



**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE**

**IMPLEMENTACIÓN DE UN VIDEOJUEGO DINÁMICO PARA  
INCENTIVAR EL APRENDIZAJE DE OPERACIONES  
ARITMÉTICAS EN EL CURSO DE MATEMÁTICA PARA EL  
PRIMER GRADO DE EDUCACIÓN PRIMARIA**

**Tesis presentada por:**

Diego Joseph Añazco Bolívar

**Asesor:**

Álvaro Fernández del Carpio

**Para obtención del Título Profesional de:**

**INGENIERO DE SOFTWARE**

Arequipa – Perú

2022

## **Dedicatoria**

A mi familia, que siempre me brindó un apoyo incondicional en todas mis metas.

## **Agradecimientos**

A mis docentes de la carrera de Ing. de Software por brindarme los conocimientos que hicieron posible este trabajo de investigación.

A los docentes del colegio San Juan Bautista de La Salle por la confianza y las facilidades para realizar las pruebas en sus instalaciones.

Seres queridos y amigos que apoyaron con su retroalimentación para mejorar la usabilidad del producto.

# Índice General

Dedicatoria .....	2
Agradecimientos .....	3
Índice General.....	4
Lista de Abreviaturas y Siglas.....	7
Índice de Tablas.....	8
Índice de Figuras .....	10
Resumen .....	12
Abstract .....	13
Palabras Clave .....	14
Capítulo 1. Planteamiento de la Investigación.....	15
1.1 Planteamiento del Problema .....	15
1.2 Objetivos de la Investigación .....	18
1.2.1 Objetivo General .....	18
1.2.2 Objetivos Específicos.....	18
1.3 Tipo de Investigación.....	18
1.3.1 Investigación Aplicada.....	18
1.3.2 Investigación Cualitativa .....	18
1.4 Nivel de Investigación.....	18
1.4.1 Investigación Descriptiva.....	18
1.5 Justificación .....	19
Capítulo 2. Revisión y Fundamentación Teórica.....	20
2.1 Estado del Arte.....	20
2.1.1 A Study of the Interactive Mathematics Mobile Application Development .....	20
2.1.2 An Augmented Reality Application for Basic Mathematics: Teaching and Assessing Kids' Learning Efficiency .....	20
2.1.3 An Online Adventure Game for Teaching Math .....	21
2.1.4 CocoGame: a funny app to learn physics and math .....	21
2.1.5 Designing a Game for Learning Math by Composing: a Finnish Primary School Case .....	22
2.1.6 Desing of Digital Game-Based Learning System for Elementary Mathematics Problem Solving .....	22
2.1.7 Game Based Learning for math learning - iFractions case study .....	23

2.1.8 Math Detective - Digital Game-based Mathematical Error Detection, Correction and Explanation .....	24
2.1.9 Developing a Mobile App for Game-based Learning in Middle School Mathematics Course .....	24
2.1.10 Math Education Game for Primary School .....	25
2.1.11 Space Chain A math game for training Geometric and Arithmetic Progressions .....	25
2.1.12 Plataforma M-learning para incentivar el aprendizaje de Matemáticas en estudiantes de tercero de primaria basado en Unity .....	26
2.1.13 Conclusiones sobre el estado del arte .....	27
2.2 Marco Teórico .....	27
2.2.1 Aprendizaje.....	27
2.2.2 Aprendizaje matemático.....	28
2.2.3 Videojuegos .....	32
2.2.4 Videojuegos educativos .....	35
2.2.4 Aplicación Web.....	35
2.2.5 Modelo Espiral de Boehm.....	36
2.2.6 Framework.....	38
Capítulo 3. Método de Investigación .....	40
3.1 Modelo en Espiral de Boehm .....	40
3.1.1 Establecimiento de Objetivos .....	40
3.1.2 Valoración y Reducción del Riesgo.....	40
3.1.3 Desarrollo y Validación .....	40
3.1.4 Planeación .....	40
3.2 Ingeniería de Requerimientos.....	41
3.2.1 Entrevistas.....	41
3.2.2 Requerimientos Funcionales .....	41
3.2.3 Requerimientos no Funcionales .....	45
3.3 Riesgos del Software .....	46
3.4 Arquitectura del Software.....	46
3.4.1 Diagrama de Casos de Uso .....	46
3.4.2 Vista Física .....	51
3.4.3 Vista de Desarrollo .....	53
3.5 Diseño de Prototipos .....	55
3.5.1 Personaje .....	56
3.5.2 Menú Principal .....	56

3.5.3 Niveles.....	58
3.5.4. Pistas y ayudas.....	62
3.6 Implementación.....	63
3.6.1. Instalación y Configuración .....	63
3.6.2. Función <i>preload</i> ( ).....	65
3.6.3. Función <i>create</i> ( ) .....	66
3.6.4. Función <i>update</i> ( ).....	70
3.6.5. Otras funciones.....	70
3.7 Pruebas del Software .....	75
3.7.1 Elementos .....	75
3.7.2 Pruebas de Sistema.....	75
3.7.3 Pruebas de Rendimiento.....	76
3.7.4 Pruebas de Carga .....	76
3.7.5 Pruebas de Usabilidad .....	77
Capítulo 4. Resultados y Discusión .....	79
4.1 Pruebas de Sistema .....	79
4.2 Pruebas de Rendimiento .....	80
4.3 Pruebas de Carga.....	81
4.4 Pruebas de Usabilidad .....	83
4.5 Experiencia de juego de los estudiantes (EJE) .....	85
4.5.1 Datos generales sobre el uso de videojuegos .....	85
4.5.2 Experiencia con MathDragon .....	87
4.5.3 Aprendizaje Matemático .....	87
Conclusiones.....	89
Trabajos Futuros .....	90
Referencias .....	91
Anexos.....	95
1. Cuestionario de Experiencia de Juego .....	95
2. Cuestionario de Usabilidad del Sistema Informático (CSUQ) .....	97
3. Manual de usuario de MathDragon .....	99
4. Ingeniería de Requerimientos: Resumen general de entrevistas a tutores....	102
5. Matriz de Trazabilidad de Pruebas de Sistema .....	105
6. Matriz de Riesgos.....	115

## Lista de Abreviaturas y Siglas

**CBL:** Challenge Based Learning

**CS:** Caso de Uso

**CSUQ:** Cuestionario de Usabilidad del Sistema Informático

**DOM:** Document Object Model

**DRE:** Direcciones Regionales de Educación

**EJE:** Experiencia de Juego de los Estudiantes

**HEI:** Laboratory of Higher Education Institution

**HTML:** HyperText Markup Language

**HTTP:** Hypertext Transfer Protocol

**HTTPS:** HyperText Transfer Protocol Secure

**IBM:** International Business Machines

**INEI:** Instituto Nacional de Estadística Informática

**IOS:** iPhone Operating System

**IP:** Internet Protocol

**JSON:** JavaScript Object Notation

**LMS:** Learning Managment System

**RAD:** Rapid Application Development

**RAE:** Real academia española

**REQF:** Requerimiento Funcional

**REQNF:** Requerimiento No Funcional

**SDK:** Software Development Kit

**SPSS:** Statistical Package for Social Sciences

**TCP:** Transmission Control Protocol

**UGEL:** Unidades de Gestión Educativa Locales

**URL:** Uniform Resource Locator

**WEBGL:** Web Graphics Library

## Índice de Tablas

Tabla 1. Centros Educativos según Nivel y Modalidad .....	15
Tabla 2. Alumnos Matriculados en Educación Primaria por Gestión .....	15
Tabla 3. Resultados Nacionales según Medida Promedio y Niveles de Logro .....	16
Tabla 4. Hogares según Acceso de las Tecnologías de la Información y Comunicación .....	16
Tabla 5. Colegios según Acceso a las Tecnologías de Información y Comunicación .....	17
Tabla 6. Docentes Según Gestión 2017 .....	17
Tabla 7. Resumen de hallazgos en el estado del arte .....	27
Tabla 8. Representación matemática.....	41
Tabla 9. Matemática contextual .....	41
Tabla 10. Mensajes motivacionales.....	42
Tabla 11. Obstáculos/retos en el videojuego .....	42
Tabla 12. Control de tiempo de juego.....	42
Tabla 13. Nivel Conseguir Comida .....	43
Tabla 14. Nivel Destruir Venenos .....	43
Tabla 15. Contador de vidas.....	43
Tabla 16. Movimiento y controles .....	44
Tabla 17. Ayudas/pistas en el problema matemático .....	44
Tabla 18. Persona y temática.....	44
Tabla 19. Aprendizaje del videojuego .....	45
Tabla 20. Diseño responsive.....	45
Tabla 21. Disponibilidad del videojuego .....	45
Tabla 22. Ejecución en navegadores web .....	45
Tabla 23. Matriz de riesgos.....	46
Tabla 24. Iniciar videojuego MathDragon .....	47
Tabla 25. Seleccionar nivel Conseguir Comida .....	47
Tabla 26. Seleccionar nivel Destruir Venenos.....	48
Tabla 27. Destrozar veneno .....	48
Tabla 28. Recoger muslo de pollo .....	49
Tabla 29. Recoger pescado.....	49
Tabla 30. Recoger bomba .....	50



<b>Tabla 31. Finalizar nivel .....</b>	<b>50</b>
<b>Tabla 32. Aprender conceptos matemáticos .....</b>	<b>51</b>
<b>Tabla 33. Resumen de Pruebas Unitarias .....</b>	<b>79</b>
<b>Tabla 34. Pruebas de Rendimiento: Red .....</b>	<b>80</b>
<b>Tabla 35. Pruebas de Rendimiento: Capacidad Computacional .....</b>	<b>80</b>
<b>Tabla 36. Rendimiento de red: Promedio de tiempo de respuesta .....</b>	<b>81</b>
<b>Tabla 37. Rendimiento computacional: Promedio de tiempo de respuesta .....</b>	<b>81</b>
<b>Tabla 38. Pruebas de Carga.....</b>	<b>82</b>
<b>Tabla 39. Pruebas de Usabilidad: Resultados por pregunta. ....</b>	<b>83</b>
<b>Tabla 40. Pruebas de Usabilidad: Resultados por grupo. ....</b>	<b>84</b>
<b>Tabla 41. EJE: Pregunta 1.....</b>	<b>85</b>
<b>Tabla 42. EJE: Pregunta 2.....</b>	<b>86</b>
<b>Tabla 43. EJE: Pregunta 3.....</b>	<b>86</b>
<b>Tabla 44. EJE: Pregunta 4.....</b>	<b>86</b>
<b>Tabla 45. EJE: Pregunta 5.....</b>	<b>86</b>
<b>Tabla 46. EJE: Resultados experiencia con MathDragon .....</b>	<b>87</b>
<b>Tabla 47. EJE: Aprendizaje Matemático .....</b>	<b>88</b>
<b>Tabla 48. Resumen de respuestas de la entrevista a tutores.....</b>	<b>102</b>
<b>Tabla 49. Matriz de trazabilidad de pruebas unitarias. ....</b>	<b>105</b>

## Índice de Figuras

<b>Fig. 1 Modelos didácticos observados en clases de matemáticas .....</b>	<b>29</b>
<b>Fig. 2 Etapas básicas del proceso de aprendizaje y .....</b>	<b>30</b>
<b>Fig. 3 Géneros de videojuego .....</b>	<b>32</b>
<b>Fig. 4 Separación de funciones .....</b>	<b>36</b>
<b>Fig. 5 Modelo en espiral de Boehm .....</b>	<b>37</b>
<b>Fig. 6 Diagrama de Casos de Uso de MathDragon.....</b>	<b>47</b>
<b>Fig. 7 Diagrama de despliegue de MathDragon .....</b>	<b>52</b>
<b>Fig. 8 Diagrama de componentes de MathDragon .....</b>	<b>53</b>
<b>Fig. 9 Kalh, personaje principal de MathDragon.....</b>	<b>56</b>
<b>Fig. 10 Escena “Menú Principal” de MathDragon.....</b>	<b>57</b>
<b>Fig. 11 Escena de selección de nivel en MathDragon .....</b>	<b>57</b>
<b>Fig. 12 Vida de Kalh en MathDragon.....</b>	<b>58</b>
<b>Fig. 13 Bomba en MathDragon .....</b>	<b>58</b>
<b>Fig. 14 Temporizador de tiempo en MathDragon .....</b>	<b>58</b>
<b>Fig. 15 MathDragon: Muslos de pollo .....</b>	<b>59</b>
<b>Fig. 16 MathDragon: Pescados.....</b>	<b>59</b>
<b>Fig. 17 MathDragon: Nivel Conseguir Comida .....</b>	<b>59</b>
<b>Fig. 18 Nivel Conseguir Comida: Cartel resumen.....</b>	<b>60</b>
<b>Fig. 19 MathDragon: Botella de veneno .....</b>	<b>60</b>
<b>Fig. 20 MathDragon: Nivel Destruir Venenos .....</b>	<b>60</b>
<b>Fig. 21 Nivel Destruir Venenos: Cartel resumen .....</b>	<b>61</b>
<b>Fig. 22 Cartel de felicitaciones en MathDragon .....</b>	<b>61</b>
<b>Fig. 23 Cartel de volver a intentarlo en MathDragon .....</b>	<b>62</b>
<b>Fig. 24 Nivel Conseguir Comida: Carteles de Pistas .....</b>	<b>62</b>
<b>Fig. 25 Nivel Destruir Venenos: Carteles de Pistas .....</b>	<b>62</b>
<b>Fig. 26 Instalación del framework Phaser 3.....</b>	<b>63</b>
<b>Fig. 27 Configuración del videojuego .....</b>	<b>64</b>
<b>Fig. 28 Ejemplo de spritesheet del movimiento del dragón .....</b>	<b>65</b>
<b>Fig. 29 Ejemplo de función preload( ) .....</b>	<b>66</b>
<b>Fig. 30 Método data.set en la función create( ) .....</b>	<b>66</b>
<b>Fig. 31 Métodos de physics en la función create( ) .....</b>	<b>67</b>
<b>Fig. 32 Métodos de anims y tweens en la función create( ) .....</b>	<b>68</b>

<b>Fig. 33 Método add en la función create( ) .....</b>	<b>69</b>
<b>Fig. 34 Método que detecta nuestro teclado .....</b>	<b>69</b>
<b>Fig. 35 Función update( ) en MathDragon .....</b>	<b>70</b>
<b>Fig. 36 Función encarga de sumar el puntaje a la escena .....</b>	<b>71</b>
<b>Fig. 37 Función encarga del manejo del temporizador .....</b>	<b>72</b>
<b>Fig. 38 Función encarga del manejo de las vidas.....</b>	<b>73</b>
<b>Fig. 39 Función encarga de la inicialización de objetos .....</b>	<b>74</b>
<b>Fig. 40 Listener “Graph Results” de JMeter en MathDragon .....</b>	<b>82</b>
<b>Fig. 41 Manual de Usuario: Ingresar a MathDragon con su enlace.....</b>	<b>99</b>
<b>Fig. 42 Manual de Usuario: Escena del Menú Principal.....</b>	<b>99</b>
<b>Fig. 43 Manual de Usuario: Escena de Niveles .....</b>	<b>100</b>
<b>Fig. 44 Manual de Usuario: Vidas y temporizador .....</b>	<b>100</b>
<b>Fig. 45 Manual de Usuario: Cartel nivel Conseguir Comida .....</b>	<b>101</b>
<b>Fig. 46 Manual de Usuario: Cartel Destruir Venenos.....</b>	<b>101</b>
<b>Fig. 47 Manual de Usuario: Cartel de fin del nivel.....</b>	<b>101</b>

## Resumen

La educación en el Perú cada vez es más accesible para todos los peruanos; sin embargo, los niveles de logro satisfactorio en el curso de matemática en los estudiantes del primer grado de primaria de los centros educativos en el Perú, han disminuido en más de un 20% durante el periodo 2017-2019 según los reportes e informes dados por DRE/GRE, UGEL.

Esta disminución se debe a muchos factores, entre ellos: problemas familiares, falta de útiles escolares, falta de metodología de enseñanza, falta de motivación por parte de los estudiantes.

Frente a este último factor, la motivación por el curso de matemática, se propuso la implementación de un videojuego matemático educativo que se ejecute en navegadores web de computadoras o laptops, con el fin de motivar al estudiante a reforzar esta materia mejorando los niveles de logro.

El videojuego de nombre MathDragon, se implementó con el *framework* de videojuegos *Phaser 3*, haciendo uso de la metodología en espiral de Boehm para su desarrollo; fue probado en 64 estudiantes del primer grado de primaria del colegio San Juan Bautista de La Salle en Arequipa – Perú.

MathDragon fue evaluado en el año 2022 mediante dos cuestionarios: CUSQ, dedicado a la usabilidad del videojuego dado a un total de 13 docentes y EJE centrándose en la jugabilidad e incentivo de seguir aprendiendo matemática con videojuegos realizado a estudiantes; obteniendo una calificación de “muy bueno” según la evaluación del CUSQ y con un 95% de respuestas aprobatorias de motivación positiva para seguir aprendiendo operaciones matemáticas con videojuegos por parte de los estudiantes.

## Abstract

Education in Peru is becoming more and more accessible to all Peruvians; However, the levels of satisfactory achievement in the mathematics course in the students of the first grade of primary school in educational centers in Peru, have decreased by more than 20% during the period 2017-2019 according to the reports and reports given by DRE/GRE, UGEL.

This decrease is due to many factors, among them: family problems, lack of school supplies, lack of teaching methodology, lack of motivation on the part of the students.

Faced with this last factor, the motivation for the mathematics course, the implementation of an educational mathematical video game that runs on computer or laptop web browsers was proposed, in order to motivate the student to reinforce this subject by improving achievement levels. .

The videogame named MathDragon, was implemented with the Phaser 3 videogame framework, making use of Boehm's spiral methodology for its development; It was tested on 64 students of the first grade of primary school of the San Juan Bautista de La Salle school in Arequipa - Peru.

MathDragon was evaluated in 2022 through two questionnaires: CUSQ, dedicated to the usability of the video game given to a total of 13 teachers, and EJE, focusing on the gameplay and incentive to continue learning mathematics with video games carried out for students; obtaining a rating of "very good" according to the CUSQ evaluation and with 95% of approving responses of positive motivation to continue learning mathematical operations with video games by the students.

## **Palabras Clave**

Videojuegos, matemáticas, educación, motivación, usabilidad, phaser3.

# Capítulo 1. Planteamiento de la Investigación

## 1.1 Planteamiento del Problema

La educación cada vez es más accesible para los niños del Perú, esto gracias al incremento de centros educativos en las distintas modalidades, educación inicial, primaria o secundaria (Tabla 1). Por consecuencia los alumnos matriculados en la educación primaria aumentaron con el pasar de los años. Siendo la gestión pública la que cuenta con mayor número de alumnos a comparación de la gestión privada. (Tabla 2)

*Tabla 1. Centros Educativos según Nivel y Modalidad*

---

Educación básica regular			
<u>Año</u>	<u>Educación Inicial</u>	<u>Educación Primaria</u>	<u>Educación Secundaria</u>
2018	53,671	38.532	14.683
2019	53,706	38.605	14.831
2020	53,278	38.411	14.942

---

Fuente: Instituto Nacional de Estadística e Informática del Perú [1]

*Tabla 2. Alumnos Matriculados en Educación Primaria por Gestión*

---

Departamento de Arequipa		
<u>Año</u>	<u>Gestión Pública</u>	<u>Gestión Privada</u>
2018	85,900	58,300
2019	90,600	59,900
2020	100,600	56,000

---

Fuente: Instituto Nacional de Estadística e Informática del Perú [2][3]

La educación evoluciona con el pasar de los días y los docentes se tienen que adaptar a nuevas metodologías para poder adquirir la atención de sus alumnos; y además obtener los logros establecidos por el Ministerio de Educación. Lamentablemente en los últimos años se vio una baja significativa en la cantidad de alumnos con resultado satisfactorio en el área de matemáticas, a comparación de los años 2015 y 2016. (Tabla 03)

*Tabla 3. Resultados Nacionales según Medida Promedio y Niveles de Logro*

---

Curso de Matemáticas. Primer grado de primaria			
<u>Año</u>	<u>Satisfactorio</u>	<u>En Proceso</u>	<u>En Inicio</u>
2016	26.6	42.3	31.0
2017	34.1	37.3	28.6
2018	14.7	30.3	55.0
2019	17.0	31.9	51.1

---

Fuente. Reportes e informes 2019 por DRE/GRE, UGEL [4]

Es notable que el acceso a las tecnologías de la información y comunicación en las familias arequipeñas tuvo un considerable aumento; siendo el acceso al internet 6.61 veces mayor y en el caso a los celulares de un 2.53 (Tabla 4). Además, los colegios siguen adquiriendo computadoras e internet para una educación de calidad. (Tabla 5)

*Tabla 4. Hogares según Acceso de las Tecnologías de la Información y Comunicación*

---

Región de Arequipa		
<u>Año</u>	<u>Internet</u>	<u>Celular</u>
2007	21.133	153.299
2017	139.893	388.404

---

Fuente: Instituto Nacional de Estadística e Informática del Perú [5]



*Tabla 5. Colegios según Acceso a las Tecnologías de Información y Comunicación*

Educación Primaria del Departamento de Arequipa		
<u>UGEL</u>	<u>Alumnos por Computadora</u>	<u>Porcentaje con Acceso a Internet</u>
Arequipa Norte	8	74.9
Arequipa Sur	8	74.9
Camaná	8	64.2
Caravelí	5	20.7
Castilla	4	18.8
Caylloma	4	15.2
Condesuyos	3	7.0
Islay	5	68.9
La Unión	3	8.2
La Joya	7	51.2

Fuente. Censo Educativo de DRE y UGEL 2017 [6]

La última encuesta realizada por el videojuego para computadora Fútbol Latino Online en el 2016 en nuestro país [7], demuestra que el 70% de niños y adolescentes entre los 8 y 13 años son los usuarios más activos. Es por ello que junto con el avance de las tecnologías de la información se podría apoyar a la educación, incentivando a los estudiantes con algo que los apasiona, los videojuegos.

