que el usuario solo tiene que volver al juego en cualquier momento.

A su vez a través de varias pruebas se determinó que el SDK de Facebook es solo compatible con sistemas Android y a pesar de que el envío de notificación es al usuario es efectivo, un usuario en iOS o en Windows Phone no podrán redirigirse a la tienda de Google.

5.3.4.3 Resultados

Los resultados son los esperados de acuerdo al sprint 3, se logró hacer que la aplicación utilice el SDK de Facebook para poder iniciar sesión y compartir el juego con los contactos del usuario. Como se puede apreciar en la figura 33 el usuario puede continuar con su sesión del dispositivo móvil cuando hace clic en la función de Facebook.

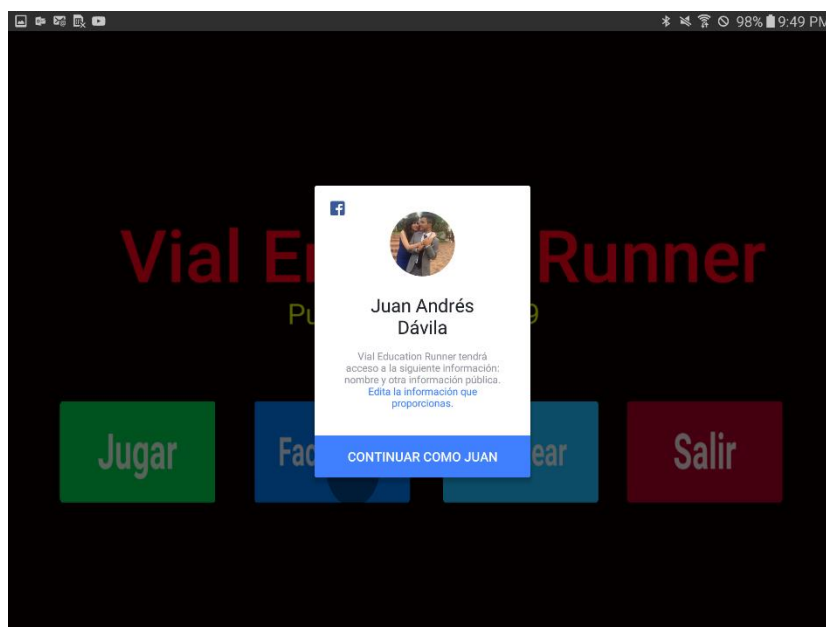


Figura 33. Resultados Sprint 3. Acceso a información de Facebook.

Una vez que se haya accedido a la información del usuario en la red social se habilitarán los botones de compartir e invitar. La figura 34 presenta como una vez iniciada sesión estos botones se muestran en pantalla (la presentación en pantalla puede variar en las siguientes iteraciones).



Figura 34. Resultados Sprint 3. Inicio de sesión en Facebook.

Si el usuario hace clic en la opción de invitar se desplegará una ventana con todos los contactos en la red social. Se puede elegir uno o varios contactos y después ingresar un mensaje personalizado con el link de la tienda de Google para descargar la aplicación. Por otro lado, en usuario también puede hacer clic en la función de compartir y puede escribir un mensaje que aparecerá en la biografía (se muestra a todos los contactos) dentro de la red social (figura 35).

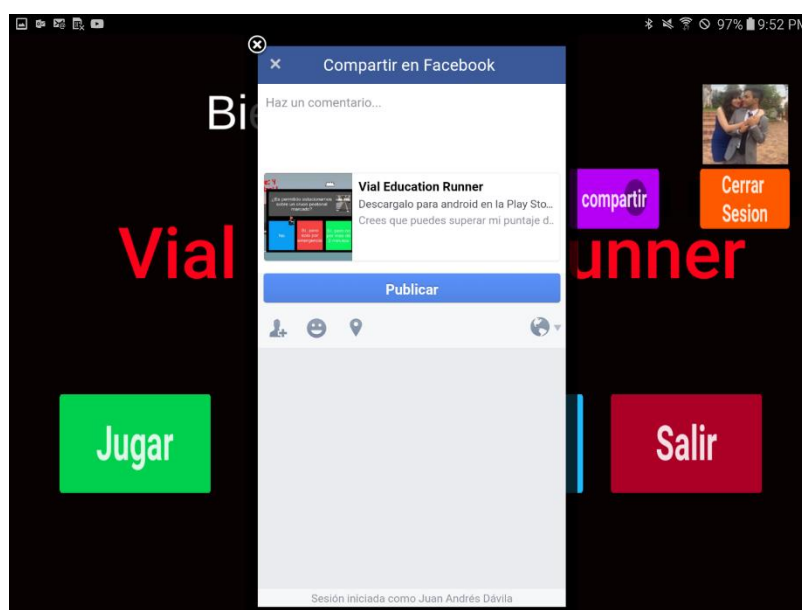


Figura 35. Resultados Sprint 3. Compartir en Facebook.

De igual manera si el usuario quiere compartir el juego en Twitter este presiona el botón correspondiente para que se abra una ventana con un tweet con el puntaje y el vínculo para descargar la aplicación en la tienda (figura 36).



Figura 36. Resultados Sprint 3. Compartir en Twitter.

EL usuario tiene una recompensa en forma de vidas al iniciar la aplicación tras haber compartida la misma en redes sociales. (figura 37).



Figura 37. Resultados Sprint 3. Aumento de vidas

En la figura 38 se puede apreciar que se pasan las tareas del sprint 3 a la sección de finalizados (Done en el tablero Kanban), ya que el aplicativo cumple con todos los requisitos.

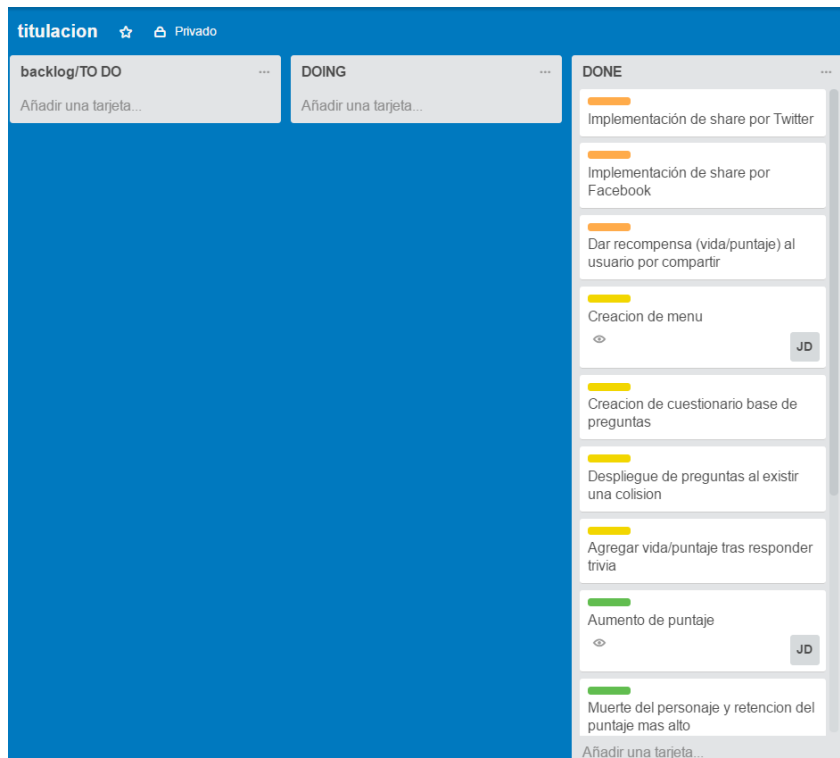


Figura 38. Actualización del cuadro Kanban tras sprint 3.

Como se puede observar en la figura 39, el Burndown de este sprint permaneció en 12 puntos durante bastante tiempo esto debido a que se destinó bastante esfuerzo a la solución de errores del sprint pasado y a la documentación.

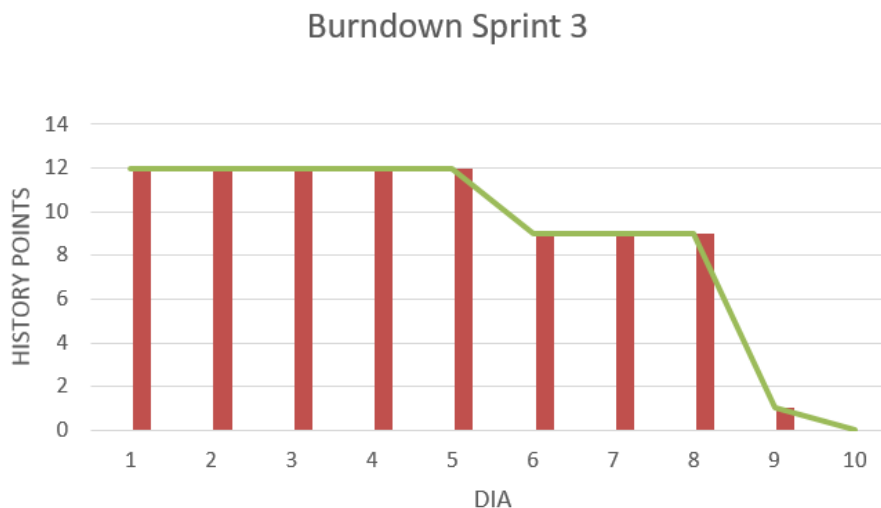


Figura 39. Burndown tercera iteración

Se actualizan los puntos ganados tras la finalización del tercer sprint. (Obsérvese Tabla 23)

Tabla 23.
Puntos ganados tras Sprint 3

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
9	Implementación de share por Facebook	3	8	8
10	Implementación de share por Twitter	3	3	3
11	Dar recompensa (vida/puntaje) al usuario por compartir	3	1	1

5.3.4.4 Retrospectiva

Se puede decir que este es el sprint de mayor dificultad ya que se realizó la tarea de mayor esfuerzo en todo el proyecto, pero gracias a que existió una planificación correcta se pudo otorgarle un mayor tiempo para poder desarrollarla.

