

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE FORMACIÓN DE TECNÓLOGOS

DESARROLLO DE APLICACIÓN PARA FORTALECER LA MOTRICIDAD FINA EN PRESCOLARES MEDIANTE JUEGOS INTERACTIVOS.

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE TECNÓLOGO SUPERIOR
EN DESARROLLO DE SOFTWARE**

CHRISTIAN PAUL LLUMIQUINGA AYALA

DIRECTOR: Ing. IVONNE MALDONADO, MSc.

DMQ, febrero 2022

CERTIFICACIONES

Yo, CHRISTIAN PAUL LLUMNIQUINGA AYALA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



CHRISTIAN PAUL LLUMIQUINGA AYALA

Christian.llumiquinga@epn.edu.ec

cristianktt@outlook.com

Certifico que el presente trabajo de integración curricular fue desarrollado por CHRISTIAN PAUL LLUMIQUINGA AYALA, bajo mi supervisión.



Ing. IVONNE MALDONADO MSc.

DIRECTOR

ivonne.maldonadof@epn.edu.ec

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

CHRISTIAN PAUL LLUMIQUINGA AYALA

DEDICATORIA

Dedico este trabajo a mi familia por el apoyo incondicional en este tiempo y ayudarme a tener la fortaleza necesaria para poder culminarlo siendo un ejemplo conmigo de superación, honestidad, perseverancia, humildad y responsabilidad y darme ese valor durante toda mi vida universitaria.

CHRISTIAN PAUL LLUMIQUINGA AYALA

AGRADECIMIENTO

Agradezco a Dios por darme esa fortaleza y ese consuelo en mis momentos más difíciles, además agradezco a mi familia y amigos que siempre han estado apoyándome en culminar esta meta, un agradecimiento a mis padres, como primer punto mi papá por siempre darme ese apoyo incondicional pese a que las circunstancias no han sido fáciles, en segundo lugar, a mi mamá por ese apoyo moral y ese refugio en los momentos más complicados de mi vida.

Un agradecimiento especial a la Escuela Politécnica Nacional, a sus profesores, administrativos y quienes conforman la institución, un agradecimiento a los profesores de la escuela de formación de Tecnólogos por todos los conocimientos impartidos y que han sido de gran aporte a mi etapa universitaria para culminar esta carrera de forma exitosa.

CHRISTIAN PAUL LLUMIQUINGA AYALA

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	III
DEDICATORIA.....	IV
AGRADECIMIENTO.....	V
ÍNDICE DE CONTENIDO.....	VI
RESUMEN	VIII
ABSTRACT	IX
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general.....	2
1.2 Objetivos específicos.....	2
1.3 Alcance	3
1.4 Marco Teórico	3
Unity 3D.....	4
2 METODOLOGÍA	6
2.1 Metodología de Desarrollo	6
Roles.....	8
Artefactos.....	9
2.2 Diseño de interfaces	10
FIGMA.....	11
2.3 Diseño de la arquitectura	11
Arquitectura de Aplicación Unity.....	12
2.4 Herramientas de desarrollo.....	13
3 RESULTADOS.....	14
3.1 <i>Sprint</i> 0. Configuración del ambiente de desarrollo y Conceptualización del juego	14
Instalación de Unity 3D	14
Autoeducación de la herramienta Unity 3D.....	15
Configuración del ambiente de desarrollo.....	15
Conceptualización de la colección de juegos.....	16
Conceptualización de las mecánicas del juego	16
3.2 <i>Sprint</i> 1. Diseño	17
Diseño de la Interfaz del juego.....	17

Creación de las diferentes escenas del juego	18
Creación de los diferentes niveles de los juegos.....	18
3.3 <i>Sprint 2</i> Codificación de los minijuegos	21
Implementación del cronometro en los diferentes juegos.....	22
Implementación del juego ABC Descubre.....	22
Implementación juego de rompecabezas.....	23
Efectos de sonido	24
Implementación de efecto de sonido <i>Game Over</i>	25
Implementación efecto de sonido al cumplir un objetivo	25
Implementación del juego TocaToca	25
3.4 <i>Sprint 3</i> Pruebas	26
Pruebas Unitarias	26
Generación de la APK.....	28
Resultados de las pruebas de compatibilidad.....	32
Resultados de las pruebas de aceptación	33
3.5 <i>Sprint 4</i> Despliegue.....	34
Publicación en Play Store	34
4 Conclusiones	37
5 Recomendaciones	38
6 REFERENCIAS BIBLIOGRÁFICAS.....	39
7 ANEXOS.....	42
ANEXO I. Certificado de Originalidad.....	43
ANEXO II. Manual Técnico.....	45
Pruebas Unitarias	62
Pruebas de Compatibilidad	64
Pruebas de aceptación	67
ANEXO III. Manual de usuario	71
ANEXO IV. Manual de instalación.....	72

RESUMEN

El presente proyecto de titulación se desarrolla con el fin de fortalecer la motricidad fina en niños de preescolar mediante juegos interactivos, buscando ser una herramienta complementaria con la educación de hoy en día. La aplicación cuenta con diferentes actividades tipo juego que van aumentando de dificultad según el avance del niño, junto con la ayuda de la tecnología que cumplen un papel importante en la educación preescolar, al permitir el desarrollo de destrezas táctiles, auditivas y visuales enriqueciendo el proceso tradicional de enseñanza, al atraer el interés de los niños por medio de la actitud espontánea de curiosidad y de manera divertida a través del juego.

Esta aplicación móvil está diseñada con el motor de Unity, que nos proporciona las herramientas necesarias para el desarrollo de juegos, mediante la ejecución de scripts C#, que son asignados a la interfaz, que será con la que el usuario interactúa directamente en su dispositivo móvil, siendo así como interactúa esta aplicación móvil, con el usuario final.

Para el desarrollo de la aplicación se ha utilizado Sum, una metodología ágil similar a Scrum, pero dedicada específicamente a la creación de videojuegos, logrando así que se cumpla con todas las fases del ciclo de vida de *software* y enfocándose al cumplimiento de las necesidades del usuario final.

PALABRAS CLAVE: aplicación móvil, Sum, educación, motricidad fina, Unity.

ABSTRACT

This degree project is developed in order to strengthen fine motors skills in preschool children through interactive games, seeking to be a complementary tool with today's education. The application has different game-type activities that increase in difficulty as the child progresses, along with the help of technology that play an important role in preschool education, by allowing the development of tactile, auditory and visual skills, enriching the process. Traditional way of teaching, by attracting the interest of children through the spontaneous attitude of curiosity and in a fun way through play.

This mobile application is designed with the Unity engine, which provides us with the necessary tools for game development, through the execution of C# scripts, which will be the one with which the user interacts directly on their mobile device, this being how this mobile application interacts with the end user.

For the development of the application, Sum has been user, an agile methodology similar to Scrum, but specifically dedicated to the creation of video games, thus achieving compliance with all phases of the software life cycle and focusing on meeting the needs of the final user.

KEYWORDS: mobile app, Sum, education, fine motor, Unity.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

La motricidad fina es uno de los temas de mayor relevancia durante el desarrollo de un niño, la misma que trata de la relación que existe entre los músculos de las manos y la vista, de manera que se trabaje de forma coordinada entre las mismas [1]. Por lo que es de suma importancia el desarrollo de la misma, sobre todo a temprana edad cuando estos se encuentran en la edad de preescolar [2].

El desarrollo de la motricidad fina es decisivo para poder aprender y experimentar sobre el entorno que, a su vez, conlleva un papel central en el desarrollo de la inteligencia del infante [3]. Se considera a la edad preescolar en el rango de edad de 3 a 6 años, etapa en la que se perfecciona la coordinación óculo-manual, es aquí donde se debe llevar una especial atención en el aspecto psicomotor que es la base de los aprendizajes preescolares posteriores [4].

Actualmente se plantea que la educación debe responder a estándares de calidad, que estén acordes con el desarrollo cronológico de los niños, por ello el hecho de que las organizaciones mundiales se preocupen por crear políticas de bienestar educativo que centren su atención en la población de preescolar, que garanticen un desarrollo eficaz, y holístico, sin excluir condiciones sociales, ideológicas, políticas o económicas. En Ecuador el ministerio de educación está acoplado políticas educativas que garantizan mejorar la educación en todos los niveles, priorizando la educación preescolar e infantil, debido a que estas son edades en las que se debe desarrollar al máximo las capacidades, destrezas y habilidades, siendo esta participativa, activa, efectuada, con estrategias innovadoras, para lograr niveles: cognitivas, motrices, sociales y afectivos eficientes [5].

Por otro lado, hoy en día las tecnologías de la información y comunicación (TIC), cumplen un papel importante en la educación preescolar, permitiendo el desarrollo de destrezas táctiles, auditivas y visuales [6]. El incorporar estas tecnologías en espacios preescolares requiere de un proceso pensado, intencionado y direccionado, de modo que se logre la interacción formativa de niños y niñas de una forma curiosa, donde pueden explotar, conocer, cuestionar, armar, desarmar y hasta investigar [7]. Su uso como recurso de apoyo para la educación enriquece el proceso tradicional de enseñanza, ya que atrae el interés de los niños por medio de la actitud espontánea de la curiosidad, más aún si se refiere a la educación preescolar, desarrollando habilidades como la visión, la audición, la atención, la memoria, la percepción y la cognición de manera divertida a través del juego y mediante una cultura tecnológica a edad temprana, preparando a los niños y niñas a enfrentarse al dinamismo de la era de la información actual [8]. Ahí el hecho de que las TIC deben ser un

recurso que se adapten a los procesos de enseñanza en la etapa educativa del preescolar [9].

El *software* como herramienta educativa, es una gran herramienta que ayuda a complementar la educación, en el caso de la educación preescolar el *software* educativo mediante juegos que complementen su aprendizaje, curiosidad y razonamiento de formas interactivas que ayudaran al preescolar a desarrollar su pensamiento.

Es por aquellos que se ha desarrollado una aplicación móvil, que contiene una colección de juegos interactivos que sirven para fomentar el razonamiento y curiosidad de los preescolares, la misma que tiene una interfaz gráfica agradable al usuario, y que de forma paulatina puede ir avanzando entre los niveles de dificultad de los diferentes juegos de la aplicación móvil.

La aplicación móvil consta de tres juegos diferentes, los mismos que poseen distintos niveles de dificultad, cada uno de estos juegos ayuda al preescolar a desarrollar diferentes habilidades; el primero, un juego de rompecabezas donde el infante genera estrategias para su resolución, el segundo juego consta de identificar diferentes animales para que vaya completando el nombre de la especie animal al que pertenezca, por último se presenta un juego donde el niño tendrá que atrapar ratones a diferentes velocidades. Cabe mencionar que la aplicación móvil está únicamente orientada a niños preescolares entre 3 a 6 años.

1.1 Objetivo general

Desarrollar una aplicación para fortalecer la motricidad fina en niños de preescolar mediante juegos interactivos.

1.2 Objetivos específicos

1. Conceptualizar la idea de los juegos interactivos, mediante el entendimiento y necesidades para fortalecer la motricidad fina en niños de preescolar basado en la experiencia de un profesional parvulario.
2. Diseñar los ejercicios específicos, tipo juegos, a ser realizados por los niños de preescolar.
3. Codificar la aplicación en base a las necesidades para fortalecer la motricidad fina en niños de preescolar.
4. Probar la aplicación hasta alcanzar lo esperado como producto final.

1.3 Alcance

En este proyecto se ha desarrollado una aplicación que sirve como herramienta para ayudar en el desarrollo de la motricidad fina de niños de preescolar mediante juegos interactivos con el uso de dispositivos móviles, en este sentido el alcance de este proyecto tiene como beneficiarios directos a los niños y niñas de preescolar que se encuentran bajo el cuidado particular de una Licenciada Parvularia en el sector norte de Quito, quien actualmente es la encargada de brindar un servicio particular en educación preescolar, tareas dirigidas y asesoramiento en técnicas de lenguaje y psicología; como una alternativa tecnológica para mejorar y fortalecer la motricidad fina de los niños, aprovechando las posibilidades que ofrece una interfaz de pantalla táctil como tocar varios punteros al mismo tiempo, es decir varios dedos para aumentar la fuerza, el control y la destreza por medio de una aplicación móvil para la plataforma Android [10], [11].

1.4 Marco Teórico

En el desarrollo de *software* existen diferentes metodologías las cuales se adaptan a las necesidades del desarrollador o equipo de desarrollo, así mismo el *software* está basado en el cumplimiento necesario de cada una de las fases del ciclo de vida, que es una serie de pasos que buscan garantizar que los sistemas y aplicaciones creadas sean: eficientes, fiables, seguros y respondan a las necesidades de los usuarios finales validando y verificando que se esté cumpliendo con los requisitos y usando procedimientos apropiados sin importar el tamaño del proyecto [12].

Con lo que respecta a proyectos de corta duración actualmente las metodologías de desarrollo ágil marcan una gran diferencia, de ahí para poder cumplir con el desarrollo de la presente propuesta se haya elegido una metodología ágil *Scrum* variación para el desarrollo de videojuegos de *Scrum*, la cual permite que el trabajo se vaya adaptando a las condiciones del proyecto y al entorno cambiante que este conlleva, así permitiendo tener una respuesta inmediata y flexible para alcanzar los objetivos con pequeñas entregas funcionales para alcanzar los objetivos planteados mediante pequeñas entregas funcionales acorde a las necesidades del cliente, siendo una solución rápida y así evitando la documentación excesiva [13], [14].

Los video juegos en su mayoría se han considerado un entretenimiento o una forma de escape para niños, jóvenes y adultos, sin embargo también se considera el hecho de aprender jugando debido a que esta es una actividad que se lo realiza de forma voluntaria sin importar la edad que se tenga, fortaleciendo habilidades como desarrollo de la

creatividad, competencia intelectual, fortaleza emocional y estabilidad personal; es decir el desarrollo integral de una persona [15].

En este contexto la tecnología usada en la educación es un recurso importante y esta enriquece el proceso de enseñanza tradicional, trayendo consigo el interés de los niños, creando en ellos una actitud espontánea de curiosidad, ampliando el hecho de que los niños logren niveles crecientes de conocimiento [16].

Es así como a lo largo de los años se han ido desarrollando aplicaciones que complementan la educación, en este caso la educación preescolar haciendo énfasis en lo que es la motricidad fina, un ejemplo de ello es la aplicación *dexteira* que contiene un conjunto de juegos con múltiples ejercicios que promueven la ejercitación y desarrollo de las habilidades motoras finas [17].

Así mismo la aplicación desarrollada para este proyecto, tiene su fuerte en ser una colección de minijuegos atractivos para los preescolares, en el cual ellos pueden desarrollar diferentes habilidades de razonamiento y además la aplicación móvil presenta diferentes niveles de dificultad los cuales pondrán a prueba el buen desarrollo de los infantes.

Esta aplicación móvil es nativa para Android por lo cual su desarrollo es específico para dicha plataforma, uno de los principales beneficios de una aplicación nativa es que puede tener un mejor rendimiento en dicha plataforma [18]. Así mismo se da el uso de la herramienta Unity que nos permite crear aplicaciones nativas de buen rendimiento para diferentes plataformas siendo el caso de este proyecto la plataforma Android [19].

Unity 3D

En la actualidad los videojuegos han tenido un gran crecimiento en la industria del *software*, en un comienzo se entendía a los videojuegos solo como algo de entretenimiento para el usuario, pero ahora existen muchos juegos orientados a la educación por lo cual, el *software* de videojuegos tiene un gran crecimiento en el sector educativo [19].

La creación de un videojuego puede ser un trabajo extremadamente grande en cuanto a diseño, historia, programación, animación, etc. Lo cual hace demoroso su desarrollo, de ahí el hecho de que existan diferentes herramientas que facilitan dicho trabajo, dado todo el soporte necesario para poder crear un juego.

Unity posee el soporte para múltiples plataformas en el ámbito de videojuegos, esto incluye los dispositivos móviles, con Android [19], por ello es considerada líder en cuanto a desarrollo de videojuegos se trata. Esta herramienta engloba entre sus beneficios motores para el renderizado de imágenes, de audio, de animaciones entre otros, además de tener

soporte para la creación de juegos multijugador, herramientas de navegación y soporte para realidad virtual [20].

El motor gráfico que usa Unity para Android y IOS es *OpenGL (Open Graphics Library)*, en el cual se hace énfasis en que es un *API (Aplicación Programming Interface)* multiplataforma de gráficos que especifica una interfaz de *software* para el hardware de procesamiento de gráficos dando así una compatibilidad de gráficos de alto rendimiento con Android [19].

Unity también puede trabajar con diferentes lenguajes de programación y su implementación está basada en *scripts* los cuales pueden ser creados mediante JavaScript o C#, además de aquello brinda un editor de código integrado [19].

Se ha escogido esta herramienta como motor gráfico de la aplicación móvil y sus diferentes juegos, por lo completo que es, para el desarrollo multiplataforma en este caso la plataforma Android, pudiendo crear diferentes escenas y partes de los juegos y luego añadir el código correspondiente para que funcionen cada uno de los elementos que conforman los diferentes juegos de la aplicación móvil.