*Tabla 6. Docentes Según Gestión 2017*

Censo Educativo 2017			
<u>Nivel</u>	<u>Gestión Pública</u>	<u>Gestión Privada</u>	<u>Total</u>
Inicial	58.953	35.963	94.946
Primaria	145.637	61.463	196.718
Secundaria	144.950	53.336	198.286

Fuente. Censo Educativo de DRE y UGEL 2017 [8]

Se observa que los estudiantes no están ajenos a esta tecnología creciente, con el apoyo de los hogares, miles de docentes (Tabla 6) y los centros educativos, se plantea hacer uso de estas tecnologías para resolver el bajo rendimiento académico en el curso de matemáticas, implementando un juego dinámico donde puedan aprender jugando.

## **1.2 Objetivos de la Investigación**

### **1.2.1 Objetivo General**

Implementar un videojuego dinámico que permita incentivar el aprendizaje de las operaciones de adición y sustracción dictado en el curso de matemática para el primer grado de educación primaria.

### **1.2.2 Objetivos Específicos**

- Adquirir los requerimientos del videojuego mediante técnicas de captura aplicadas a docentes expertos en el área.
- Codificar el videojuego en una plataforma que sea accesible para todos los usuarios finales.
- Elaborar un cuestionario que mida el incentivo de los estudiantes para aprender matemáticas con un videojuego.
- Elaborar pruebas de software de sistema, rendimiento, carga y usabilidad.
- Evaluar el videojuego con las pruebas de software.

## **1.3 Tipo de Investigación**

### **1.3.1 Investigación Aplicada**

La investigación propone el uso de un videojuego desplegado en navegadores web en los estudiantes de primer grado de educación primaria. Durante el proceso de la investigación se adquieren nuevos conocimientos para lograr un software que cumpla con el objetivo de la investigación.

### **1.3.2 Investigación Cualitativa**

En el proceso de la investigación se obtienen datos mediante cuestionarios sobre usabilidad, sentimientos y reacciones, y aprendizaje matemático de los estudiantes y docente frente al videojuego.

## **1.4 Nivel de Investigación**

### **1.4.1 Investigación Descriptiva**

Se realizará un diseño, desarrollo, implementación y validación de un videojuego con el *framework* Phaser 3 que tiene como finalidad incentivar el aprendizaje de las operaciones de adición y sustracción del curso de matemática. La descripción

de las actividades y procesos que realizan los alumnos son obtenidos mediante entrevistas a sus tutores; docentes encargados de los alumnos a evaluar.

### **1.5 Justificación**

La disminución de los niveles de logro Satisfactorio es más de un 20% (Tabla 3) en el curso de matemática. Un modelo de videojuego incentiva el aprendizaje de los estudiantes, incrementando su pensamiento lógico, mejorando sus calificaciones y su motivación por aprender [9] [10].

Los docentes tienen que ingeniar nuevas metodologías de enseñanza para captar la atención de los estudiantes; esto conlleva adaptarse a las nuevas tecnologías, a la realidad en la que viven actualmente los estudiantes. Es por ello que utilizar un modelo de videojuego como herramienta de enseñanza, facilitará el proceso de enseñanza del docente.

## Capítulo 2. Revisión y Fundamentación Teórica

### 2.1 Estado del Arte

#### 2.1.1 A Study of the Interactive Mathematics Mobile Application Development [11]

En Taiwán se utilizó la realidad aumentada como herramienta para la implementación de un videojuego, que mejora el aprendizaje de matemáticas en el área de geometría de plano simple y forma tridimensional. El videojuego se implementó con el motor de videojuegos Unity 3D y el kit de desarrollo de software de realidad aumentada Vuforia SDK. Se dividió el videojuego en dos secciones con los siguientes objetivos:

- Diversión de aprendizaje: Entender la estructura tridimensional y fortalecer temas de volumen.
- Diversión de desafío: Evaluar al estudiante, verificando los resultados del aprendizaje

El videojuego fue probado por 102 estudiantes del 5to grado de primaria. Los estudiantes rindieron un examen de conocimientos antes y después de su experiencia con el videojuego obteniendo en promedio el puntaje de 88.19 y 92.59 respectivamente.

Los resultados indican un aumento de 3.4 del puntaje promedio en el aprendizaje de matemáticas en los estudiantes evaluados, además, el uso de la realidad aumentada como herramienta genera una mayor motivación y aceptación.

#### 2.1.2 An Augmented Reality Application for Basic Mathematics: Teaching and Assessing Kids' Learning Efficiency [12]

Mathify es un videojuego matemático de realidad aumentada diseñado para estudiantes de 3 a 5 años que tiene como objetivo enseñar el sistema numérico del 1 al 9. El videojuego ha sido implementado con el motor de videojuegos Unity 3D. El aprendizaje con Mathify se divide en dos fases con los siguientes objetivos:

- Reconocimiento: Enseñar a los estudiantes el sistema numérico del 1 al 9 mediante una analogía entre el número y diferentes frutas.

- Evaluación: Evaluar el conocimiento adquirido por los estudiantes.

La eficiencia del videojuego para el aprendizaje del sistema numérico fue evaluada mediante dos exámenes: con el método tradicional (lápiz y papel) y con Mathify; el puntaje obtenido con el videojuego de Mathify fue mayor con un promedio total de 8.5 contra 6.2 en una escala de evaluación del 1 al 10; donde 10 es el mayor puntaje.

### **2.1.3 An Online Adventure Game for Teaching Math [13]**

Ghost Detective es un videojuego del género aventuras diseñado para la enseñanza del curso de matemática con los temas de: adición, sustracción, multiplicación y división, para estudiantes entre 9 y 12 años. El videojuego ha sido implementado como una aplicación web.

El jugador interactúa con los objetos que se encuentran en el escenario, cada objeto presenta un desafío matemático que debe resolverse para continuar con el videojuego.

El videojuego clasifica a los estudiantes en tres niveles: principiante, intermedio y avanzado. Para la clasificación se realiza un registro de cada uno de los estudiantes, guardando las respuestas enviadas por acertijo, clasificándolas en correctas e incorrectas.

Fue probado por una cantidad reducida de estudiantes que dieron una retroalimentación positiva, resaltando la narrativa de la historia del protagonista. El trabajo de investigación ha demostrado efectos positivos acerca los juegos de aventura para el aprendizaje

### **2.1.4 CocoGame: a funny app to learn physics and math [14]**

En México se implementó el videojuego Coco Game para dispositivos móviles y navegadores web, con el objetivo de mejorar los conocimientos en física y matemática de los estudiantes.

Implementado con el *framework* Meteor de *JavaScript* se utilizó la metodología ágil *Rapid Application Development* (RAD) que prioriza las versiones e

iteraciones de prototipos. Se escogió *MongoDB* como base de datos, optando por el formato de almacenamiento *JSON*.

Coco es el personaje principal del videojuego; en un inicio es pequeño e ignorante, conforme se progresa en el videojuego, Coco crece. La estructura del videojuego es la siguiente: Universos, mundos, niveles y elementos.

CocoGame cuenta con una interfaz para los docentes. Esta interfaz permite cargar y editar el contenido de las preguntas en varios formatos, también se puede realizar un seguimiento a los estudiantes, viendo su clasificación y su eficiencia en porcentaje de aciertos.

El videojuego ha sido probado únicamente por los implicados en su desarrollo, sin embargo, se tiene como trabajo futuro aplicarlo a una escuela de Ciudad de México.

### **2.1.5 Designing a Game for Learning Math by Composing: a Finnish Primary School Case [15]**

Diseñado en Finlandia, Harmony Hippo es un videojuego musical para enseñar matemáticas a alumnos de primaria. Para su implementación se utilizó el motor de videojuego Unity 3D con un enfoque iterativo de 3 ciclos basado en el Design Science in Information Systems Research [16].

Se presentó el videojuego a 6 estudiantes y 2 docentes. Las características más resaltantes por parte de los estudiantes fue su interés por seguir jugando y crear sus propias melodías. Los docentes resaltaron esta iniciativa de los estudiantes por tratar de ganar el desafío que presenta Harmony Hippo. Sin embargo, no se presentaron pruebas para medir si el desempeño en matemáticas mejoró después de interactuar con el videojuego.

### **2.1.6 Desing of Digital Game-Based Learning System for Elementary Mathematics Problem Solving [17]**

Los problemas matemáticos se presentan en términos de lenguajes, y no solo como operaciones matemáticas. Es por ello que en Taiwan se realizó un videojuego para

convertir y simular estos problemas matemáticos de multiplicación y división verbales en imágenes.

El estudiante debe comprender primero la narrativa y luego, los conocimientos matemáticos. Esta comprensión se logró mediante imágenes, ya que una imagen puede ser absorbida y procesada con mayor rapidez por los estudiantes.

Para el experimento se utilizó el método de evaluación cuasiexperimental, teniendo un total de 23 estudiantes entre 11 y 12 años de edad, cursando el cuarto grado de primaria. Después de una preprueba y posprueba del videojuego se lograron resultados positivos en el grupo experimental de un total de 79.806 de media contra un 70.933 del grupo control.

### **2.1.7 Game Based Learning for math learning - iFractions case study [18]**

iFractions es un videojuego con el objetivo de enseñar fracciones matemáticas a estudiantes del nivel primario en colegios de Brasil y Perú. Fue implementado con el *framework* de JavaScript Phaser.js, usando el módulo LMS (Learning Management System) para la evaluación de los estudiantes.

La evaluación del videojuego se realizó tanto en docentes como en estudiantes, logrando los siguientes resultados:

- Docentes: Fueron evaluados 29 docentes. Todos los docentes no tuvieron una explicación previa al videojuego, de los cuales el 76% completó todos los niveles, y el 24% uno o dos niveles.
- Estudiantes: Fueron evaluados 78 niños, de los cuales el 75% logró completar satisfactoriamente el primer nivel, el segundo y tercer nivel con un 19% y 6% respectivamente.

Los estudiantes jugaron 14,5 minutos en promedio, logrando un 63,62% de éxito en los niveles que ingresaron.

### **2.1.8 Math Detective - Digital Game-based Mathematical Error Detection, Correction and Explanation [19]**

Aunque los errores pueden parecer un fracaso para los estudiantes, estos resultan ser un benéfico potencial para el aprendizaje. Math Detective es un videojuego para mejorar el aprendizaje en matemáticas, donde los estudiantes toman el rol de detectives descubriendo los errores en una operación de multiplicación.

El videojuego fue implementado siguiendo la metodología de Fisher y Lipson [20] que consta de tres fases principales: Detección de errores y corrección, explicación del error e intercambio de explicación y modificación.

Siendo evaluados 24 estudiantes del cuarto grado de primaria en una escuela de Taiwán se encontraron dificultades en la segunda fase debido a que muchos estudiantes detectaron el error en la multiplicación, pero no sabían explicar la razón del mismo.

Math Detective incentivó a un conflicto cognitivo entre los estudiantes, donde ellos defendían su postura e intentaban averiguar porqué de la postura de sus compañeros, y por consecuencia, se aprendía mejor el procedimiento de multiplicación.

### **2.1.9 Developing a Mobile App for Game-based Learning in Middle School Mathematics Course [21]**

Para mejorar el aprendizaje de matemáticas en los temas: Números y cantidad, forma y espacio, probabilidad y estadística, y álgebra; se implementó un videojuego para celular, teniendo como público objetivo niños entre 5 a 7 años de edad.

El videojuego fue evaluado en una escuela de Taiwán, optando por un diseño cuasiexperimental. Fueron tres clases de estudiantes de quinto grado de primaria, siendo un total de 64 estudiantes como grupo experimental.

Después de un examen de matemáticas, el grupo experimental tuvo un aumento de 3.9 puntos a comparación del grupo de control. Además, se realizaron cuestionarios, donde la mayoría de los estudiantes prefieren materiales de



aprendizaje digitales interactivos, ya que aumenta su interés en el curso de matemáticas.

#### **2.1.10 Math Education Game for Primary School [22]**

Tiki-Taka Matematika es un videojuego implementado para estudiantes del primer grado de primaria en un colegio de Indonesia. Se tratan temas como la adición y la sustracción de acuerdo a la malla actual de la escuela.

El videojuego ha sido implementado para computadoras o laptops, en 2D. Después de una investigación de la metodología de enseñanza por parte de los docentes, se adaptó la metodología ágil SCRUM para poder desarrollar el videojuego logrando las siguientes etapas:

- Datos previos al desarrollo: Incluye el concepto principal, diseño básico, materiales y jugabilidad básica del videojuego.
- Datos adicionales: Datos que no fueron considerados en la anterior etapa, además se identifican que elementos del videojuego deben mejorarse y/o tener mayor prioridad.
- Datos de entrada del probador de videojuego: Datos recibidos mediante la presentación de prototipos a jugadores, recibiendo retroalimentación de los mismos.
- Datos de perfeccionamiento: Última inspección y realización de pruebas para garantizar la calidad del videojuego.

Después de la evaluación a los estudiantes, se demostró que los niños pudieron jugar y superar los desafíos de Tiki-Taka Matematika con mucha facilidad en la jugabilidad del videojuego (controles, cambios escenarios) debido a su familiaridad con los mismos.

#### **2.1.11 Space Chain A math game for training Geometric and Arithmetic Progressions [23]**

En España se diseñó Space Chain, un videojuego para dispositivos móviles IOS, con el objetivo principal de enseñar la progresión aritmética y geométrica, logrando la formación de secuencias.

El videojuego fue implementado con el *framework* CBL (*Challenge Based Learning*) para la validación del mismo como herramienta didáctica para la enseñanza de conceptos matemáticos.

Este *framework* explica que el alumno adquiere conocimiento de acuerdo a su área de interés, contando con tres fases principales: participar, investigar y actuar.

Se utilizó la heurística de Nielsen para validar el videojuego contando con los siguientes puntos: desarrollo de habilidades, estrategia del tutorial, desafío, inmersión, historia, usabilidad y mecánica del videojuego.

Estos puntos fueron evaluados por seis docentes del *Laboratory of Higher Education Institution* (HEI), obteniendo buenos resultados y críticas constructivas sobre la aplicación:

- El videojuego es atractivo para el público objetivo que es presentado debido a sus colores y originalidad de diseños.
- Presenta los conceptos y metodología para la enseñanza del tema que se trata.
- El uso de tutoriales con imágenes y animaciones obtiene mejores resultados a comparación del uso de textos y descripciones extensas.

#### **2.1.12 Plataforma M-learning para incentivar el aprendizaje de Matemáticas en estudiantes de tercero de primaria basado en Unity [24]**

Micco es un videojuego implementado para incentivar el aprendizaje de matemáticas en estudiantes del tercer grado de primaria, siendo guiados por el Diseño Curricular Nacional de Educación Básica Regular del Perú.

El videojuego ha sido desarrollado usando *Machine Learning* y haciendo uso de la metodología V para el ciclo de vida del software. Micco es el personaje principal por el que pasa por varios retos en los cuales los estudiantes tienen que resolverlos para lograr que Micco llegue a su meta.

La evaluación se realizó mediante un cuestionario a los estudiantes de tres colegios de la ciudad de Arequipa, siendo un total de 116 alumnos participantes que completaron los cuestionarios correctamente y además jugaron Micco;

mostrando una gran aceptación del uso de los videojuegos como herramienta de estudio.

### 2.1.13 Conclusiones sobre el estado del arte

En base a la revisión realizada se encontraron diferentes soluciones que abordan conceptos matemáticos en los colegios haciendo el uso de videojuegos en diferentes plataformas, obteniendo resultados satisfactorios sobre el incentivo que generan los videojuegos en los estudiantes, tal como se muestra en la Tabla 7.

Tabla 7. Resumen de hallazgos en el estado del arte

	<u>Aplicaciones Móviles</u>	<u>Aplicaciones Web</u>	<u>Aplicaciones de Escritorio</u>
Rango de Edad	3-11 años	9-15 años	7-10 años
Herramienta de Implementación	<i>Unity 3D</i>	<i>Meteor, Phaser</i>	<i>Unity 3D</i>
Conceptos matemáticos	Geometría, sistema numérico	Física, operaciones matemáticas	Matemática contextual, operaciones matemáticas
Resultados	Positivos*	Positivos*	Positivos*

Fuente: Elaboración Propia

\* Se presentan en todos los trabajos

## 2.2 Marco Teórico

### 2.2.1 Aprendizaje

El concepto de aprendizaje es importante, sin embargo, visto de diferentes puntos de vista sobre las causas, proceso y/o consecuencias del mismo, es difícil encontrar una definición aceptada por todos.

La RAE define al aprendizaje como “Acción y efecto de aprender algún arte, oficio u otra cosa”, y en términos psicológicos como: “Adquisición por la práctica de una conducta duradera” [25].

Dale H. Schunk [26] define al aprendizaje como: “Un cambio perdurable en la conducta o en la capacidad de comportarse de cierta manera, el cual es resultado de la práctica o de otras formas de experiencia”. Además, el aprendizaje es

inferencial, es decir, no se puede observar de manera directa, se observa mediante resultados. Se definen los siguientes criterios del aprendizaje:

- Implica un cambio en la conducta, las personas que aprenden adquieren la capacidad de realizar alguna acción de distinta forma a los demás.
- Perdura a lo largo del tiempo, durante el resto de nuestra vida; es por ello que no se puede calificar como aprendizaje cambios de poca duración.
- Ocurre por medio de la experiencia, entre más se utiliza lo aprendido, mayor es el aprendizaje. Por ejemplo, el lenguaje, una persona más letrada podrá comunicarse mejor con las otras, a comparación de un niño en su etapa de la infancia.

La motivación es un factor muy importante, ya que puede afectar en todas las fases del aprendizaje. Tanto motivación como aprendizaje se encuentran relacionados, pero no son codependientes; una persona puede aprender sin estar motivado, sin embargo, se destaca que la motivación aumenta la atención e influye de manera positiva la forma en la que se procesa la información.

### **2.2.2 Aprendizaje matemático**

La matemática se ha convertido como parte de la formación integral del ser humano, debido a que se encuentra presente durante toda nuestra vida desde muy temprana edad. La matemática no solo puede ser aprendida mediante una formación escolar, con mallas curriculares establecidas; sino también, en nuestra vida cotidiana que se encuentran acordes con nuestra necesidades e intereses.

Es por ello que los profesores se encuentran en constante investigación de nuevas metodologías didácticas para el aprendizaje de matemática en los estudiantes, Mora [27] muestra dos modelos didácticos que se observaron alrededor del mundo (Fig. 1), siendo más popular el MODELO A.