- **Qué cosas han funcionado bien.**

EL uso de SKD de Facebook ha facilitado la incorporación de funciones como el invitar o compartir.

- **Qué se puede mejorar.**

Se puede mejorar los tiempos de desarrollo si se busca las fuentes oficiales de cualquier tecnología, es decir no buscar información en fuentes externas como primera alternativa.

- **Qué se ha aprendido.**

El manejo del SDK de Facebook y como llamar a los principales métodos que este ofrece. Acceder a la documentación es vital para este tipo de complementos, donde se puede ahorrar mucho tiempo al leer las recomendaciones del creador del SDK.

- **Cuáles son los problemas que podrían impedir progresar adecuadamente.**

La compatibilidad del SDK con la versión 5.6 de Unity ha sido un problema por lo que se tuvo que bajar a las versiones 5.5. A pesar de que es una versión antigua puede causar problemas por temas de compatibilidad con otros componentes que se quieran agregar a futuro.

5.3.5 Sprint 4

El sprint cuatro se centra en el diseño del juego. Al tratarse de un videojuego y no de una aplicación normal, se debe poner mucho más esfuerzo a la parte gráfica o visual.

5.3.5.1 Sprint Backlog

Pensando en cumplir con el objetivo de tener una interfaz y una jugabilidad atractiva para el usuario se han propuesto las siguientes historias (tabla 24)

Tabla 24.
Sprint 4 Backlog

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
12	Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)	4	3	0
13	Creación e implementación de Sprites (gráficos) de UI (botones, menú)	4	3	0
14	Creación e implementación de sonidos	4	1	0
15	Animación del personaje, obstáculos y fondo	4	3	0
16	Animación de interfaz	4-5	3	0

Se ponen las historias de usuario en el tablero de Trello, la etiqueta roja señala que las tareas pertenecen al sprint 4, figura 40.

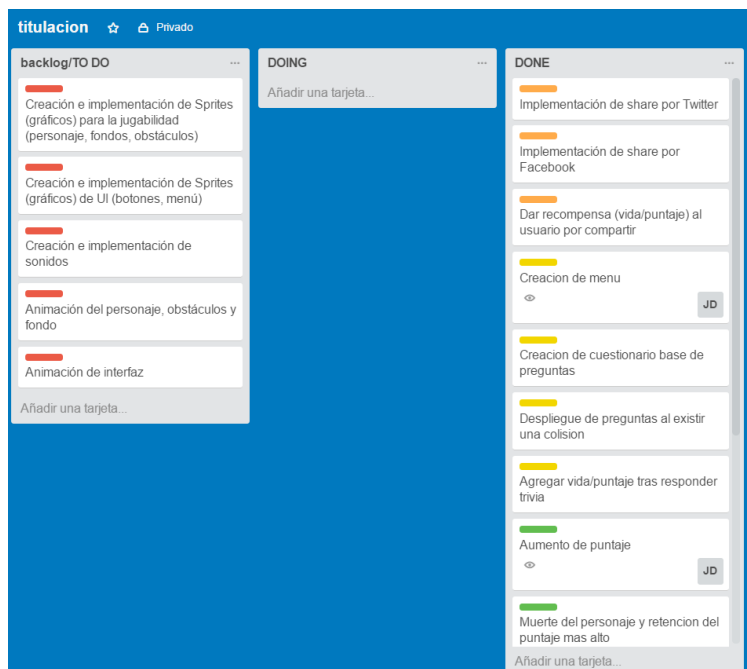


Figura 40. Cuadro de tareas Kanban del Sprint 4

5.3.5.2 Historias de usuario

Tabla 25.

Historia de Usuario 12: Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)

Número	12	Usuario	Jugador
Nombre de la Historia	Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)		
Prioridad	Media	Riesgo	Bajo
Puntos estimados	3	Iteración	4
Descripción	Como cliente quiero que el juego tenga gráficos en 2D vistosos para llamar la atención del jugador		
Validación	El jugador debe encontrarse con un juego que le llame la atención, que tenga colores vivos, dibujos acordes al tema y que mantenga cierta armonía.		

Tabla 26.

Historia de Usuario 13: Creación e implementación de Sprites (gráficos) de UI (botones, menú)

Número	13	Usuario	Jugador
--------	----	---------	---------

Nombre de la Historia	Creación e implementación de Sprites de UI (botones, menú)		
Prioridad	Media	Riesgo	Medio
Puntos estimados	3	Iteración	4
Descripción	Como cliente y jugador quiero que la aplicación tenga botones y menús simples pero llamativos para poder utilizarla fácilmente.		
Validación	Los botones mantienen un diseño uniforme en todos los menús, se presentan colores claros y vivos que reflejen la visión del juego.		

Tabla 27.

Historia de Usuario 14: Creación e implementación de sonidos

Número	14	Usuario	Jugador
Nombre de la Historia	Creación e implementación de sonidos		
Prioridad	Media	Riesgo	Bajo
Puntos estimados	1	Iteración	4
Descripción	Como jugador quiero que el juego tenga sonidos y música de acuerdo a la temática casual y educativa del juego para poder disfrutar más y orientarme dentro del juego.		
Validación	<ol style="list-style-type: none"> 1. El personaje comienza a desplazarse y cuando salta se reproduce un sonido. 2. Se reproduce un sonido tras una colisión. 3. Se reproduce el sonido de reloj cuando se abre la trivía para alertar del paso del tiempo 4. Existe música de fondo 5. Existe musca de fondo en el menú. 		

Tabla 28.

Historia de Usuario 15: Animación del personaje y fondo

Número	15	Usuario	Jugador
Nombre de la Historia	Animación del personaje y fondo		
Prioridad	Media	Riesgo	Bajo

Puntos estimados	3	Iteración	4
Descripción	Como cliente quiero que los personajes y los obstáculos se muevan y demuestren vida al usuario para llamar su atención.		
Validación	1. El personaje tiene la animación de correr y saltar. 2. Los obstáculos se mueven (no necesariamente todos) EL fondo se mueve lateralmente con el tiempo.		

Tabla 29.

Historia de Usuario 16: Animación de interfaz (botones, menú y obstáculos)

Número	16	Usuario	Jugador
Nombre de la Historia	Animación de interfaz (botones, menú y obstáculos)		
Prioridad	Media	Riesgo	Medio
Puntos estimados	3	Iteración	4
Descripción	Como cliente quiero que la interfaz gráfica tenga animación para que el jugador se sienta atraído por la aplicación.		
Validación	Deseo que el menú y los botones tengan animaciones simples que brinden de más calidez y vida a la interfaz		

5.3.5.3 Riesgos

Al tratarse del diseño gráfico del juego en esta iteración es la que se presenta menor riesgo. La complicación de este sprint viene por el tiempo que se le dedica ya que la animación y el diseño puede complicarse cuando el equipo es de una sola persona.

- Diseños centrados en el gusto de una sola persona, no existe un consenso de la paleta de colores o de la temática con algún cliente.
- Tiempo de ejecución. Este sprint presenta actividades que son normales para un diseñador gráfico, por ende, pueden tomar más tiempo para el estudiante.

5.3.5.4 Evaluación del estado del videojuego

El juego cumple con los requisitos propuestos en el sprint. El personaje corre y salta, la trivia, el fondo se mueve con el pasar del tiempo, el menú se cambió para que sea más atractivo y animado. Sin embargo, durante el desarrollo del sprint se observó que podían añadirse más cosas que podrían mejorar la apariencia y la jugabilidad. Se planea realizar dichos cambios como trabajo extra durante los siguientes Sprints.

5.3.5.5 Resultados

El sprint se cumplió en su totalidad a pesar de que pueden existir mejoras que se incluirán posteriormente se mejoró notablemente el aspecto grafico del juego. A continuación, podemos ver cómo se creó diferentes Sprints para suplantar los antiguos (figura 41)

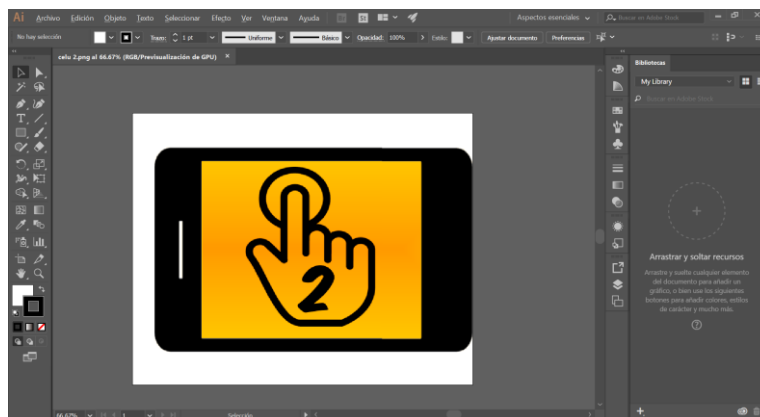


Figura 41. Resultados Sprint 4. Material Gráfico.

A su vez se descargó de sitios gratuitos sonidos y música que se agregó fácilmente al juego debido a que Unity permite incorporar audio a los objetos e inicializarlos en el código de manera natural y poco compleja. (figura 42).