2 METODOLOGÍA

Para alcanzar los objetivos propuestos dentro de este proyecto se debe seguir una serie de tareas las cuales tienen distintas prioridades para que así las actividades y objetivos propuestos sean exitosos, a este proceso se le conoce como metodología, siendo de suma importancia y aplicada en diferentes áreas de la ciencia y en la vida diaria, ya que ayuda a evitar que el resultado final de los objetivos este condicionado por factores aleatorios.

Para aplicar una metodología de trabajo se debe tener un conocimiento previo o seguir y cumplir los pasos correctamente, de manera que se alcance los objetivos planteados de forma concreta al ejecutar el método aplicado en un caso en particular [21].

El estudio de caso consiste en una técnica o método de investigación que se caracteriza por tener un proceso de búsqueda, indagación y análisis sistemático de uno o varios casos, a diferencia de otros tipos de investigación empírica, esta metodología es considerada como una técnica de investigación cualitativa, puesto que aquí se centra en el estudio exhaustivo de un fenómeno en específico [22].

En este proyecto los estudios de caso son un modelo de investigación útil que sirve para la recopilación de información en específico como es el de la motricidad fina en niños de preescolar mediante el uso de la tecnología por medio de juegos dentro de la educación.

2.1 Metodología de Desarrollo

Esta aplicación móvil consta de videojuegos educativos, por lo que se la considera como un videojuego. El desarrollo de videojuegos se considera un trabajo arduo, que conlleva tiempo es así que muchos de los equipos de programación optan por usar los métodos tradicionales del desarrollo de *software*, sin embargo existen metodologías que pueden ser adaptados a este concepto de desarrollos como lo es la metodología *Sum* adaptándose al entorno de trabajo y a los recursos que se tiene a disposición [23].

Sum no es más que una variante de *Scrum*, pero enfocada en el desarrollo de videojuegos, por lo que brinda flexibilidad al definir el ciclo de vida del *software* y se adapta fácilmente a distintas realidades del equipo [24].

En cuanto a los roles, son los mismos que se tiene en *Scrum*, es decir se tiene: *Scrum Master*, *Product Owner* y *Development Team*, con la diferencia de que se debe contar con sub-roles como: artista gráfico, artista sonoro y diseñador de juego, ya que se requiere una alta especialidad para satisfacer la demanda de un videojuego [24]. Así también, en *Sum* se tiene el rol de verificador beta, necesario e indispensable en la industria de los

videojuegos ya que es el encargado de realizar las pruebas de funcionalidad finales, y comunicar el resultado obtenido [24].

El ciclo de vida del *software* según la metodología *Sum* se divide en fases secuenciales las cuales van incrementando según avance el desarrollo y surgen como adaptación al desarrollo de videojuegos (*pre-game, game, post-game*). *Pre-game* y *game* coinciden con las fases de planificación y elaboración mientras que *post-game* corresponde a la fase beta y cierre, pero con división ya que como se menciona la fase beta es una fase propia de la industria de los videojuegos [24].

Las fases secuenciales son:

- **Concepto:** en esta fase se desarrolla la idea y el tipo de juego a crear, así como las características, en este punto también se toma en cuenta el *gameplay* [23]. Es aquí donde se centra la idea del negocio y al público objetivo, además se define las herramientas con las que se va a trabajar; este punto finaliza cuando se tiene aprobado todo el concepto del videojuego entre todas las partes involucradas [24].
- **Planificación:** este punto pone énfasis en el cronograma y avances propuestos, siempre tomando en cuenta las capacidades del equipo de desarrollo [23]. En esta fase también es donde se realiza la recopilación de requerimientos funcionales y no funcionales, siendo los funcionales una función propia del juego y los no funcionales una propiedad del juego a desarrollar, y con ello poder crear la planificación de cómo se va desarrollando cada uno de los elementos que componen el videojuego; la planificación puede irse modificando para irse adaptando a los cambios y al estado actual en que se encuentra el proyecto [24].
- **Elaboración:** esta es la fase más importante en el desarrollo del videojuego ya que aquí se define el éxito o el fracaso del mismo, esta fase se subdivide en tres fases; la primera es el punto en el que se plantean los objetivos los cuales definen el avance del proyecto que se van cumpliendo en base a tareas designadas y al cronograma desarrollado en la fase anterior [23], en la segunda se desarrollan las características del juego en base a las tareas especificadas que se ha planificado con el equipo del proyecto, mantiene un control de la misión y visión del proyecto para llevar el control en cada interacción, y la tercera fase evalúa el estado del videojuego y de todo lo ocurrido durante la interacción, para que así se pueda actualizar el plan del proyecto según el estado real del mismo [24].

- **Beta:** en esta fase se desarrolla una versión de prueba del videojuego la cual sirve para generar una retroalimentación y con lo que se corrigen errores, además de que se puede publicar el videojuego y obtener una retroalimentación de jugadores reales [23].

Mientras esto sucede se pueden generar reportes con los errores y las características evaluadas para realizar ajustes al videojuego si es necesario. Se pueden liberar diferentes versiones beta y así mismo analizar errores y realizar ajustes pertinentes, esta iteración finaliza una vez alcanzado los criterios de funcionalidad que se tengan en la planificación en conjunto con los interesados [24].

- **Cierre:** es la fase final, cuando finalmente se lanza la versión terminada del videojuego, durante esta fase se evalúan los objetivos cumplidos, los problemas presentados, las soluciones propuestas y finalmente se obtiene la retroalimentación de la creación de todo el videojuego, obteniendo información para futuras decisiones [23].

Con las conclusiones obtenidas se aprende de los fallos y así se mejora esta metodología para hacerla de una forma más eficiente en futuros proyectos [24].

Roles

Ya que la aplicación móvil se ha tomado como un videojuego por los componentes que tiene para poder alcanzar los objetivos planteados, *Sum* define de forma similar a *Scrum* los roles, pero enfocándolo al desarrollo de videojuegos [23], definiendo y asignando a los miembros del equipo un rol específico, que para este proyecto son:

Product Owner: es quien se encarga de gestionar el desarrollo del proyecto, además de ser el representante del cliente final; es encargado de hacer llegar todos los requerimientos para que puedan ser interpretadas en el desarrollo [25].

Scrum Máster: es el encargado de garantizar que todos los miembros del equipo lleven a cabo el correcto uso de metodología y que se cumplan todos los procesos de esta. Debe supervisar que todo el proyecto avance sin contratiempos y asegurarse que todos los miembros del equipo tengan las herramientas necesarias para poder sacar adelante el proyecto [25].

Development Team: es el grupo de personas o equipo que está encargado de entender los requerimientos del usuario, las especificaciones del funcionamiento del negocio, estimar las Historias de Usuario y crear los *Sprint* del proyecto [25].

La asignación de dichos roles se presenta a continuación en la TABLA I: Asignación de Roles.

TABLA I: Asignación de Roles.

NOMBRE	ROL
Lcda. Tania Soliz	<i>Product Owner</i>
Ing. Ivonne Maldonado MSc.	<i>Scrum Máster</i>
Sr. Christian Llumiquinga	<i>Development Team</i>

Artefactos

Son todos aquellos elementos que garantizan una transparencia en el registro de la información, mejorando la calidad y la productividad de la metodología [26].

Recopilación de Requerimientos

Es la identificación de requerimientos de usuario para luego gestionarlos y transfórmalos en casos de uso [27].

La recopilación de requerimientos se ha logrado en base a entrevistas con el principal interesado en el producto, toda esa lista esta detallada en el **ANEXO II - Recopilación de Requerimientos**.

Historias de Usuario

Es la descripción de la funcionalidad que debe incorporar un *software* [28]. En este caso la jugabilidad de cada uno de los juegos de la aplicación móvil; es así como las historias de usuario describen lo que el jugador puede hacer, así como lo que el equipo de desarrollo necesita para lograr la implementación del videojuego.

Las historias de usuario que se han elaborado en este proyecto están basadas en la recopilación de requerimientos y son la clave para poder empezar con la fase de elaboración. En la **ANEXO II - Historias de Usuario**.

TABLA II se detalla un ejemplo de las 18 historias de usuario que se encuentran en **ANEXO II - Historias de Usuario**.

TABLA II – Ejemplo historias de usuario.

Historia de Usuario	
Identificador: HU1	Usuario: Desarrollador.

Nombre de la Historia: Realizar Autoeducación en la herramienta Unity.	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Medio
Días Estimados: 7	Sprint Asignados: 0
Programador Responsable: Llumiyinga Christian.	
Descripción: El desarrollador de la aplicación móvil podrá utilizar la herramienta Unity para la construcción de los juegos de la aplicación móvil.	
Observaciones: El desarrollador puede iniciar un proyecto para Android en la herramienta Unity, que sirve para el desarrollo de juegos para diferentes plataformas.	

Product Backlog

Es una lista ordenada con todo lo que se necesita para cumplir con el desarrollo del producto de *software* [26]. En ella se muestran todas las funcionalidades requeridas para la implementación del proyecto, en una forma ordenada, donde cada iteración corresponde a cada fase del desarrollo de la metodología *Sum* para poder llevar a cabo la construcción del componente. Es así como la etapa de concepto y planificación corresponde al *sprint* 0 y 1, la etapa de elaboración que es la que lleva la codificación del juego es el *sprint* 2, la etapa de pruebas es la etapa beta que corresponde al *sprint* 3 y finalmente la parte del cierre corresponde al *sprint* 4.

Además, también se indica el identificador único por cada historia de usuario, los días estimados y la prioridad que representa para el cumplimiento del proyecto, el *Product Backlog* se representa en el **ANEXO II - TABLA XXV**– Product Backlog.

Sprint Backlog

Es un subconjunto del *product backlog*, actividades que son elegidas para abordarse en un *sprint*, que es un periodo de tiempo delimitado donde se puede alcanzar un avance del proyecto, dividido el mismo en pequeños proyectos que se pueden entregar y acoplar a la funcionalidad completa del proyecto [26]. El *sprint backlog* de este proyecto está detallado en el **ANEXO II** -¡Error! No se encuentra el origen de la referencia..

2.2 Diseño de interfaces

Un *mockup*, es un montaje que se realiza antes del desarrollo, para así poder mostrar al cliente una vista previa del producto final, esto ayuda al cliente a echar un vistazo a cómo

será la aplicación y como se interactúa con ella, pudiendo así pedir cambios o aceptarlos antes de ser codificados [29].

FIGMA

Es una aplicación web para el diseño de interfaces, esta herramienta brinda múltiples herramientas vectoriales capaces de ilustrar y generar prototipos de cómo se verá la aplicación móvil en funcionamiento, también permite crear bibliotecas reutilizables que pueden ser usadas a futuro, otra de la ventaja es el poder generar simulaciones en base a puntos de acceso en el diseño para dar un vistazo a como el usuario navegaría por la aplicación móvil [30]. El diseño básico de las diferentes pantallas de la aplicación móvil se encuentra detallado en el **ANEXO II. Manual Técnico**

La **Fig. 1** muestra la pantalla principal del videojuego.



Fig. 1 Diseño escena principal.

2.3 Diseño de la arquitectura

Este proyecto se trata de una aplicación móvil que contiene una colección de minijuegos, por ello el hecho de manejarse como un videojuego, en este contexto se usa el modelo vista controlador como guía ya que es posible usar este patrón arquitectónico para separar los datos con la lógica del videojuego, sin ignorar la naturaleza de las vistas o escenas de un videojuego, sin embargo esto puede generar un colapso debido a los cambios frecuentes que naturalmente requieren los videojuegos [31]. Por ello se opta por una variación de esta arquitectura, donde por un lado se puede cumplir con la separación en capas para lograr calidad de *software* como lo es la estabilidad y mantenibilidad que requiere un videojuego sin dejar de lado la arquitectura que maneja Unity [31].

Arquitectura de Aplicación Unity

Para poder realizar la estructuración del videojuego se ha utilizado arquitectura en capas junto con componentes, es así como los componentes de diferentes capas se están comunicado entre si a través de las interfaces o de instancias de clases, las cuales se comunican con el *Game Manager* de Unity [31]. Definiendo las siguientes capas:

Capa principal: en esta se encuentra el controlador principal del juego siendo el *GameManger*, el *State Machine*, el *Sound Manager* y el *Scene Manager*.

- El *Game Manager* que es el controlador principal del sistema que se implementa como un objeto de Unity el cual se encuentra en todas las escenas [31].
- El *State Machine* es el que se encarga de guardar los estados del juego en los diferentes estados que este pueda presentar, siendo así como son los niveles de la colección de juegos [31].
- El *Sound Manager* permite llevar el control del sonido del juego y el *Scene Manager* es el que permite el cambio de escenas como se realiza en este proyecto [31].

Capa de caracteres: en esta capa se lleva los diferentes elementos del proyecto, los que pueden ser jugables o no, es decir con los que el usuario puede tener interacción o no [31].

Capa de interacción: como último punto del proyecto se tiene la capa de interacción donde agregan los diferentes elementos dentro de la escena siendo así que los que modifican el videojuego se agregan aquí, como en el caso del proyecto se opta por añadir otras mecánicas al juego como lo es un cronometro que llevara el límite de tiempo en el cual un nivel de la colección de juegos puede ser superado [31].

La **Fig. 2** muestra el modelo de arquitectura a seguir según la estructura propia de Unity, para el desarrollo de esta aplicación móvil.

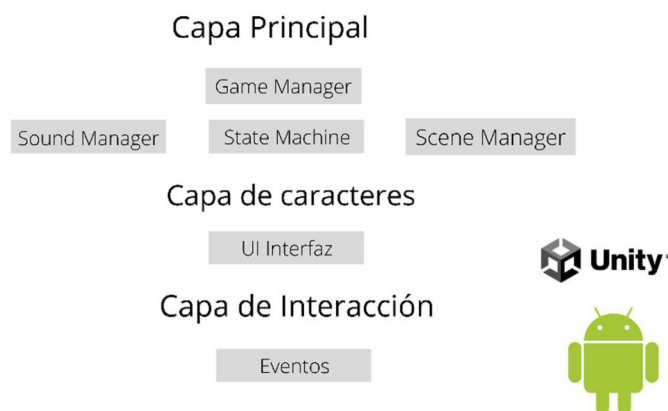


Fig. 2: Patrón de Arquitectura Unity para la aplicación móvil.

2.4 Herramientas de desarrollo

Este proyecto se enfoca en la creación de una aplicación móvil, con una colección de juegos que ayudan al desarrollo de la motricidad fina de los prescolares, es así como se ha optado por usar el motor gráfico de Unity, que proporciona las herramientas para la creación de un juego 2D, así mismo mediante la herramienta se crea la apk que sirve para su instalación en el sistema Android. A continuación, en la **TABLA III** se muestran las herramientas que se han utilizado en la creación de este proyecto.

TABLA III: Herramientas de Uso para el desarrollo de la aplicación móvil.

Herramienta	Justificación
Unity	Es la herramienta escogida para el desarrollo de esta aplicación móvil. Con una colección de juegos interactivos, debido a la lógica del negocio de este proyecto se lo debe tratar como un videojuego, es así como Unity es la herramienta ideal para el desarrollo de videojuegos en 2D y 3D [32].
Visual Studio	Viene integrado con el motor de Unity, proporcionando las herramientas idóneas para el desarrollo de código fuente de manera rápida y precisa, siendo así un IDE (Entorno de desarrollo Integrado), completo para Unity [33].
GitHub	Es la plataforma por defecto para el almacenamiento de código y trabajo remoto.
C#	Es un lenguaje de programación completo basado en C++ y C, es un lenguaje compilado, por lo que debe ejecutarse previamente, este lenguaje está integrado a Unity, siendo este lenguaje el que permite darle la funcionalidad a los diferentes módulos desarrollados en Unity [33].

3 RESULTADOS

En la siguiente sección se muestran los resultados obtenidos en cada una de las tareas que provienen de los *Sprint*, además también se muestran los resultados de las pruebas realizadas según la metodología *Sum*.

3.1 *Sprint* 0. Configuración del ambiente de desarrollo y Conceptualización del juego

En base a la planificación el primer *Sprint*, tiene como resultado:

- Instalación de Unity 3D.
- Configuración del ambiente de desarrollo.
- Autoeducación de la herramienta de desarrollo.
- Conceptualización de la colección de juegos.
- Conceptualización de las mecánicas del juego.

Instalación de Unity 3D

Unity 3D es una herramienta nueva para el desarrollador, por lo que está se coloca como una actividad del *Sprint* 0, en la siguiente ilustración se detalla cómo se descarga Unity 3D y sus principales características siendo más detallado en el **ANEXO II - Descripción del entorno Unity 3D**.

Para la instalación de Unity 3D se muestran la página oficial de Unity, junto con Unity Hub en la **Fig. 3** y **Fig. 4** respectivamente, permitiendo la instalación de una versión específica [32].



Fig. 4: Página Oficial de Unity.

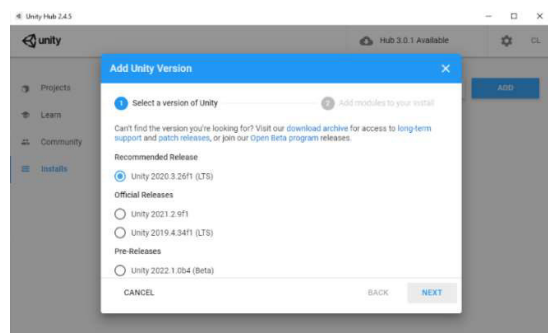


Fig. 3: Unity Hub.