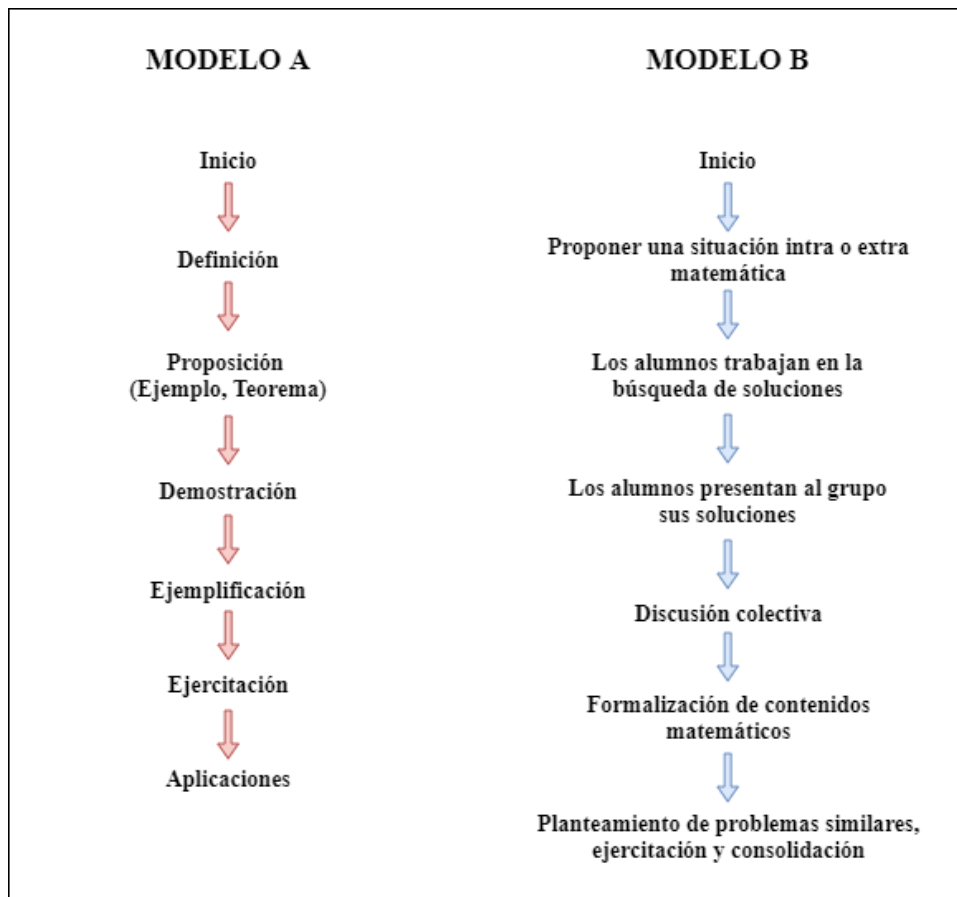


Fig. 1 Modelos didácticos observados en clases de matemáticas [27]  
Fuente: Elaboración propia

Estudiantes y docentes influyen en el éxito del proceso de aprendizaje de las matemáticas. Ambos son responsables para obtener los resultados y logros esperados. Durante la aplicación de los modelos presentados anteriormente se encuentran siete etapas básicas del proceso de aprendizaje y enseñanza de las matemáticas:



Fig. 2 Etapas básicas del proceso de aprendizaje y enseñanza de las matemáticas [27]  
Fuente: Elaboración propia

**1. Introducción didáctica:** Mención breve del tema que se va enseñar. Una de las maneras de realizar esta etapa es preguntando a los estudiantes acerca del tema vinculándolo con nuestra vida cotidiana, logrando e incentivando el debate o reflexión.

Este vínculo crea una realidad ficticia, donde el lenguaje natural, manipulación de objetos se convierten en el tema matemático a tratar.

**2. Desarrollo:** El objetivo de esta etapa es hacer que los estudiantes aprendan nuevos conocimientos matemáticos, y/o aprendan el procedimiento matemático del tema a tratar.

La etapa normalmente es controlada en su totalidad por el docente, realizando preguntas y esperando respuestas de los estudiantes (que normalmente no participan, al desconocer el tema); de igual forma se suelen formar grupos o parejas para lograr una retroalimentación.

**3. Vinculación con otros conceptos matemáticos:** El objetivo es que los estudiantes deben comprender la importancia de la conexión de los conceptos matemáticos.

Normalmente esta etapa suele estar implícita durante el desarrollo de las demás etapas; un ejemplo común son las figuras geométricas, que pueden ser tratados en temas de geometría básica, hasta temas de trigonometría o sólidos.

**4. Consolidación de los nuevos conocimientos matemáticos:** Comprender y reflexionar sobre el concepto matemático aprendido y hacerlo perdurar por el resto de nuestra vida. Muchas veces los docentes utilizan la repetición para lograr este objetivo; sin embargo, es mejor profundizar y razonar en pocos ejercicios que realizar muchos de manera mecánica.

La consolidación está unida a la calidad de los contenidos matemáticos, las estrategias de enseñanzas y sobre todo la relación entre matemática y realidad [28].

**5. Profundización de los conocimientos matemáticos:** Seguida de la fase de consolidación de los conocimientos matemáticos, los docentes tienen la tarea de encontrar qué estudiantes necesitan con mayor urgencia profundizar algunos conceptos, explicándolos de distintas formas hasta se logre la comprensión del mismo.

**6. Inspección de los nuevos conocimientos matemáticos:** Evaluar al estudiante sobre los conceptos matemáticos enseñados; no obstante, aún no existe una respuesta completamente satisfactoria a cómo determinar si los estudiantes realmente han logrado las metas planteadas en su plan curricular.

Para lograr el éxito del aprendizaje matemático no solo depende del estudiante y como resolvió su evaluación; el docente tiene un gran protagonismo realizando su trabajo de manera más didáctica generando motivación y como consecuencia, mejores resultados.

**7. Corrección, eliminación de errores y concepciones erróneas:** Trabajar el error del estudiante como una oportunidad para mejorar su aprendizaje. Los errores forman parte del trabajo matemático, sin embargo, no se aprovecha de

esta forma; la enseñanza tradicional penaliza y castiga el error del estudiante, sin generar un tipo de autoevaluación para mejorar el aprendizaje.

### 2.2.3 Videojuegos

Se encuentran múltiples definiciones sobre los videojuegos, debido a su constante evolución e incorporación de características nuevas gracias al avance tecnológico. La RAE define un videojuego como “Dispositivo electrónico que permite, mediante mandos apropiados, simular juegos en las pantallas de un televisor, una computadora u otro dispositivo electrónico” [29]

Juárez y Mombiela [30] definen a un videojuego como un programa informático diseñado para el entretenimiento y la diversión que se pueden utilizar a través de varios soportes como las videoconsolas, los ordenadores o los teléfonos móviles.

A comparación de la definición de la RAE, Juárez y Mombiela especifican que son diseñados para el entretenimiento y la diversión del usuario que lo utilice; esta definición es más acertada para la actualidad.

#### 2.2.3.1 Clasificación de los videojuegos

Una manera de clasificar los videojuegos es mediante su género, en el siguiente mapa conceptual (Fig. 3) se observa una clasificación acertada e incluso algunos con subgéneros del mismo.

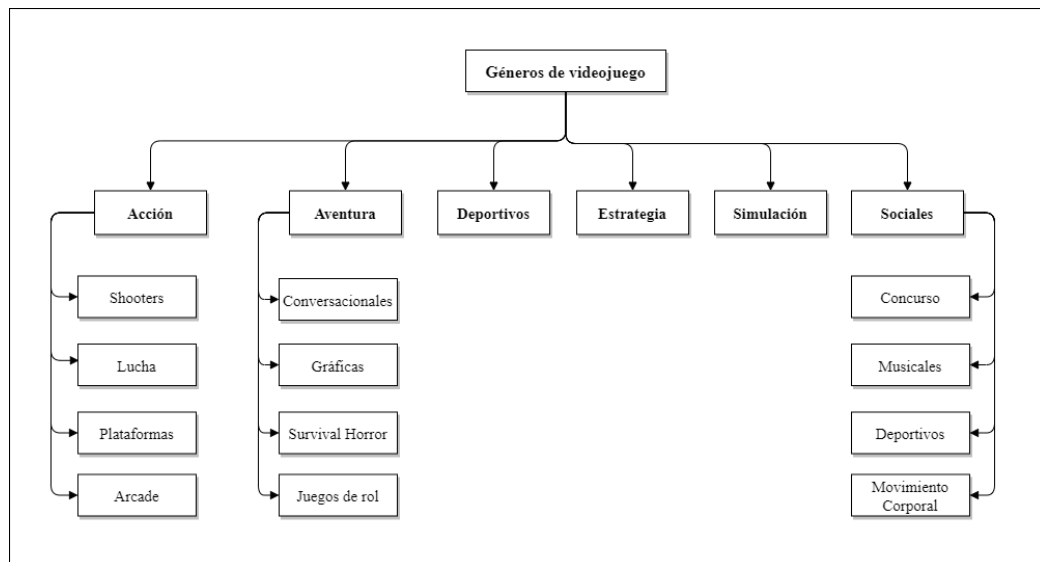


Fig. 3 Géneros de videojuego [29]

Fuente: Elaboración propia

A continuación, se detallarán cada uno de los géneros y subgéneros de los mismos:



1. **Videojuegos de acción:** Los videojuegos de acción se caracterizan por retar las habilidades psicomotrices del jugador, mostrando estímulos en la pantalla. Para mantener la motivación del jugador es necesario que se demuestre una mejora gracias a la práctica constante del videojuego. En este género se encuentran los siguientes subgéneros:
  - *Shooters*: también llamados juegos de disparo. El objetivo principal es derribar al otro jugador para continuar con el videojuego. Por ejemplo: Valorant, Counter Strike Go.
  - Lucha: Dos personajes dentro del videojuego se enfrentan en combate uno contra uno, o en equipos. El objetivo es vencer a tu adversario. Por ejemplo: Dragon Ball Z Fighters, Mortal Kombat.
  - Plataformas: El personaje principal debe alcanzar la meta superando múltiples plataformas que cada vez lo llevan más alto. Por ejemplo: Mario vs Donkey Kong.
  - Arcade: Subgénero para los juegos clásicos de las máquinas que llevan el mismo nombre. Por ejemplo: Pacman, Space Invaders.
2. **Videojuegos de aventura:** Este subgénero presenta la historia de un personaje que resuelve distintos desafíos y pruebas para finalizar con su historia; el jugador se siente identificado con el mismo, investigando y explorando distintos escenarios para lograr todos los objetivos. Cuenta con los siguientes subgéneros:
  - Conversacionales: Se le presentan diferentes diálogos al jugador, dependiendo del diálogo escogido, la historia cambia completamente para el protagonista. Por ejemplo: The Walking Dead.
  - Gráficas: Este subgénero presenta diferentes rompecabezas para ser resueltos por el jugador.
  - Supervivencia: El jugador tiene la misión de sobrevivir a todos los obstáculos que se le presentan mientras juega. Por ejemplo: Left 4 Dead.
  - Juegos de rol: El jugador inventa su propio personaje, desde lo físico hasta las actividades que se encarga, por ejemplo, de un policía. Al ser un personaje propio tiene total libertad de acción e interacción con otros jugadores. Por ejemplo: GTA V *Roleplay*.

3. **Videojuegos deportivos:** El videojuego presenta un deporte real (fútbol, básquet, tenis), respetando todas las reglas del mismo. La mayor atracción de estos videojuegos es el realismo y la actualización; los videojuegos se encuentran actualizados a las ligas y campeonatos deportivos actuales, entrenadores, estadios, fichajes. Por ejemplo: FIFA, PES.
4. **Videojuegos de estrategia:** El objetivo principal de este género es planificar, organizar y ejecutar acciones de manera inteligente mediante la obtención de información presentada en los escenarios. El razonamiento estratégico es clave en este género, debido a que todos los jugadores cuentan con la misma información, pero distinta forma de utilizarla. Por ejemplo: Age of Empires, StarCraft.
5. **Videojuegos de simulación:** Este género de videojuegos presenta de forma muy realista algún tipo de actividad. El jugador simula realizar esta actividad mediante los escenarios y controles del videojuego. Algunas de las simulaciones más conocidas son el manejo de una nave, o un auto de carrera. Por ejemplo: Rally Point, Star Conflict.
6. **Videojuegos sociales:** Tiene como objetivo el compartir con un grupo de personas, donde se puede fomentar la competitividad, o la expresión de emociones y sentimientos mientras se juega. Cuenta con los siguientes subgéneros:
  - Concursos: Subgénero donde se presentan preguntas al jugador principal, que, si son contestadas correctamente, logran el premio mayor. Por ejemplo: ¿Quién quiere ser millonario?
  - Musicales: Subgénero que simula diferentes escenarios donde se utiliza la música (canto e instrumentos) como medio de entretenimiento. Por ejemplo: Guitar Hero.
  - Deportivos: Este género aprovecha los controles especializados para simular las acciones corporales de un deporte real (tenis, ping pong, fútbol). Por ejemplo: Super Mario Wii.
  - Movimiento corporal: El jugador imita el movimiento físico del personaje del videojuego. Se utilizan bailes y canciones de moda para incentivar su jugabilidad. Por ejemplo: Just Dance.

### 2.2.4 Videojuegos educativos

Actualmente existen muchas aplicaciones que se despliegan en computadoras, consolas o celulares que sirven para enseñar conceptos que se aprenden normalmente en la escuela, sin embargo, muchas de estas aplicaciones se pueden considerar como videojuegos educativos. La taxonomía de Lepper y Malone [31] nos indica qué características debe tener un videojuego:

1. **Desafío:** El videojuego debe motivar al jugador a ganar, a seguir compitiendo para lograr los objetivos. Esta motivación se logra mediante los niveles que ofrece el videojuego, siendo la retroalimentación una pieza clave para lograr superarlos.
2. **Curiosidad:** Despertar la curiosidad del jugador tanto sensorialmente (mediante efectos visuales del videojuego que se presentan en la pantalla, pueden ser luces, imágenes, videos, explosiones) como cognitivamente (acertijos, misterios, rompecabezas sin resolver).
3. **Control:** El jugador debe contar sentir que controla al personaje en su totalidad, al tomar decisiones para que el mismo logre el objetivo del nivel.
4. **Fantasía:** El videojuego debe contar con analogías entre el mundo real y el mundo del videojuego.

Una vez el videojuego cuenta con estas características, para que sea considerado un videojuego educativo, se tiene que enseñar algo, un concepto; esto quiere decir que los jugadores aprenderán nuevas cosas jugando; pero, sobre todo, tienen que aprender sin darse cuenta, de forma implícita.

### 2.2.4 Aplicación Web

El acceso al internet para una persona natural ya no es algo ajeno, todos se encuentran familiarizados con su uso, es por ello que es importante el concepto de las aplicaciones web; ya que son utilizadas en nuestro día a día.

Las aplicaciones web en el fondo son aplicaciones de arquitectura cliente/servidor, donde el cliente solicita servicios y el servidor responde a esta solicitud. Luján S. indica que esta arquitectura permite la separación de funciones en tres niveles: Lógica de presentación, negocio y datos [32].

**Lógica de presentación:** Es la encargada de la comunicación directa con el usuario final. Tiene como principal objetivo el obtener la información del usuario,

enviarla a la lógica de negocio para ser procesada y finalmente comunicar los resultados de este procesamiento al usuario.

**Lógica de negocio:** Es el intermediario entre el usuario y los datos. Gestiona la información a nivel de procesamiento, es decir, interactúa con la lógica de datos para cumplir la petición del usuario y entregarla a la lógica de presentación.

**Lógica de datos:** Gestiona los datos a nivel de almacenamiento, es decir, almacena registros, y asegura su integridad.

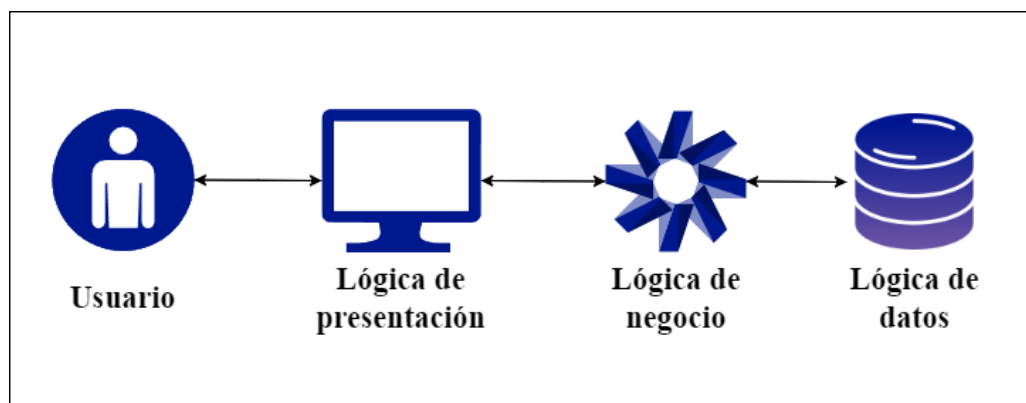


Fig. 4 Separación de funciones  
Fuente: Elaboración propia

En las aplicaciones web, el cliente suele ser el navegador web (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera) y el servidor un servidor web. Estos se comunican a través del protocolo HTTP.

El protocolo HTTP es un protocolo de comunicación TCP/IP que se emplea en el uso del internet y es el intermediario entre el cliente y el servidor.

### 2.2.5 Modelo Espiral de Boehm

Boehm, B. [33] propuso un modelo para desarrollar software que se basa en el riesgo: el modelo espiral.

A diferencia de modelos como el cascada que son secuenciales, este modelo es representado como una espiral; es decir, con varias iteraciones donde en cada iteración contiene las siguientes etapas: establecimiento de objetivos, valoración y reducción del riesgo, desarrollo y validación, y planeación.

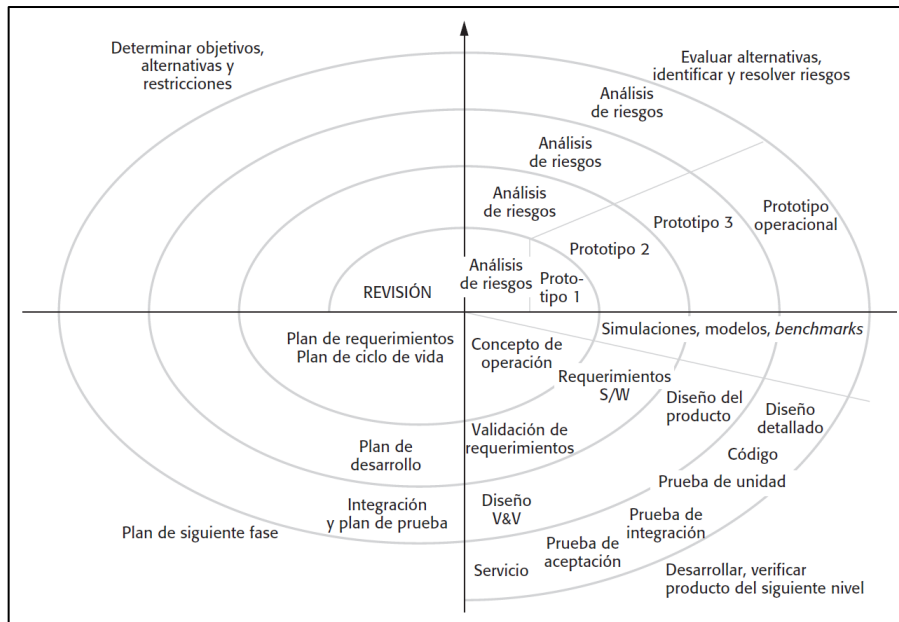


Fig. 5 Modelo en espiral de Boehm [34]  
Fuente: Elaboración propia

**Establecimiento de objetivos:** En esta etapa se recopilan los objetivos, alternativas y restricciones que puede tener el software que se desarrollará. Los objetivos se relacionan con la iteración actual y con el producto final del software. Las alternativas y restricciones que tiene el software son identificadas en esta etapa.

**Valoración y reducción del riesgo:** Se identifican los riesgos teniendo como base las alternativas y restricciones de la primera etapa. Estos riesgos son analizados para crear un plan de acciones reduciendo el impacto de los mismos. Los riesgos pueden incluir factores humanos (falta de experiencia de los desarrolladores) o materiales (falta de equipos como laptops, monitores, etc.)

**Desarrollo y validación:** Se diseña, desarrolla, y se prueba el producto de software. Todas estas actividades tienen que estar basadas en la identificación y planificación de los riesgos encontrados en la etapa anterior.

**Planeación:** Se realiza una evaluación al proyecto y a la actual iteración. Si el desarrollo del software lo requiere, se planifica una siguiente iteración con los aprendizajes obtenidos de la actual.

### 2.2.6 Framework

Actualmente la industria de la tecnología es muy acelerada, por lo que un software exitoso tiene que ser capaz de adaptarse a este avance tecnológico sin sufrir un cambio que sea muy costoso en recursos y tiempo. Muchos desarrolladores redescubren y reinventan funcionalidades que se adaptan a un problema muy común, y la adapta estrictamente a su propio software, volviéndolo así monolítico y poco modular; generando una escalabilidad escasa. Para evitar este gasto de recursos es mejor utilizar *frameworks*.