Figura 42. Resultados Sprint 4. Música del videojuego

Si se observa la figura 43 podemos ver cómo se agregó la animación al fondo del juego, el cual mediante un objeto 3D llamado Quad permite repetir una imagen que mediante programación podamos definir una velocidad. De esta forma se utilizó una imagen creada en la historia de usuario 12 que tenga inicio y final similar para que se pueda dar el efecto de movimiento constante.

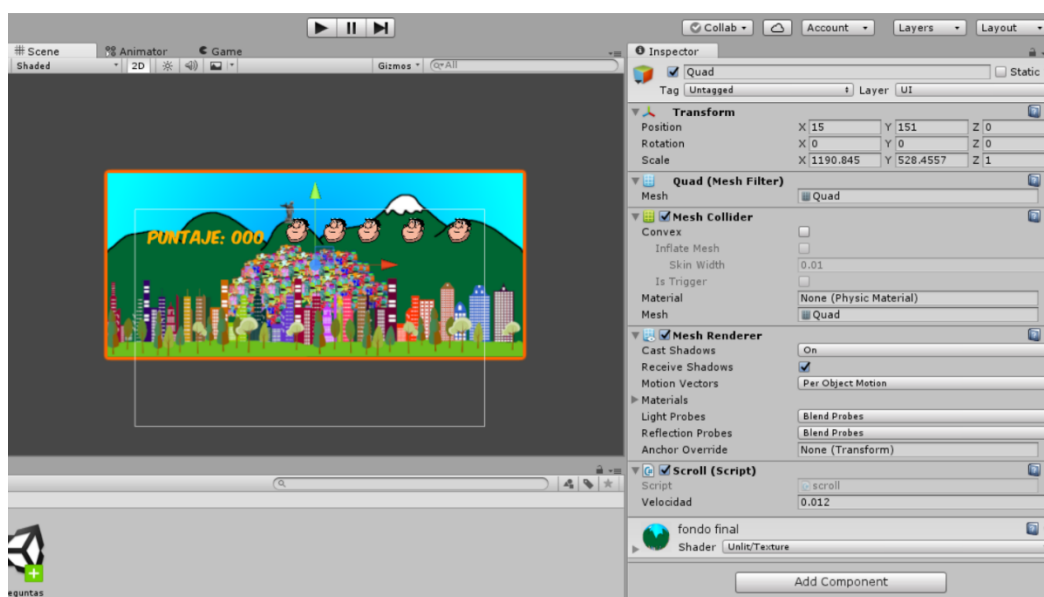


Figura 43. Resultados Sprint 4. Fondo

La animación del personaje resultó más difícil ya que se crearon varias partes que sean capaces de moverse independientemente y dar la sensación de que el personaje este saltando y que este corriendo mediante código. Figura (44)

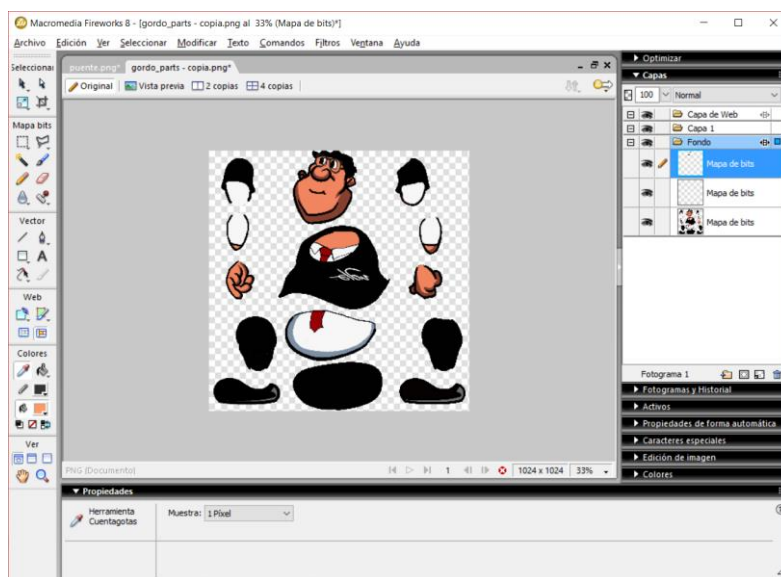


Figura 44. Resultados Sprint 4. Animación y diseño del personaje

El menú y los botones fueron cambiados se usó fuentes gratis de botones para mantener una similitud en todo el juego. (Figura 45)



Figura 45. Resultados Sprint 4. Menú del videojuego

Se puede ver en la figura 46 como el Burndown no termina en 0 ya que la última historia de usuario no se terminó y se incluyó como parte del siguiente sprint.

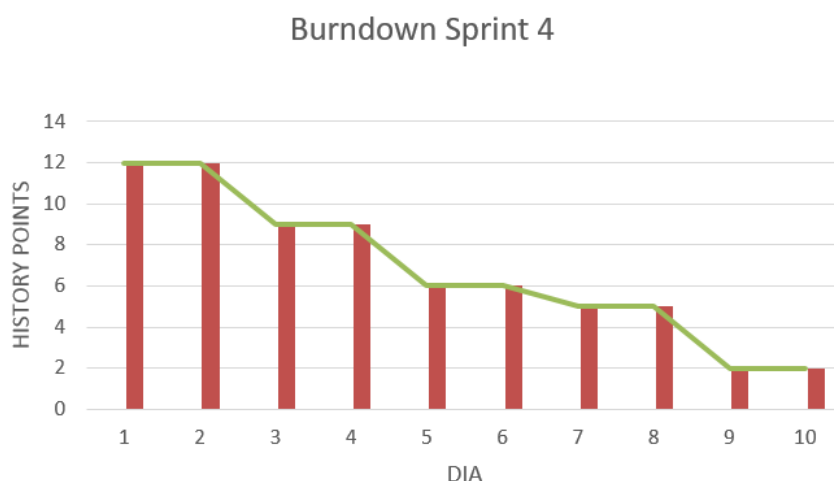


Figura 46. Resultados Sprint 4. Burndown Sprint 4

Se actualizó el tablero Kanban para mover las tareas que ya se ejecutaron a la sección de finalizadas. En este caso en la figura 47 vemos como la última historia de usuario permanece en la sección de en ejecución (Doing).

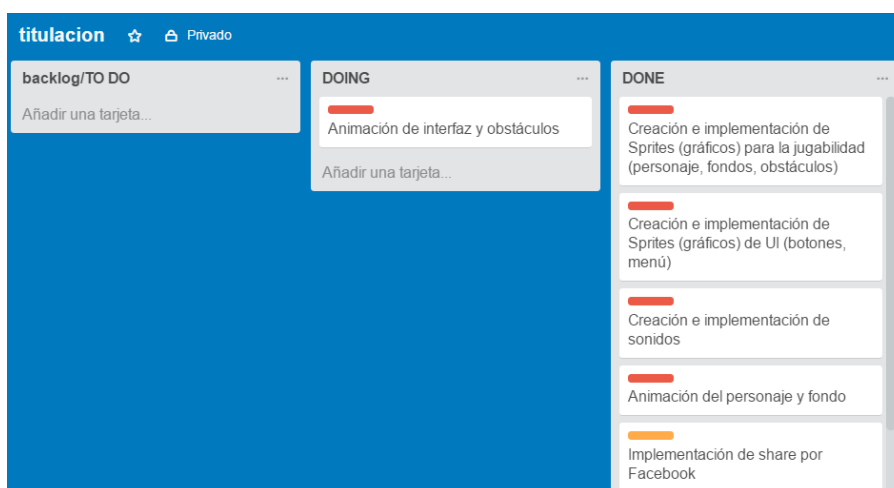


Figura 47. Actualización del cuadro Kanban tras sprint 4.

Se presenta la tabla de historias de usuario para ver los puntos ganados una vez concluidas las dos semanas de este sprint (tabla 30).

Tabla 30.
Sprint 4 Backlog

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados

12	Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)	4	3	3
13	Creación e implementación de Sprites (gráficos) de UI (botones, menú)	4	3	3
14	Creación e implementación de sonidos	4	1	1
15	Animación del personaje, obstáculos y fondo	4	3	3
16	Animación de interfaz	4-5	3	1

5.3.5.6 Retrospectiva

- **Qué cosas han funcionado bien.**

A pesar de que la mayoría de gráficos se parte de una base existente fue de gran ayuda usar una tableta digitalizadora con lápiz ya que así se puede dibujar de manera más natural.

- **Qué se puede mejorar.**

Se puede agregar una animación que muestre el tiempo restante por cada pregunta.

- **Qué se ha aprendido.**

Se aprendió a utilizar básicamente los programas de diseño como Fireworks o Adobe Illustrator, la creación de capas, e iluminación.

- **Cuáles son los problemas que podrían impedir progresar adecuadamente.**

La complejidad de los gráficos aumenta el tamaño en megas del juego, entre mejor definición y entre más imágenes, mayor el tamaño.

5.3.6 Sprint 5

El sprint cinco trata acerca de la finalización del proyecto para pasar a la etapa de beta. Básicamente se termina las tareas del sprint anterior y se prepara el APK (instalador de la aplicación para Android) de la aplicación y se maneja el release en la tienda de Google.

5.3.6.1 Sprint Backlog

Este sprint es el más corto, como la metodología puede adaptarse a las necesidades de cada negocio, en este proyecto se planeó dejar un sprint más corto para poder atender solución de bugs o mejora de las funcionalidades y del aspecto gráfico. El sprint está compuesto de 3 historias de usuario (tabla 31).

Tabla 31.
Sprint Backlog 5

Núm.	Historia de usuario	Sprint	Esfuerzo	Puntos Ganados
16	Animación de interfaz (botones, menú y obstáculos)	4-5	3	1
17	Manejo de reléase en tienda	5	5	0
18	Publicación para beta	5	3	0

Se actualizó el tablero de Kanban en Trello con las nuevas tareas, se puede ver que quedo una tarea pendiente del anterior sprint. (figura 48).