Autoeducación de la herramienta Unity 3D

Ha sido necesario entender las mecánicas de la herramienta en si, por lo que para cumplir con esta tarea se ha seguido tutoriales correspondientes a Unity, junto con la documentación oficial, misma que se encuentra en la referencia [34].

A continuación, se presenta la lista de elementos estudiados.

1. Event System.
2. SceneManager.
3. Box Collider.
4. Audio Source.
5. Canvas Ui.
6. UnityEngineUi.
7. TouchRotate.
8. GameObject.
9. UnityEngineEventSystems.
10. Interfaces Drag and Drop.
11. Transform.

Ha sido de suma importancia conocer estos elementos para el desarrollo de la aplicación móvil, los mismos que se pueden encontrar en la documentación de Unity [34].

Configuración del ambiente de desarrollo

La configuración del ambiente de desarrollo se complementa con la instalación de las herramientas necesarias para trabajar, como ya es el caso de Unity, realizando la configuración de Visual Studio, para poder usar el código de Unity con C# y que compile dentro de Unity, esta configuración se la hace en el instalador de Visual Studio como se muestra en la **Fig. 5**.

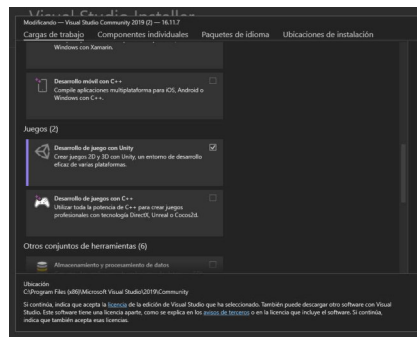


Fig. 5: Instalador de los módulos de Unity para Visual Studio.

Conceptualización de la colección de juegos

Para completar esta tarea se ha basado en desarrollar la colección de juegos de la aplicación móvil, según los ejercicios adecuados para un precolar, juegos referentes al desarrollo motriz, lógico y de aprendizaje, teniendo como resultado el diseño de la colección de juegos de un rompecabezas, aprendizaje de palabras mediante sonidos y ejercicio de dedos mediante interacción con la pantalla táctil.

En la **Fig. 6** se muestran la pantalla de selección del tipo de juego.



Fig. 6: Juegos implementados en la aplicación móvil.

Conceptualización de las mecánicas del juego

Para la conceptualización de las mecánicas del juego se toman en cuenta los juegos de la colección definidos, dando como resultado tres niveles iniciales, avanzar de acuerdo con cierto nivel de dificultad, etc., teniendo como principales características las siguientes:

- **Cronometro:** Esta mecánica indica que cada nivel de cada uno de los juegos se debe superar en un tiempo estimado, generando mejor desarrollo de un pensamiento rápido en los niños precolares según el nivel que vayan superando.
- **Reinicio:** Por cada nivel que no se haya superado o se haya complicado de más el precolar puede reiniciar el nivel del juego y empezar de nuevo, en tanto el nivel sea fallido por el precolar este tiene la opción de reintentar el juego.
- **Siguiente:** Otra de las mecánicas del juego es que con cada nivel superado se habilita el nivel de siguiente, aumentando el nivel de dificultad.
- **Descripción del objetivo del juego:** Como ya se ha indicado cada una de las mecánicas del juego, el objetivo en sí será que esta aplicación móvil sirva para ejercicio de niños precolares en el desarrollo de sus habilidades motrices finas

y todo lo relacionado, siendo así que el preescolar deberá superar los diferentes niveles de dificultad del juego, por cada uno de la colección de juegos pertenecientes a la aplicación móvil.

3.2 *Sprint 1*. Diseño

En base a lo planeado en el *Sprint 1* tiene como resultado lo siguiente:

- Diseño de la Interfaz del juego.
- Creación de las diferentes escenas del juego.
- Creación de los diferentes niveles de cada juego.

Diseño de la Interfaz del juego

En base a los parámetros propios de la lógica del juego, el diseño de la interfaz va acorde con la atracción de un niño preescolar, por lo cual para conseguir el diseño más adecuado se ha hecho uso de colores que llaman la atención del preescolar. La **Fig. 7** y **Fig. 8** muestran los diseños de la pantalla principal y la pantalla que abre la colección de juegos.



Fig. 7: Interfaz principal del juego.



Fig. 8: Interfaz selección de juego.

Creación de las diferentes escenas del juego

Para esta tarea se hace uso de las escenas de Unity, para dividir las diferentes interfaces de la aplicación móvil, por lo que estas escenas son destinadas a ser los diferentes niveles además de ser también los diferentes juegos, cada una de estas escenas están representando un punto en específico de la aplicación móvil, ya especificado en la lista de requerimientos, a continuación, en la **Fig. 9** se presentan las escenas creadas para la aplicación móvil, listadas en la carpeta de escenas del directorio del proyecto Unity.

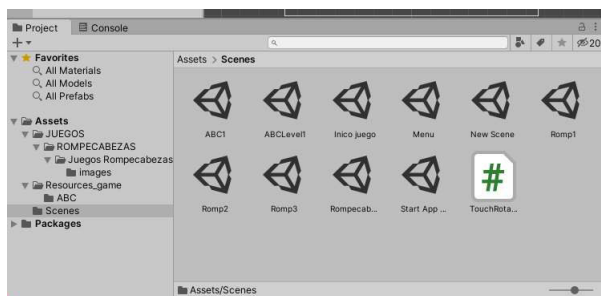


Fig. 9: Escenas de la aplicación móvil.

Creación de los diferentes niveles de los juegos

Para esta tarea se debe recordar que la aplicación móvil contiene una colección de juegos, donde cada juego, según el tipo, tiene un objetivo principal que se debe cumplir en el tiempo establecido, es así como el jugador debe superar cada uno de los niveles y pasar al siguiente nivel, hasta poder cumplir la colección de ejercicios, de los diferentes juegos.

El primer juego es un rompecabezas donde el usuario debe cumplir el objetivo de armarlo en el tiempo establecido.

Nivel 1 - Juego rompecabezas:

El usuario debe completar un mini rompecabezas en un tiempo límite de 10 segundos para de esa forma pasar al siguiente nivel, en la **Fig. 10** se muestra el nivel 1.

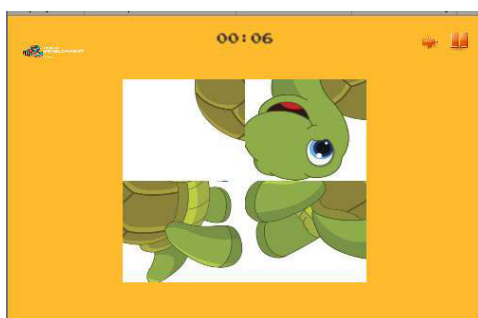


Fig. 10: Nivel 1 Rompecabezas.

Nivel 2 - Juego Rompecabezas:

El nivel 2 del rompecabezas consiste en que el usuario debe superar en 45 segundos un rompecabezas con mayor cantidad de piezas, por ello se lo considera un nivel más complejo. La **Fig. 11** presenta el respectivo nivel.

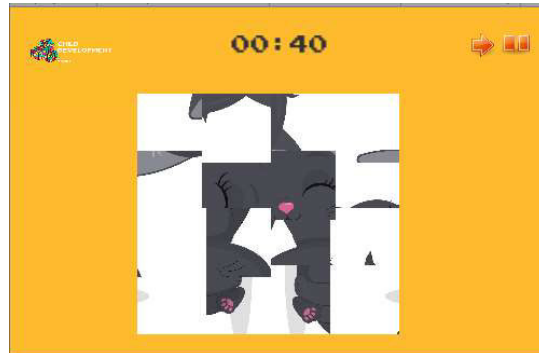


Fig. 11: Nivel 2 Rompecabezas.

Nivel 3 - Juego Rompecabezas:

El nivel 3 del juego del rompecabezas aumenta su dificultad debido a una mayor cantidad de piezas, cabe mencionar que este es el nivel final de este tipo de juego, siendo que al completar este objetivo el prescolar puede volver al menú principal y repetir el juego; este rompecabezas se debe cumplir en un tiempo límite de 60 segundos. En la **Fig. 12** se muestra el nivel 3 del juego de rompecabezas.



Fig. 12: Nivel 3 Rompecabezas.

El segundo juego consiste en escuchar un sonido en este caso de un animal; el prescolar debe determinar de qué animal se trata y colocar las letras de la palabra que describe a ese animal, escogiendo cada letra de una pequeña sopa de letras.

Nivel 1 - Juego ABC Descubre:

El primer nivel de este juego consiste en una palabra sencilla, siendo el objetivo colocar las letras en su casilla adecuada dentro del tiempo límite de 20 segundos, para poder convertirlo en un reto, pasando así al siguiente nivel. En la **Fig. 13** se muestra la pantalla del nivel 1 del juego ABC Descubre.



Fig. 13: Nivel 1 ABC Descubre.

Nivel 2 - Juego ABC Descubre:

En este nivel el objetivo a completar es una palabra mucho más compleja, teniendo como pista el rugido de un animal, el precolar debe buscar entre la sopa de letras cuales son las que cuadran en los casilleros, y describen el nombre del animal del que viene el sonido. En la **Fig. 14** se muestra la pantalla del nivel 2 del juego ABC Descubre.

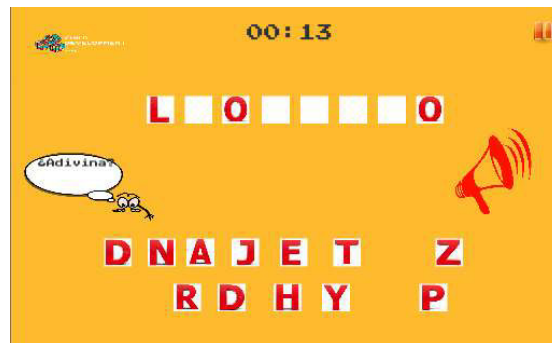


Fig. 14: Nivel 2 ABC Descubre.

Nivel 3 Juego ABC Descubre:

El nivel 3 del juego ABC Descubre el precolar debe cumplir con el mismo objetivo, completar la palabra y saber que animal es el que emite el sonido que se toca en la bocina, así mismo el precolar una vez complete con cada letra en la casilla correcta, con un nivel de dificultad mayor respecto al nombre del animal a ser completado, cabe

mencionar que con este nivel el tipo de juego termina. La **Fig. 15** muestra la interfaz del nivel 3 del juego ABC Descubre.

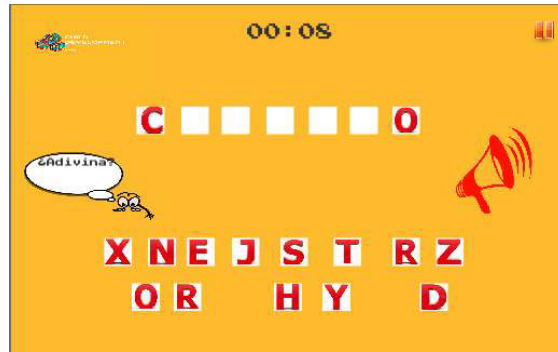


Fig. 15: Nivel 3 ABC Descubre.

Juego 3 - TocaToca:

El tercer juego es donde el precolar debe atrapar a los diferentes ratones que se presentan en un terreno rocoso, siendo el objetivo final que el usuario interactúe con la pantalla para poder eliminar a todos los ratoncitos que aparecen en ella. En la **Fig. 16** se muestra la Interfaz de este juego.



Fig. 16: Juego 3 TocaToca.

3.3 Sprint 2 Codificación de los minijuegos

En base a la planificación en este *Sprint* se realiza la codificación del proyecto en sí, teniendo como resultados:

- Implementación del cronometro en los diferentes juegos.
- Implementación juego ABC Descubre.
- Implementación juego de rompecabezas.
- Efectos de sonido.

- Implementación efecto de sonido *Game Over*.
- Implementación efecto de sonido al cumplir un objetivo.
- Implementación del juego TocaToca.

Implementación del cronometro en los diferentes juegos

Para cumplir con los objetivos de cada nivel, el prescolar debe superar en un tiempo límite los diferentes niveles para llegar al final de cada juego, es así como, para este punto, el cronometro se encuentra en la parte superior de la interfaz gráfica para cada nivel, en la **Fig. 17**, se muestra la ubicación del cronometro en funcionamiento.

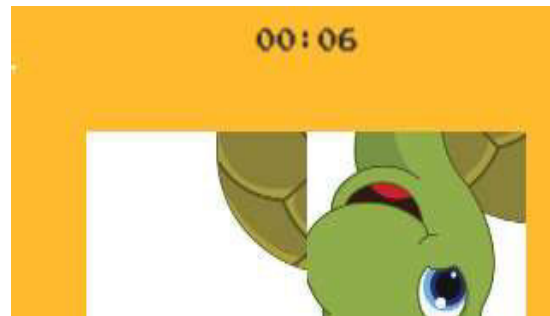


Fig. 17: Cronometro en rompecabezas nivel 1.

Implementación del juego ABC Descubre

Esta tarea cumple directamente con una de las funcionalidades propias de la aplicación móvil, el juego ABC Descubre consta de la sopa de letras, las casillas donde se ubican cada letra para formar la palabra, una bocina donde se presenta el sonido del animal que deben escribir, el cronometro y finalmente el botón de avance para el siguiente nivel.

La **Fig. 18**, **Fig. 19** y **Fig. 20** muestran las funcionalidades completas de esta implementación.

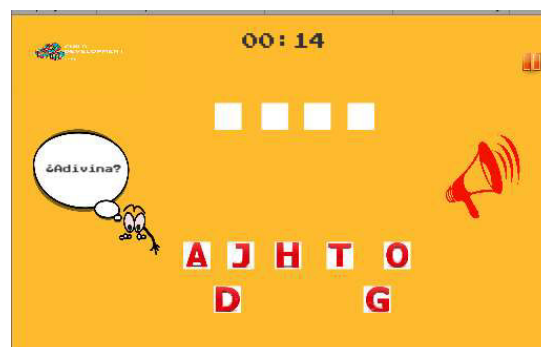


Fig. 18: Implementación juego ABC Descubre.



Fig. 19: Juego completo y pasar al siguiente nivel.



Fig. 20: Juego finalizado sin cumplir objetivo.

Implementación juego de rompecabezas

Esta tarea cumple directamente con una de las principales funcionalidades de la aplicación móvil, el rompecabezas es un juego de habilidad y paciencia que consiste en recomponer una imagen combinando de manera correcta unas determinadas piezas. La **Fig. 21** muestra la interfaz principal de este juego. La **Fig. 22** muestra la interfaz al completar uno de los niveles de este juego. Finalmente, la **Fig. 23** muestra la interfaz al terminarse el tiempo de resolución de uno de los rompecabezas.

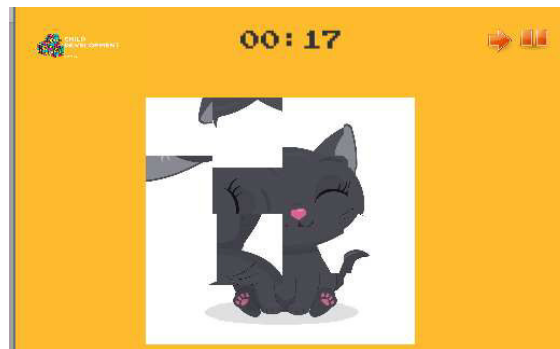


Fig. 21: Interfaz principal del juego de rompecabezas.

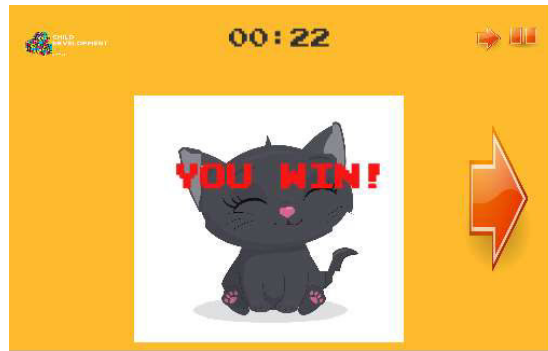


Fig. 22: Interfaz al momento de cumplir con el objetivo.



Fig. 23: Interfaz con tiempo terminado.

Efectos de sonido

Al ser un juego, se requiere que las diferentes escenas tengan efecto de sonido al momento de interactuar con la pantalla, ganar una partida o al perder la misma. El manejo de sonido se hace mediante objetos propios del juego como lo es el *Audio Source*, y con la ayuda de algunos *Scripts*, que complementan esta tarea tanto para que el usuario tenga efecto de sonido por la interacción con la aplicación móvil, así mismo como música de escenas, desde que inicia el juego. En la **Fig. 24** muestra el *gameObject* que posee el audio de la escena.



Fig. 24: *AudioScene*.

Implementación de efecto de sonido *Game Over*

Para esta tarea se requiere un efecto de sonido para el momento en el que el prescolar no logra cumplir el objetivo de pasar de nivel, es así que este sonido se presenta al momento de que el tiempo se agote, indicando al prescolar que debe volver a intentarlo. En la **Fig. 25** se muestra el sonido ya implementado en la interfaz de Unity.



Fig. 25: Game Over.

Implementación efecto de sonido al cumplir un objetivo

Para esta tarea en cambio se requiere que el juego tenga un efecto de sonido al momento en el que el prescolar logre cumplir los objetivos del nivel avanzado, indicándole que puede ir por el siguiente nivel. En la **Fig. 26** se indica el audio usado en la implementación de esta tarea ya en Unity.