Un *framework* es un conjunto de componentes, clases, objetos que colaboran entre sí para proporcionar una arquitectura reutilizable para un grupo de aplicaciones que comparten un propósito [35]. Reutilizando el diseño, implementación y validación.

Actualmente existen muchos *frameworks* para el desarrollo de videojuegos, algunos de ellos son:

**Pygame**, tiene como lenguaje principal Python. Su principal función es el desarrollo de videojuegos 2D orientado al manejo de animaciones (*sprites*). El *framework* es de código abierto.

**Pandas3D**, este *framework* tiene como especialidad la creación de videojuegos en 3D. Es de código abierto teniendo como principal lenguaje de programación a Python aunque su motor esté escrito en C++.

**Allegro**, *framework* de código abierto desarrollado en lenguaje C. Con el mismo puede manipular las imágenes y dispositivos de entrada para la creación de videojuegos.

**LWJGL**, Lightweigh Java Game Library es un *framework* para videojuegos escritos en el lenguaje Java. Permite la utilización de recursos externos como joysticks o simuladores de volantes.

**Phaser**, *framework* que permite desarrollar videojuegos en *HTML5* para dispositivos de escritorio y móviles. Utiliza *Javascript* como principal lenguaje para sus funciones y renderizado, como por ejemplo el renderizado *WebGL*, que es una renderización de gráficos en 3D.

**Love2D**, *framework* para el desarrollo de videojuegos 2D en el lenguaje de programación Lua. Permite la creación de videojuegos en distintos sistemas operativos como: Windows, Mac OS X, Linux, Android y IOS.

## **Capítulo 3. Método de Investigación**

### **3.1 Modelo en Espiral de Boehm**

La implementación del videojuego se realizó haciendo uso del modelo en espiral de Boehm. El modelo realiza tantos ciclos sean necesarios para finalizar el software a implementar. Cada ciclo se divide en cuatro etapas, cada una de ellas con sus respectivos objetivos específicos para MathDragon.

#### **3.1.1 Establecimiento de Objetivos**

- Definir/redefinir requerimientos con los docentes y expertos encargados de la educación de los estudiantes del primer grado de primaria.
- Diseñar/rediseñar una arquitectura de software adecuada para la implementación de MathDragon.
- Identificar los riesgos que se encuentran en el videojuego a implementar/implementado.

#### **3.1.2 Valoración y Reducción del Riesgo**

- Desarrollar prototipos (mockups) que satisfagan los requerimientos encontrados. Los prototipos serán evaluados por docentes y expertos.
- Analizar los riesgos, especificando su valor y nivel de cada uno de ellos.
- Describir un plan de acción frente a los riesgos encontrados, detallando su estrategia de contingencia.

#### **3.1.3 Desarrollo y Validación**

- Desarrollar versión del videojuego teniendo como guía los prototipos evaluados por los docentes y expertos.
- Validar y realizar pruebas de sistema a la versión actual del videojuego.
- Presentar la versión actual del videojuego a los estudiantes del primer grado de primaria.

#### **3.1.4 Planeación**

- Revisar el proyecto y evaluar si es necesario otro ciclo espiral.
- Desarrollar un nuevo plan para la siguiente iteración del ciclo espiral / fin del ciclo.



## 3.2 Ingeniería de Requerimientos

### 3.2.1 Entrevistas

MathDragon será probado con los alumnos del primer grado de primaria del colegio San Juan Bautista de La Salle de Arequipa – Perú. Es por ello que se contactó con los tutores del grado.

Las entrevistas tuvieron como objetivo principal entender el método de enseñanza, la problemática de los estudiantes con el curso de matemática, y opiniones sobre el uso de los videojuegos en la enseñanza. En el Anexo 4 se encuentra un resumen de las respuestas por pregunta de los tutores entrevistados.

Una vez realizadas las entrevistas se capturaron los requerimientos de MathDragon.

### 3.2.2 Requerimientos Funcionales

*Tabla 8. Representación matemática*

Título	REQF01 – Representación matemática
Función	Enseñar el concepto de intra-matemática a los estudiantes.
Descripción	El videojuego debe ser capaz de reforzar el concepto de intra-matemática realizando operaciones matemáticas presentando por el videojuego
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 9. Matemática contextual*

Título	REQF02 – Matemática contextual
Función	Enseñar mediante el enfoque contextual a los estudiantes.
Descripción	El videojuego debe presentar un problema matemático contextual, de tal manera que el estudiante sea capaz de reconocer la temática del videojuego y resolver la problemática de forma natural.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 10. Mensajes motivacionales*

Título	REQF03 – Mensajes motivacionales
Función	Presentación de mensajes de motivación al estudiante.
Descripción	El videojuego contará con mensajes motivacionales, cuando logre el objetivo se le presentará un mensaje ganador. En el caso que el estudiante pierda, se incentivará a mejorar su desempeño.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 11. Obstáculos/retos en el videojuego*

Título	REQF04 – Obstáculos/retos en el videojuego
Función	Presentar retos en el videojuego.
Descripción	El videojuego debe contar con un nivel de dificultad incentivando la competitividad del estudiante.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 12. Control de tiempo de juego*

Título	REQF05 – Control de tiempo de juego
Función	Contar con un control de tiempo de juego.
Descripción	El videojuego debe tener un tiempo límite de jugabilidad. Cuando se supere el tiempo de jugabilidad, el videojuego debe terminar.
Prioridad	Media.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 13. Nivel Conseguir Comida*

Título	REQF06 – Nivel Conseguir Comida
Función	Etapas donde se refuerce el concepto de adiciones.
Descripción	El videojuego debe contar con una etapa que refuerce el concepto de adición siendo representada la intra-matemática y matemática contextual.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 14. Nivel Destruir Venenos*

Título	REQF07 – Nivel Destruir Venenos
Función	Etapas donde se refuerce el concepto de adiciones.
Descripción	El videojuego debe contar con una etapa que refuerce el concepto de sustracción siendo representada la intra-matemática y matemática contextual.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 15. Contador de vidas*

Título	REQF08 – Contador de vidas
Función	Visualizar el contador de vidas/oportunidades del jugador.
Descripción	El videojuego deberá contar con un contador de vidas. El contador debe contar con por lo menos 3 vidas; debe ser actualizado y visible al estudiante en todo momento.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 16. Movimiento y controles*

Título	REQF9 – Movimiento y controles
Función	Realizar el movimiento del personaje principal en el videojuego.
Descripción	El videojuego debe contar con un movimiento/controles del personaje lo más sencillo posible para el estudiante.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 17. Ayudas/pistas en el problema matemático*

Título	REQF10 – Ayudas/pistas en el problema matemático
Función	Brindar ayuda al estudiante para la resolución del problema.
Descripción	El videojuego debe presentar ayudas o pistas al estudiante. Si el estudiante responde de manera incorrecta, se deberán presentar diferentes ayudas que lo ayuden a resolver el problema con mayor facilidad.
Prioridad	Media.
Tipo	Funcional.

Fuente: Elaboración propia

*Tabla 18. Persona y temática*

Título	REQF11 – Personaje y temática
Función	El videojuego cuenta con personajes y temática.
Descripción	El videojuego debe tener un personaje y temática que incentive al estudiante a jugarlo.
Prioridad	Alta.
Tipo	Funcional.

Fuente: Elaboración propia

### 3.2.3 Requerimientos no Funcionales

Tabla 19. Aprendizaje del videojuego

REQNF01 – Aprendizaje del videojuego	
Función	El aprendizaje del videojuego debe ser eficiente.
Descripción	El tiempo de aprendizaje del videojuego (controles, desafíos, objetivos) por el estudiante debe ser menor a 30 minutos.
Prioridad	Alta.
Tipo	No funcional - Usabilidad

Fuente: Elaboración propia

Tabla 20. Diseño responsive

REQNF02 – Diseño responsive	
Función	Correcta visualización del videojuego en diferentes resoluciones.
Descripción	El videojuego debe poseer un diseño responsive para garantizar que pueda ser visualizado y utilizado correctamente desde diferentes resoluciones de pantalla de los computadores o laptops.
Prioridad	Media.
Tipo	No funcional - Usabilidad

Fuente: Elaboración propia

Tabla 21. Disponibilidad del videojuego

REQNF03 – Disponibilidad del videojuego	
Función	Brindar disponibilidad de uso del videojuego.
Descripción	El videojuego debe estar disponible siempre que el estudiante/usuario intente acceder a él. Debe contar con un hosting estable.
Prioridad	Alta.
Tipo	No funcional - Dependibilidad

Fuente: Elaboración propia

Tabla 22. Ejecución en navegadores web

REQNF04 – Ejecución en navegadores web	
Función	Soporte de navegadores web en el videojuego.
Descripción	El videojuego debe ser implementado para que sea ejecutado sin problemas por navegadores web que soporten HTML5.
Prioridad	Alta.
Tipo	No funcional – Dependibilidad

Fuente: Elaboración propia

### 3.3 Riesgos del Software

Se realizó una matriz (Anexo 6) donde se analiza y evalúa cada uno de los riesgos encontrados, dando a conocer su probabilidad, gravedad, valor, nivel y estrategia de contingencia. La matriz es guiada por la leyenda de la Tabla 23 [37].

Tabla 23. Matriz de riesgos

			Gravedad				
			Muy baja	Baja	Media	Alta	Muy alta
			1	2	3	4	5
Probabilidad	Muy alta	5	5	10	15	20	25
	Alta	4	4	8	12	16	20
	Media	3	3	6	9	12	15
	Baja	2	2	4	6	8	12
	Muy baja	1	1	2	3	4	5

<b>Riesgo muy grave</b>		Requiere medidas preventivas urgentes. No se debe iniciar el proyecto sin la aplicación de medidas preventivas urgentes y sin acotar sólidamente el riesgo.
<b>Riesgo importante</b>		Medidas preventivas obligatorias. Se deben controlar fuertemente las variables de riesgo durante el proyecto.
<b>Riesgo apreciable</b>		Estudiar económicamente si es posible introducir medidas preventivas para reducir el nivel de riesgo. Si no fuera posible, mantener las variables controladas.
<b>Riesgo marginal</b>		Se vigilará, aunque no requiera, medidas preventivas de partida.

Fuente: Region Administrativa y de Planeación Especial - RAPE

### 3.4 Arquitectura del Software

#### 3.4.1 Diagrama de Casos de Uso

La realización de diagramas de casos de uso ayuda a entender la forma en la que trabaja el *software* y cómo responde a los requerimientos establecidos por el cliente.

Después de un análisis de requerimientos, se tiene el siguiente Diagrama de Casos de Uso (Fig. 6)

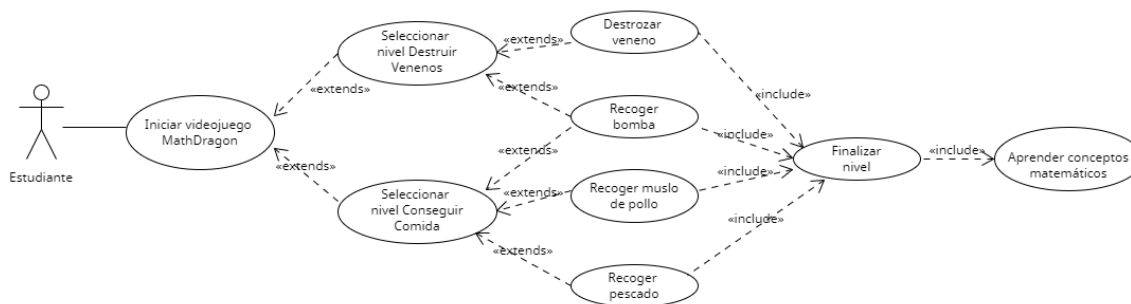


Fig. 6 Diagrama de Casos de Uso de MathDragon.  
Fuente: Elaboración propia

Tabla 24. Iniciar videojuego MathDragon

ID - Nombre	CS01 – Iniciar videojuego MathDragon
Descripción	El estudiante iniciará el videojuego MathDragon desde su computador o laptop. El videojuego se ejecutará una vez ingrese al URL que lo contiene.
Actor	Estudiante
Precondiciones	Contar con un dispositivo que tenga instalado un navegador web.
Flujo normal	<ol style="list-style-type: none"> <li>1. Abrir navegador web.</li> <li>2. Ingresar la URL del videojuego MathDragon en la barra de direcciones.</li> </ol>
Flujo alternativo	Ninguno
Postcondiciones	MathDragon ejecutado correctamente

Fuente: Elaboración propia

Tabla 25. Seleccionar nivel Conseguir Comida

ID - Nombre	CS02 – Seleccionar nivel Conseguir Comida
Descripción	El estudiante escoge el nivel Conseguir Comida haciendo clic en su tablero correspondiente. El nivel corresponde a la operación matemática de adición. Se cargan los recursos necesarios para el nivel.
Actor	Estudiante
Precondiciones	MathDragon ejecutado correctamente.
Flujo normal	<ol style="list-style-type: none"> <li>1. Videojuego MathDragon en ejecución.</li> </ol>

	<ol style="list-style-type: none"> <li>2. Selección de nivel Conseguir Comida en MathDragon.</li> <li>3. Inicio de nivel Conseguir Comida.</li> </ol>
Flujo alternativo	Ninguno
Postcondiciones	Ninguna

Fuente: Elaboración propia

*Tabla 26. Seleccionar nivel Destruir Venenos*

ID - Nombre	CS03 – Seleccionar nivel Destruir Venenos
Descripción	El estudiante escoge el nivel Destruir Venenos haciendo clic en su tablero correspondiente. El nivel corresponde a la operación matemática de sustracción. Se cargan los recursos necesarios para el nivel.
Actor	Estudiante
Precondiciones	MathDragon ejecutado correctamente.
Flujo normal	<ol style="list-style-type: none"> <li>4. Videojuego MathDragon en ejecución.</li> <li>5. Selección de nivel Destruir Venenos en MathDragon.</li> <li>1. Inicio de nivel Destruir Venenos.</li> </ol>
Flujo alternativo	Ninguno
Postcondiciones	Ninguna

Fuente: Elaboración propia

*Tabla 27. Destrozar veneno*

ID - Nombre	CS04 – Destrozar veneno
Descripción	El dragón hace colisión con el objeto de las botellas de veneno en el nivel Destruir Venenos. Cuando se realiza esta colisión el puntaje se actualiza en el tablero de puntajes del nivel actual.
Actor	Estudiante
Precondiciones	Nivel Destruir Venenos elegido.
Flujo normal	<ol style="list-style-type: none"> <li>1. El dragón realiza una colisión con la botella de veneno.</li> <li>2. Desaparece la botella de veneno.</li> <li>3. Se actualiza el tablero de puntajes.</li> </ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia



*Tabla 28. Recoger muslo de pollo*

ID - Nombre	CS05 – Recoger muslo de pollo
Descripción	El dragón hace colisión con el objeto del muslo de pollo en el nivel Conseguir Comida. Cuando se realiza esta colisión el puntaje se actualiza en el tablero de puntajes del nivel actual.
Actor	Estudiante
Precondiciones	Nivel Conseguir Comida elegido.
Flujo normal	<ol style="list-style-type: none"> <li>1. El dragón realiza una colisión con el muslo de pollo.</li> <li>2. Desaparece el muslo de pollo.</li> <li>3. Se actualiza el tablero de puntajes.</li> </ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia

*Tabla 29. Recoger pescado*

ID - Nombre	CS06 – Recoger pescado
Descripción	El dragón hace colisión con el objeto del pescado en el nivel Conseguir Comida. Cuando se realiza esta colisión el puntaje se actualiza en el tablero de puntajes del nivel actual.
Actor	Estudiante
Precondiciones	Nivel Conseguir Comida elegido.
Flujo normal	<ol style="list-style-type: none"> <li>1. El dragón realiza una colisión con el pescado.</li> <li>2. Desaparece el pescado.</li> <li>3. Se actualiza el tablero de puntajes.</li> </ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia

*Tabla 30. Recoger bomba*

ID - Nombre	CS07 – Recoger bomba
Descripción	El dragón hace colisión con el objeto de la bomba en el nivel Conseguir Comida o Destruir Venenos. Cuando se realiza esta colisión el número de vidas del dragón disminuye en uno. Además, se activa la animación de daño.
Actor	Estudiante
Precondiciones	Nivel Conseguir Comida/Destruir Venenos elegido.
Flujo normal	<ol style="list-style-type: none"><li>1. El dragón realiza una colisión con la bomba.</li><li>2. Desaparece la bomba.</li><li>3. Se actualiza el contador de vidas en el tablero.</li></ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia

*Tabla 31. Finalizar nivel*

ID - Nombre	CS08 – Finalizar nivel
Descripción	El nivel es finalizado si el dragón no cuenta con más vidas o si el temporizador de tiempo llegó a 0.
Actor	Estudiante
Precondiciones	Contador de vidas en 0 / Temporizador finalizado
Flujo normal	<ol style="list-style-type: none"><li>1. Temporizador finalizado o contador de vidas finalizado.</li><li>2. Inicio del tablero con el problema matemático</li></ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia

Tabla 32. *Aprender conceptos matemáticos*

ID - Nombre	CS09 – Aprender conceptos matemáticos
Descripción	<p>Se presentarán los conceptos de intra-matemática y matemática contextual en cada uno de los niveles del videojuego.</p> <p>En el nivel Conseguir Comida, la intra-matemática es representada en la adición de muslos de pollo y pescados que tiene que realizar el estudiante. Mientras que la matemática contextual se representa con la temática del personaje principal en conseguir su alimento, realizando la siguiente pregunta: ¿Cuánta comida ha recolectado Kalh en total?</p> <p>En el nivel Destruir Venenos, la intra-matemática es representada en la sustracción de botellas de veneno objetivo y botellas de veneno destrozadas. La matemática contextual es representada de la misma forma que en el nivel Conseguir Comida, la diferencia es en la pregunta: ¿Cuántas botellas de veneno le faltó destrozarse a Kalh?</p>
Actor	Estudiante
Precondiciones	Nivel Conseguir Comida/Destruir Venenos finalizado.
Flujo normal	<ol style="list-style-type: none"> <li>3. Videojuego MathDragon en ejecución.</li> <li>4. Selección de nivel en MathDragon.</li> <li>5. Inicio de nivel.</li> <li>6. Fin de nivel.</li> </ol>
Postcondiciones	Ninguna

Fuente: Elaboración propia

### 3.4.2 Vista Física

La vista física está conformada por dos nodos: Cliente web y servidor web. Del lado del servidor web se hace el uso del *framework* Phaser, debido a la familiaridad con su lenguaje principal programación *JavaScript*, y también por su comunidad activa de desarrolladores.

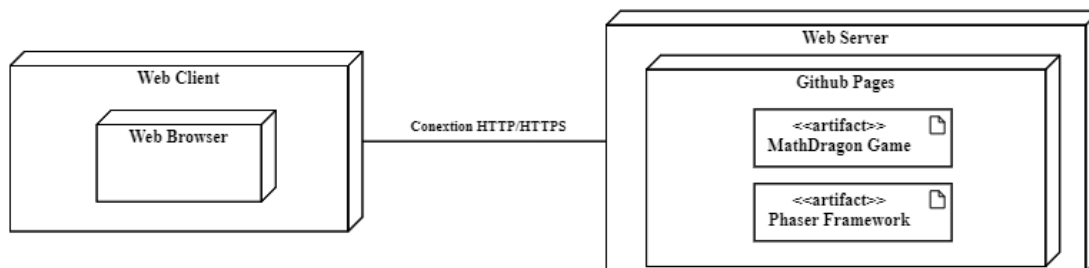


Fig. 7 Diagrama de despliegue de MathDragon  
Fuente: Elaboración propia

#### Nodos:

- *Web Client*: Entidad que se encarga del nodo de navegador web. En este nodo se ejecuta el navegador web utilizado por el estudiante.
- *Web Server*: Entidad que contiene dos artefactos: *MathDragon Game* y *Phaser Framework*. Este nodo ejecuta el videojuego *MathDragon* además hace uso del *framework* de *Phaser* para sus diferentes utilidades.
- *Github Pages*. Este nodo actúa como *host* para el videojuego *MathDragon*.

#### Artefactos:

- *MathDragon Game*: Contiene el proceso de desarrollo del videojuego. Las utilidades utilizadas se encuentran en el artefacto.
- *Phaser Framework*: Contiene el proceso de desarrollo del *framework Phaser*, encargado de la representación de gráficos, animación, efectos, entradas, y físicas.