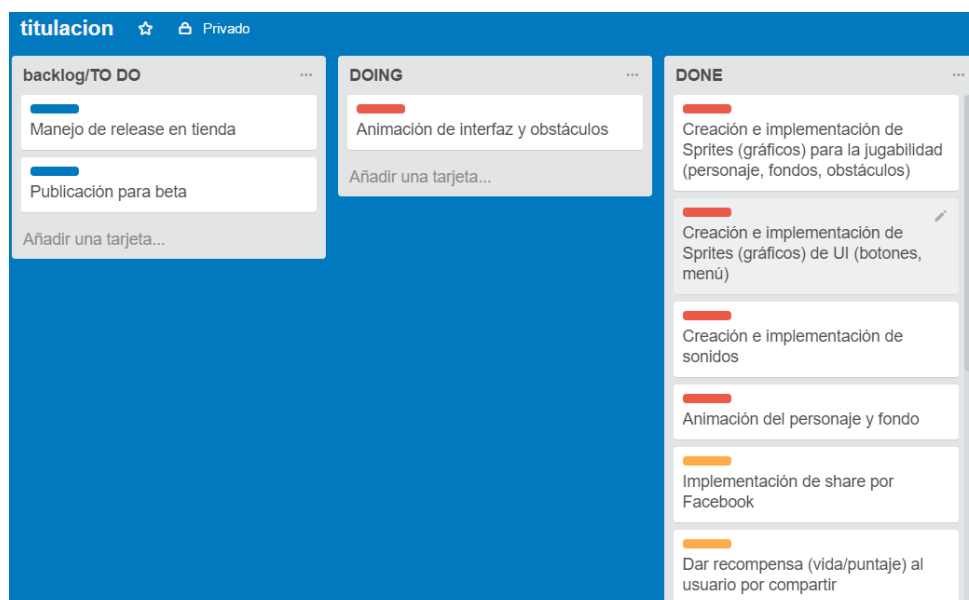


Figura 48. Actualización del tablero de historias del sprint 5.

5.3.6.2 Historias de usuario

La historia de usuario 16 se encuentra en el sprint anterior (ver Sprint 4)

Tabla 32.

Historia de Usuario 17: Manejo de reléase en tienda

Número	17	Usuario	Jugador
Nombre de la Historia	Manejo de reléase en tienda		
Prioridad	Alta	Riesgo	Bajo
Puntos estimados	5	Iteración	5
Descripción	Como cliente quiero que el juego esté disponible en la tienda de Android y que se pueda ir lanzando actualizaciones para poder realizar cualquier cambio o corrección.		
Validación	<ol style="list-style-type: none"> 1. Existe una página de la aplicación en Google donde se puede manejar toda su información. 2. Existe una carpeta donde se localicen fácilmente los aplicativos de la aplicación APK para poder subirlos a la tienda. 		

Tabla 33.

Historia de Usuario 18: Publicación para beta

Número	18	Usuario	Cliente
Nombre de la Historia	Publicación para beta		
Prioridad	Alta	Riesgo	Bajo
Puntos estimados	3	Iteración	5
Descripción	Como cliente quiero que la gente pueda acceder al juego para que lo pruebe y poder detectar errores u obtener recomendaciones.		
Validación	La aplicación debe poder descargarse en la tienda de Google.		

5.3.5.3 Riesgos

El riesgo en esta etapa es mínimo ya que el juego se encuentra en la parte final de su desarrollo y en este sprint se prepara lo necesario para que el juego pueda entrar a la fase de beta según SUM.

5.3.5.4 Evaluación del estado del videojuego

Se procedió a la publicación de una beta que exponga lo realizado durante los cuatro Sprints anteriores. Durante este periodo se siguieron encontrando errores so mejoras que han sido corregidas a la par del lanzamiento a la tienda para la beta.

5.3.6.5 Resultados

Este sprint abarcaba una historia de usuario del sprint anterior, aquí se terminó de animar y diseñar la interfaz, lo botones y los obstaculos, se puede apreciar en la figura 49 que se agregó más dibujos y animaciones al menú, además de que los botones tienen un efecto si se presionan



Figura 49. Resultados Sprint 5. Animación de interfaz

Se incluyeron nuevos obstaculo y la animacion de los mismos, así como de los antiguos. Este proceso se pudo realizar facilmente usando el animador de Unity que permite por ejemplo grabar la posición y el movimiento de un objeto y reproducirlo cuando es instanciado.

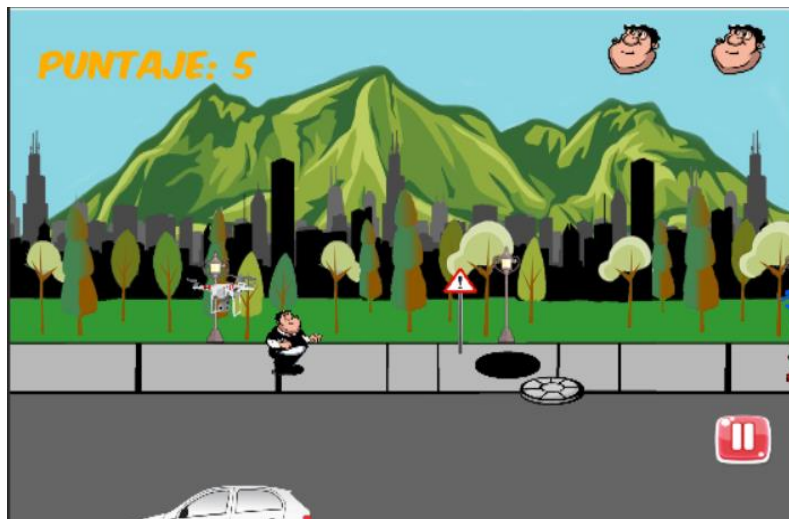


Figura 50. Resultados Sprint 5. Animación de obstáculos

Para el manejo del release o lanzamiento en la tienda se permite lanzar versiones alfa y beta, en este caso se lanzó tanto una versión en producción y la beta. Esto para lograr que se pueda acceder a la aplicación fácilmente (la beta solo permite invitados por correo electrónico, la versión de producción aparece en la tienda para todo el público) (figura 51).

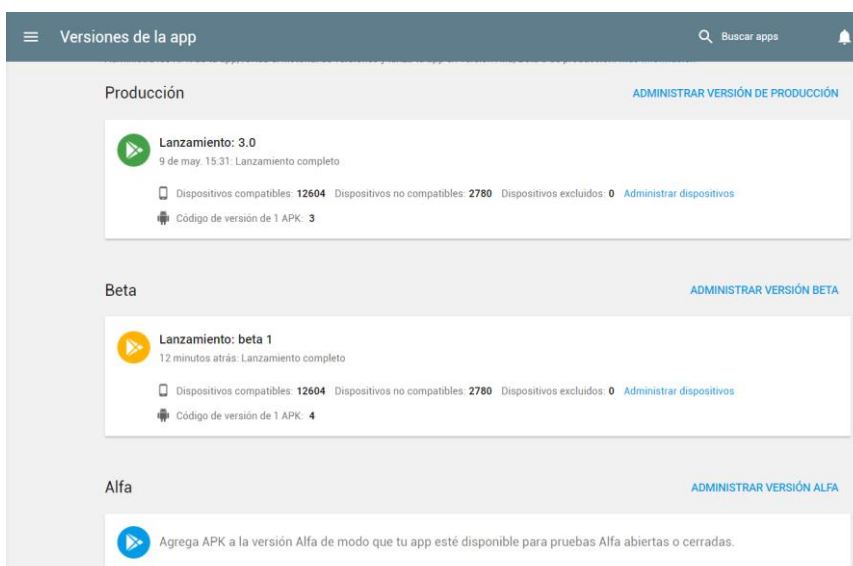


Figura 51. Resultados Sprint 5. Manejo del release (beta).

Al momento de manejar un reléase en la tienda como se trata de un videojuego Google dispone que este tiene que obtener una calificación de contenido que permita saber qué clase de contenido ofrece el mismo. (Figura 52.)



Figura 52. Resultados Sprint 5. Manejo del release (calificación juego).