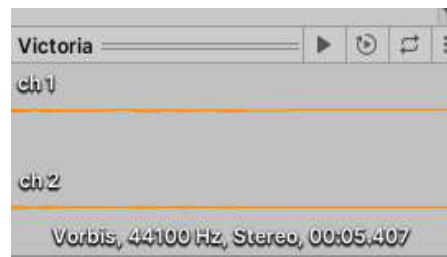


Fig. 26: Audio Victoria.

Implementación del juego TocaToca

Esta tarea cumple con una de las funcionalidades propias de la aplicación móvil, siendo este el tercer juego de la colección de juegos que se presentan en la aplicación móvil.

Este juego consiste en que el prescolar debe atrapar los ratoncitos que aparecen en pantalla, siendo de esta forma un método entretenido para que el prescolar logre ejercitar su coordinación ojo vista, de esta manera el objetivo se cumple cuando el prescolar logra atrapar a todos los ratoncitos en un tiempo determinado. En la **Fig. 27** se muestra la interfaz principal. En la **Fig. 28** se muestra la interfaz al momento de no lograr cumplir el objetivo. Finalmente, en la **Fig. 29** muestra la Interfaz que se presenta al momento de que el prescolar logra cumplir con el objetivo.



Fig. 27: Interfaz principal juego TocaToca.



Fig. 28: Interfaz *GameOver* juego TocaToca.



Fig. 29: Interfaz Victoria juego TocaToca.

3.4 *Sprint 3* Pruebas

En base a las tareas descritas en el *Sprint 3*, los resultados obtenidos son:

- Pruebas Unitarias.
- Generación de la APK.
- Prueba de compatibilidad.
- Pruebas de aceptación.

Pruebas Unitarias

Las pruebas unitarias consisten en probar un fragmento de código para corroborar su correcto funcionamiento, para este caso se han realizado este tipo de pruebas en base a cada escena de videojuego ya que ahí se representa cada funcionalidad de la colección de juegos.

Las pruebas unitarias de cada uno de los módulos de las funcionalidades directas se las realiza mediante las herramientas propias de Unity, con el paquete de extensión de Unity Automated QA, mismo que permite realizar pruebas mediante el uso de la aplicación móvil, ejecutándola de la manera en la que esta funcionaria. En la **Fig. 30** se muestra el paquete que permite generar estas pruebas, además de eso al finalizar la creación de la prueba por escena se corre el proyecto en modo de prueba con la herramienta Test Runner, en la pestaña *Play Mode*, que permite evaluar las funcionalidades de un juego, como se muestra en la **Fig. 31**.

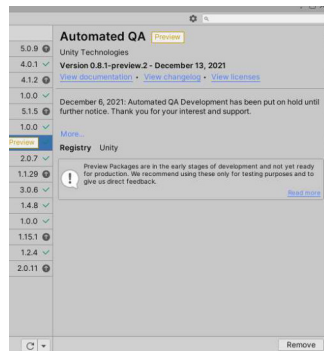


Fig. 30: Paquete Automated QA propio de Unity.

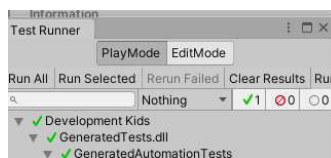


Fig. 31: Test Runner.

En el orden que las escenas van tomando vida en el videojuego se realizan las respectivas pruebas, en la **Fig. 32**, se muestra el resultado de la escena 1, el resto de las escenas se detalla en el **ANEXO II - Pruebas**.

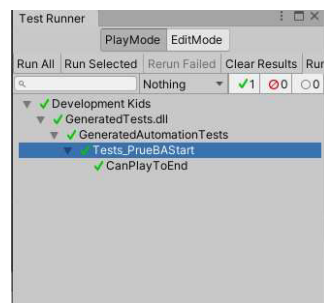


Fig. 32: Resultado prueba escena 1.

Una vez terminadas con las pruebas unitarias se concluye que las escenas no presentan fallos a nivel de código y funcionalidad directa de los minijuegos de la aplicación móvil.

Generación de la APK

Esta tarea consiste en realizar la APK y Bundle, para la instalación en los diferentes dispositivos Android, antes de realizar la puesta en producción, es importante tener la APK, y probarla en diferentes dispositivos, por lo que esta tarea es de suma importancia antes de llevar a cabo la siguiente tarea de este *sprint*.

En la **Fig. 33** se muestra la creación de la APK, en el *Build Settings* de Unity.

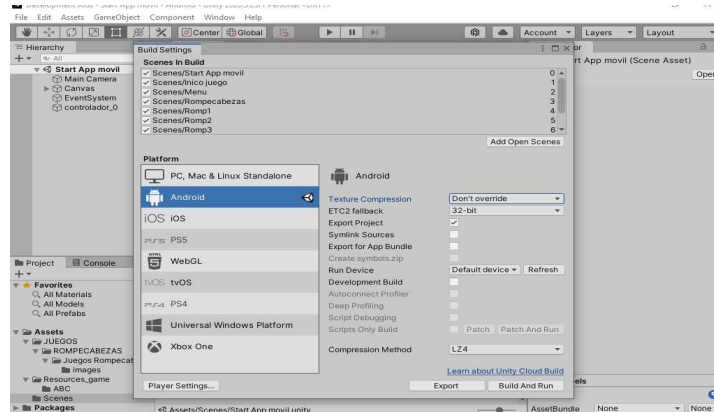


Fig. 33: Generación APK.

La **Fig. 34** muestra el uso de la opción de *build settings*, donde se puede configurar la exportación del proyecto de esta forma se genera un proyecto de Android por la nueva configuración de Unity.

Unity proporciona una herramienta para poder exportar el proyecto como ya se ha mencionado en la **Fig. 35**, y dentro de esta se tiene la interfaz de *Player Settings*, donde se debe hacer algunas configuraciones más para poder exportar el proyecto correctamente.

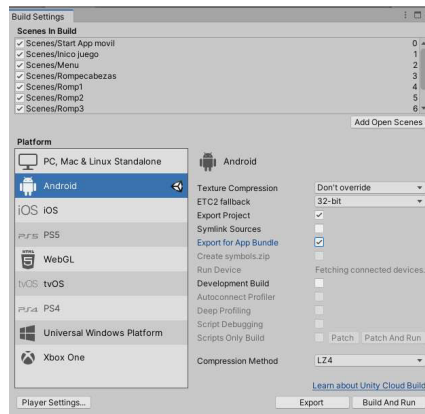


Fig. 34: Opción Build Settings.

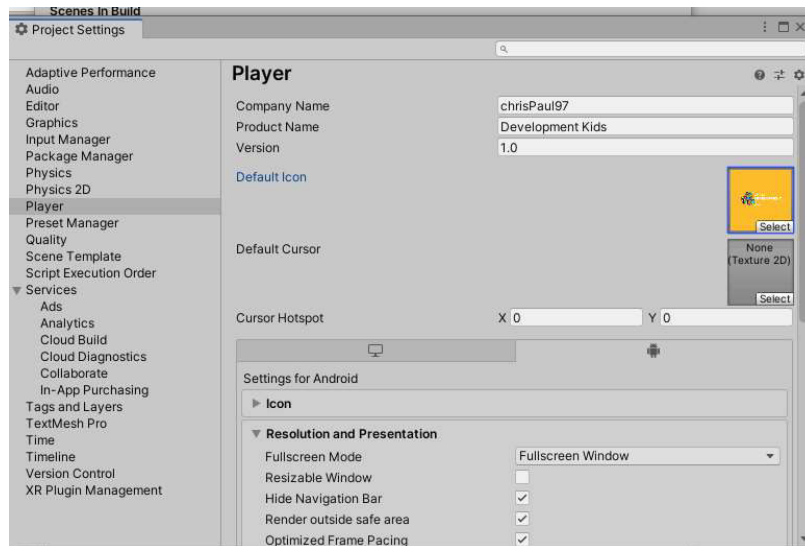


Fig. 35: Configuración icono y nombre.

En la misma interfaz se realiza la configuración de la presentación de la aplicación móvil, la misma que se configura para que siempre se presente de manera horizontal en el dispositivo móvil, ya sea rotativa, pero de forma horizontal con la propiedad que tiene el nombre *landscape* como se muestra en la **Fig. 36**.

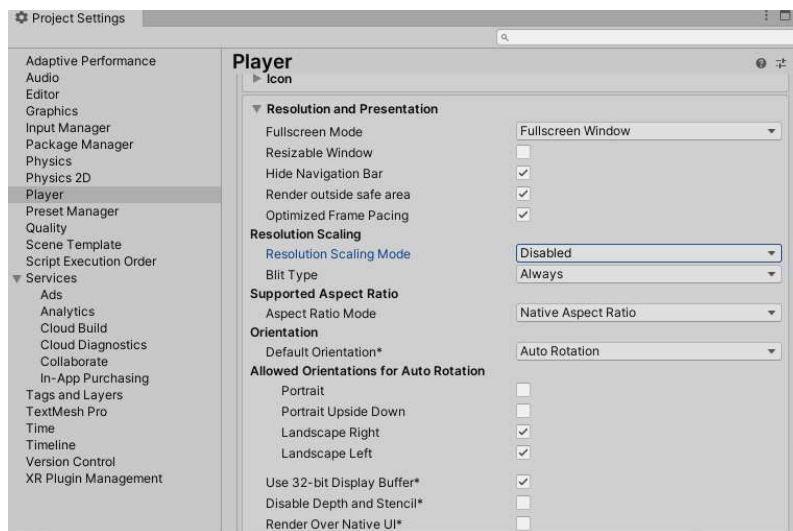


Fig. 36: Configuración presentación de la aplicación móvil.

En *Other Settings* se realiza la configuración del nombre del paquete el cual será identificado en la *Play Store*, este mismo se asigna de forma automática con el nombre del proyecto, así también se realiza la configuración del nivel del API de Android al que se apunta, por política de Google este siempre debe ser el último nivel, esta parte se aprecia en la **Fig. 37**.

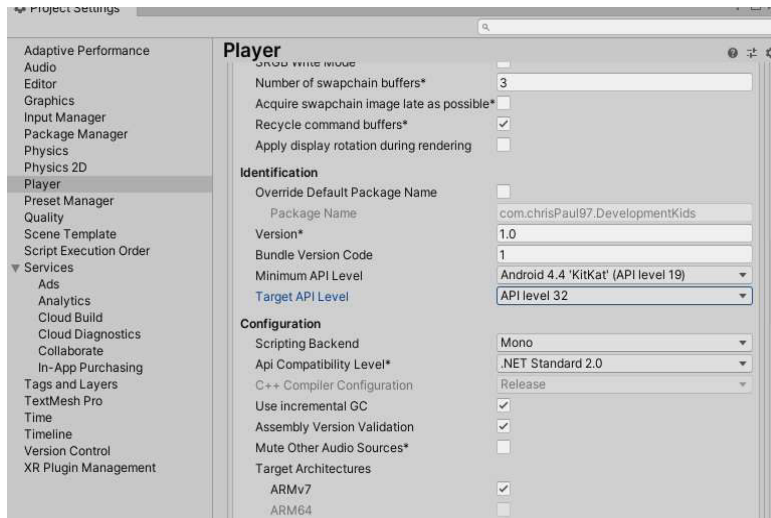


Fig. 37: Configuración Api y package.

En la misma opción de *Other Settings* se debe generar *Bundle's* para dos arquitecturas diferentes para que Google detecte automáticamente la arquitectura del teléfono que está descargando la aplicación móvil y se le asigne adecuadamente el *Bundle* específico para dicho dispositivo móvil, la sección de las arquitecturas se muestra en la **Fig. 38**.

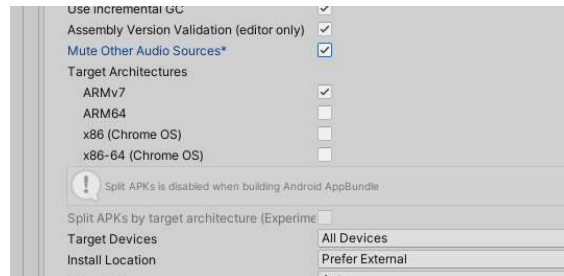


Fig. 38: Sección de arquitecturas.

Para terminar la configuración, al momento de publicar en la opción *Publishing Settings* se tiene que realizar la firma de la aplicación donde Google identificará el propietario de la aplicación y se pueda subir actualizaciones de la aplicación móvil, la colocación y creación de la firma se puede ver en la **Fig. 39** y **Fig. 40**.

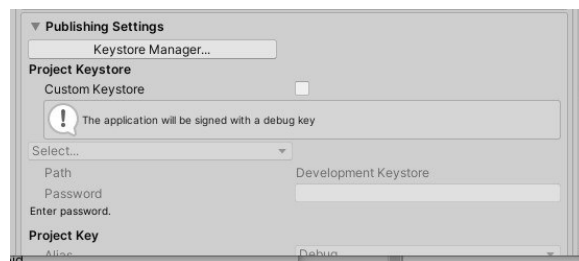


Fig. 39: Ubicación firma.

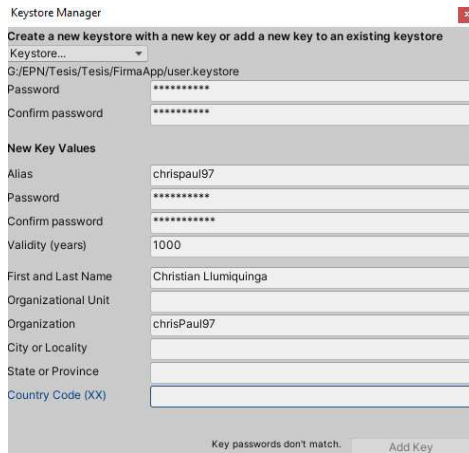


Fig. 40: Generación de la firma.

Una vez completado todas las configuraciones se procede a exportar el proyecto para crear el ejecutable, como se muestra en la **Fig. 41**. Una vez terminado este proceso se genera una carpeta con los archivos de Android, donde se procede a crear el APK y *Bundle*, mediante *Android Studio*, en la pestaña *Build*, se presenta las opciones como se muestra en la **Fig. 42**.

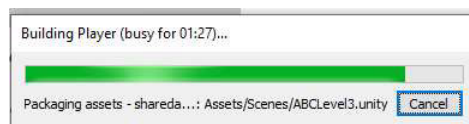


Fig. 41: Proceso de exportación del juego.

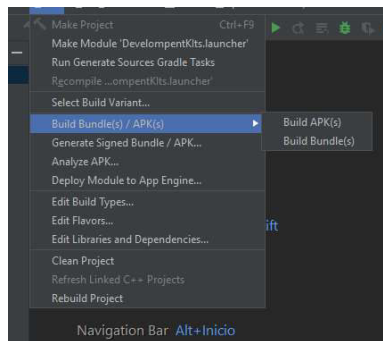


Fig. 42: APK y Bundle en Android estudio.

A continuación, se muestra el *Bundle* y la APK de la aplicación móvil ya generada por *Android Studio* (**Fig. 43**).

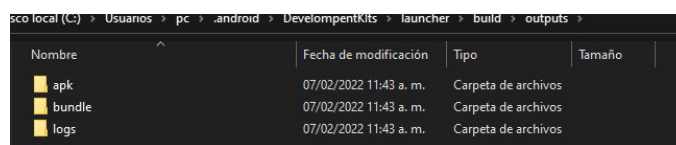


Fig. 43: Carpetas de APK y Bundle.

Finalmente, para comprobar que la APK se ha generado de forma exitosa la **Fig. 44** muestra la aplicación móvil ya instalada en un dispositivo móvil.

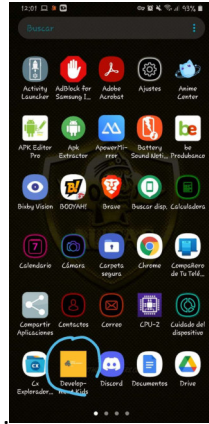


Fig. 44: Apk instalada en dispositivo móvil.

Resultados de las pruebas de compatibilidad

La aplicación móvil con una colección de juegos para ayudar en la mejora de la motricidad fina de niños y niñas de edad preescolar puede ser usado en dispositivos Android con una versión de Android mínima 4.4 hasta una versión máxima de Android de API nivel 32.

A continuación, en la **TABLA IV** se presenta una lista de algunos de los dispositivos en los cuales se ha probado la aplicación móvil.

TABLA IV: Lista de dispositivos donde se ha probado la aplicación móvil.

Dispositivos	Características.
Samsung A7 2018	Android:10, arquitectura ARM64, procesador Exynos 7885
Xiaomi redmi note 8.	Android:11, arquitectura ARM64, Procesador Snapdragon 665.
Xiaomi redmi 9	Android:10, arquitectura ARM64, Procesador Helio G8 MediaTeck
Samsung A20	Android: 9, arquitectura ARM64, Procesador Exynos 7884.

En conclusión, en todos los dispositivos mencionados la aplicación móvil, no presenta errores y ni fallas por lo cual se procede a la siguiente fase de prueba.

Las evidencias de las pruebas de compatibilidad en los ya mencionados dispositivos se pueden evidenciar en el **ANEXO II - Pruebas**.