#### Comunicación:

- HTTP/HTTPS: Los nodos “Cliente Web” y “Servidor Web” se conectan mediante el protocolo de transferencia de hipertexto HTTP/HTTPS según corresponda al hosting utilizado.

### 3.4.3 Vista de Desarrollo

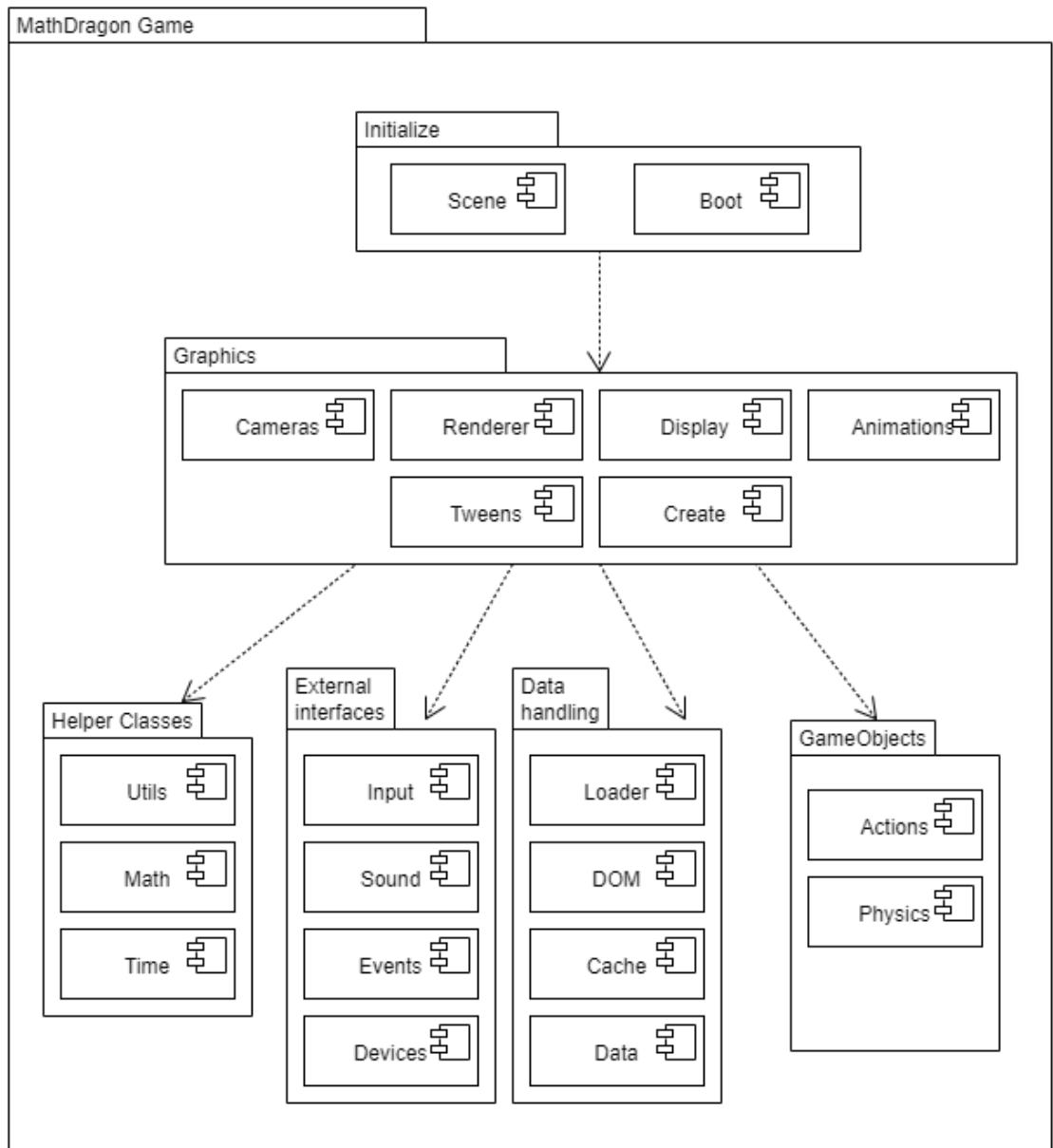


Fig. 8 Diagrama de componentes de MathDragon

Fuente: Elaboración propia

Paquetes y componentes:

- *Initialize*: Encargado de administrar todo lo relacionado con los aspectos visuales del videojuego.
  - *Scene*: Contiene todas las escenas implementadas en el videojuego.
  - *Boot*: Responsable de la configuración del videojuego, incluido los recursos externos. Encargado también de administrar el componente “*Scene*”

- *Graphics*: Es ejecutado por el componente de “*Scene*”, este paquete es responsable de visualizar el contenido gráfico en *Phaser*.
  - *Renderer*: Componente principal. Encargado de administrar todos los objetos visibles en pantalla.
  - *Cameras*: Es tu visión del mundo del juego. Tiene una posición y tamaño que representa solo aquellos objetos que se encuentran en el campo de visión.
  - *Animations*: Contiene la animación y los controles necesarios para reproducirla.
  - *Tweens*: Permite modificar las propiedades de un objeto durante un tiempo definido por el desarrollador.
  - *Create*: Permite generar el contenido del videojuego de manera rápida y sencilla, sin la necesidad de algún archivo externo.
- *Helper classes*: Contiene funciones básicas de utilidad que facilita la implementación en el *framework Phaser*.
  - *Utils*: Permite modificar e inspeccionar los objetos y *strings*.
  - *Math*: Encargado de la interacción con los objetos, como por ejemplo las rotaciones.
  - *Time*: Reloj interno del videojuego. Gestiona el tiempo transcurrido y todo tipo de operaciones con esta variable.
- *External interfaces*: Encargado de la administración y conexión de todas las entradas externas (software y hardware)
  - *Input*: Encargado de todos los dispositivos de entrada de *Phaser*, estos pueden ser el mouse, teclado, *touch* en los dispositivos móviles, etc.
  - *Sound*: Sirve para manipular el sonido en el videojuego; se puede reproducir, pausar y detener. Además, permite ajustar su volumen.
  - *Events*: Manejador de los eventos desencadenados y/o implementados en el videojuego.
  - *Device*: Comprueba la configuración del sistema durante su ejecución/arranque y las almacena para su futura manipulación.
- *Data handling*: Paquete encargado de manejar todos los datos entrantes que utiliza el *framework Phaser*.

- *Loader*: Tiene como función cargar todos los archivos externos en el videojuego (imágenes, audios, archivos *JSON*, etc.) y los agrega al caché.
- *DOM*: Contiene los métodos específicos del *DOM*. Estos métodos son: el *offset handling*, la calibración de la ventana, la verificación de los bordes/limites.
- *Cache*: Almacena y entrega todos los archivos cargados por el componente *Loader* de *Phaser*.
- *Data*: Almacena datos específicos en el videojuego. Estos datos pueden ser manipulados de manera general.
- *GameObjects*: Encargado de manejar los diferentes tipos de objetos que son representados en el videojuego.
  - *Actions*: Conjunto de funciones de acciones que puede realizar un objeto en el videojuego, por ejemplo, gestionar la velocidad y aceleración de un objeto por un tiempo determinado.
  - *Physics*: Responsable de las propiedades físicas de los objetos, por ejemplo, la gravedad, o colisión de un objeto con otro.

### 3.5 Diseño de Prototipos

MathDragon debe contar con las siguientes características dadas por Lepper y Malone [31]:

- Desafío, el videojuego ofrece dos niveles: recoger comida (adición) y destruir venenos (sustracción); presentando una retroalimentación de pistas de ayuda si el estudiante no logra responder la operación matemática para superar el nivel en el que se encuentra.
- Curiosidad, se utilizan diferentes animaciones para indicar diferentes acciones del estudiante, por ejemplo: vuelo del dragón, daño del dragón, muerte del dragón, contador de vidas.
- Control, el movimiento del dragón se realiza con las flechas direccionales y puede moverse por todo el escenario presentado sin ninguna restricción.
- Fantasía, la analogía se presenta en un animal ficticio que es el dragón y la recolección de comida que lo realiza cualquier animal en el mundo real.

Finalmente, para que MathDragon sea considerado un videojuego educativo se presenta el concepto matemático implícitamente en la recolección de comida o la destrucción de venenos de los niveles de adición y sustracción respectivamente.

Los prototipos fueron diseñados con las herramientas Adobe Illustrator y Adobe Photoshop; posteriormente presentados a los tutores de los estudiantes con el fin de recibir retroalimentación de los mismos.

### **3.5.1 Personaje**

Al tratarse de niños con edades entre 7 y 8 años se buscaron dibujos animados con los que se sintieran a gusto. Es por ello que se optó por usar el diseño de un dragón, nombrándolo Kalh. (Fig. 9)



Fig. 9 Kalh, personaje principal de MathDragon.  
Fuente: Elaboración propia

### **3.5.2 Menú Principal**

MathDragon cuenta con menús de navegación: Menú principal (Fig. 10) y selección de nivel (Fig. 11).



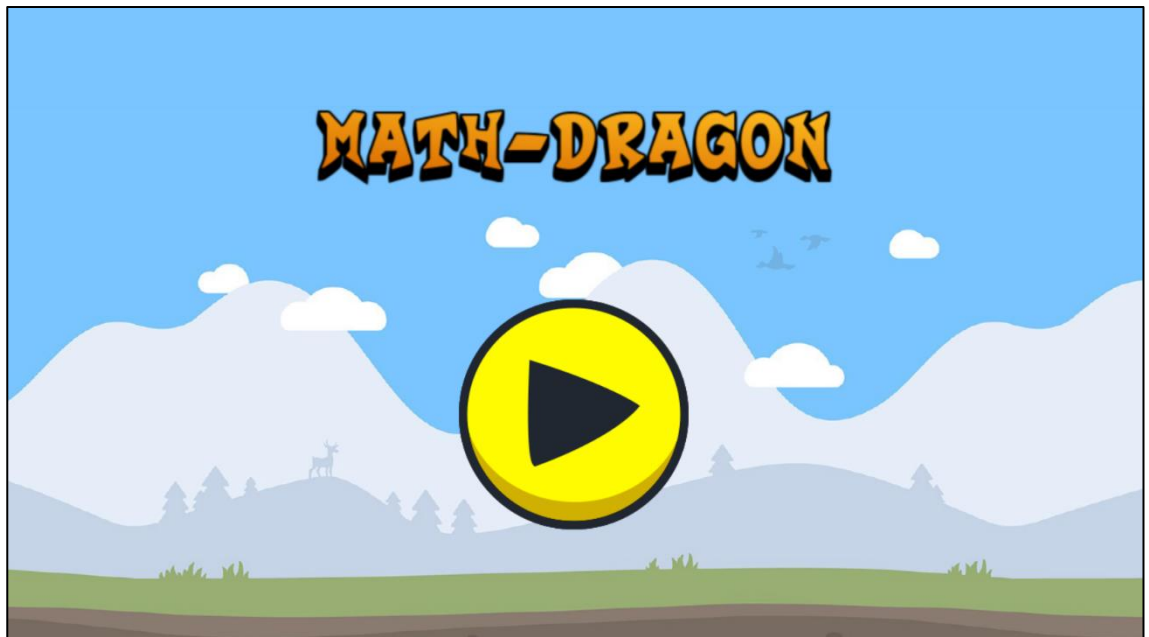


Fig. 10 Escena "Menú Principal" de MathDragon  
Fuente: Elaboración propia



Fig. 11 Escena de selección de nivel en MathDragon  
Fuente: Elaboración propia

El estudiante podrá regresar a cualquiera de estos menús mediante el cartel que se presenta al finalizar un nivel.

### 3.5.3 Niveles

El estudiante se enfrentará con dos dificultades al realizar un nivel de MathDragon.

Kalh contará con un total de 3 vidas durante el nivel del videojuego, estas vidas se irán perdiendo si Kalh colisionó contra una bomba. Las bombas son generadas de manera aleatoria cada 4 segundos.



Fig. 12 Vida de Kalh en MathDragon

Fuente: Elaboración propia



Fig. 13 Bomba en MathDragon

Fuente: Elaboración propia

El estudiante tendrá que cumplir con su objetivo sin que el temporizador del nivel finalice. Para ello se utilizó la imagen de un reloj para que indique el tiempo restante del nivel.



Fig. 14 Temporizador de tiempo en MathDragon

Fuente: Elaboración propia

#### 3.5.3.1 Nivel Conseguir Comida

En el nivel Coseguir Comida, Kalh tiene que obtener la mayor cantidad de muslos de pollo y pescados mientras que el temporizador del tiempo no finalice, evitando las bombas que se presenten. (Fig. 15, 16, 17)



Fig. 15 MathDragon: Muslos de pollo

Fuente: Elaboración propia



Fig. 16 MathDragon: Pescados

Fuente: Elaboración propia

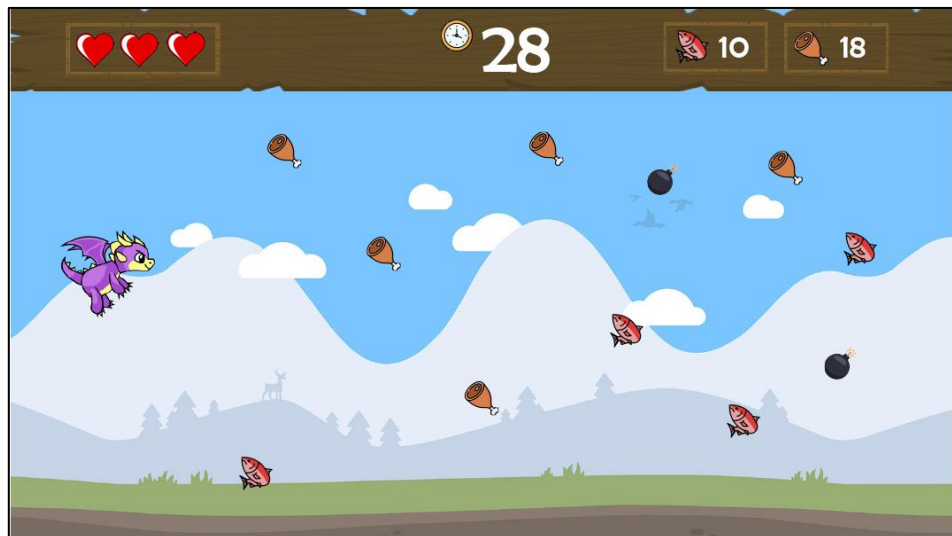


Fig. 17 MathDragon: Nivel Conseguir Comida

Fuente: Elaboración propia

Una vez finalizado el nivel se presentará un cartel resumen, en el mismo se pueden encontrar: Razón de la finalización del nivel, cantidad de muslos de carne y pescados adquiridos, pregunta del problema matemático a resolver, input de respuesta. (Fig. 18)

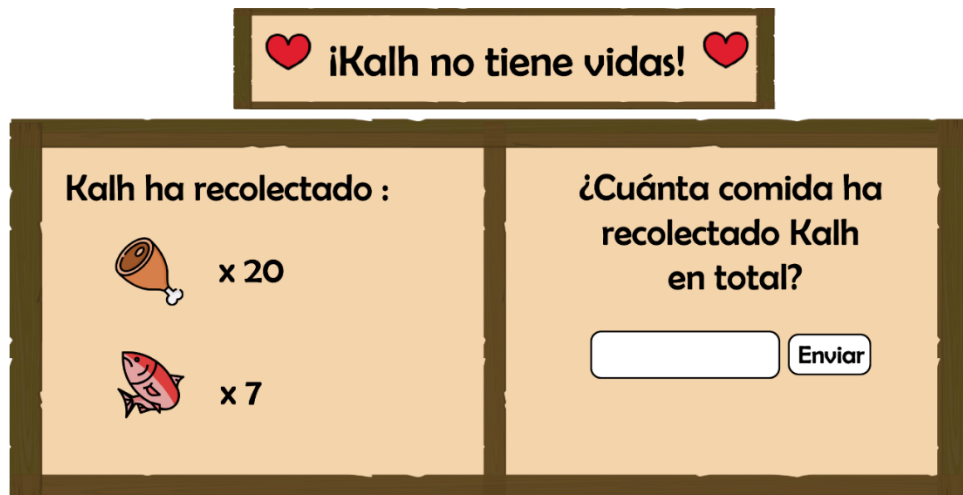


Fig. 18 Nivel Conseguir Comida: Cartel resumen  
Fuente: Elaboración propia

### 3.5.3.2 Nivel Destruir Venenos

En el nivel Destruir Venenos, Kalh tiene que destrozarse la mayor cantidad de botellas de veneno mientras que el temporizador del tiempo no finalice, evitando las bombas que se presenten. (Fig. 19, 20)



Fig. 19 MathDragon: Botella de veneno  
Fuente: Elaboración propia

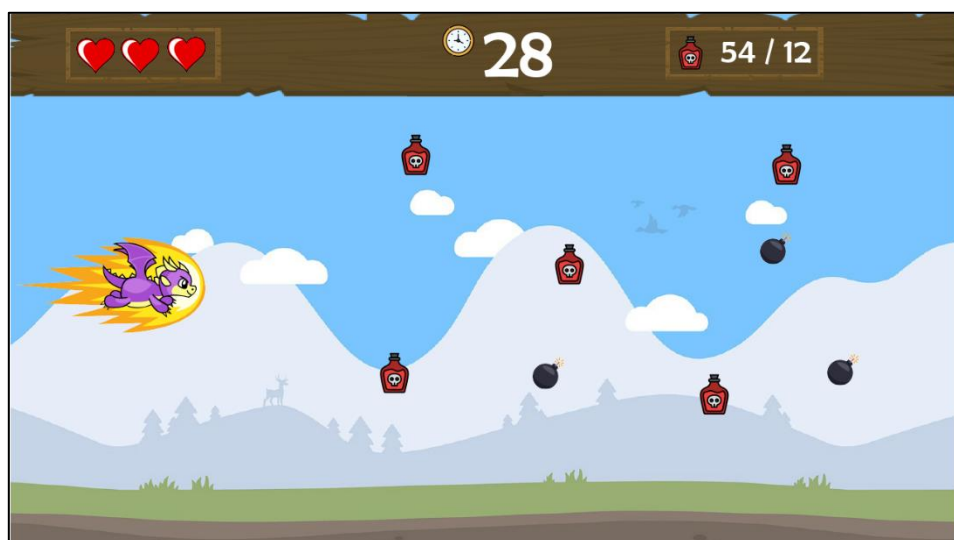


Fig. 20 MathDragon: Nivel Destruir Venenos  
Fuente: Elaboración propia

Al igual que en el nivel de Conseguir Comida, se presentará un cartel resumen. (Fig. 21)

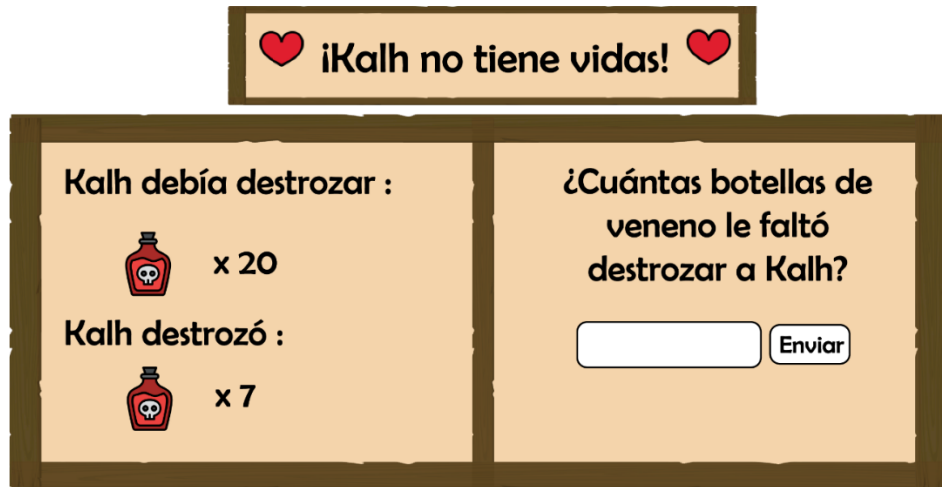


Fig. 21 Nivel Destruir Venenos: Cartel resumen  
Fuente: Elaboración propia

### 3.5.3.3 Fin del nivel

En ambos niveles el estudiante tendrá que resolver el problema llenando el input y haciendo clic en el botón de enviar. Si el estudiante responde correctamente se mostrará un cartel de felicitaciones (Fig. 22).