La beta quedo abierta para el público en general si accede a la tienda y busca la aplicación y también es posible enviar la misma a testers a través de una lista de correo electrónico (figura 53)

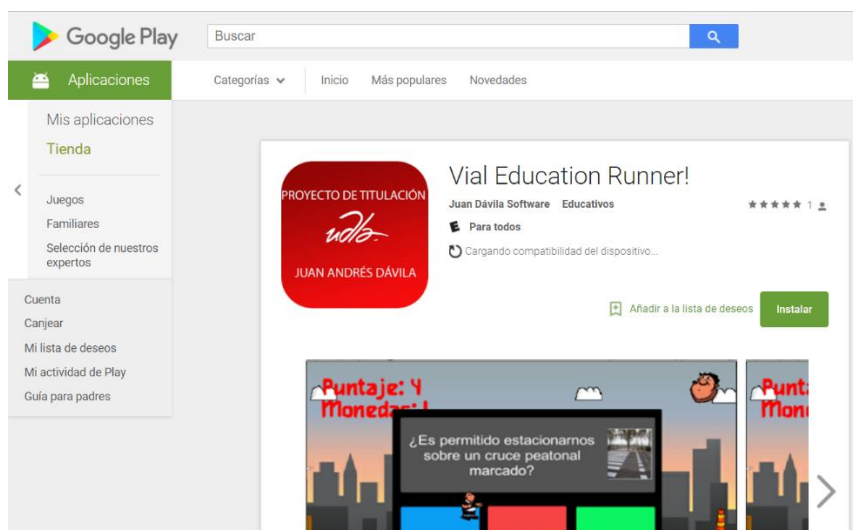


Figura 53. Resultados Sprint 5. Publicación de beta

Este sprint era el que menor esfuerzo representaba y por eso las historias se pudieron cumplir inclusive antes de lo esperado, tiempo el cual se aprovechó para mejorar el acabado y jugabilidad de la aplicación. (figura 54)

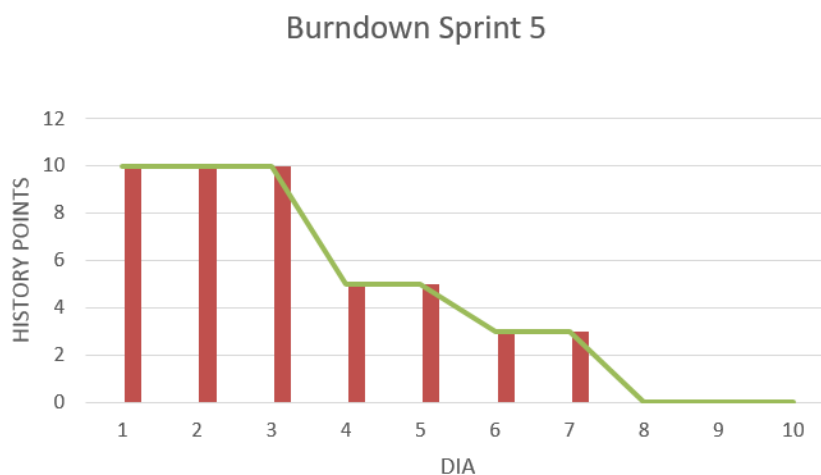


Figura 54. Resultados Sprint 5. Burndown quinta iteración

Se actualizó el tablero y se dan por terminadas todas las tareas de los Sprints para proceder a una fase de beta donde se corrijan errores que son comunes en este tipo de software. (figura 55)

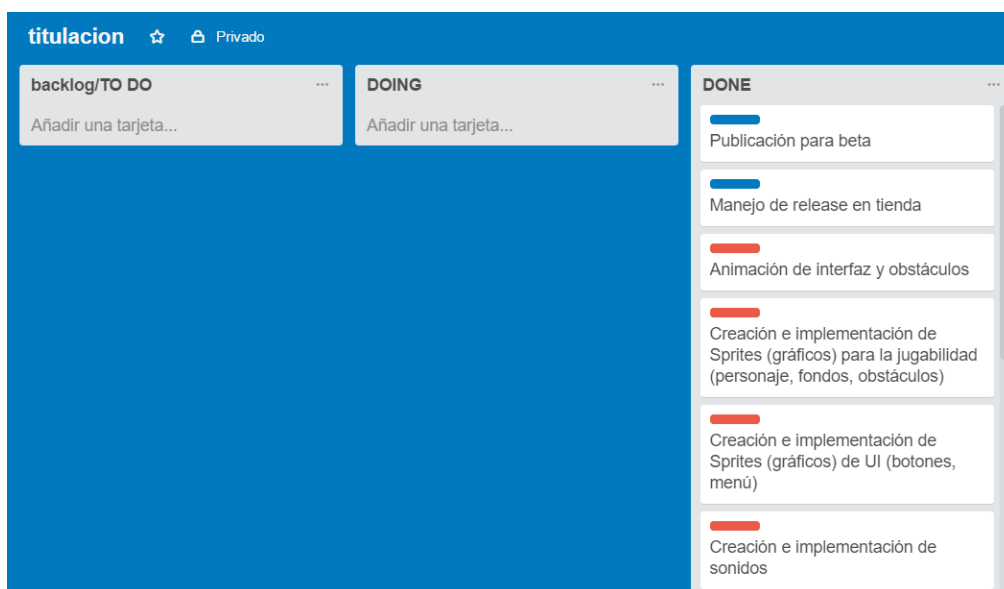


Figura 55. Resultados Sprint 5. Actualización del cuadro Kanban tras sprint 5.

5.3.6.6 Retrospectiva

- **Qué cosas han funcionado bien.**

El utilizar el libre acceso a la aplicación y no el uso de una beta a través de correos electrónicos, permite mejor acceso a la retroalimentación de un proyecto de esta índole.

- **Qué se puede mejorar.**

Se puede todavía agregar mejoras y correcciones a futuro simplemente se puede actualizar el aplicativo subiendo un nuevo APK a la tienda.

- **Qué se ha aprendido.**

El proceso de lanzamiento para una aplicación de Android a través de la tienda de Google.

- **Cuáles son los problemas que podrían impedir progresar adecuadamente.**

En este caso ninguno, simplemente se necesita una versión del aplicativo funcional y una cuenta de desarrollador.

5.4 Beta

5.4.1 Verificación del videojuego

Para la verificación del videojuego se procedió a realizar los casos de prueba de cada historia de usuario para verificar que el desarrollo se haya realizado sin ningún problema.

5.4.1.1 Casos de prueba sprint 1

Tabla 34.

Caso de prueba 1

ID:1	Historia a probar: Salto y desplazamiento del jugador	Fecha: 09/ 05/17
Pre Requisitos:	*Haber iniciado la aplicación *El personaje tiene que estar representado de alguna manera visible (un dibujo) y no necesariamente estar animado	
Pasos o secuencia lógica:	1. Presionar la pantalla una vez para saltar 2. Presionar la pantalla de nuevo para saltar de nuevo	

Resultados esperados:	<ol style="list-style-type: none"> 1. El personaje salta máximo dos veces cuando el jugador presiona la pantalla 2. El personaje se mueve hacia la derecha constantemente mientras la cámara lo sigue.
Resultados obtenidos:	La prueba se realiza de manera exitosa ya que solo se puede saltar dos veces, y solo después de haber topado el suelo (Figura 56)
Observaciones:	Se incluyó el manejo de velocidad del usuario



Figura 56. Caso de prueba 1.

Tabla 35.

Caso de prueba 2

ID:2	Historia a probar: Generación de obstáculos y escenario	Fecha: 09/ 05/17
Pre Requisitos:	<ul style="list-style-type: none"> *Haber iniciado la aplicación *Tener al menos una imagen o Sprite para el personaje. *Existe una imagen o Sprite que representa el suelo y los obstáculos 	
Pasos o secuencia lógica:	1. El personaje inicia la aplicación, el resto se genera automáticamente, tanto los personajes como los obstáculos	
Resultados esperados:	<ol style="list-style-type: none"> 3. Se generan un suelo donde el personaje puede desplazarse constantemente a hacia la derecha. 4. Se genera obstáculos en el escenario conforme avanza el tiempo. 	
Resultados obtenidos:	La prueba se realiza de manera exitosa, el escenario donde el personaje camina automáticamente se genera con el paso de los segundos mientras el jugador trata de evadir los obstáculos que aparecen de forma aleatoria. (Figura 57)	

Observaciones: Se necesitan crear varios bloques que aparecerán en el escenario como obstáculos.



Figura 57. Caso de prueba 2

Tabla 36.

Caso de prueba 3

ID:3	Historia a probar: Aumentar puntaje	Fecha: 09/ 05/17
Pre Requisitos:	*Haber iniciado la aplicación *Jugar al menos durante 3 segundos	
Pasos o secuencia lógica:	1. Con el desplazamiento constante del personaje, el puntaje que se muestra en pantalla aumentará.	
Resultados esperados:	1. Existe un texto en la parte superior de la pantalla que indica el puntaje del jugador, el mismo que aumenta de 1 en 1 conforme pasan los segundos.	
Resultados obtenidos:	Mientras el jugador se mantenga corriendo el puntaje aumenta de forma uniforme. (Figura 58)	
Observaciones: Ninguna		

Tabla 37.

Caso de prueba 4

ID:4	Historia a probar: Muerte del personaje, disminución de vidas y retención del puntaje más alto	Fecha: 09/ 05/17
-------------	---	-------------------------

Pre Requisitos:	*Existen obstáculos con los que el personaje puede chocar. *Disponer de puntos,
Pasos o secuencia lógica:	1. El personaje inicia una nueva partida 2. Conforme pasa el tiempo el puntaje aumenta. 3. Aparecen obstáculos en el escenario que si el personaje los toca pierda una vida (se definen 3 vidas de inicio)
Resultados esperados:	3. Una vez perdidas todas las vidas se para el juego finalizando la partida. 4. Se muestra un menú con el puntaje máximo obtenido en todas las partidas jugadas.
Resultados obtenidos:	Todos los resultados se dieron como se esperaba, el personaje muere cuando se queda sin vidas y se acaba la partida, y se registra una variable estática que guarda la mayor puntuación obtenida hasta el momento. (Figura 58)
Observaciones:	Se debe ofrecer al jugador la oportunidad de volver a jugar en el menú que aparece al finalizar la partida.



Figura 58. Caso de prueba 3 y 4

5.4.1.1 Casos de prueba sprint 2

Tabla 38.

Caso de prueba 5

ID:5	Historia a probar: Creación de menú	Fecha: 15/ 05/17
Pre Requisitos:	*Debe cumplirse con las historias de usuarios anteriores.	
Pasos o secuencia lógica:	<ol style="list-style-type: none"> 1. El jugador inicia la aplicación 2. El menú aparece después del logotipo de la universidad. 3. Si el personaje muere aparece la opción de volver al menú 	
Resultados esperados:	<ol style="list-style-type: none"> 1. Aparece un menú con la opción de salir, jugar, compartir en Facebook y compartir en Twitter. 2. Aparece un menú después de morir que permite volver a jugar y volver al menú principal 	
Resultados obtenidos:	Existe un menú con los botones especificados. (Figura 58)	
Observaciones: Ninguna		

Tabla 39.