Resultados de las pruebas de aceptación

Las pruebas de aceptación tienen el objetivo de validar que un sistema cumple con el funcionamiento esperado, y que así el cliente acepte el producto de *software* basado en su funcionamiento, cumpliendo con lo solicitado al inicio del proyecto [35].

En la **TABLA V: Prueba de aceptación 1: Menú juegos**, se presenta la primera prueba de aceptación que ha sido desarrollada de acuerdo con los requerimientos funcionales que se han solicitado al inicio del proyecto, mientras que en el **ANEXO II - Pruebas** de este documento se detalla las demás pruebas.

TABLA V: Prueba de aceptación 1: Menú juegos.

Prueba de Aceptación	
Identificador (ID): PA01	Identificador de requerimiento: RR001
Nombre: Menú de juegos.	
Descripción: Como usuario de la aplicación este deberá ver un menú con los juegos disponibles de la aplicación móvil, el cual puede acceder a cualquiera de estos.	
Pasos de ejecución: <ul style="list-style-type: none"> • Abrir la aplicación móvil. • Iniciar la aplicación móvil. • Dar clic en el botón jugar. • Se presenta el menú con los diferentes juegos. 	
Resultado deseado: El usuario podrá ver un menú con los tres tipos de juegos indicados en los requerimientos funcionales.	
Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.	

Una vez que estas pruebas de aceptación se han concluido y en base a los resultados se determina que hay una aceptación total en el aspecto de la funcionalidad de la aplicación móvil, directamente con lo que tiene que ver con la funcionalidad de los juegos.

3.5 Sprint 4 Despliegue

El resultado de este *sprint* es:

- Publicación en *Play Store*.

Publicación en Play Store

Una vez terminado el proceso de pruebas es hora de que la aplicación pase a producción, en este caso la aplicación es desplegada en el servicio de Google *Play Store*, siendo así disponible para todos los dispositivos Android. Desde la **Fig. 45** hasta la **Fig. 51** se puede observar este proceso.

Primero, en Google Play Console se crea una nueva aplicación la cual tiene el nombre de la aplicación móvil que se presenta como “Development Kids” (**Fig. 45**).

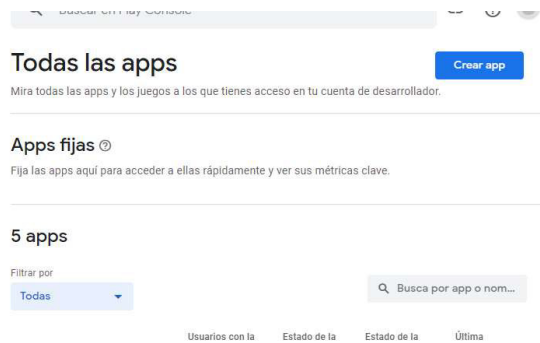


Fig. 45: Creación de una nueva aplicación.

Seguido, se llena la ficha de Play Store que es lo que se presenta en la Tienda de Google *Play Store* (**Fig. 46**).

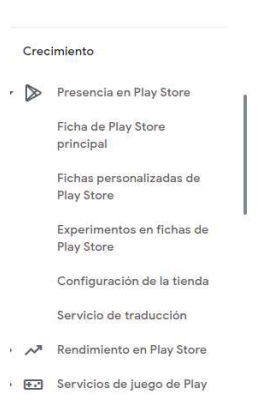


Fig. 46: Ficha de *Play Store*.

La ficha se llena con información básica de la aplicación para que pueda ser entendida por los usuarios (**Fig. 47**).

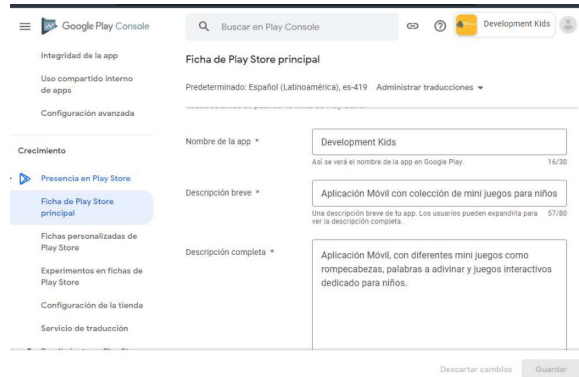


Fig. 47: Ficha llena con información básica.

Es momento de agregar el icono de la aplicación móvil, que es una imagen representativa en este caso del videojuego (**Fig. 48**Fig. 48), además de una serie de imágenes que muestran de forma breve la intención del videojuego (**Fig. 49**).

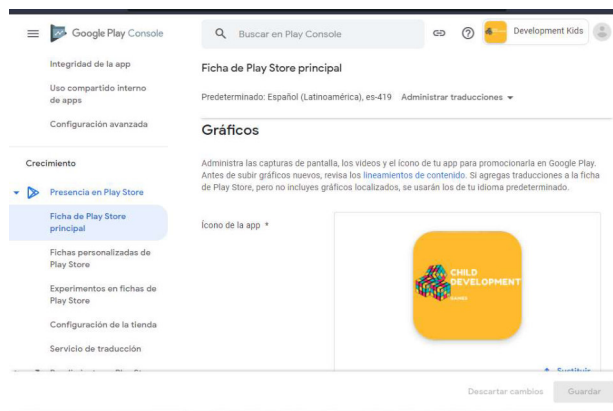


Fig. 48: Icono de la Aplicación móvil.

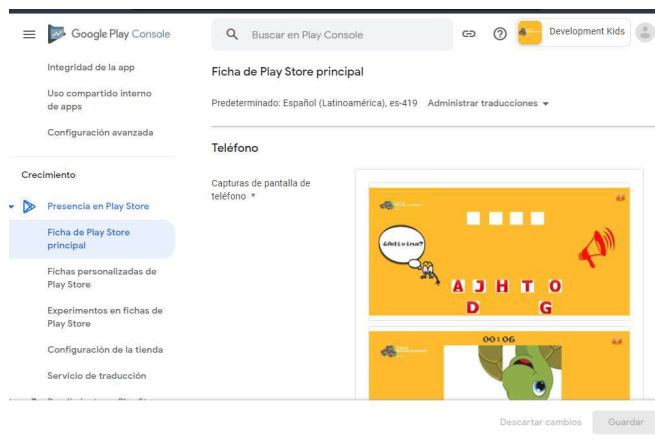


Fig. 49: Imágenes de la Aplicación Móvil.

Una vez creada la versión de producción se debe agregar el archivo Bundle (**Fig. 50**) y finalmente es hora de esperar que se apruebe la revisión para proceder a publicarse en *Play Store* (**Fig. 51**).

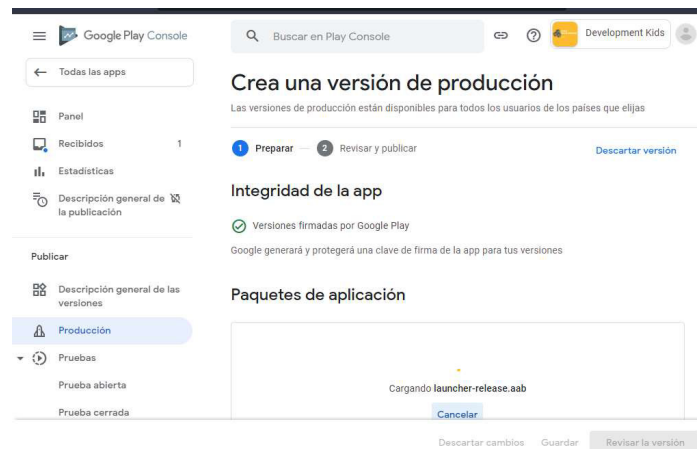


Fig. 50: Agregar el archivo Bundle.

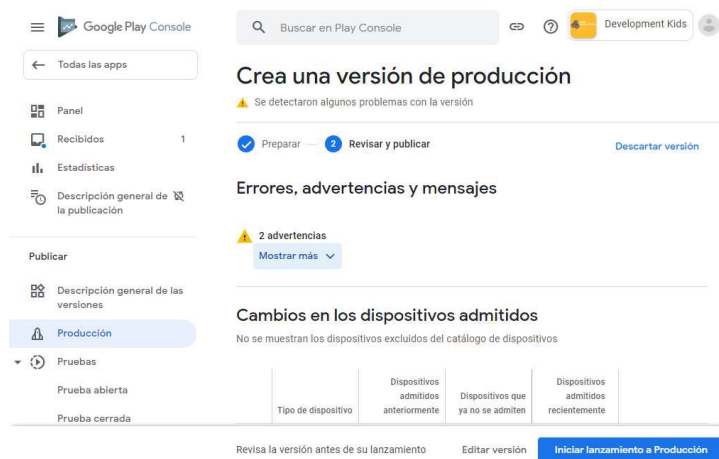


Fig. 51: Revisión para proceder a publicarse en *Play Store*.

4 CONCLUSIONES

Al finalizar el proyecto se ha propuesto conclusiones referentes al manejo de la herramienta Unity, y sus temas de aprendizaje como lo es el manejo de la herramienta en general y los objetivos planteados en un inicio con el proyecto, a continuación, se presenta las conclusiones:

- Se ha implementación de un videojuego con tres juegos que ayudan en la motricidad fina de preescolares, desarrollado enteramente en Unity 3D, herramienta que ha permitido ir alcanzado los objetivos planteados desde el inicio del proyecto.
- El que la metodología *Sum* sea una variante de *Scrum* ha permitido que el equipo se acople a la misma de manera fácil y eficiente, por lo que se la considera una metodología apta para futuros proyectos de esta misma temática de desarrollo de videojuegos.
- La arquitectura en Unity, ha permitido utilizar los conocimientos de MVC y poder variarlos de acuerdo a las necesidades. En general la arquitectura de datos y el manejo de los mismos cambia, siendo el factor principal el cambio la lógica del negocio, por lo que el tener una arquitectura de referencia aporta significativamente al ser adaptada a lo que demanda el proyecto.
- Las pruebas Unitarias en cada uno de estos módulos han permitido detectar posibles fallas de código en el juego, logrando solucionarlas a tiempo y que a la larga no se generen *bugs*; con lo que se asegura la entrega de un producto de calidad al mercado.
- Además del aprendizaje de una nueva herramienta como lo es Unity, también se ha realizado el aprendizaje del lenguaje de programación C#, el cual aplicando los fundamentos de la carrera se ha logrado obtener una buena retroalimentación acerca de este nuevo lenguaje para el desarrollador.

5 RECOMENDACIONES

En esta sección se presentan las recomendaciones acerca de este proyecto:

- La herramienta Unity, permite crear aplicaciones complejas, pero es necesario tener un tiempo prudente para el aprendizaje de la misma y todas sus funciones, por lo que se recomienda que en la planificación de proyectos se tome en cuenta dicho tiempo.
- Se recomienda seguir trabajando en futuras actualizaciones, como nuevos niveles y nuevos desafíos.
- La aplicación móvil ha sido probada en dispositivos funcionales, por lo que es recomendable trabajar en el diseño adaptativo a diferentes tipos de pantallas.
- Para la fase de elaboración, es recomendable llevar una buena lógica de programación, para que en producción no se generen bugs, que luego pueden ser mucho más difícil de encontrar y corregir.
- Esta aplicación también puede ser pensada de forma *online* para la utilización de diferentes servicios en la nube, siempre y cuando se tome la lógica correcta y lo que se quiere con este proyecto.
- Como recomendación final es importante el trabajo organizado y el correcto procedimiento de la metodología *Sum* para alcanzar los tiempos estimados.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] Ser, Espacios de, 20 noviembre 2020. [En línea]. Available: <https://www.robotica.com.py/que-es-motricidad-fina-en-que-ayuda-al-nino-o-nina/>. [Último acceso: 21 Diciembre 2020].
- [2] M. E. Vilaro, «Padres Hoy,» 27 Marzo 2017. [En línea]. Available: <https://www.elobservador.com.uy/nota/la-escritura-en-el-nino-preescolar-2017327500>. [Último acceso: 9 Diciembre 2020].
- [3] M. G. H. y. L. M. B. García, «El desarrollo de la motricidad fina en los niños y las niñas de la primera infancia,» *Revista Atlante: Cuadernos de Educación y Desarrollo*, vol. 2, p. 11, 2018.
- [4] C. M. M. - S. C. Saéz, «DESARROLLO FÍSICO Y PSICOMOTOR EN LA ETAPA IFANTIL,» Avda. Tolosa 70, San Sebastian.
- [5] P. M. P. Tumipamba, Artist, *La motrocidad fina en el aprendizaje de la pre-escritura en los niños y niñas*. [Art]. UNIVERSIDAD CENTRAL DEL ECUADOR, 2017.
- [6] M. Matilde, «Eres Mamá,» 30 Abril 2019. [En línea]. Available: <https://eresmama.com/la-importancia-de-las-tic-en-preescolar/>. [Último acceso: 21 Diciembre 2020].
- [7] M. M. G. G. V. Camacho Alvarez, «PRINCIPIOS PARA LA INCORPORACION DE LA TECNOLOGIA EN ESPACIOS ESCOLARES,» *InterSedes: Revista de las Sedes Regionales*, vol. IX, nº 17, pp. 47-50, 2008.
- [8] R. Chatel, «Los Equipos que se usan en prescolar,» *Nueva Inglaterra Reading Associaton Journal*, vol. II, nº 41, pp. 49-52, 2005.
- [9] «UNIR: La Universidad en Internet,» 28 Enero 2020. [En línea]. Available: <https://www.unir.net/educacion/revista/tic-educacion-infantil/#:~:text=Las%20TICS%20en%20Infantil%20deben,cada%20alumno%20y%20sus%20necesidades>. [Último acceso: 21 Diciembre 2020].
- [10] BynaryLabs, «AppStore,» [En línea]. Available: <https://apps.apple.com/es/app/dexterity/id420464455>. [Último acceso: 27 Febrero 2021].
- [11] android developers, «developers,» 25 Mayo 2020. [En línea]. Available: <https://developer.android.com/training/gestures/multi?hl=es-419>.
- [12] R. S. Pressman, *Ingenieria de software un enfoque practico*, México: McGraw-Hill, 2010.
- [13] t. Technology, 16 Octubre 2018. [En línea]. Available: <https://www.tithink.com/es/2018/10/16/metodologias-agiles-que-son-y-para-que-sirven/>. [Último acceso: 23 Febrero 2021].
- [14] T. C.-R. Balet, «SINNAPS,» 2020. [En línea]. Available: <https://www.sinnaps.com/blog-gestion-proyectos/metodologia-scrum>. [Último acceso: 21 Enero 2021].

- [15 C. L. Reventos, «El videojuego como herramienta educativa. Posibilidades y problemáticas acerca los serious games,» *apertura*, vol. 8, nº 1, p. 3, 2016.
- [16 G. A. Y. Murcia, Artist, *USO DEL SOFTWARE COMO HERRAMIENTA DE APRENDIZAJE EN NIÑOS DEL*. [Art]. UNIVERSIDAD DEL TOLIMA, 2015.
- [17 G. C. Chinguel, «RecetasTic,» 16 Abril 2016. [En línea]. Available:
] <https://www.comunidad.recetastic.com/15-aplicaciones-para-desarrollar-la-motricidad-fina-en-los-ninos/>. [Último acceso: 19 febrero 2021].
- [18 L. Herazo, «Anincubator,» [En línea]. Available: <https://anincubator.com/que-es-una-aplicacion-movil/>. [Último acceso: 27 12 2021].
- [19 T. A. Baeza, Artist, *Desarrollo de un videojuego para móviles con Unity*. [Art]. Escuela Politecnica Superior, 2015.
- [20 D. E. García, «OpenWebinars,» 10 Junio 2019. [En línea]. Available:
] <https://openwebinars.net/blog/que-es-unity/>. [Último acceso: 11 Noviembre 2021].
- [21 «Importancia,» [En línea]. Available: <https://www.importancia.org/metodologia.php>.
] [Último acceso: 28 Diciembre 2021].
- [22 I. R. Salvador, «Psicología y Mente,» [En línea]. Available:
] <https://psicologiaymente.com/psicologia/estudio-de-caso>. [Último acceso: 30 Diciembre 2021].
- [23 A.-L. G.-R. A.-W. I.-I. J.-A. Q.-P. G. S.-A. F.-R. D.-P. Xavier-Alexis MURILLO-SANCHEZ, Artist, *Implementación de la metodología SUM modificada para el*. [Art]. Universidad Católica Boliviana “San Pablo”, 2018.
- [24 A. C. G. M. A. V. Nicolás Acerenza, Artist, *Una Metodología para Desarrollo de Videojuegos*. [Art]. Universidad de la República, Uruguay, 2009.
- [25 R. Buergo, «Gdp Master,» 23 Junio 2021. [En línea]. Available:
] <https://www.gestiondeproyectos-master.com/roles-y-responsabilidades-en-un-proyecto-scrum/>. [Último acceso: 15 Noviembre 2021].
- [26 VIEWNEXT, «VIEWNEXT,» 27 Noviembre 2019. [En línea]. Available:
] <https://www.viewnext.com/artefactos-scrum/>. [Último acceso: 15 Noviembre 2021].
- [27 M. T. Gallego, Artist, *Gestión de Proyectos Informáticos Metodología Scrum*. [Art]. Universidad Europea Online.
- [28 Scrum Manager, «Scrum Manager,» 26 Abril 2021. [En línea]. Available:
] https://www.scrummanager.net/bok/index.php/Historia_de_usuario. [Último acceso: 15 Noviembre 2021].
- [29 E. Rodríguez, «seoestudios,» 21 Enero 2021. [En línea]. Available:
] <https://www.seoestudios.es/blog/que-es-un-mockup/>. [Último acceso: 6 Enero 2022].