Fig. 22 Cartel de felicitaciones en MathDragon  
Fuente: Elaboración propia

El estudiante cuenta con tres oportunidades para poder responder correctamente, al finalizar sus tres oportunidades, se muestra un cartel de volver a intentarlo (Fig. 23)



Fig. 23 Cartel de volver a intentarlo en MathDragon  
Fuente: Elaboración propia

Ambos carteles cuentan con dos botones: Un botón para reiniciar el nivel y un botón para volver a la selección de niveles de MathDragon.

#### 3.5.4. Pistas y ayudas

Si el estudiante responde de manera incorrecta se presentará el cartel de pistas en la parte inferior del cartel resumen. El estudiante cuenta con un total de tres intentos. Si no responde correctamente se mostrará un cartel de volver a intentarlo (Fig. 24, 25).

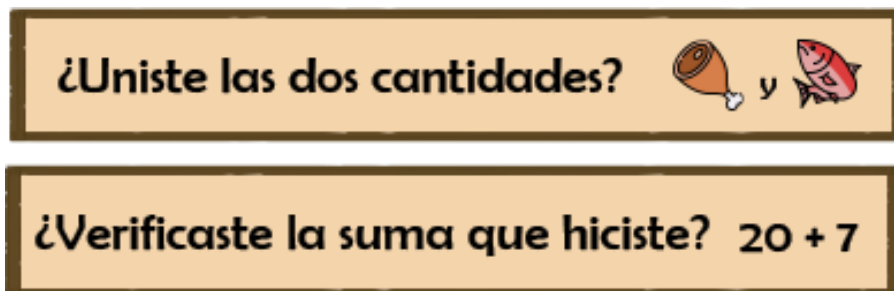


Fig. 24 Nivel Conseguir Comida: Carteles de Pistas  
Fuente: Elaboración propia

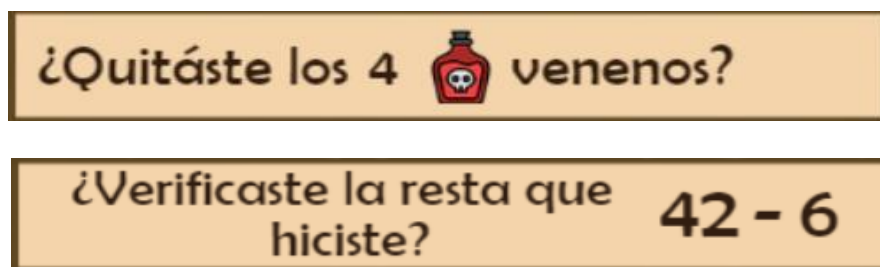


Fig. 25 Nivel Destruir Venenos: Carteles de Pistas  
Fuente: Elaboración propia

### 3.6 Implementación

La programación de MathDragon se realizó en el editor de código *Visual Studio Code* (VsCode), por su usabilidad con el usuario. VsCode permitió la instalación del *framework Phaser 3* con facilidad, además de extensiones como *CodeSnap*, que sirvió para tomar capturas de pantalla al código desarrollado.

Se utilizó *Git* como software de control de versiones, y como alojamiento del repositorio la plataforma *GitHub*. El *framework* utilizado fue *Phaser 3* basado en el lenguaje de programación JavaScript.

La implementación del código siguió el estándar ECMAScript 2015 que se encarga de registrar como debe ser interpretado y funcionar el lenguaje JavaScript.

#### 3.6.1. Instalación y Configuración

La instalación de *Phaser 3* se realizó en el archivo *index.html*. Importando el *script* de *Phaser 3* (Fig. 26, línea 5) en el *body*, además de las escenas utilizadas en el videojuego.

```
1 <body>
2   <div class="fuentes">.</div>
3   <div class="contenedor"></div>
4
5   <script src="https://cdn.jsdelivr.net/npm/phaser@3.51.0/dist/phaser-arcade-physics.min.js">
6   </script>
7   <script src="./src/scenes/main_menu.js"></script>
8   <script src="./src/scenes/niveles_scene.js"></script>
9   <script src="./src/scenes/add_scene.js"></script>
10  <script src="./src/scenes/sus_scene.js"></script>
11  <script src="./src/init.js" type="module"></script>
12 </body>
```

Fig. 26 Instalación del *framework Phaser 3*

Fuente: Elaboración propia

Una vez instalado el *framework* se debe configurar el videojuego, se crea un archivo llamado *init.js*, en donde se contendrá la configuración principal del videojuego. (Fig. 27)

```

1 const config = {
2   type: Phaser.AUTO,
3   scale: {
4     mode: Phaser.Scale.FIT,
5     width: 1280,
6     height: 720,
7   },
8   parent: "contenedor",
9   physics:{
10    default: "arcade"
11  },
12  dom: {
13    createContainer: true
14  },
15  scene: [
16    Main_menu,
17    Niveles_scene,
18    Add_scene,
19    Sus_scene,
20  ]
21 }
22
23 new Phaser.Game(config)

```

Fig. 27 Configuración del videojuego  
Fuente: Elaboración propia

**type (Fig. 27, línea 2):** Establece el contexto de renderizado que se quiere en el videojuego. Se puede elegir entre CANVAS, WEBGL o AUTO. Es recomendado enviar la propiedad AUTO, ya que automáticamente intenta renderizar nuestro juego en WEBGL, pero si el navegador web que lo está ejecutando no lo soporta, recurrirá al renderizado por CANVAS.

**scale (Fig. 27, línea 3):** Indica la resolución recomendada para nuestro videojuego, en el caso que el computador no cuente con esta resolución, se le envía en *Phaser.Scale.FIT* en la propiedad *mode* (Línea 4), para que el videojuego se ajuste a la resolución automáticamente.

**parent (Fig. 27, línea 8):** Indica la clase principal de nuestro videojuego, en este caso “contenedor”.

**physics y DOM (Fig. 27, línea 10 y 12):** Encargados de la inicialización del videojuego.

**scene (Fig. 27, línea 15):** En esta propiedad se envía el *array* de las escenas que contiene el videojuego.



### 3.6.2. Función *preload*( )

*Phaser 3* utiliza la función *preload*( ) para cargar todos los recursos que se necesita en una escena de nuestro videojuego. Esta función se llama automáticamente al iniciar una escena y carga cualquier imagen, *input*, video, sonido, etc. que se encuentre definido dentro de esta función.

MathDragon utiliza los siguientes métodos en la función de *preload*( ) de todas las escenas:

***load.image*** (Fig. 29, línea 2): Carga imágenes .png o .jpg en la escena actual.

Argumentos:

1. Alias de la imagen para poder ser llamada en otras funciones.
2. Ubicación del archivo en nuestro proyecto.

***load.spritesheet*** (Fig. 29, línea 3): Carga una imagen conformada por pequeñas imágenes, simulando así una animación. Argumentos:

1. Alias de la animación para ser utilizada en otras funciones.
2. Ubicación del archivo en nuestro proyecto
3. Definición del alto y ancho de cada *frame* de nuestra animación. Ver ejemplo de *spritesheet* en la Fig. 28

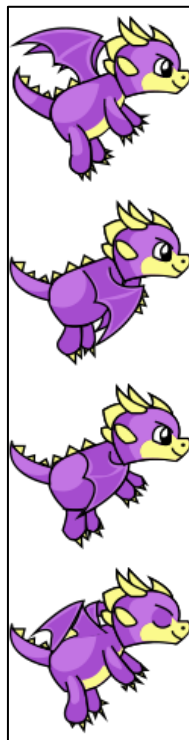


Fig. 28 Ejemplo de *spritesheet* del movimiento del dragón  
Fuente: Elaboración propia

**load.html (Fig. 29, línea 4):** Carga una interfaz externa de código HTML puro.

Argumentos:

1. Alias de la imagen para poder ser llamada en otras funciones.
2. Ubicación del archivo en nuestro proyecto.

```
1 preload() {  
2   this.load.html('form', './assets/respuesta.html')  
3   this.load.image("dragon", "./assets/dragon.png")  
4   this.load.spritesheet("mov_dragon", "./assets/mov_sprite.png", { frameWidth: 120, frameHeight: 120})  
5 }
```

Fig. 29 Ejemplo de función *preload()*  
Fuente: Elaboración propia

### 3.6.3. Función *create()*

Función encarga de la manipulación de los recursos inicializados en la función *preload()*. En esta función se define la posición de los recursos y sus respectivas animaciones, los datos globales que tenga la escena y se hacen las llamadas a las funciones personalizadas que sean necesarias.

El videojuego utiliza los siguientes métodos:

**data.set (Fig. 30, línea 5):** Inicializa variables a nivel de escena para su uso en otras funciones.

Argumentos:

1. Alias de la variable.
2. Valor inicial (*int* o *float*) de la variable.

```
1 this.data.set('puntaje', 0)  
2 this.data.set('puntajeCarnes', 0)  
3 this.data.set('puntajePescados', 0)  
4 this.data.set('vida', 3)  
5 this.data.set('temporizador', Phaser.Math.Between(20,40))  
6 this.data.set('pistas', 0)
```

Fig. 30 Método *data.set* en la función *create()*  
Fuente: Elaboración propia

En este caso en especial, para el temporizador del juego, se utilizó el método *Math.Between* (Fig. 30, línea 5), la misma devuelve un valor aleatorio en un rango especificado en sus argumentos.

En los siguientes métodos se encuentra el prefijo de *physics*, en *Phaser 3* indica que el recurso agregado tendrá responderá a las leyes de la física que le asigne.

***physics.add.image* (Fig. 31, línea 1):** Inserta una imagen en cierta posición de la escena actual. Argumentos:

1. Posición en sentido horizontal.
2. Posición en sentido vertical.
3. Alias del recurso que se desea insertar, previamente definida en la función *preload*( ).

Este método recibe además una función de *Phaser*: *setImmovable*. Esta función indica que la imagen será inamovible.

***physics.add.sprite* (Fig. 31, línea 2):** Inserta un *spritesheet* en la posición indicada de la escena actual. Sus argumentos son los mismos que el método *physics.add.image*, explicado anteriormente.

***physics.add.collider* (Fig. 31, línea 3):** Crea una relación de colisión entre dos recursos inicializados. Recibe como argumentos, los dos recursos que se relacionarán.

```
1 this.marco = this.physics.add.image(center_width, center_height-305, "marco").setImmovable(true)
2 this.dragon = this.physics.add.sprite(200, center_height, 'mov_dragon')
3 this.physics.add.collider(this.dragon, this.marco)
```

Fig. 31 Métodos de *physics* en la función *create*( )  
Fuente: Elaboración propia

***anims.create* (Fig. 32, línea 1):** Crea la animación de un *spritesheet* inicializado con anterioridad. Recibe un objeto como argumento, este objeto tiene los siguientes pares (*key/value*):

1. *key*, alias de la animación que se creará.

2. *frames*, encargado de capturar el *spritesheet* de la función *preload( )* e indicar cuantos *frames* tiene la misma.
3. *repeat*, indica el número de veces que se repetirán los *frames*, si el valor es -1, se repetirá de manera infinita.
4. *frameRate*, indica la velocidad que tendrá la animación.

***anims.play* (Fig. 32, línea 9):** Pone en marcha una animación previamente creada. Recibe como único argumento el alias de la animación.

***tweens.createTimeline* (Fig. 32, línea 10):** Inicializa una animación nativa de *Phaser 3* para su uso. En la línea 11 de la Fig. 32, se puede ver cómo poner en marcha la misma, recibe objeto como único argumento, este objeto tiene los siguientes pares:

1. *targets*, indica a qué variable va dirigida la animación.
2. *duration*, indica la cantidad de duración de la animación en milisegundos.
3. *ease*, indica el tipo de animación.
4. *y*, indica la posición en el eje vertical que tiene que finalizar la animación

```

1 this.anims.create({
2   key: 'dragon_mov',
3   frames: this.anims.generateFrameNumbers('mov_dragon', {
4     frames: [0, 1, 2, 3]
5   }),
6   repeat: -1,
7   frameRate: 8
8 })
9 this.dragon.anims.play('dragon_mov')
10 this.tweenPuntaje = this.tweens.createTimeline()
11 this.tweenPuntaje.add({
12   targets: this.contPuntajeTemp,
13   duration: 1500,
14   ease: 'Power1',
15   y: 0,
16 })

```

Fig. 32 Métodos de *anims* y *tweens* en la función *create( )*  
Fuente: Elaboración propia

***add.text*** (Fig. 33, línea 1): Crea un texto en la escena actual. Argumentos:

1. Posición en sentido horizontal.
2. Posición en sentido vertical.
3. Frase del texto a insertar.
4. Objeto con las propiedades que se desea que tenga el texto insertado.

***add.dom*** (Fig. 33, línea 6): Crea en la escena un interfaz externa previamente inicializada en la función *preload*( ). Argumentos:

1. Posición en sentido horizontal.
2. Posición en sentido vertical.
3. Método nativo de *Phaser 3* para crear una interfaz externa, este recibe el alias de nuestra interfaz.

***add.container*** (Fig. 33, línea 7): Agrupa varios elementos creados y los asigna a un solo contenedor. Argumentos:

1. Posición en sentido horizontal.
2. Posición en sentido vertical.
3. *Array* de elementos creados que se desea asignar al contenedor.

```
1 this.textRazonTiempo = this.add.text(center_width - 118 , center_height - 130, "¡Se acabó el tiempo!", {
2   fontFamily: 'Berlin_Sans',
3   fontSize: '28px',
4   color: '#311D0F'
5 })
6 this.respuestaFinal = this.add.dom(center_width + 135, center_height + 55 ).createFromCache('form')
7 this.contRazonTiempo = this.add.container(0, -700 , [
8   this.puntosRazonTiempo,
9   this.tiempoRazon01,
10  this.tiempoRazon02,
11  this.textRazonTiempo
12 ])
```

Fig. 33 Método *add* en la función *create*( )

Fuente: Elaboración propia

***input.keyboard.createCursorKeys*** (Fig. 34, línea 1): Función nativa de *Phaser 3*, para detectar nuestro teclado de la computadora.

```
1 this.cursor = this.input.keyboard.createCursorKeys()
```

Fig. 34 Método que detecta nuestro teclado

Fuente: Elaboración propia

### 3.6.4. Función *update*( )

*Phaser 3* utiliza la función *update*( ) como un oyente constante, es decir que se llama en un *bucle* infinito durante la ejecución de la escena. En *MathDragon* se utiliza esta función para el movimiento del dragón, ya que se está en constante escucha de lo que presione el usuario.

```
1 if(this.cursor.down.isDown){
2     this.dragon.body.setVelocityY(150)
3 } else if (this.cursor.up.isDown) {
4     this.dragon.body.setVelocityY(-150)
5 } else if (this.cursor.left.isDown){
6     this.dragon.body.setVelocityX(-250)
7 } else if (this.cursor.right.isDown){
8     this.dragon.body.setVelocityX(250)
9 } else{
10    this.dragon.body.setVelocityX(0)
11    this.dragon.body.setVelocityY(0)
12 }
```

Fig. 35 Función *update*( ) en *MathDragon*

Fuente: Elaboración propia

### 3.6.5. Otras funciones

Además de las funciones predefinidas por *Phaser 3* (*preload*, *create* y *update*) se utilizaron otras funciones necesarias para el correcto desarrollo de la escena principal.

**Adición de puntos:** Función encarga de sumar puntos en el tablero principal dependiendo si el dragón realiza una colisión con el objeto actual. Argumentos:

1. dragón, objeto inicializado en la función *create*( )
2. carne, objeto inicializado en la función *create*( )

```

1 puntoCarne (dragon, carne){
2     carne.disableBody(true, true)
3
4     //Puntaje Carnes
5     this.data.setValue('puntajeCarnes', this.data.get('puntajeCarnes')+1)
6     this.scoreCarnes.setText(' ' + this.data.get('puntajeCarnes'))
7     this.carneProblema.setText('= ' + this.data.get('puntajeCarnes'))
8     this.pistaCarne.setText(this.data.get('puntajeCarnes'))
9
10    //Puntaje total
11    this.data.setValue('puntaje', this.data.get('puntaje')+1)
12
13 }

```

Fig. 36 Función encarga de sumar el puntaje a la escena  
Fuente: Elaboración propia

**Temporizador:** Controla el tiempo del nivel. El valor del tiempo es aleatorio definido por la variable “temporizador” en la Fig. 30, línea 6.

Si el temporizador llega a 0 o las vidas del dragón se terminaron, detiene el juego, poniendo en marcha las animaciones de resumen del nivel.

```

1 funTemporizador(){
2   var puntaje_temp = this.data.get('puntaje')
3   var respuesta_temp
4   var temp_pistas = 0
5   var cartelPista01 = this.tweenPista01
6   var cartelPista02 = this.tweenPista02
7   var cartelPista01Retirada = this.tweenPista01Retirada
8   var cartelPista02Retirada = this.tweenPista02Retirada
9   var retirada_puntaje = this.tweenPuntajeRetirada
10  var retirada_tiempo = this.tweenTiempoRetirada
11  var dragonGanador = this.dragonGanador
12  var marcoGanador = this.tweenGanador
13  var dragonReset = this.dragonReset
14  var marcoReset = this.tweenReset
15  var dragon = this.dragon
16  if(this.data.get('vida') > 0){
17    if(this.data.get('temporizador') > 0){
18      this.data.setValue('temporizador', this.data.get('temporizador') - 1)
19      this.scoreTemporizador.setText(this.data.get('temporizador'))
20      this.time.delayedCall(1000, this.funTemporizador, [], this);
21      if (this.data.get('temporizador') <= 0){
22        this.dragon.body.setEnabled(false)
23        this.tweenPuntaje.play()
24        this.tweenTiempo.play()
25        this.respuestaFinal.addListener('click')
26        this.respuestaFinal.on('click', function (event) {
27          if(event.target.name === 'enviar'){
28            respuesta_temp = this.getChildByName('respuesta').value
29            if(puntaje_temp == respuesta_temp){
30              retirada_puntaje.play()
31              retirada_tiempo.play()
32              cartelPista01Retirada.play()
33              cartelPista02Retirada.play()
34              marcoGanador.play()
35              dragonGanador.anims.play('dragon_mov')
36              dragon.body.reset(1400,400)
37              console.log("GANASTE")
38            } else {
39              cartelPista01.play()
40              if(event.target.name === 'enviar'){
41                temp_pistas ++
42                respuesta_temp = this.getChildByName('respuesta').value
43                if (temp_pistas == 1)
44                  {
45                    console.log("VUELVE A INTENTARLO")
46                  } else if ( temp_pistas == 2) {
47                    cartelPista02.play()
48                    console.log("VUELVE A INTENTARLO")
49                  } else {
50                    console.log("JUEGO TERMINADO")
51                    retirada_puntaje.play()
52                    retirada_tiempo.play()
53                    cartelPista01Retirada.play()
54                    cartelPista02Retirada.play()
55                    dragon.body.reset(1400,400)
56                    marcoReset.play()
57                    dragonReset.anims.play('dragon_mov')
58                  }
59                }
60              }
61            }
62          })
63        }
64      }
65    }
66  }
67 }

```

Fig. 37 Función encarga del manejo del temporizador  
Fuente: Elaboración propia



**Contador de vidas:** Controla el número de vidas del dragón. Las vidas se encuentran definidas por la variable “vida” en la Fig. 30, línea 4.