Caso de prueba 6

ID:6	Historia a probar: Despliegue de preguntas al existir una colisión	Fecha: 15/ 05/17
Pre Requisitos:	*Existen obstáculos con los que el personaje puede chocar para iniciar la trivía. *Debe existir una lista con preguntas del tema. *El desarrollador puede fácilmente agregar preguntas	
Pasos o secuencia lógica:	<ol style="list-style-type: none"> 1. El personaje inicia una nueva partida 2. Conforme pasa el tiempo aparece un obstáculo que indica que se puede activar la trivía. 	
Resultados esperados:	<ol style="list-style-type: none"> 1. Al topar el obstáculo aparece un recuadro en pantalla con una pregunta, una imagen, un contador y tres opciones. 2. El contador indica que se tiene 10 segundos para responder 	

	3. Si aparece de nuevo la trivía, la pregunta debe ser otra, nunca se pueden repetir.
Resultados obtenidos:	Todos los resultados se dieron como se esperaba, el personaje colisiona con un rectángulo que tiene un dibujo de signo de interrogación y aparece un rectángulo con las preguntas. (Figura 59)
Observaciones: Se debe asegurar que el jugador no colisione varias veces con el bloque.	

Tabla 40.
Caso de prueba 7

ID:7	Historia a probar: Agregar vida/puntaje tras responder trivía	Fecha: 15/ 05/17
Pre Requisitos:	*El personaje activo la trivía * Se seleccionó una opción de respuesta.	
Pasos o secuencia lógica:	1. El personaje activo la trivía y apareció un panel con tres opciones.	
Resultados esperados:	3. Si se eligió la opción incorrecta se desactiva el panel y el juego continua. 4. Si se eligió la respuesta correcta se continua con el juego, pero se otorga una vida extra o 10 puntos 5. Si pasan los 10 segundos se desactiva el panel y se continua con el juego	
Resultados obtenidos:	Todos los resultados se dieron como se esperaba, se otorga vida y puntaje de acuerdo a lo esperado. Se incluyeron sonidos para indicar al jugador si su respuesta fue la correcta. (Figura 59)	
Observaciones: Ninguna		

Como se puede observar en la figura 59 existe un panel de trivía que aparece tras colisionar y que desaparece después de diez segundos. Al contestar correctamente se agrega una vida

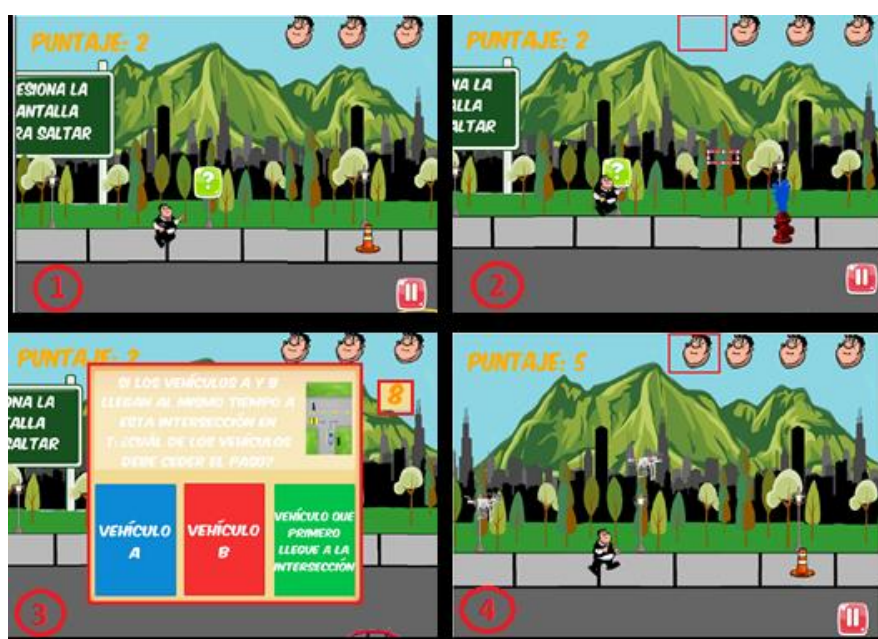


Figura 59. Caso de prueba 5-7

1. Aparece un objeto que activa la trivia.
2. El jugador colisiona con el objeto.
3. La trivia aparece con una pregunta y tres opciones.
4. Se recompensa al jugador con una vida si respondió correctamente

5.4.1.1 Casos de prueba sprint 3

Tabla 41.

Caso de prueba 8

ID:8	Historia a probar: Implementación de share por Facebook	Fecha: 15/ 05/17
Pre Requisitos:	*el usuario tiene una cuenta en Facebook	
Pasos o secuencia lógica:	<ol style="list-style-type: none"> 1. El personaje ingresa al juego 2. Hace clic en el botón de Facebook. 3. Acepta condiciones 	
Resultados esperados:	<ol style="list-style-type: none"> 4. Se habilitan dos botones nuevos, uno para compartir y otro para invitar. 5. Si el usuario comparte el juego aparecer en su muro de Facebook. 	

	6. Si el usuario invita a sus amigos estos reciben una invitación con el link de la aplicación para descárgala en su Android.
Resultados obtenidos:	Se obtuvo los siguientes resultados: 1. El usuario invita sus amigos y comparte en su muro correctamente 2. Cuando inicia sesión aparecen los botones respectivos y se desactiva el de iniciar sesión (Figura 60)
Observaciones: Se debe observar si es normal que no aparezca una petición de Facebook para acceder a los datos.	



Figura 60. Caso de Prueba 8

1. Si se inicia sesión el usuario verá los botones de invitar y compartir.
2. En caso de elegir compartir, el usuario puede escribir un mensaje.
3. En el caso de invitar el usuario elige a sus contactos.

Tabla 42.

Caso de prueba 9

ID:9	Historia a probar: Implementación de share por Twitter	Fecha: 15/ 05/17
Pre Requisitos:	*el usuario tiene una cuenta en Twitter	
Pasos o secuencia lógica:	<ol style="list-style-type: none"> 1. El personaje ingresa al juego 2. Hace clic en el botón de Twitter. 	
Resultados esperados:	<ol style="list-style-type: none"> 1. Se abre una ventana con texto predeterminado indicando el puntaje obtenido y el enlace a la tienda para descargar la app. 	

Resultados obtenidos:	Todos los resultados se dieron como se esperaba al hacer clic en el botón aparece una ventana con el puntaje más alto obtenido (Figura 61)
Observaciones: Ninguna	

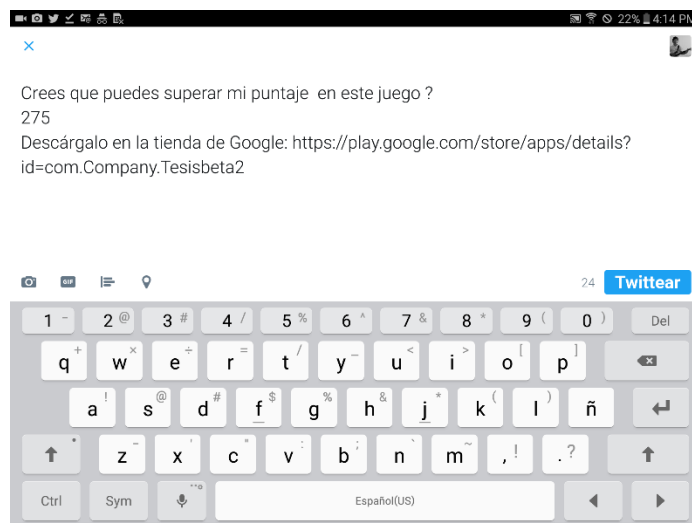


Figura 61. Caso de prueba 9

Tabla 43.

Caso de prueba 10

ID:10	Historia a probar: Dar recompensa (vida/puntaje) al usuario por compartir	Fecha: 15/05/17
Pre Requisitos:	*el jugador compartió el juego o invitó contactos a descargar la app	
Pasos o secuencia lógica:	1. El jugador hace clic en el botón de Facebook/Twitter y comparte o invita el juego.	
Resultados esperados:	3. Una vez compartido el juego el usuario inicia una partida nueva con una vida más si compartió por Facebook, Twitter o si invito amigos. 4. Si hizo las tres acciones se le agregara 10 puntos y 2 vidas.	
Resultados obtenidos:	Se inicia la partida con una o dos vidas y hasta con 10 puntos extra. (Figura 62)	
Observaciones: Ninguna		



Figura 62. Caso de prueba 10

5.4.1.1 Casos de prueba sprint 4

Tabla 44.

Caso de prueba 11

ID:11	Historia a probar: Creación e implementación de Sprites (gráficos) para la jugabilidad (personaje, fondos, obstáculos)	Fecha: 09/05/17
Pre Requisitos:	*Existen Sprites donde se puedan colocar las nuevas imágenes.	
Pasos o secuencia lógica:	1. Iniciar el juego	
Resultados esperados:	1. Se crean y dibujan nuevos diseños para el personaje el fondo y los obstáculos, se mantiene una estética 2D simplista y en forma de cartoon.	
Resultados obtenidos:	Se cambiaron los Sprints y el juego mejoro indiscutiblemente su presentación.	
Observaciones: Se debe corregir errores de coloración en los archivos PNG de fondo y de los obstáculos (algunos tienen detalles sin pintar)		

Tabla 45. Caso de prueba 12

ID:12	Historia a probar: Creación e implementación de Sprites (gráficos) de UI (botones, menú)	Fecha: 09/05/17
--------------	---	------------------------

Pre Requisitos:	*existe un menú para el juego que contenga varios botones
Pasos o secuencia lógica:	1. Iniciar el juego
Resultados esperados:	1. Los botones mantienen un diseño uniforme en todos los menús, se presentan colores claros y vivos que reflejen la visión del juego. 2. El menú tiene un fondo nuevo.
Resultados obtenidos:	Se han cambiado los botones de manera correcta manteniendo la uniformidad. A su vez el menú presenta una imagen de fondo similar a la que se muestra en los paneles de trivia o de pausa. (Figura 63)
Observaciones: Ninguna	

Tabla 46.