- [30 K. Bracey, «evantotuts,» 26 Noviembre 2018. [En línea]. Available:
] <https://webdesign.tutsplus.com/es/articulos/what-is-figma--cms-32272>. [Último acceso: 6 Enero 2020].
- [31 J. A. D. F. A. A. P. C. Andy Hernández Paez, Artist, *Arquitectura de software para el desarrollo de videojuegos en el motor de juegos unity 3d*. [Art]. Universidad de las ciencias informaticas Vertex, 2018.
- [32 Unity, «Unity,» [En línea]. Available: <https://unity3d.com/es/get-unity/download>.
]
- [33 Microsoft, «Microsoft,» Enero 2022. [En línea]. Available:
] <https://visualstudio.microsoft.com/es/vs/unity-tools/>. [Último acceso: 20 Enero 2022].
- [34 U. Documents, «Unity,» [En línea]. Available:
] <https://docs.unity3d.com/Manual/index.html>.
- [35 M. Cillero, «Manuel.Ciller.es,» [En línea]. Available:
] <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/pruebas/acceptacion/>.
- [36 L. Carvajal, Metodología de la Investigación Científica. Curso general y aplicado, 28 ed.,
] Santiago de Cali: U.S.C., 2006, p. 139.
- [37 L. Castillo, «IMPORTANCIA DE LAS TECNOLOGIAS EN LA EDUCACIÓN INICIAL,» 2 Junio
] 2013. [En línea]. Available:
<http://tecnologiasaplicadaslaurimar.blogspot.com/2013/06/la-importancia-del-uso-de-las-tics-en.html>. [Último acceso: 23 Febrero 2021].
- [38 pandorafms_blogs, «pandorafms,» 18 Noviembre 2015. [En línea]. Available:
] <https://pandorafms.com/blog/es/nosql-vs-sql-diferencias-y-cuando-elegir-cada-una/>.

7 ANEXOS

En este apartado, se presentan los anexos que son información relevante y que transparenta la elaboración del presente proyecto de titulación.

- **ANEXO I.** Certificado de Originalidad
- **ANEXO II.** Manual Técnico
- **ANEXO III.** Manual de Usuario
- **ANEXO IV.** Manual de Instalación

ANEXO I. Certificado de Originalidad

A continuación, se presenta el certificado de originalidad emitido por la directora del proyecto.



ESCUELA POLITECNICA NACIONAL
ESCUELA DE FORMACIÓN DE TECNÓLOGOS

CERTIFICADO DE ORIGINALIDAD

Quito, D.M. 22 de febrero de 2022

De mi consideración:

Yo, IVONNE FERNANDA MALDONADO SOLIZ, en calidad de Director del Trabajo de Integración Curricular titulado **DESARROLLO DE APLICACIÓN PARA FORTALECER LA MOTRICIDAD FINA EN PRESCOLARES MEDIANTE JUEGOS INTERACTIVOS** elaborado por el estudiante **CHRISTIAN PAUL LLUMIQUINGA AYALA** de la carrera en **TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE**, certifico que he empleado la herramienta Turnitin para la revisión de originalidad del documento escrito completo (sin anexos), producto del Trabajo de Integración Curricular indicado.

El documento escrito tiene un índice de similitud del 7%.

Es todo cuanto puedo certificar en honor a la verdad, pudiendo el interesado hacer uso del presente documento para los trámites de titulación.

NOTA: Se adjunta el informe generado por la herramienta Turnitin.

Atentamente,

Ivonne Maldonado

Técnico Docente

ESFOT

INFORME DE ORIGINALIDAD

7%

INDICE DE SIMILITUD

5%

FUENTES DE INTERNET

0%

PUBLICACIONES

4%

TRABAJOS DEL
ESTUDIANTE

ANEXO II. Manual Técnico

1. Recopilación de Requerimientos

En la **TABLA VI**, se listan los requerimientos de usuario y en la **TABLA VII** se listan los requerimientos no funcionales, que se ha recopilado mediante reuniones con la Parvularia experta en el tema de motricidad fina en niños menores de 6 años, lo que ha servido de guía para el inicio del proyecto.

TABLA VI – Recopilación de requerimientos funcionales.

Recopilación de Requerimientos funcionales de la aplicación móvil	
ID-RR	Detalle
RR001	Como usuario de la aplicación móvil al arrancar la misma se le presentara una pantalla de inicio con tarjetas indicando los juegos disponibles a probar.
RR002	Como usuario de la aplicación móvil al arrancar la misma deberá escoger entre los diferentes juegos que se presenten.
RR003	Dentro de la aplicación móvil el usuario podrá ingresar a otro juego el que será un rompecabezas.
RR004	Dentro de la aplicación móvil el usuario podrá ingresar a un juego el que contará con una bocina y en base al sonido del animal, deberá adivinar el animal al que pertenece el sonido y completar su nombre con la sopa de letras.
RR005	Cada juego de la aplicación móvil deberá contar con música de ambiente por juego.
RR006	Cada juego de la aplicación móvil puede ser terminado en cualquier momento por el usuario.
RR007	Cada juego de la aplicación móvil tendrá un tiempo límite para culminar con el mismo.
RR008	Al finalizar el tiempo de cada juego de la aplicación móvil se deberá presentar un sonido de game over, para que el usuario entienda que perdió.
RR009	Por cada juego de la aplicación móvil culminada con éxito se deberá emitir un sonido de felicitaciones para que el usuario se dé cuenta de lo que está pasando.
RR010	Al finalizar el juego de la aplicación móvil sea cual sea este regresará al menú principal, donde podrá escoger un juego diferente.
RR011	La aplicación móvil junto con los juegos que contendrá será atractiva para el usuario en el aspecto visual.

TABLA VII– Recopilación de requerimientos no funcionales

Recopilación de Requerimientos no funcionales de la aplicación móvil	
ID-RR	Detalle
RR001	Desarrollar la aplicación móvil tipo juego con la herramienta Unity y usar todas las herramientas que esta nos brinda para el desarrollo de juegos en 2d y la iteración con el dispositivo.
RR002	Diseñar las escenas de la aplicación móvil, como lo será el menú principal y la interfaz de los juegos descritos.
RR003	Diseño de la aplicación móvil atractiva para niños con imágenes y efectos de sonido.
RR004	Diseño del menú del rompecabezas.
RR005	Diseño de la interfaz del juego que permite iteración con la pantalla.
RR006	Diseño de la interfaz del juego de completar palabras.
RR007	Programación de los diferentes juegos de la aplicación móvil acordes con los requerimientos funcionales.
RR008	Utilización de los efectos de sonido al momento de ganar una partida, y la música ambiente que tendrán los juegos.
RR009	Programación optima y reutilizable de los componentes de los juegos que componente la aplicación móvil.
RR010	Pruebas de compatibilidad en dispositivos Android.
RR011	Depuración de errores.
RR012	Desarrollo de la aplicación móvil con enfoque a un posible crecimiento.
RR013	Publicación en <i>Play Store</i> .

2. Historias de Usuario

A continuación, desde la **TABLA VIII** hasta la **TABLA XXIV**; Error! No se encuentra el origen de la referencia. se presenta las historias de usuarios basadas en los requerimientos no funcionales de la aplicación móvil que dan la pauta de las funcionalidades del proyecto.

TABLA VIII: Historia de Usuario 2 – Conceptualizar los niveles de dificultad en los diferentes juegos de la aplicación móvil.

Historia de Usuario	
Identificador: HU2	Usuario: Desarrollador.
Nombre de la Historia: Conceptualizar los niveles de dificultad en los diferentes juegos de la aplicación móvil.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Baja
Días Estimados: 4	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: El desarrollador creara los diferentes niveles de los juegos de la aplicación móvil, teniendo en cuenta la dificultad adecuada para un niño, y no causar frustración en el infante.	
Observaciones:	

Una vez que los niveles de los juegos de la aplicación móvil estén conceptualizados estarán listos para el desarrollo.

TABLA IX: Historia de usuario 3: Conceptualizar las mecánicas del juego.

Historia de Usuario	
Identificador: HU3	Usuario: Desarrollador.
Nombre de la Historia: Conceptualizar las mecánicas del juego	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 3	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: Conceptualización la temática, diseño y mecánicas de los juegos a ser implementados.	
Observaciones: Las mecánicas de los juegos deben ser conceptualizadas correctamente para su posterior implementación.	

TABLA X: Historia de usuario 4 – Crear la interfaz principal de la aplicación móvil.

Historia de Usuario	
Identificador: HU4	Usuario: Desarrollador.
Nombre de la Historia: Crear de la interfaz principal de la aplicación móvil	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 2	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: Diseño y creación de la interfaz de la aplicación móvil según los requerimientos recopilados debe ser agradable a la vista de los niños.	
Observaciones: La interfaz deberá ser diseñada en figma y luego creada como una escena en Unity.	

TABLA XI: Historia de Usuario 5 – Crear las escenas del menú y los juegos de la aplicación móvil.

Historia de Usuario	
Identificador: HU5	Usuario: Desarrollador.
Nombre de la Historia: Crear las escenas del menú y los juegos de la aplicación móvil.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 2	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: El desarrollador creara las escenas necesarias para los juegos y los niveles de la aplicación móvil.	

TABLA XII: Historia de usuario 6 – Implementar la interfaz de cada juego de la aplicación móvil con música de ambiente por juego.

Historia de Usuario	
Identificador: HU6	Usuario: Desarrollador.
Nombre de la Historia: Implementar la Interfaz de cada juego de la aplicación móvil con música de ambiente por juego	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 2	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: El desarrollador deberá crear las interfaces de los juegos con música ambiente por cada juego además al iniciar la aplicación móvil.	
Observaciones: Se deberá conceptualizar y escoger la música correcta que sea atractiva para los niños y acorde con los juegos y temática de la aplicación móvil.	

TABLA XIII: Historia de usuario 7 – Conceptualizar e implementar las imágenes para el juego del rompecabezas.

Historia de Usuario	
Identificador: HU7	Usuario: Desarrollador.
Nombre de la Historia: Conceptualizar e implementar las imágenes para el juego del rompecabezas.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Días Estimados: 3	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: El desarrollador deberá escoger las imágenes que irán en el rompecabezas el cual el niño deberá completar, y también realizar la implementación al juego.	
Observaciones: Con la implementación de las imágenes se estará completando uno de los juegos lo que deja pendiente una implementación del cronometro en la HU15.	

TABLA XIV: Historia de usuario 8 – Implementar la interfaz del juego de interacción con la pantalla táctil.

Historia de Usuario	
Identificador: HU8	Usuario: Desarrollador.
Nombre de la Historia: Implementar la interfaz del juego de interacción con la pantalla táctil.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 3	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador deberá diseñar e implementar el juego donde el niño debe interactuar con la pantalla táctil.	
Observaciones:	

La implementación de este juego de la aplicación móvil es de vital importancia para alcanzar con el objetivo y ser una herramienta para ayudar al desarrollo de la motricidad fina.

TABLA XV: Historia de usuario 9 – Conceptualizar las palabras a completar en el juego de completa el nombre.

Historia de Usuario	
Identificador: HU9	Usuario: Desarrollador.
Nombre de la Historia: Conceptualizar las palabras a completar en el juego de completa el nombre.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Bajo
Días Estimados: 1	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador debe idear las palabras correctas que serán implementadas en el juego de completa el nombre de la imagen, estas deben estar acorde con la edad de los niños a los que se les distribuirá la aplicación móvil.	
Observaciones: Las palabras junto con las imágenes de esta historia de usuario deben ser acordes a la edad de un niño entre 7 y 4 años.	

TABLA XVI: Historia de usuario 10 – Implementar el cronometro en el juego del rompecabezas.

Historia de Usuario	
Identificador: HU10	Usuario: Desarrollador.
Nombre de la Historia: Implementar el cronometro en el juego del rompecabezas.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 1	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador deberá implementar un cronometro dentro del juego del rompecabezas y así darle un tiempo límite para ser armado,	
Observaciones: El tiempo límite para completar el rompecabezas se dará acorde a la edad del niño para no generar frustración, según el nivel de dificultad.	

TABLA XVII: Historia de usuario 11 – Implementar efectos de sonido, diferente por cada juego.

Historia de Usuario	
Identificador: HU11	Usuario: Desarrollador.
Nombre de la Historia: Implementar efectos de sonido diferente por cada juego.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 1	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción:	

El desarrollador deberá escoger los audios correctos y realizar la implementación por cada juego.
Observaciones: Los audios para escoger deben cumplir con los requerimientos de usuario de que sean atractivos para niños.

TABLA XVIII: Historia de usuario 12 – Implementar efecto de sonido de game over cuando él no pueda completar un juego.

Historia de Usuario	
Identificador: HU12	Usuario: Desarrollador.
Nombre de la Historia: Implementar efecto de sonido de <i>game over</i> cuando el niño no pueda completar un juego	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Medio
Días Estimados: 1	Sprint Asignados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador debe implementar un efecto de sonido que le dé a entender al niño que ha fallado en el juego.	
Observaciones: Este efecto de sonido debe ser implementado en todos los juegos de la aplicación móvil.	

TABLA XIX: Historia de usuario 13: Implementar efecto de sonido de felicitaciones cuando el niño pueda completar un juego.

Historia de Usuario	
Identificador: HU13	Usuario: Desarrollador.
Nombre de la Historia: Implementar efecto de sonido de felicitaciones cuando el niño pueda completar un juego.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Media
Días Estimados: 1	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian	
Descripción: El desarrollador debe implementar un efecto de sonido que felicite al niño al momento de superar un nivel de los diferentes juegos de la aplicación móvil.	
Observaciones: Este efecto de sonido debe ser implementado en todos los juegos.	

TABLA XX: Historia de usuario 14 – Conceptualizar y cargar las imágenes del juego de completar la palabra en la aplicación móvil.

Historia de Usuario	
Identificador: HU14	Usuario: Desarrollador.
Nombre de la Historia: Conceptualizar y cargar las imágenes del juego de completar la palabra en la aplicación móvil.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 2	Sprint Agnados: 0

Programador Responsable: Llumiquinga Christian.
Descripción: El desarrollador debe escoger las imágenes correctas las cuales luego serán las que permitan al niño saber cómo deben completar las palabras.
Observaciones: Se debe tomar en cuenta la edad de los niños para escoger correctamente las palabras y las imágenes.

TABLA XXI: Historia de usuario 15 – Realizar Prueba beta de la aplicación móvil.

Historia de Usuario	
Identificador: HU15	Usuario: Desarrollador.
Nombre de la Historia: Realizar Prueba beta de la aplicación móvil.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 2	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador debe realizar las pruebas de correcto funcionamiento de la aplicación móvil con los juegos integrados.	
Observaciones: En caso de entrar errores se debe corregirlos, y volver a realizar las pruebas.	

TABLA XXII: Historia de usuario 16 – Realizar Prueba de compatibilidad en los dispositivos Android.

Historia de Usuario	
Identificador: HU16	Usuario: Desarrollador.
Nombre de la Historia: Realizar Prueba de compatibilidad en los dispositivos Android.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 1	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador debe comprobar y listar los dispositivos con los cuales será compatible la aplicación móvil.	
Observaciones: Se debe tomar en cuenta los dispositivos actuales, sobre todo.	

TABLA XXIII: Historia de usuario 17 – Generar la Apk para instalar en dispositivos Android.

Historia de Usuario	
Identificador: HU17	Usuario: Desarrollador.
Nombre de la Historia: Generar la Apk para instalar en dispositivos Android.	
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alta
Días Estimados: 1	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción:	

El desarrollador debe generar el instalador de la aplicación móvil, este caso el archivo apk desde la herramienta Unity.
Observaciones: Con la HU23 se debe listar los dispositivos donde puede ser instalado.

TABLA XXIV: Historia de usuario 18 – Publicar en play store.

Historia de Usuario	
Identificador: HU18	Usuario: Desarrollador.
Nombre de la Historia: Publicar en <i>Play Store</i> .	
Prioridad en Negocio: Media	Riesgo en Desarrollo: Alta
Días Estimados: 1	Sprint Agnados: 0
Programador Responsable: Llumiquinga Christian.	
Descripción: El desarrollador debe realizar la publicación en play store de la aplicación móvil, para que así esté disponible en su revisión.	
Observaciones: No hay observación.	

3. Product Backlog

En la **TABLA XXV** TABLA XXV– Product Backlog se muestra el *Product backlog* con respecto a las funcionalidades de la aplicación móvil con los juegos interactivos.