Si las vidas llegan al valor de 0, la escena termina y se muestran las animaciones del resumen del nivel, así como un cartel indicando que ya no te quedan más vidas.

```
1 contVidas(){
2   var puntaje_temp = this.data.get('puntaje')
3   var respuesta_temp
4   var temp_pistas = 0
5   var cartelPista01 = this.tweenPista01
6   var cartelPista02 = this.tweenPista02
7   var retirada_puntaje = this.tweenPuntajeRetirada
8   var retirada_vida = this.tweenVidasRetirada
9   var cartelPista01Retirada = this.tweenPista01Retirada
10  var cartelPista02Retirada = this.tweenPista02Retirada
11  var dragonGanador = this.dragonGanador
12  var marcoGanador = this.tweenGanador
13  var dragonReset = this.dragonReset
14  var marcoReset = this.tweenReset
15  var dragon = this.dragon
16  if (this.data.get('vida') === 2){
17    this.corazon_3.anims.play('corazon_3', true)
18  } else if (this.data.get('vida') === 1){
19    this.corazon_2.anims.play('corazon_2', true)
20  } else if (this.data.get('vida') === 0){
21    this.corazon_1.anims.play('corazon_1', true)
22    this.dragon.anims.play('dragon_muer', true)
23    this.dragon.body.setEnabled(false)
24    this.tweenPuntaje.play()
25    this.tweenVidas.play()
26    this.respuestaFinal.addListener('click')
27    this.respuestaFinal.on('click', function (event) {
28      if(event.target.name === 'enviar'){
29        respuesta_temp = this.getChildByName('respuesta').value
30        if(puntaje_temp == respuesta_temp){
31          retirada_puntaje.play()
32          retirada_vida.play()
33          cartelPista01Retirada.play()
34          cartelPista02Retirada.play()
35          marcoGanador.play()
36          dragonGanador.anims.play('dragon_mov')
37          dragon.body.reset(1400,400)
38          console.log("GANASTE")
39        } else {
40          cartelPista01.play()
41          if(event.target.name === 'enviar'){
42            temp_pistas ++
43            respuesta_temp = this.getChildByName('respuesta').value
44            if (temp_pistas == 1)
45            {
46              console.log("VUELVE A INTENTARLO")
47            } else if ( temp_pistas == 2) {
48              cartelPista02.play()
49              console.log("VUELVE A INTENTARLO")
50            } else {
51              console.log("JUEGO TERMINADO")
52              retirada_puntaje.play()
53              retirada_vida.play()
54              cartelPista01Retirada.play()
55              cartelPista02Retirada.play()
56              dragon.body.reset(1400,400)
57              marcoReset.play()
58              dragonReset.anims.play('dragon_mov')
59            }
60          }
61        }
62      })
63    })
64  }
65 }
```

Fig. 38 Función encarga del manejo de las vidas  
Fuente: Elaboración propia

**Inicialización de objetos:** Función encarga de mostrar los diferentes objetos durante el nivel. Estos objetos pueden ser: bombas, carne, pescado, venenos. Los objetos son generados en una posición aleatoria en el mapa. Si el temporizador o las vidas llegan a 0, los objetos dejan de inicializarse.

```
1 nuevaCarne() {
2   if(this.data.get('vida') > 0 && this.data.get('temporizador')> 0){
3     this.carne.create(Phaser.Math.Between(1200,1280), Phaser.Math.Between(150,680), 'carne');
4     this.carne.setVelocityX(-200);
5     this.carne.checkWorldBounds = true;
6     this.carne.outOfBoundsKill = true;
7     this.time.delayedCall(2500, this.nuevaCarne, [], this);
8     this.physics.add.overlap(this.dragon, this.carne, this.puntoCarne, null, this);
9   }
10
11 }
```

Fig. 39 Función encarga de la inicialización de objetos  
Fuente: Elaboración propia

### **3.7 Pruebas del Software**

Las pruebas son prácticas que se realizan durante el ciclo de vida del software, estas son realizadas tanto por el desarrollador, como los usuarios finales. Tienen como objetivo demostrar que el programa hace lo que se intenta que haga, además de encontrar defectos antes de usarlo por los usuarios. Para MathDragon se realizó un plan de pruebas detallado a continuación:

#### **3.7.1 Elementos**

- Escena “Menú Principal” de MathDragon.
- Escena “Niveles” de MathDragon.
- Escena “Nivel Conseguir Comida” de MathDragon.
- Escena “Nivel Destruir Venenos” de MathDragon.

#### **3.7.2 Pruebas de Sistema**

Se realizaron pruebas de sistema a las diferentes escenas de MathDragon comprobando que funcionen correctamente.

##### **3.7.2.1 Criterio de Finalización**

Cada funcionalidad probada en todas las escenas de MathDragon, funciona correctamente y no se encuentra ningún error.

##### **3.7.2.2 Herramienta**

Para facilitar la trazabilidad de las pruebas de sistema se elaboró una matriz que contiene los siguientes elementos: Número de prueba, título de prueba, descripción, módulo/escena, datos de entrada, resultado esperado, resultado obtenido, error, comentarios.

La matriz de trazabilidad de pruebas de sistema se utilizó con el fin de documentar cada una de las pruebas de sistema realizadas durante el desarrollo de MathDragon.

##### **3.7.2.3 Métrica**

Finalizada la matriz de trazabilidad de pruebas de sistema se cuenta el número total de errores encontrados en las diferentes pruebas realizadas a MathDragon. Se realizaron un total de 30 pruebas.

##### **3.7.2.4 Entregable**

- Matriz de trazabilidad de pruebas de sistema (Anexo 5)

### **3.7.3 Pruebas de Rendimiento**

MathDragon será jugado por distintos estudiantes con diferentes ordenadores en capacidad computacional y capacidad de red.

La capacidad de hardware de una computadora o laptop debe ser capaz de ejecutar el videojuego sin ningún problema. Además, las conexiones de internet en Arequipa son muy variadas. Es por ello que se realizaron pruebas de rendimiento a MathDragon.

#### **3.7.3.1 Criterio de Finalización**

MathDragon es capaz de ejecutarse sin problemas en computadoras de baja potencia computacional y baja potencia de red.

#### **3.7.3.2 Herramienta**

Herramienta para desarrolladores del navegador Google Chrome. Esta herramienta permite la simulación de un computador con bajos recursos computacionales. De igual forma se puede alterar la velocidad de la red.

#### **3.7.3.3 Métrica**

Tabla comparativa de tiempos de las diferentes opciones de red y potencia computacional ofrecidas por Google Chrome.

- Potencia computacional: 4x slowdown y 6x slowdown.
- Red: Fast 3G, Slow 3G.

#### **3.7.3.4 Entregable**

Tabla comparativa de tiempos de respuesta en las diferentes escenas de MathDragon de recursos computacionales y red alterados.

### **3.7.4 Pruebas de Carga**

El videojuego es implementado para todos los estudiantes del primer grado de primaria del colegio San Juan Bautista de La Salle, siendo un total de 123 estudiantes. MathDragon debe ser capaz de funcionar correctamente por lo menos con la llegada de todos estos usuarios simultáneamente. Actualmente MathDragon encuentra desplegado en el host gratuito de GitHub.

#### **3.7.4.1 Criterio de Finalización**

MathDragon es capaz de ejecutarse simultáneamente sin problemas frente a un estrés mínimo de 123 peticiones sin mostrar ningún problema.

#### **3.7.4.2 Herramienta**

JMeter es un proyecto de Apache que se utiliza generalmente como una herramienta de pruebas de carga para diferentes aplicaciones web.

Gracias a esta herramienta se realiza peticiones GET a la página web en donde se encuentra desplegado MathDragon.

#### **2.7.4.3 Métrica**

HTTP Request Sample de JMeter al dominio de Github donde se encuentra alojado MathDragon. Se enviarán un total de 150 peticiones en un segundo superando el valor mínimo de 123 para los estudiantes del colegio San Juan Bautista de La Salle.

#### **3.7.4.4 Entregable**

Gráfica y tabla resumen del rendimiento entregado por JMeter que tuvo MathDragon con la prueba de carga.

### **3.7.5 Pruebas de Usabilidad**

Las pruebas de usabilidad son necesarias en MathDragon ya que es un videojuego diseñado para niños de edades entre 7 y 8 años. MathDragon debe ser de fácil uso, aprendizaje y generar una satisfacción al estudiante.

#### **3.7.5.1 Criterio de Finalización**

MathDragon cuenta con un promedio aprobatorio en las siguientes categorías:

- Calidad del sistema
- Calidad de la información
- Calidad de la interfaz

#### **3.7.5.2 Herramienta**

Se utilizó el Cuestionario de Usabilidad del Sistema Informático (CSUQ). El cuestionario mide las reacciones de desempeño de los participantes con el videojuego. Cuenta con tres secciones:

- Calidad del sistema: preguntas del 1 al 4.
- Calidad de la información: preguntas del 5 al 9.
- Calidad de la interfaz: preguntas del 10 al 13.

CUSQ se encuentra estandarizado y adaptado por Hedleft en el año 2015; por lo que cuenta con un Alfa de Cronbach de 0.94 [36].

### **3.7.5.3 Métrica**

Cada pregunta del cuestionario CUSQ es evaluada en la escala de Likert. 1 punto significa “Totalmente en desacuerdo” y 5 puntos significa “Totalmente de acuerdo”.

Una vez obtenidos los resultados se obtendrá el promedio por cada una de las secciones del cuestionario, siendo evaluadas de la siguiente forma:

- Mala: promedio mayor que 0 y menor que 1.
- Regular: promedio mayor que 1 y menor que 2.
- Buena: promedio mayor que 2 y menor que 4.
- Muy Buena: promedio mayor que 5.

### **3.7.5.4 Entregable**

Informe de resultados obtenidos después de promediar cada una de las secciones del cuestionario.

## Capítulo 4. Resultados y Discusión

### 4.1 Pruebas de Sistema

La matriz de trazabilidad de pruebas de sistema (Anexo 5) cuenta con los siguientes elementos:

- Número de prueba: Identificador para el caso de prueba de sistema que se realizará.
- Título de prueba: Título del caso de prueba de sistema, tiene que ser especificado de manera general.
- Descripción: Breve descripción de la prueba de sistema que se está realizando.
- Módulo/escena: Se indica en que escena de MathDragon se está realizando la prueba de sistema.
- Datos de entrada: Se entiende como datos de entrada toda acción que realizas o envías al videojuego.
- Resultado esperado: Descripción breve sobre el resultado que se espera con los datos de entrada detallados anteriormente.
- Resultado obtenido: Descripción breve sobre el resultado real obtenido con los datos de entrada detallados anteriormente.
- Error: Valor booleano. Si el resultado esperado es igual al obtenido, el valor se define como “SI”, caso contrario, se define como “NO”.
- Comentarios: Breves comentarios sobre el caso de prueba de sistema que ayude a una mejor documentación.

Documentadas todas las pruebas en la matriz de trazabilidad de pruebas de sistema se realiza un conteo de las respuestas “SI” en el elemento “Error”. Obteniendo los siguientes resultados:

*Tabla 33. Resumen de Pruebas de Sistema*

---

Porcentaje de errores según la cantidad de casos de prueba		
<u>Casos de Prueba</u>	<u>Errores</u>	<u>Porcentaje</u>
30	0	0 %

---

Fuente: Elaboración Propia

Se observa que el porcentaje de errores según el número de casos de pruebas es nulo, por lo cual se concluye que las diferentes funcionalidades que cuenta MathDragon funcionan correctamente.

## 4.2 Pruebas de Rendimiento

Para las pruebas de rendimiento se hizo uso de la pestaña “Performance” de las herramientas para desarrolladores de Google Chrome.

“Performance” permite simular diferentes escenarios de red y de capacidad computacional que pueden tener varios estudiantes en nuestros casos.

Se midió los resultados según el tiempo de respuesta de MathDragon. Las funcionalidades evaluadas fueron las más críticas durante el videojuego, comenzando por la renderización de alguna escena hasta la finalización de un nivel.

Tabla 34. Pruebas de Rendimiento: Red

Tiempo de respuesta en segundos según funcionalidad en MathDragon			
	<u>Normal</u>	<u>Fast 3G</u>	<u>Slow 3G</u>
Menú Principal	0.33s	4.86s	16.97s
Niveles	0.52s	2.17s	7.79s
Nivel Conseguir Comida	0.42s	2.35s	8.45s
Nivel Destruir Venenos	0.52s	3.37s	12.09s

Fuente: Elaboración propia

Tabla 35. Pruebas de Rendimiento: Capacidad Computacional

Tiempo de respuesta en segundos según funcionalidad en MathDragon			
	<u>Normal</u>	<u>4x slowdown</u>	<u>6x slowdown</u>
Menú Principal	0.32s	0.9s	1.26s
Niveles	0.93s	0.19s	0.32s
Nivel Conseguir Comida	0.27s	0.82s	1.21s
Nivel Destruir Venenos	0.29s	0.81s	1.24s

Fuente: Elaboración propia

Los datos obtenidos después de las pruebas de rendimiento serán evaluados respecto a la diferencia de tiempo entre los diferentes escenarios de recursos (red y capacidad computacional). Ver Tabla 34 y 35.



Tabla 36. Rendimiento de red: Promedio de tiempo de respuesta

Promedio de tiempo en segundos de cada uno de los escenarios de recursos de red		
<u>Normal</u>	<u>Fast 3G</u>	<u>Slow 3G</u>
0.4475s	3.1875s	11.325s

Fuente: Elaboración propia

Tabla 37. Rendimiento computacional: Promedio de tiempo de respuesta

Promedio de tiempo en segundos de cada uno de los escenarios de recursos de red		
<u>Normal</u>	<u>4x slowdown</u>	<u>6x slowdown</u>
0.4525s	0.68s	1.0075s

Fuente: Elaboración propia

Se utiliza el método de desviación media para comparar los resultados en ambas pruebas de rendimiento. Si la desviación es elevada, quiere decir que existe mucha variabilidad en los datos, mientras que, si se aproxima a cero, los datos de los valores son iguales. Para la interpretación del videojuego, se busca un dato no muy elevado para que MathDragon sea apto para todo tipo de computadoras.

El rendimiento de red la diferencia es de 4.2256, y para el rendimiento computacional la diferencia entre los tiempos es de un 0.1961. Estos resultados indican que se puede jugar MathDragon sin problemas aún en computadoras o laptops con bajos recursos computacionales, pero que existe una pequeña latencia cuando los recursos de red son muy escasos.

### 4.3 Pruebas de Carga

Las pruebas de carga fueron realizadas con la herramienta *JMeter* de *Apache*. Se creó primeramente un grupo de usuarios, siendo un total de 150 que envían la petición al servidor en un segundo.

Se creó un *sampler* de *HTTP Request*, para ingresar el dominio de MathDragon, configurando como petición principal un GET.

Se utilizaron dos *listener*: “*Summary Report*” y “*Graph Result*” obteniendo los siguientes resultados (Tabla 38) (Fig. 40)

Tabla 38. Pruebas de Carga

Datos obtenidos del *listener* “*Summary Report*” y “*Graph Result*” de *JMeter* al dominio de MathDragon

	<u>Resultado</u>
<i>Samples</i>	150 request
<i>Average</i>	114ms
<i>Min</i>	92ms
<i>Max</i>	169ms
<i>Estandar Deviation</i>	12.5
<i>Error</i>	0%
<i>Throughput</i>	120.675/ms

Fuente: Elaboración propia

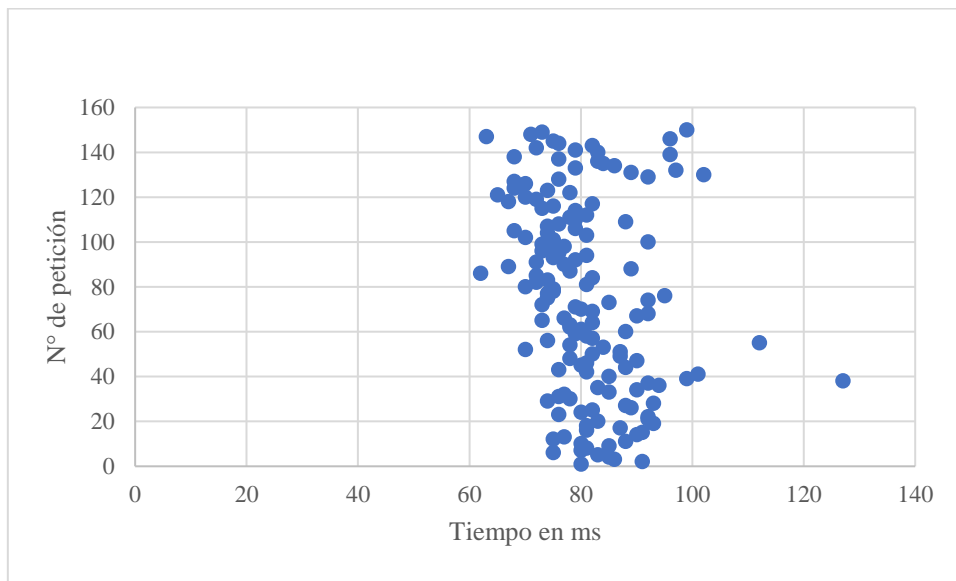


Fig. 40 *Listener* “*Graph Results*” de *JMeter* en MathDragon  
Fuente: Elaboración propia

**Samples:** número de peticiones que se realizó.

**Average:** promedio de respuesta de petición en milisegundos.

**Min:** mínimo tiempo de respuesta de una petición en el caso de prueba en milisegundos

**Max:** máximo tiempo de respuesta de una petición en el caso de prueba en milisegundos

**Standar Deviation:** Desviación estándar del tiempo de respuesta de las peticiones.

**Error:** porcentaje de error en las peticiones.

**Throughput:** Representa la cantidad de procesamiento completado satisfactoriamente en un periodo de tiempo.

Los resultados obtenidos indican que las pruebas de carga fueron superadas satisfactoriamente, y MathDragon está preparado para soportar una cantidad considerable de peticiones simultaneas, mostrando así que los alumnos del primer grado de instrucción primaria pueden jugar sin problemas.

#### 4.4 Pruebas de Usabilidad

Los cuestionarios de CSUQ fueron enviados a un total de 13 docentes del grado de instrucción primaria del colegio San Juan Bautista de La Salle.

Los docentes cumplen el rol de tutores de todos los grados de primaria, siendo un total de 12 docentes. Se evaluó también al director de primaria. Finalmente contando con 13 docentes/expertos para evaluar a MathDragon. Antes de la evaluación del videojuego se les presentó un video tutorial ([https://www.youtube.com/watch?v=hT38Ln\\_dMFA](https://www.youtube.com/watch?v=hT38Ln_dMFA) [38]) de las funcionalidades y controles de MathDragon, para su mayor entendimiento. De igual forma se encuentra el manual de usuario en el Anexo 4.

Primeramente, se calificó cada pregunta con un puntaje que responde a la escala de Likert, obteniendo 1 punto si la respuesta es “Totalmente en desacuerdo” y 5 puntos si la respuesta es “Totalmente de acuerdo”. Se obtuvieron los siguientes resultados por pregunta. Ver Tabla 39.

*Tabla 39. Pruebas de Usabilidad: Resultados por pregunta.*

Promedio por pregunta del cuestionario CSUQ	
<u>Pregunta</u>	<u>Promedio</u>
Estoy satisfecho con lo fácil que es utilizar el videojuego.	4.307
Soy capaz de completar mis tareas rápidamente utilizando el videojuego.	4.461
Fue fácil aprender a utilizar el videojuego.	4.538
Creo que me volví experto rápidamente utilizando el videojuego.	4.076
El videojuego muestra mensajes de error que me dicen claramente cómo resolver los problemas.	4.23

Cada vez que cometo un error utilizando el videojuego, lo resuelvo fácil y rápidamente.	4.461
Es fácil encontrar en el videojuego la información que necesito.	4.615
La información que proporciona el videojuego fue efectiva ayudándome a completar las tareas.	4.461
La organización de la información del videojuego fue clara.	4.538
La interfaz del videojuego fue placentera.	4.153
Me gustó utilizar el videojuego.	4.384
El videojuego tuvo todas las herramientas que esperaba que tuviera.	4
En general, estuve satisfecho con el videojuego.	4.461

Fuente: Elaboración Propia

Una vez adquiridos los promedios y desviaciones estándar por cada una de las preguntas realizadas en el cuestionario CSUQ, se agrupó las preguntas en tres grupos:

- Calidad del sistema: preguntas del 1 al 4.
- Calidad de la información: preguntas del 5 al 9.
- Calidad de la interfaz: preguntas del 10 al 13.

Se evaluó el promedio y las desviaciones estándar por cada uno de los grupos para evaluar los criterios de usabilidad. Ver Tabla 40.

*Tabla 40. Pruebas de Usabilidad: Resultados por grupo.*

Promedio por grupos.	
	<u>Promedio</u>
Calidad del sistema	4.346
Calidad de la información	4.461
Calidad de la interfaz	4.25

Fuente: Elaboración propia

Finalmente, con los promedios de cada una de las categorías se evaluaron bajo la siguiente métrica:

- Mala: promedio mayor que 0 y menor que 1.

- Regular: promedio mayor que 1 y menor que 2.
- Buena: promedio mayor que 2 y menor que 4.
- Muy buena: promedio mayor que 4.

Obteniendo los siguientes resultados:

- Calidad del sistema se obtuvo un promedio de 4.346 por lo que se califica como “Muy buena”.
- Calidad de la información, se obtuvo el mayor promedio de las tres categorías siendo de 4.461 siendo calificada como “Muy buena”.
- Calidad de la interfaz, un promedio de 4.25 siendo calificada como “Muy Buena”

Todos los resultados fueron dados por las respuestas al cuestionario CSUQ realizado por expertos (docentes del grado de instrucción de primaria). Gracias a estos resultados se concluye que MathDragon es un videojuego amigable y usable.