Caso de prueba 13

ID:13	Historia a probar: Creación e implementación de sonidos	Fecha: 09/ 05/17
Pre Requisitos:	*Jugabilidad básica: existe un personaje que salta y se desplaza y aparece obstáculos con los que el personaje puede chocar.	
Pasos o secuencia lógica:	1. Iniciar el juego	
Resultados esperados:	6. El personaje comienza a desplazarse y cuando salta se reproduce un sonido. 7. Se reproduce un sonido tras una colisión. 8. Se reproduce el sonido de reloj cuando se abre la trivia para alertar del paso del tiempo 9. Existe música de fondo 10. Existe musca de fondo en el menú.	
Resultados obtenidos:	Se reproducen los sonidos correctamente tanto en el menú en el juego, al saltar, al abrir la trivia y al colisionar.	
Observaciones: Se debe mejorar el sonido de colisión.		

Tabla 47.
Caso de prueba 14

ID:14	Historia a probar: Animación del personaje, obstáculos y fondo	Fecha: 09/ 05/17
Pre Requisitos:	<ul style="list-style-type: none"> *Existe un personaje *Existe un fondo *Existen obstáculos con los que el personaje puede chocar. 	
Pasos o secuencia lógica:	1. Iniciar el juego	
Resultados esperados:	<ul style="list-style-type: none"> 3. El personaje tiene la animación de correr y saltar. 4. Los obstáculos se mueven (no necesariamente todos) 5. EL fondo se mueve lateralmente con el tiempo. 	
Resultados obtenidos:	El personaje y los obstáculos se mueven e interactúan, pero el fondo del escenario no responde de acuerdo a lo esperado (Figura 63)	
Observaciones: Ninguna		

Tabla 48.
Caso de prueba 15

ID:15	Historia a probar: Animación de interfaz	Fecha: 09/ 05/17
Pre Requisitos:	Existe un menú de inicio	
Pasos o secuencia lógica:	1. Iniciar el juego	
Resultados esperados:	<ul style="list-style-type: none"> 1. Se animó la interfaz del menú existen gráficos que se mueven. 2. Los botones tienen el efecto de ser presionados 	
Resultados obtenidos:	La interfaz se mejoró de acuerdo a lo predicho. (Figura 63)	
Observaciones: Tener en cuenta de la resolución de la pantalla y como se los objetos animados del menú se posicionan.		

Se puede apreciar en la figura 61 el cambio a la apariencia general del videojuego. Animaciones, cambios de Sprites.



Figura 63.Caso de prueba 12-16

1. Se puede observar la mejora del menú.
2. Los Sprites están animados.
3. Se incluyeron nuevos obstáculos y se mejoró su diseño.
4. Se añadió animación a los obstáculos.

5.4.2 Corrección del videojuego

Tras la liberación del Beta es cuando se procede a escuchar la opinión y el reporte de errores de los testers. Para esto se realizó pequeñas preguntas a los jugadores sobre como encontraron el videojuego, que no les gusto, que les pareció divertido y que creen que podría agregarse. Las siguientes correcciones fueron tomadas en cuenta para incluirse en el proyecto:

- Incluir un “Power UP”. Esto quiere decir incluir una forma de que el personaje gane puntos más rápido o sea invulnerable por cierto tiempo.
- Mejorar la velocidad del personaje, a veces es muy fácil el juego debido a que el personaje puede evadir los obstáculos fácilmente.

Otra herramienta importante para la corrección del videojuego son los comentarios o reseñas que hacen los usuarios a la aplicación a través de la tienda de Google, el desarrollador puede ver estos mensajes y obtener una valiosa retroalimentación. (Figura 64).

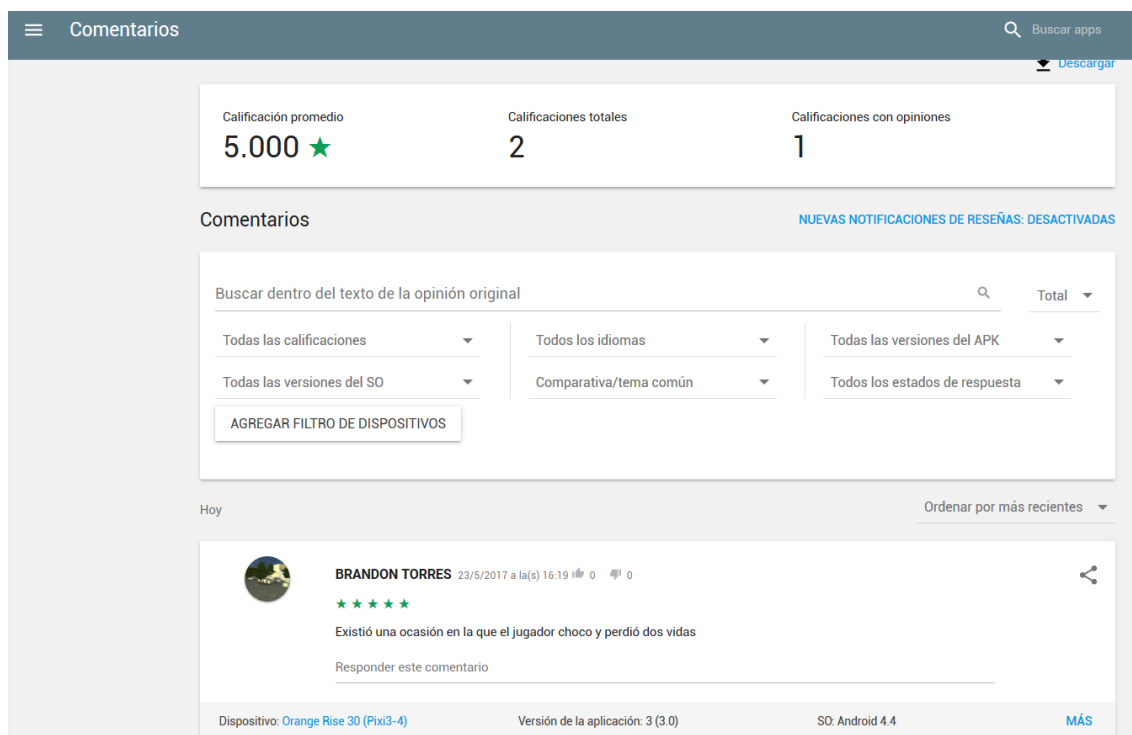


Figura 64. Comentarios para la corrección del juego

5.5 Cierre

5.5.1. Liberación

El proceso de liberación del videojuego es un paso relativamente simple ya que después del proceso de beta el juego puede ser liberado oficialmente al público. En este proyecto esto se realizó fácilmente gracias a la tienda de Google que permite promover la versión de beta al lanzamiento o subir un aplicativo nuevo.

En este caso la aplicación se encuentra disponible a través del siguiente enlace: <https://play.google.com/store/apps/details?id=com.Company.Tesisbeta2&hl=es> (figura 65)

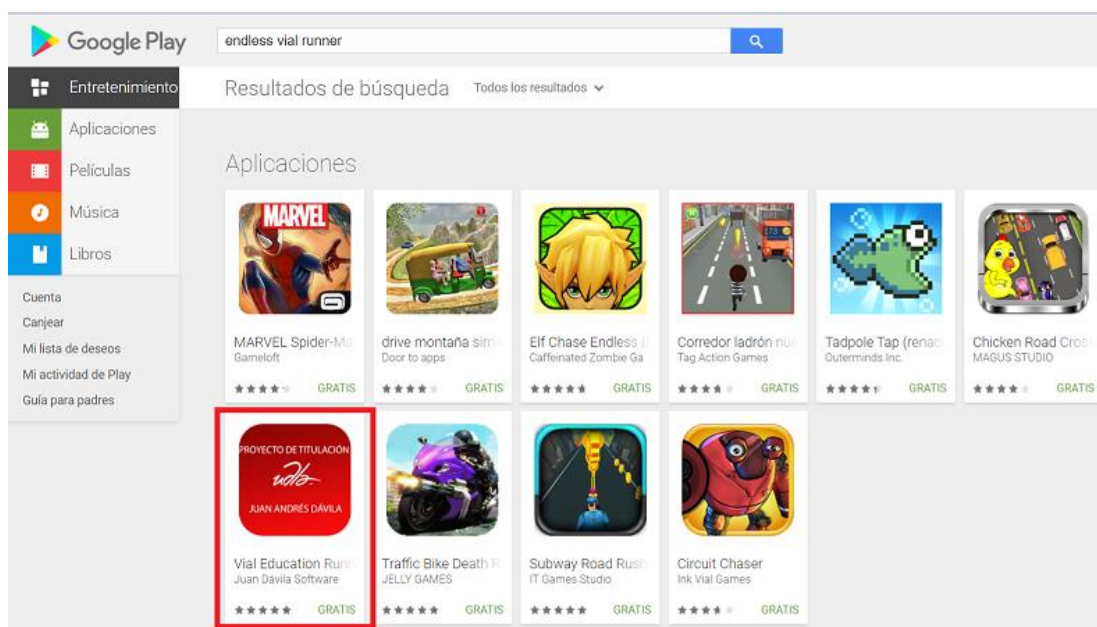


Figura 65. Publicación de la aplicación

Cabe recalcar que como se trata de un proyecto de software lo más probable es que esta sea la versión inicial y que ante la aparición de errores o si se desea mejorar el juego se puede lanzar nuevas versiones del mismo. Igualmente, gracias a la consola de versionamiento que ofrece Google, este proceso es sencillo ya que solo hay que limitarse a subir un aplicativo nuevo y describir los cambios realizados.