TABLA XXV– Product Backlog

Identificador	Historia	Iteración	Días estimados	Prioridad
HU1	Realizar la Autoeducación en la herramienta Unity.	0	7	Alta
HU2	Conceptualizar los niveles de dificultad en los diferentes juegos de la aplicación móvil.	0	4	Alta
HU3	Conceptualización de las mecánicas del juego.	0	3	Alta
HU4	Diseño y creación de la interfaz principal de la aplicación móvil.	1	2	Alta
HU5	Creación de las escenas del menú y los juegos de la aplicación móvil.	1	2	Alta
HU6	Interfaz de cada juego de la aplicación móvil con música de ambiente por juego.	1	2	Alta
HU7	Conceptualización e implementación de las imágenes para el juego del rompecabezas.	2	3	Alta
HU8	Diseño e implementación de la interfaz del juego de	2	2	Alta

	interacción con la pantalla táctil.			
HU9	Conceptualización de las palabras a completar en el juego de completa el nombre.	2	1	Alta
HU10	Implementación del cronometro en el juego del rompecabezas.	2	1	Alta
HU11	Efectos de sonido diferente por cada juego.	2	1	Media
HU12	Efecto de sonido de game over cuando el niño no pueda completar un juego.	2	1	Media
HU13	Efecto de sonido de felicitaciones cuando el niño pueda completar un juego.	2	1	Media
HU14	Conceptualización y carga de las imágenes del juego de completar la palabra en la aplicación móvil.	2	1	Alta
HU15	Prueba beta de la aplicación móvil.	3	2	Alta
HU16	Prueba de compatibilidad en los dispositivos Android.	3	2	Alta
HU17	Prueba de aceptación	3	3	Alta
HU17	Generación de la apk para instalar en dispositivos Android.	3	1	Alta
HU18	Publicación en <i>Play store</i> .	4	2	Media

4. Sprint Backlog

En la **Tabla XXVI** Error! No se encuentra el origen de la referencia. TABLA XXV– Product Backlog se muestra el *Sprint backlog* con cada una de las tareas a realizar para poder cumplir cada una de las Historias de Usuario.

Tabla XXVI – Sprint Backlog

Etapas de la metodología	ID	Nombre	Tareas	Duración en horas
Concepto y Planificación	0	Configuración del ambiente de desarrollo y Conceptualización del juego	<ul style="list-style-type: none"> • Instalación de Unity y sus Herramientas de desarrollo. • Configuración del ambiente de desarrollo. • Autoeducación en la herramienta de desarrollo. • Conceptualización de los niveles de dificultad de la aplicación móvil. • Conceptualización de las mecánicas de este. 	50
	1	Diseño	<ul style="list-style-type: none"> • Diseño de la interfaz del juego. • Creación de las diferentes escenas del juego. • Creación de los niveles de cada juego. 	40
Elaboración	2	Codificación de los minijuegos	<ul style="list-style-type: none"> • Conceptualización e implementación de imágenes para el rompecabezas. • Implementación del cronometro en los diferentes juegos de la aplicación móvil. • Efectos de sonido de la aplicación móvil. • Implementación efecto de sonido <i>Game Over</i> en los juegos de la aplicación móvil. • Implementación juego ABC Descubre. • Implementación juego de rompecabezas. • Diseño e implementación del juego de contacto atrápame. 	100

Beta	3	Pruebas	<ul style="list-style-type: none"> • Pruebas Unitarias. • Generación de la Apk. • Prueba de compatibilidad. • Pruebas de aceptación. 	30
Cierre	4	Despliegue y entrega	<ul style="list-style-type: none"> • Publicación en Play Store. • Documentación. 	20
Total, en horas				240

5. Diseño de interfaces

Aquí se muestra el diseño básico de cada una de las interfaces que vienen a ser las escenas del juego en Unity, indicados desde la **Fig. 52** hasta la **Fig. 63**.



Fig. 52: Interfaz de inicio.



Fig. 55: Interfaz menú niveles de rompecabezas.



Fig. 53: Interfaz Menú de opciones.



Fig. 56: Interfaz rompecabezas nivel 1.



Fig. 54: Interfaz Menú juegos.

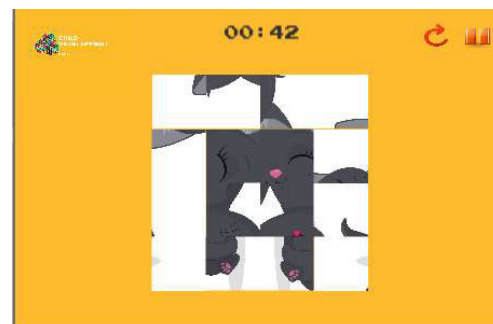


Fig. 57: Interfaz rompecabezas nivel 2.



Fig. 58: Interfaz rompecabezas nivel 3.

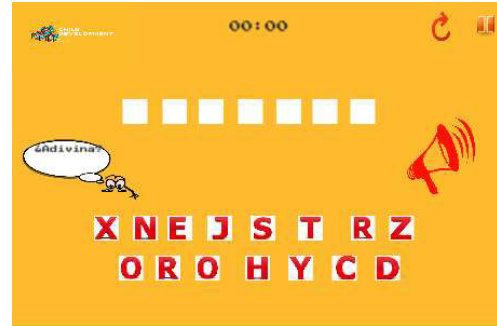


Fig. 61: Interfaz ABC Descubre nivel 3.



Fig. 59: Interfaz ABC Descubre nivel 1.

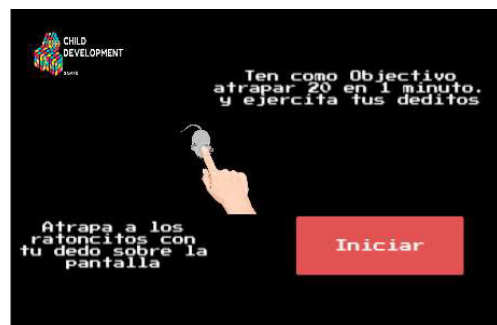


Fig. 62: Interfaz introducción juegos TocaToca.

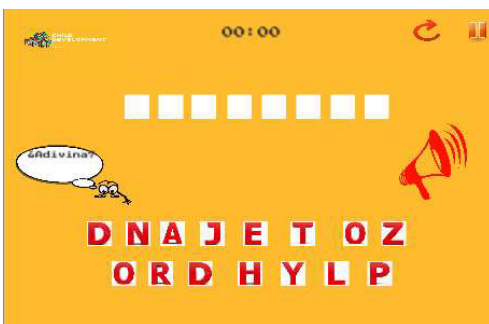


Fig. 60: Interfaz ABC Descubre nivel 2.



Fig. 63: Interfaz juego TocaToca.

6. Descripción del entorno Unity 3D

A continuación, se describe el entorno de Unity que se ha usado para este proyecto.

En la Fig. 64 se puede apreciar el directorio Assets que es donde se guardan las escenas del juego y los paquetes de recursos usados en el desarrollo de la aplicación móvil.

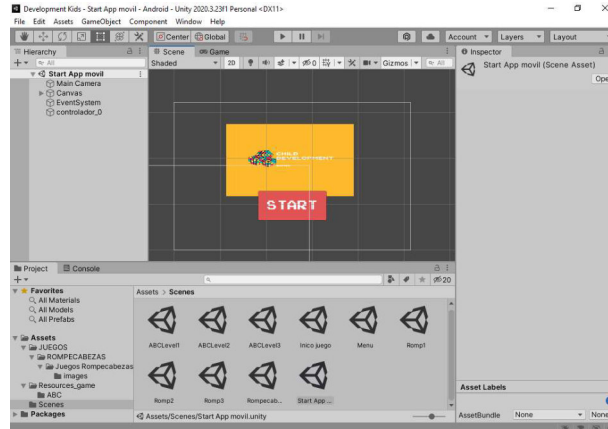


Fig. 64: Presentación Interfaz Unity.

En la Fig. 65 y Fig. 66 se puede observar por Escena el inspector donde se puede interactuar y modificar las propiedades de cada uno de los elementos mostrados en pantalla.

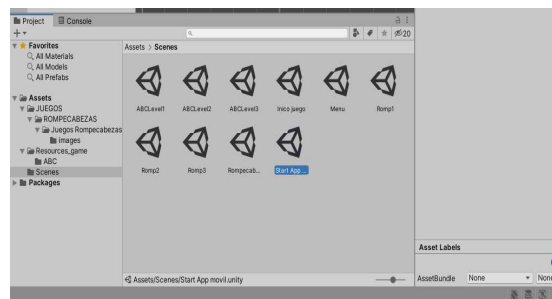


Fig. 65: Diferentes escenas usadas en la aplicación móvil.

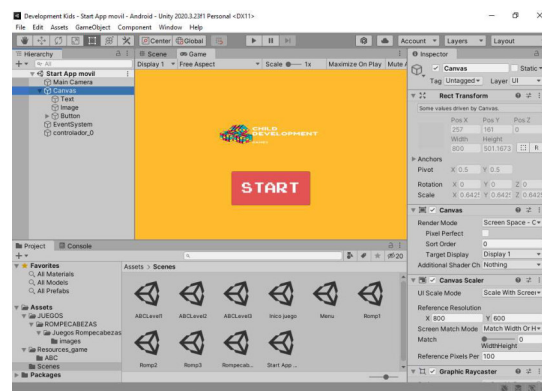


Fig. 66: Interfaz principal Unity.

La Fig. 67 muestra el inspector donde se cargan los *scripts* para dar el comportamiento a los elementos de la aplicación móvil.



Fig. 67: Muestra del inspector de Unity.

Descripción de los elementos usados en esta aplicación móvil

Desde la Fig. 68 hasta la Fig. 71, se muestran los elementos que se han usado en el desarrollo de la aplicación móvil.

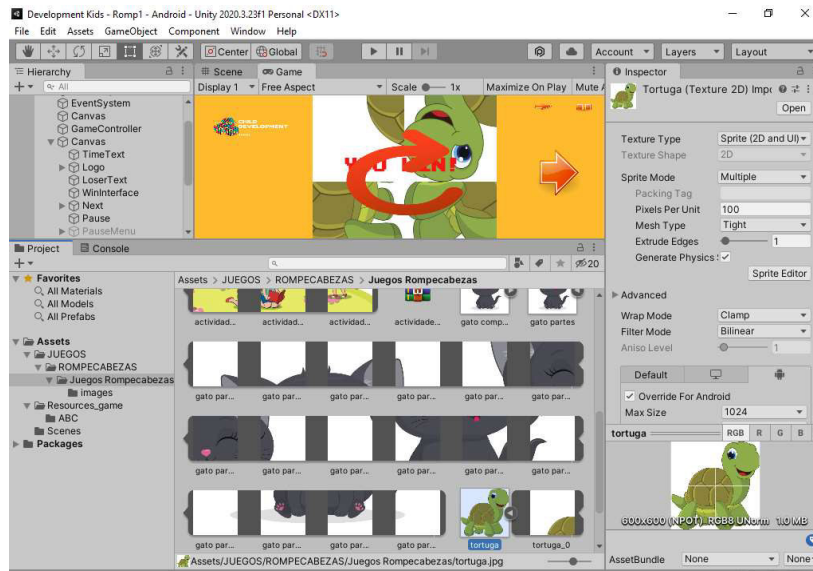


Fig. 68: Imágenes usadas en el rompecabezas con sus diferentes Sprite (que sirven para poder interactuar con los elementos)



Fig. 69: Recursos usados durante el desarrollo de la aplicación, mostrados en la carpeta correspondiente.

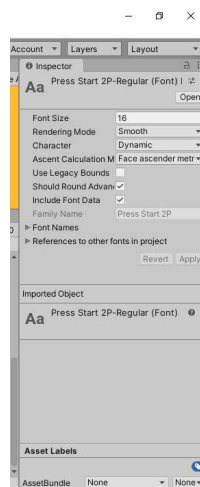


Fig. 70: Tipo de letra usada para esta aplicación móvil.

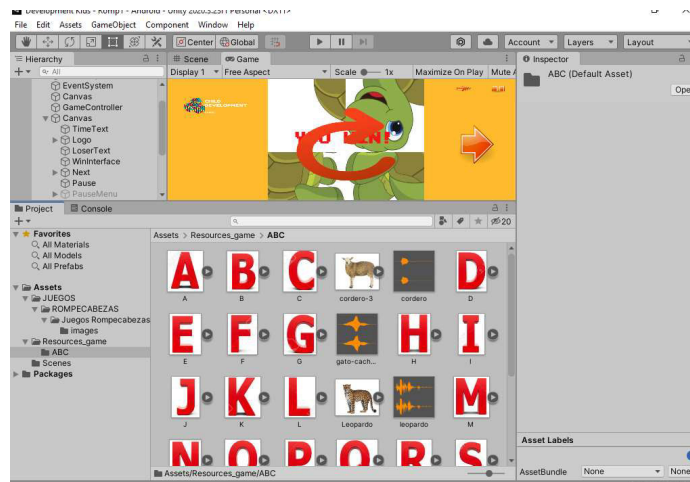


Fig. 71: Muestra Sprite de abecedario para el juego ABC Descubre.

Estructura de escenas

El manejo de escenas en Unity 3D es de suma importancia debido a que es la forma en la que los juegos diseñados en esta herramienta se complementan, según la lógica en la que estos estén basados, a continuación, desde la **Fig. 72** hasta la **Fig. 81** se presentan las escenas de la aplicación móvil.

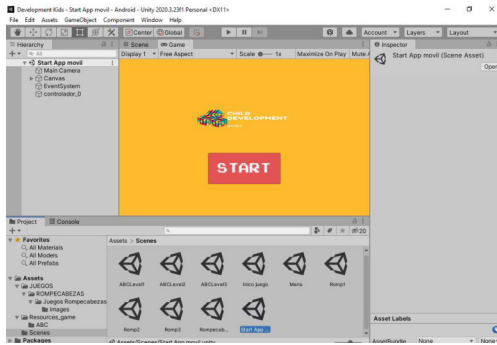


Fig. 72: Escena Inicio.



Fig. 73: Escena Interfaz Inicio.



Fig. 74: Escena Menú.

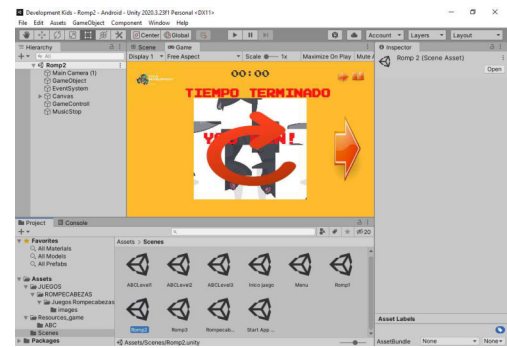


Fig. 76: Escena Nivel2 – rompecabezas.

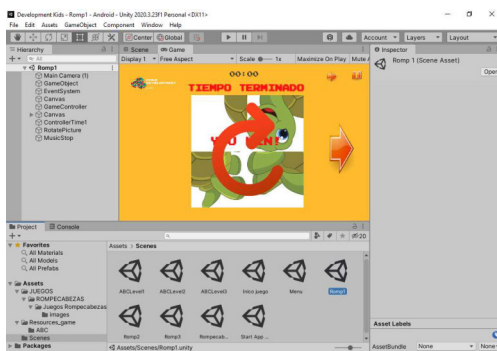


Fig. 75: Escena Nivel 1- rompecabezas.

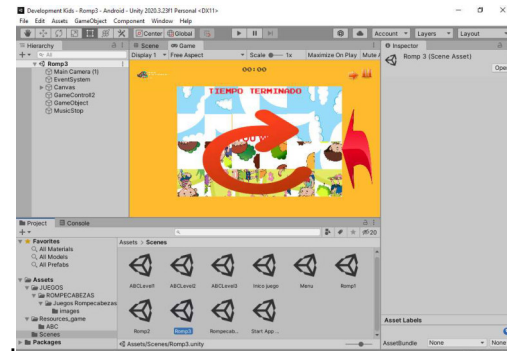


Fig. 77: Escena Nivel 3 - rompecabezas.



Fig. 78: Escena Menú - rompecabezas.

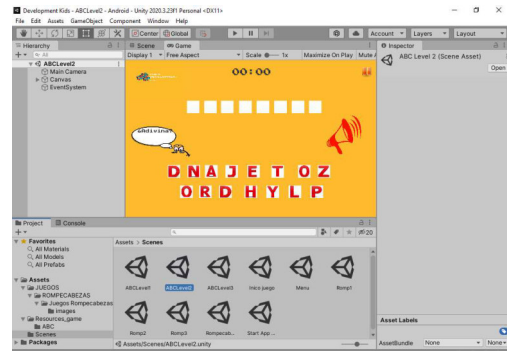


Fig. 80: Escena Nivel 2 - ABC Descubre.

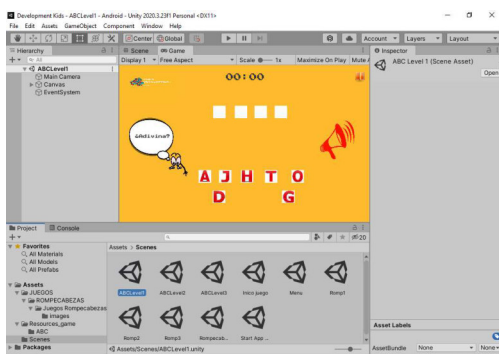


Fig. 79: Escena Nivel 1 - ABC Descubre.

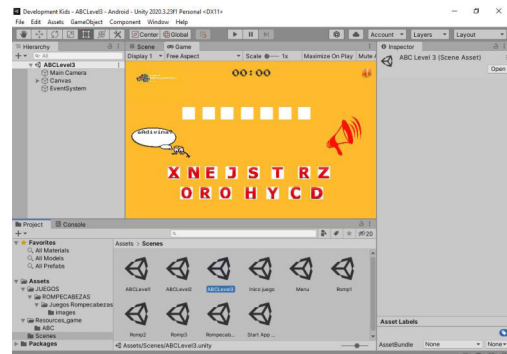


Fig. 81: Escena Nivel 3 - ABC Descubre.