6. CONCLUSIONES Y RECOMENDACIONES

6.1 Conclusiones

- La gran cantidad de usuarios que utilizan Android y el bajo costo de las licencias, hacen de esta plataforma la ideal para lanzar videojuegos en la actualidad, por ende, es importante crear contenido para poder concientizar y educar a la gente. En este ámbito se concluye que para la educación vial es importante manejar temas básicos y de sentido común que son los que más pueden evitar siniestros.
- Como se pudo observar al realizar este proyecto, la concepción de material gráfico vistoso y simple que ayude al jugador a entender lo que pasa en pantalla es muy importante. El videojuego puede ser divertido, pero si no tiene una buena presentación, éste no tendrá acogida entre el público.
- Unity facilitó la implementación del módulo de trivia de educación vial que incluye bonificaciones de vidas y puntaje al jugador, haciendo que el juego integre mecánicas educativas e intuitivas para el usuario además de motivar a que siga jugando y aprendiendo para romper su propio récord.
- La integración de redes sociales en los videojuegos es relativamente simple y aporta mucho a la difusión de una aplicación. En este caso se logró que el usuario pueda compartir su puntaje e invitar a sus contactos en redes sociales a descargar la app. Gracias a que hoy en día los usuarios pasan gran parte de su tiempo en estas redes, una idea será mejor difundida por estos medios.

6.2 Recomendaciones

- A futuro se podría añadir más preguntas acerca de seguridad vial con mayor dificultad pero que a su vez otorguen más beneficios al jugador, de esta forma se puede aumentar la variedad en la trivia. A parte de esto es

recomendable ir incluyendo mecánicas nuevas a través de actualizaciones para que el jugador tenga un motivo para volver a jugar el juego.

- Si se plantea lanzar un videojuego al mercado de dispositivos móviles es importante que este se promueva a través de publicidad o redes sociales. A pesar de que no estar incluido dentro del alcance del proyecto, la promoción y la publicidad son beneficiosos para la difusión de un videojuego. Si se dispone de recursos, la publicidad pagada en medios como Facebook o YouTube es altamente recomendable.
- Al querer lanzar un videojuego en Android se recomienda realizar la etapa de beta con la mayor cantidad de público posible. Muchas veces se requieren varias pruebas de un juego para detectar un error o bug, por lo que es recomendable recibir consejos o retroalimentación de varias fuentes.

REFERENCIAS

- Adobe Systems Incorporated. (2017). Descubre la experiencia de Creative Cloud. Recuperado el 02 de Abril de 2017 de https://creative.adobe.com/es/plans?single_app=photoshop&promoid=5JRH8K43&mv=other&store_code=mx
- Abad, D. (2016). ¿Qué sistemas operativos prefieren los usuarios en América Latina? Recuperado el 08 de <https://www.latamclick.com/que-sistemas-operativos-prefieren-los-usuarios/>
- Adams, E. (2014). *Fundamentals of Game Design*. Pearson Education. Recuperado el 30 de marzo de 2017 de https://books.google.com.ec/books?id=Lm1jAgAAQBAJ&pg=PA45&dq=video+game+design+process&hl=es-419&sa=X&ved=0ahUKEwiP_7qmvf_SAhWIZCYKHT89BJwQ6AEII DAB#v=onepage&q=video%20game%20design%20process&f=false
- Banco Galicia. (2014). ¿Qué es un modelo de negocios freemium? Recuperado el 30 de marzo de 2017 de <http://www.buenosnegocios.com/notas/768-que-es-un-modelo-negocios-freemium>
- Calle, J. D. (2012). S.A.B.I.A. Sistemas Adaptativos y Bioinspirados en Inteligencia Artificial. Recuperado el 21 de marzo de 2017 de <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/ProgramacionVideoJuegos/Unity3D/introunity.html>
- Castelán, Y. G. (2013). Multimedia en la educación, una necesidad. Recuperado el 30 de junio de 2017 de <https://www.uaeh.edu.mx/scige/boletin/prepa4/n1/e6.html>
- Diario La Hora. (2011). 546 millones de dólares se pierden por accidentes de tránsito. Recuperado el 02 de abril de 2017 de [https://lahora.com.ec/noticia/648739/546-millones-de-dc3b3lares-se-pierden-por-accidentes-de-trc3a1nsito](https://lahora.com.ec/noticia/648739/546-millones-de-dolares-se-pierden-por-accidentes-de-trc3a1nsito)

- Definicion.De. (2015). Definición de Trivia. Recuperado el 22 de Mayo de 2017 de <http://definicion.de/trivia/>
- Estudio Wam. (2017). Desarrollo De Aplicaciones Móviles. Recuperado el 30 de marzo de 2017 de <http://estudiowam.com/desarrollo-de-aplicaciones-moviles/>
- Gallego, M. (2014). La priorización de historias de usuario (versión reducida). Recuperado el 10 de Julio de <https://es.slideshare.net/micaelgallego/la-priorizacin-de-historias-de-usuario-versin-reducida>
- GamerDic. (2017). *Endless runner*. Recuperado el 26 de marzo de 2017 de <http://www.gamerdic.es/termino/endless-runner>
- Gartner. (2016). *Gartner Says Five of Top 10 Worldwide Mobile Phone Vendors Increased Sales in Second Quarter of 2016*. Recuperado el 20 de marzo de 2017 de <http://www.gartner.com/newsroom/id/3415117>
- Google. (2017). Configurar pruebas Alfa o Beta. Recuperado el 21 de marzo de 2017 de <https://support.google.com/googleplay/android-developer/answer/3131213>
- Low, A. (2017). *Code And Web*. Recuperado el 28 de marzo de 2017 de <https://www.codeandweb.com/texturepacker/tutorials/how-to-create-a-sprite-sheet>
- Marcos, D. (2012). Motor gráfico... ¿y eso qué es? Recuperado el 29 de marzo de 2017 de <https://marcosdiez.wordpress.com/2012/11/14/motor-grafico-y-eso-que-es/>
- McDonald, E. (2017). *The global games market will reach \$108.9 billion in 2017 with mobile taking 42%*. Recuperado el 1 de Julio de 2017 <https://newzoo.com/insights/articles/the-global-games-market-will-reach-108-9-billion-in-2017-with-mobile-taking-42/>

nVidia. (2017). Juegos de pc. Recuperado el 3 de Julio de 2017 de <http://www.nvidia.es/object/pc-games-home-es.html>

Páginas Amarillas CanTV. (2011). Importancia de la seguridad vial. Recuperado el 29 de marzo de 2017 de <http://www.pac.com.ve/contenido/seguridad-y-proteccion/importancia-de-la-seguridad-vial/8605/90>

Petit, A. (2017). Sprint 0, clave en la gestión de proyectos ágiles. Recuperado el 03 de abril de 2017 de <https://www.paradigmadigital.com/techbiz/sprint-0-clave-la-gestion-proyectos-agiles/>

ProyectosAgiles.org. (2017). Lista de objetivos / requisitos priorizada (Product Backlog). Recuperado el 03 de abril de 2017 de <https://proyectosagiles.org/lista-requisitos-priorizada-product-backlog/>

Revista Líderes. (2013). La factura que dejan los accidentes de tránsito. Recuperado el 03 de abril de 2017 de <http://www.revistalideres.ec/lideres/factura-dejan-accidentes-transito.html>

Unity Technologies. (2017). Visión general del Sistema de Animación. Recuperado el 20 de marzo de 2017 de <https://docs.unity3d.com/es/current/Manual/AnimationOverview.html>

Unity Technologies. (2017). Aprendiendo la Interfaz. Recuperado el 20 de marzo de 2017 de <https://docs.unity3d.com/es/current/Manual/LearningtheInterface.html>

Unity Technologies. (2017). Visión general del Sistema de Animación. Recuperado el 21 de marzo de 2017 de <https://docs.unity3d.com/es/current/Manual/AnimationOverview.html>

4RSoluciones. (2019). 4R. Recuperado el 22 de Mayo de 2017 de <http://www.4rsoluciones.com/blog/que-es-un-kit-de-desarrollo-de-software-sdk-2/>

ANEXOS

Anexo 1

Términos y definiciones

- **Tester:** Persona encargada de probar el funcionamiento de una aplicación, de un software o de un producto antes de que sea lanzado al público con el fin de detectar errores.
- **Depurar:** Se refiere a la acción de buscar, y corregir errores dentro del código fuente de un software.
- **Asset:** Un activo es la representación de cualquier elemento que se puede utilizar en su juego o proyecto.
- **Sprite:** Es una textura o imagen usada para trabajar en juegos 2D
- **Trigger:** Son aquellos eventos que suceden cuando se cumple una o varias condiciones.
- **Trivia:** Una trivia es una serie de preguntas que se plantean en el marco de un concurso o de un juego, cuyas respuestas deben ser escogidas entre diferentes opciones.
- **Gameplay:** Jugabilidad, forma de interactuar de un videojuego con el jugador.
- **SDK:** De las siglas (Software Development Kit), o kit de desarrollo de software, es un conjunto de herramientas que ayudan a la programación de aplicaciones para un entorno tecnológico particular.