7. Pruebas

Para tener una mayor certeza de que el videojuego funcione correctamente, se han realizado una serie de pruebas.

Pruebas Unitarias

Terminada la etapa de codificación se ha establecido una serie de pruebas Unitarias para verificar que cada juego de la aplicación móvil no presente errores a nivel de código que luego puedan resultar en diferentes bugs, es así como a continuación se presenta el resultado de las pruebas unitarias que están indicados desde la **Fig. 82** hasta la **Fig. 87**.

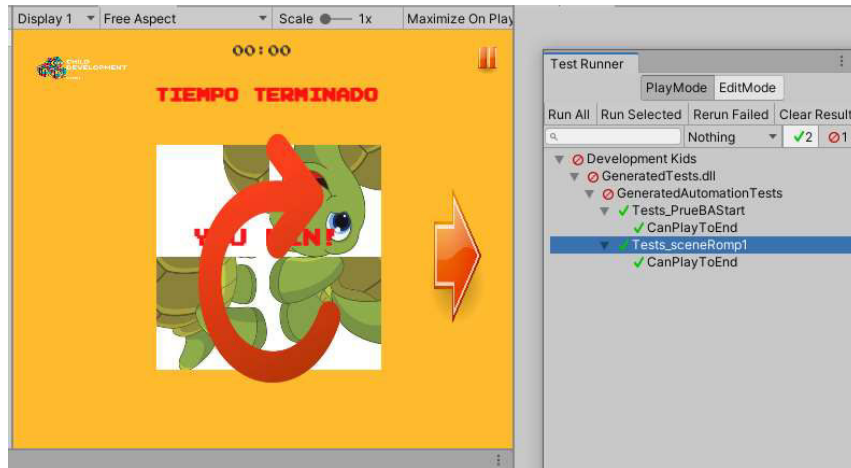


Fig. 82: Prueba Unitaria Rompecabezas nivel 1.



Fig. 83: Prueba unitaria rompecabezas nivel 2.

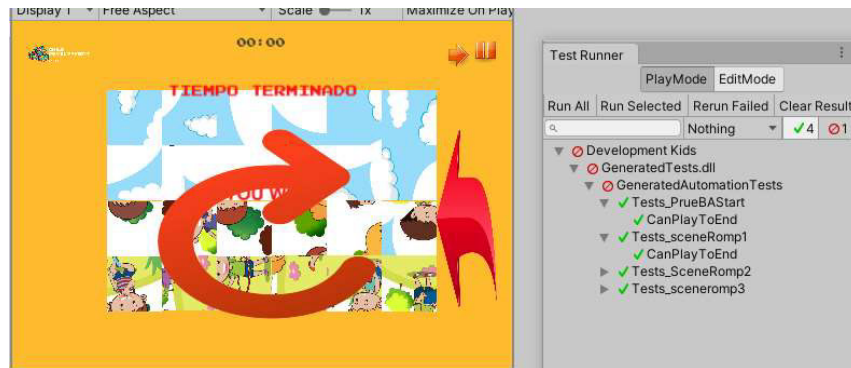


Fig. 84: Prueba Unitaria rompecabezas nivel 3.

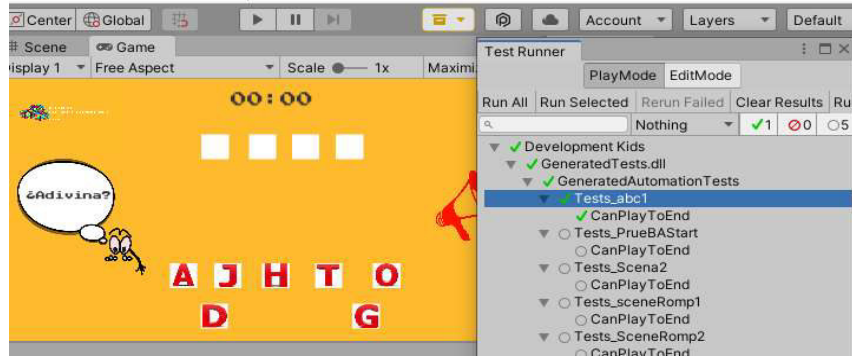


Fig. 85: Prueba Unitaria ABC nivel 1.

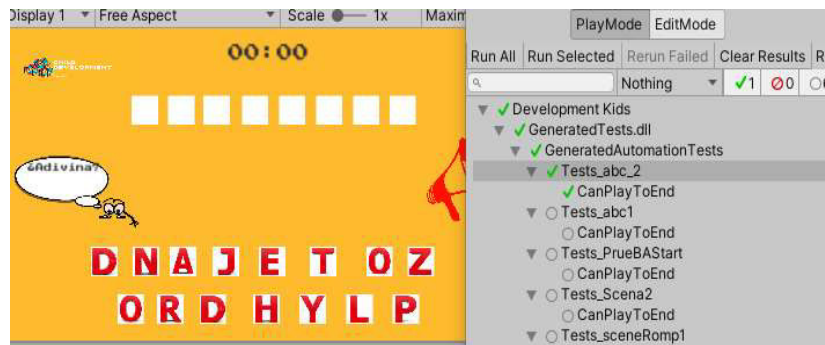


Fig. 86: Prueba Unitaria ABC nivel 3.

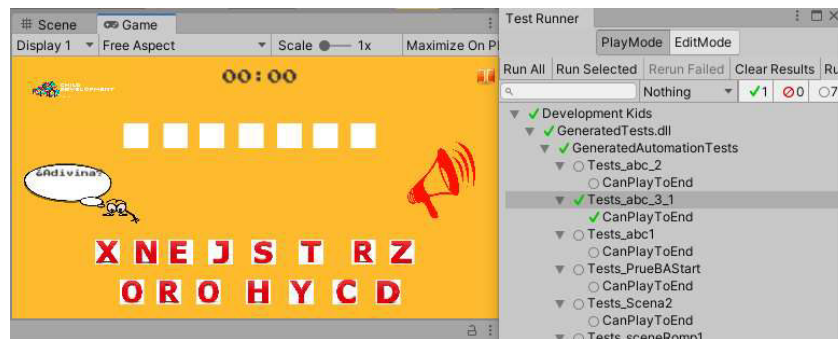


Fig. 87: Prueba Unitaria ABC nivel 3.

Pruebas de Compatibilidad

A continuación, se muestra la aplicación ejecutándose en diferentes dispositivos móviles Android.



Fig. 88: Nivel 1 ABC Aprende en SAMSUNG Galaxy A7 2018.

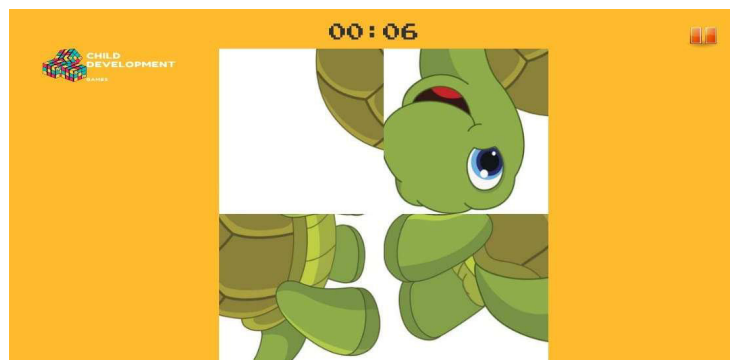


Fig. 89: Nivel 1 juego Rompecabezas en SAMSUNG Galaxy A8 2018.

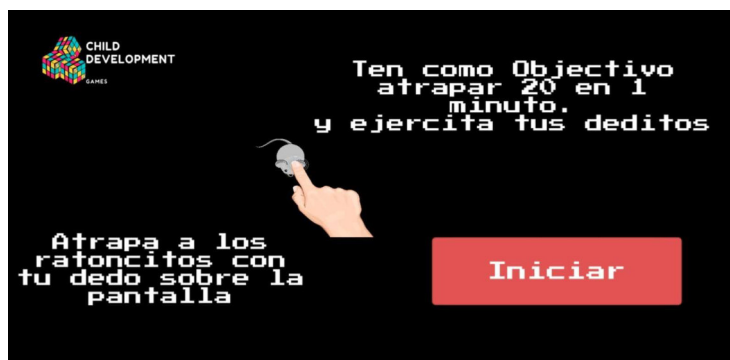


Fig. 90: Introducción al juego TocaToca en SAMSUNG A20.



Fig. 91: Nivel 3 rompecabezas SAMSUNG A20.



Fig. 92: Nivel 1 ABC Descubre en Xiaomi Redmi Note 8.



Fig. 93: Menú inicial en Xiaomi Redmi Note 8.

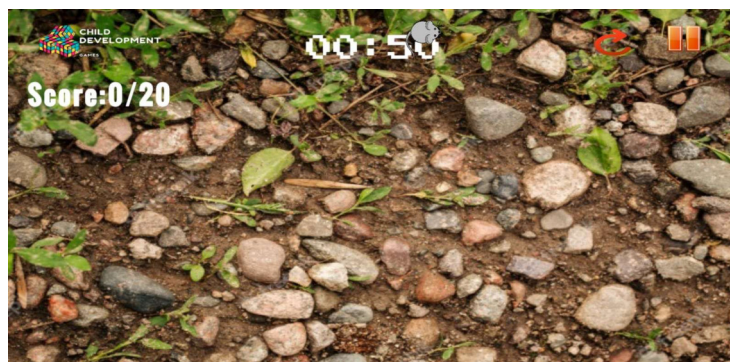


Fig. 94: Juego TocaToca Xiaomi Redmi 9.

Pruebas de aceptación

A continuación se presentan las pruebas de aceptación desde la **TABLA XXVII** hasta **TABLA XXXIII**, las mismas que a su vez se indica el procedimiento de cada uno y de lo que el usuario debe realizar para la correcta ejecución de estas funcionalidades y su posterior aprobación.

TABLA XXVII-Prueba de aceptación 2: Escoger cualquier juego.

Prueba de Aceptación	
Identificador (ID): PA02	Identificador de requerimiento: RR002
Nombre: Escoger cualquier juego.	
Descripción: Como usuario una vez que se presente dentro de la aplicación móvil este podrá escoger cualquiera de los juegos que desee probar.	
Pasos de ejecución: <ul style="list-style-type: none"> • Entrar al menú de juegos. • Escoger el primer juego rompecabezas. • Escoger un nivel. • Jugar nivel 1 del juego rompecabezas. • Escoger pasar al siguiente nivel. • Usar el botón de pause, y dar clic en salir al menú principal. • Escoger el siguiente juego que desee. 	
Resultado deseado: El usuario podrá escoger entre los diferentes juegos, y en caso de no querer jugarlos el mismo podrá salir y escoger otro.	
Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.	

TABLA XXVIII-Prueba de aceptación 3: Ingresar al segundo juego.

Prueba de Aceptación	
Identificador (ID): PA03	Identificador de requerimiento: RR004
Nombre: Ingresar al segundo juego.	
Descripción:	

<p>Como usuario podrá jugar el segundo juego que cuenta con una bocina y una sopa de letras la cual al presionar la bocina esta me dará un sonido de un animal que luego será completado por medio de la sopa de letras presente en la parte de abajo.</p>
<p>Pasos de ejecución:</p> <ul style="list-style-type: none"> • Entrar al menú de juegos. • Escoger el segundo juego dando clic en el mismo. • Ingresar al juego ABC Descubre. • Presionar el botón con la imagen de una bocina, y escuchar el sonido. • Arrastrar las letras a su casilla correspondiente en base al nombre del animal que emite el sonido. • Se completa el nivel al rellenar correctamente los casilleros.
<p>Resultado deseado: El usuario puede jugar el segundo juego ABC Descubre escuchando claramente el sonido emitido por la bocina para poder completar correctamente el nombre al que pertenece el animal que emite el sonido.</p>
<p>Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.</p>

TABLA XXIX-Prueba de aceptación 4: Salir de cualquier juego

Prueba de Aceptación	
Identificador (ID): PA04	Identificador de requerimiento: RR006
Nombre: Salir de cualquier juego.	
Descripción: El usuario podrá salir de cualquier juego en cualquier momento que este lo requiera además de ponerle pause, el usuario podrá regresar al menú principal si así lo desea.	
Pasos de ejecución: <ul style="list-style-type: none"> • Escoger cualquier juego que el usuario desee. • Dar clic en el botón de pause. • Se abre el menú de pause. • Escoger la opción salir. • El usuario regresara al menú principal. 	
Resultado deseado: El usuario puede salir del juego en el momento en que este lo desea a través del menú pausa y regresar al menú principal para escoger otro juego.	
Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.	

TABLA XXX-Prueba de aceptación 5: Cronometro.

Prueba de Aceptación	
Identificador (ID): PA05	Identificador de requerimiento: RR007
Nombre: Cronometro.	
Descripción: El usuario debe cumplir el reto del juego en un tiempo limite el cual a no ser cumplido el mismo no podrá avanzar de forma linear al siguiente nivel y tendrá que repetirlo, si el usuario cumple con el objetivo el tiempo establecido este podrá superar el desafío del juego.	
Pasos de ejecución:	

<ul style="list-style-type: none"> • Escoger cualquier juego de la aplicación en el menú de juegos. • El cronometro iniciara automáticamente al iniciar el juego. • Completar el juego y el cronometro se detendrá habilitando el siguiente nivel. • Fallar el juego, el cronómetro llega a 0 y se reinicia el nivel.
<p>Resultado deseado: Al momento de ganar o perder el cronómetro da el resultado esperado de que el juego habilita el siguiente nivel o que, si debe repetir, al fallar en cumplir el objetivo en el tiempo establecido.</p>
<p>Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.</p>

TABLA XXXI-Prueba de aceptación 6: Sonido *Game Over*.

Prueba de Aceptación	
Identificador (ID): PA06	Identificador de requerimiento: RR008
Nombre: Sonido GameOver.	
Descripción: Al momento de que el usuario no cumpla con uno de los retos presentes en los juegos, el cronómetro llegará a 0 y el mismo tendrá un sonido de advertencia indicando de que el tiempo está terminado y el nivel no fue aprobado.	
Pasos de ejecución: <ul style="list-style-type: none"> • Escoger cualquiera de los juegos. • Fallar en los retos presentes en los juegos. • El cronómetro llega a 0 y se presenta el audio de Game Over. 	
Resultado deseado: Si el usuario no cumple con los retos en el tiempo indicado se le presenta un sonido de advertencia para repetir el nivel, el sonido es presentado correctamente conjuntamente cuando el cronómetro llega a su fin.	
Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.	

TABLA XXXII-Prueba de aceptación 7: Sonido de victoria.

Prueba de Aceptación	
Identificador (ID): PA07	Identificador de requerimiento: RR009
Nombre: Sonido de Victoria.	
Descripción: El usuario al completar un nivel de los juegos de la aplicación móvil este recibirá un sonido de advertencia indicando que supero el nivel en el tiempo establecido y podrá pasar al siguiente nivel.	
Pasos de ejecución: <ul style="list-style-type: none"> • Escoger cualquier juego de la aplicación móvil. • Completar cualquiera de los diferentes niveles del juego. • Al completar el usuario automáticamente escuchara el audio de victoria indicando que cumplió el reto en el tiempo estimado. 	
Resultado deseado: Al momento de completar el reto de un nivel de los juegos de la aplicación móvil, el usuario recibe una advertencia de audio correctamente indicando que ha superado el nivel y puede seguir al siguiente, el audio se presenta de forma correcta con el comportamiento de la aplicación móvil.	

<p>Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.</p>

TABLA XXXIII-Prueba de aceptación 8: Retornar al menú principal.

Prueba de Aceptación	
Identificador (ID): PA08	Identificador de requerimiento: RR010
Nombre: Retornar al menú principal.	
Descripción: Cuando el usuario complete todos los niveles del juego que hayan escogido este mismo regresara al menú principal, pudiendo así escoger un nuevo juego.	
Pasos de ejecución: <ul style="list-style-type: none"> • Escoger cualquiera de los juegos de la aplicación móvil. • Completar los niveles satisfactoriamente hasta llegar al último. • Se presenta el sonido de victoria y se abre el botón para volver al menú principal. 	
Resultado deseado: El usuario al escoger superar todos los niveles de un mismo juego de la aplicación móvil, podrá pasar al menú principal y seguir con el siguiente juego.	
Evaluación de prueba: Se comprueba el resultado esperado Aprobación 100%.	

ANEXO III. Manual de usuario

A continuación, se muestra el enlace con el video del manual de usuario de la Aplicación Móvil.

- <https://youtu.be/2Wudehv6loE>

ANEXO IV. Manual de instalación

Repositorio del código fuente:

El código fuente de todo el proyecto, se encuentra alojado en el repositorio *GitHub*, el cual se puede acceder a través de la siguiente URL:

- <https://github.com/chrispaul20191610/DevKidsGame.git>

La instalación de esta aplicación se la puede realizar desde la tienda de aplicaciones de Android, Google Play Store, y se instala automáticamente en su dispositivo, el siguiente URL, contiene la aplicación móvil en Play Store.

- <https://play.google.com/store/apps/details?id=com.chrisPaul97.DevelopmentKids>.