#### **4.5 Experiencia de juego de los estudiantes (EJE)**

El cuestionario de experiencia de juego fue entregado a un total de 64 estudiantes entre el primer grado de primaria del colegio San Juan Bautista de La Salle. Este cuestionario fue evaluado con la herramienta IBM SPSS *Statistics* 21, obteniendo un Alfa de Cronbach de 0.88.

Todos los estudiantes se encontraban en el salón de audiovisuales con sus respectivas computadoras cada uno; en primera instancia se explicó el videojuego a grandes rasgos y se les dio la libertad de escoger alguno de los niveles.

Fueron un total de 13 preguntas que se dividen en las siguientes categorías:

- Datos generales sobre el uso de videojuegos: Preguntas del 1 al 5
- Experiencia con MathDragon: Preguntas del 6 al 9
- Aprendizaje matemático: Preguntas del 10 al 12

##### **4.5.1 Datos generales sobre el uso de videojuegos**

La primera categoría tiene el fin de conocer la familiaridad de los estudiantes con los videojuegos. Obteniendo los siguientes resultados:

*Tabla 41. EJE: Pregunta 1*

<u>Pregunta</u>	<u>Si</u>	<u>No</u>
-----------------	-----------	-----------

¿Es usuario de videojuegos?	100%	0%
-----------------------------	------	----

Fuente: Elaboración propia

Tabla 42. EJE: Pregunta 2

<u>Pregunta</u>	<u>Menos de 1</u>	<u>1-2</u>	<u>3-4 horas</u>	<u>4 o más</u>
¿Cuántas horas al día dedica al uso de videojuegos?	37.5%	50%	12.5%	0%

Fuente: Elaboración propia

Tabla 43. EJE: Pregunta 3

<u>Pregunta</u>	<u>Casa</u>	<u>Colegio</u>
¿Dónde utiliza videojuegos regularmente?	100%	0%

Fuente: Elaboración propia

Tabla 44. EJE: Pregunta 4

<u>Pregunta</u>	<u>Consola</u>	<u>PC</u>	<u>Celular</u>
¿En qué plataforma has utilizado los videojuegos regularmente?	29.6%	29.6%	40.8%

Fuente: Elaboración propia

Tabla 45. EJE: Pregunta 5

<u>Pregunta</u>	<u>Si</u>	<u>No</u>
¿Has utilizado videojuegos para aprender algún tema educativo?	28%	75%

Fuente: Elaboración propia

Con estos resultados se concluye que los estudiantes no son ajenos a los videojuegos, pero al analizar la pregunta 5 del cuestionario (Tabla 45) se puede notar que los videojuegos educativos no son muy utilizados por ellos.

#### 4.5.2 Experiencia con MathDragon

Las preguntas de la presente categoría fueron evaluadas bajo la escala de Likert, teniendo 5 opciones: Totalmente en desacuerdo, en desacuerdo, neutral, de acuerdo y totalmente de acuerdo. Obteniendo los siguientes resultados:

Tabla 46. EJE: Resultados experiencia con MathDragon

<u>Pregunta</u>	<u>Totalmente</u> <u>en</u> <u>desacuerdo</u>	<u>En</u> <u>desacuerdo</u>	<u>Neutral</u>	<u>De</u> <u>acuerdo</u>	<u>Totalmente</u> <u>de acuerdo</u>
El videojuego te pareció divertido	0	0	2	20	42
Fue fácil adaptarse a la forma de control del videojuego	0	0	2	33	29
Fue fácil cumplir con el objetivo del videojuego	0	0	25	21	18
Deseaste volver a jugar para pasar más niveles	0	0	6	19	39

Fuente: Elaboración Propia

Gracias a la tabla se concluye que los estudiantes tuvieron una buena experiencia jugando MathDragon, no tuvieron dificultades para adaptarse a la jugabilidad con respecto a los controles.

#### 4.5.3 Aprendizaje Matemático

Las siguientes preguntas van relacionadas con el curso de matemáticas, respondiendo si es que el estudiante logró reforzar o aprender el concepto de adiciones y sustracciones que se presentan en los niveles de Recoger Comida y Destrozar venenos respectivamente. Adicionalmente se evalúa si el estudiante se

siente motivado de aprender más conceptos matemáticos jugando. Los resultados fueron los siguientes:

*Tabla 47. EJE: Aprendizaje Matemático*

<u>Pregunta</u>	<u>Totalmente</u> <u>desacuerdo</u>	<u>En</u> <u>desacuerdo</u>	<u>Neutral</u>	<u>De</u> <u>acuerdo</u>	<u>Totalmente</u> <u>de acuerdo</u>
Fue fácil resolver el problema matemático presentado	0	2	15	6	41
Deseo jugar más niveles con diferentes operaciones matemáticas	0	0	5	17	42
Deseo aprender matemáticas con videojuegos	0	0	2	18	44

Fuente: Elaboración Propia

Se concluye lo siguiente con los resultados del cuestionario de Experiencia de Juego en los Estudiantes (EJE)

- Los estudiantes no son ajenos a los videojuegos, utilizan diferentes plataformas en sus casas con un promedio máximo de 2 horas al día.
- MathDragon fue un videojuego divertido para ellos, con los diferentes obstáculos y forma de jugabilidad que presenta.
- Los estudiantes pudieron resolver los problemas planteados por el videojuego y además desearon seguir aprendiendo con la misma metodología.



## Conclusiones

Gracias a todos los resultados e información brindados a lo largo de este trabajo, se concluye lo siguiente:

1. Se implementó MathDragon, un videojuego dinámico que incentivó el aprendizaje de operaciones de adición y sustracción a estudiantes del primer grado de primaria con más de un 95% de respuestas positivas en su evaluación.
2. Se capturó los requerimientos funcionales y no funcionales con los docentes expertos en el área adquiriendo retroalimentación.
3. MathDragon fue codificado para que funcione en navegadores web. Se utilizó el *framework* Phaser 3 de JavaScript debido a las distintas funcionalidades que brinda para las animaciones y manejo de las físicas de los objetos de las escenas.
4. El cuestionario EJE se elaboró con éxito con ayuda de los docentes expertos para los estudiantes del primer grado de primaria obteniendo un Alfa de Cronbach de 0.88; mostrando una positividad del 95% en las respuestas de las preguntas “El videojuego te pareció divertido” y “Deseaste volver a jugar para pasar más niveles”
5. Se elaboraron pruebas de sistema, rendimiento, carga y usabilidad para MathDragon, haciendo uso de distintas herramientas como: Herramientas del desarrollador de Google Chrome, cuestionarios validados con Alfa de Cronbach, JMeter y el uso de matrices de trazabilidad.
6. Se evaluó el videojuego en las diferentes pruebas, siendo exitosas, dando como resultado que el videojuego puede ser ejecutado en computadoras con bajos recursos y de conexión a internet lenta. Además, los docentes evaluaron la usabilidad del videojuego como muy buena.

Gracias a las encuestas realizadas a los estudiantes, se concluye que el videojuego incentivó el aprendizaje de las operaciones de adición y sustracción del curso de matemáticas en los estudiantes, pidiendo más niveles para poder resolver más problemas con distintas operaciones matemáticas.

## Trabajos Futuros

MathDragon actualmente cuenta con dos niveles: Recoger comida y Destruir Venenos, que incentivan el aprendizaje de conceptos con adiciones y sustracciones respectivamente.

El videojuego tiene la capacidad de ser escalable e implementar nuevos niveles para diferentes operaciones matemáticas, tales sean multiplicaciones o divisiones, según sea el alcance de conocimiento del estudiante según su grado de instrucción.

Para la implementación de estos nuevos niveles se debería de crear una nueva escena e importarla en la configuración principal de MathDragon, en el archivo *init.js*. Posteriormente en esta escena se haría uso de las funciones predefinidas por *Phaser 3*: *preload( )*, *create( )* y *update( )*, para insertar los *assets*, inicializaciones y funciones respectivamente en cada nivel.

Se propone implementar una licencia *Creative Commons* para que el videojuego sea de acceso abierto, lo cual permitiría que cualquier usuario pueda descargar, copiar, distribuir, traducir, reutilizar, adaptar y desarrollar sin costo alguno.

## Referencias

- [1] Instituto Nacional de Estadística e Informática (2022). Centros educativos, según nivel y modalidad [Online]. Available: [https://m.inei.gob.pe/media/MenuRecursivo/indices\\_tematicos/5.35a\\_1.xlsx](https://m.inei.gob.pe/media/MenuRecursivo/indices_tematicos/5.35a_1.xlsx)
- [2] Instituto Nacional de Estadística e Informática (2022). Alumnos matriculados en educación primaria, por gestión pública, según departamento [Online]. Available: [https://m.inei.gob.pe/media/MenuRecursivo/indices\\_tematicos/5.11a\\_1.xlsx](https://m.inei.gob.pe/media/MenuRecursivo/indices_tematicos/5.11a_1.xlsx)
- [3] Instituto Nacional de Estadística e Informática (2022). Alumnos matriculados en educación primaria, por gestión privada, según departamento [Online]. Available: [https://m.inei.gob.pe/media/MenuRecursivo/indices\\_tematicos/5.12a\\_1.xlsx](https://m.inei.gob.pe/media/MenuRecursivo/indices_tematicos/5.12a_1.xlsx)
- [4] Oficina de Medición de la Calidad de los Aprendizajes, Ministerio de Educación (2019). Evaluaciones de logros de aprendizajes: Resultados 2019 [Online]. Available: <http://umc.minedu.gob.pe/wp-content/uploads/2020/06/PPT-web-2019-15.06.19.pdf>
- [5] Instituto Nacional de Estadística e Informática (2017). Hogares según cobertura de las tecnologías de información y comunicación [Online]. Available: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjBka\\_Cqc3rAhUBGLkGHfj6B7kQFjAAegQIBBAB&url=https%3A%2F%2Fwww.inei.gob.pe%2Fmedia%2FMenuRecursivo%2Fpublicaciones\\_digitaless%2FEst%2FLib1539%2Fcap07.pdf&usg=AOvVaw1worM9TcbIcWXV4c72L5fp](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjBka_Cqc3rAhUBGLkGHfj6B7kQFjAAegQIBBAB&url=https%3A%2F%2Fwww.inei.gob.pe%2Fmedia%2FMenuRecursivo%2Fpublicaciones_digitaless%2FEst%2FLib1539%2Fcap07.pdf&usg=AOvVaw1worM9TcbIcWXV4c72L5fp)
- [6] Ministerio de Educación, Estadística de la Calidad Educativa (2017). Decisiones basadas en Evidencias: Presentación del proceso censal 2017 – Arequipa [Online]. Available: [https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwivm5zoqM3rAhWtIrkGHcRSDMAQFjAAegQIAxAB&url=http%3A%2F%2Fescale.minedu.gob.pe%2Fc%2Fdocument\\_library%2Fget\\_file%3Fuuid%3Df415133e-d5b7-4a00-ba43-3c3d02ad89f7%26groupId%3D10156&usg=AOvVaw1bLJ3IE8kQawkd83gzMwIp](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwivm5zoqM3rAhWtIrkGHcRSDMAQFjAAegQIAxAB&url=http%3A%2F%2Fescale.minedu.gob.pe%2Fc%2Fdocument_library%2Fget_file%3Fuuid%3Df415133e-d5b7-4a00-ba43-3c3d02ad89f7%26groupId%3D10156&usg=AOvVaw1bLJ3IE8kQawkd83gzMwIp)
- [7] Pichihua, S. (2016). El 70% de los niños son usuarios activos de videojuegos. El Peruano [Online]. Available: <https://elperuano.pe/noticia-el-70-los-ninos-son-usuarios-activos-videojuegos-37461.aspx>

- [8] Ministerio de Educación, Estadística de la Calidad Educativa (2017). Presentación del Proceso Censal 2017: Resultados del Censo Educativo 2017: Matricula, Docentes, Recursos y Local Educativo. Resultados del Censo de DRE y UGEL 2017 [Online]. Available: [http://escale.minedu.gob.pe/c/document\\_library/get\\_file?uuid=d524d4b5-0dd3-4706-a1e8-c65fb18a3d77&groupId=10156](http://escale.minedu.gob.pe/c/document_library/get_file?uuid=d524d4b5-0dd3-4706-a1e8-c65fb18a3d77&groupId=10156)
- [9] A. Domínguez, J. Sáenz de Navarrete, L. De Marcos, L. Fernández Sanz, C. Pagés, and J.J. Martínez-Herráiz, Gamifying learning experiences: Practical implications and outcomes. *Computers and Education*, 63, pp. 380–392, 2013.
- [10] A. Iosup, and D. Epema, An experience report on using gamification in technical higher education, *Proc. 45th ACM Tech. Symp. Comput. Sci. Educ. - SIGCSE*, pp. 27–32, 2014.
- [11] W. Chao, C. Yang and R. Chang, "A Study of the Interactive Mathematics Mobile Application Development," 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII), Jeju, Korea (South), pp. 248-249, 2018.
- [12] Manisha and A. Mantri, "An Augmented Reality Application for Basic Mathematics: Teaching and Assessing Kids' Learning Efficiency," 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, pp. 1-4, 2019.
- [13] S. Papadimitriou and M. Virvou, "An online adventure game for teaching math," 7th International Conference on Information, Intelligence, Systems & Applications (IISA), Chalkidiki, Greece, pp. 1-5, 2016.
- [14] V. Robledo-Rella, R. M. Guadalupe García-Castelán, L. Medina, J. M. Ramírez de Arellano and I. Guerrero, "CocoGame: A funny app to learn physics and math," *IEEE Frontiers in Education Conference (FIE)*, Indianapolis, IN, USA, pp. 1-4, 2017.
- [15] S. Laato, T. H. Laine, J. Seo, W. Ko and E. Sutinen, "Designing a Game for Learning Math by Composing: A Finnish Primary School Case," *IEEE 17th International Conference on Advanced Learning Technologies (ICALT)*, Timisoara, Romania, pp. 136-138, 2017.
- [16] A. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

- [17] H. Chen, K. Liao and J. Chang, "Design of digital game-based learning system for elementary mathematics problem solving," 8th International Conference on Ubi-Media Computing (UMEDIA), Colombo, Sri Lanka, pp. 303-307, 2015.
- [18] M. J. Ibarra et al., "Game Based Learning for math learning: iFractions case study," International Conference on Virtual Reality and Visualization (ICVRV), Hong Kong, China, pp. 208-211, 2019.
- [19] H. N. H. Cheng, Y. Lin, M. Wang and T. Chan, "Math Detective: Digital Game-Based Mathematical Error Detection, Correction and Explanation," IEEE 15th International Conference on Advanced Learning Technologies, Hualien, Taiwan, pp. 122-126, 2015.
- [20] K. M. Fisher, and J. I. Lipson, "Twenty Questions about Student Errors," Journal of Research in Science Teaching, vol. 23(9), pp. 783-803, 1986.
- [21] R. Chang and C. Yang, "Developing a mobile app for game-based learning in middle school mathematics course," International Conference on Applied System Innovation (ICASI), Okinawa, Japan, pp. 1-2, 2016.
- [22] M. Hartono, M. A. Candramata, K. N. Adhyatmoko and B. Yulianto, "Math Education Game for primary school," International Conference on Information Management and Technology (ICIMTech), Bandung, Indonesia, pp. 93-96, 2016.
- [23] A. C. Gomes, B. Ryos, G. Rodrigues and J. P. Filho, "Space chain: A math game for training geometric and arithmetic progressions," IEEE Global Engineering Education Conference (EDUCON), Santa Cruz de Tenerife, Spain, pp. 1997-2002, 2018.
- [24] M. Quiroz and E. Paredes "Plataforma M-learning para incentivar el aprendizaje de Matemáticas en estudiantes de tercero de primaria basado en Unity", Tesis de Titulación, Arequipa, Universidad Católica Santa María, Arequipa, Perú, 2016.
- [25] Real Academia Española. (2022, septiembre 29). Aprendizaje. En Diccionario de la lengua española [Online]. Available: <https://dle.rae.es/aprendizaje>
- [26] D. H. Schunk. "Teorías del aprendizaje". Pearson educación, 2012.
- [27] C. D. Mora. "Estrategias para el aprendizaje y la enseñanza de las matemáticas" Revista de pedagogía, pp. 181-272, 2003.

- [28] P. Neshier. “Problemas relacionados entre lenguaje natural y lenguaje matemático” Gorgorió, N., Deulofeu, J. y Bishop, A. (Coords.). Matemáticas y educación. Retos y cambios desde una perspectiva internacional, pp. 109-124, 2000.
- [29] Real Academia Española. (2022, septiembre 29). Videojuego. En Diccionario de la lengua española [Online]. Available: <https://dle.rae.es/videojuego>
- [30] A. G. Juárez, and T. V. Mombiola. “Los videojuegos (Vol. 14)”. Editorial UOC. 2011.
- [31] M.R. Lepper, and T.W. Malone. “Intrinsic motivation and instructional effectiveness in computer-based education” R. E. Snow, & M. J. Farr (Eds.), Aptitude, learning, and instruction: III. Cognitive and affective process analysis, pp. 255-286, 1987.
- [32] S. Luján-Mora. “Programación de aplicaciones web: historia, principios básicos y clientes web”. Editorial Club Universitario, 2002.
- [33] B. W. Boehm. “A spiral model of software development and enhancement”. Computer, 21(5), pp. 61-72, 1988.
- [34] I. Sommerville. “Ingeniería de Software”. PEARSON EDUCACIÓN, 2011.
- [35] D. Schmidt, A. Gokhale, and R. Natarajan. “Frameworks: Why They Are Important and How to Apply Them Effectively”, 2004.
- [36] M. Hedlefs, A. Garza, M. Sánchez, and A. Garza. “Adaptación al español del Cuestionario de Usabilidad de Sistemas Informáticos CSUQ”, 2015.
- [37] Region Administrativa y de Planeación Especial RAPE Region Central. “Matriz de riesgos por procesos”, 2017.
- [38] D. Añazco. "Tutorial MathDragon" 27 de septiembre de 2022. [Online]. Available: [https://www.youtube.com/watch?v=hT38Ln\\_dMFA](https://www.youtube.com/watch?v=hT38Ln_dMFA).

## Anexos

### 1. Cuestionario de Experiencia de Juego

#### 1. ¿Es usuario de videojuegos?

Sí                       No *-fin del cuestionario-*                       Ocasionalmente

#### 2. ¿Cuántas horas al día dedica al uso de videojuegos?

Menos de 1               1-2                       3-4                       4 o más

#### 3. ¿Dónde utiliza videojuegos regularmente? (puede marcar más de una opción)

Casa                       Colegio                       Otro: \_\_\_\_\_

#### 4. ¿En qué plataforma has utilizado los videojuegos regularmente?

Consola               PC                       Celular                       Otro: \_\_\_\_\_

#### 5. ¿Has utilizado videojuegos para aprender algún tema educativo?

Si                       No

#### 6. El videojuego te pareció divertido.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

#### 7. Fue fácil adaptarse a la forma de control del videojuego.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

#### 8. Fue fácil cumplir con el objetivo del videojuego.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

#### 9. Deseaste volver a jugar para pasar más niveles.

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

#### 10. Fue fácil resolver el problema matemático presentado

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

**11. Deseo jugar más niveles con diferentes operaciones matemáticas**

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

**12. Deseo aprender matemáticas con videojuegos**

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo



## 2. Cuestionario de Usabilidad del Sistema Informático (CSUQ)

### 1. Estoy satisfecho con lo fácil que es utilizar el videojuego.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 2. Soy capaz de completar mis tareas rápidamente utilizando el videojuego.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 3. Fue fácil aprender a utilizar el videojuego.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 4. Creo que me volví experto rápidamente utilizando el videojuego.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 5. El videojuego muestra mensajes de error que me dicen claramente cómo resolver los problemas.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 6. Cada vez que cometo un error utilizando el videojuego, lo resuelvo fácil y rápidamente.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 7. Es fácil encontrar en el videojuego la información que necesito.

( )	( )	( )	( )	( )
Totalmente en desacuerdo	En desacuerdo	Neutral	De acuerdo	Totalmente de acuerdo

### 8. La información que proporciona el videojuego fue efectiva ayudándome a completar las tareas.

( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

**9. La organización de la información del videojuego fue clara.**

( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

**10. La interfaz del videojuego fue placentera.**

( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

**11. Me gustó utilizar el videojuego.**

( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

**12. El videojuego tuvo todas las herramientas que esperaba que tuviera.**

( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

**13. En general, estuve satisfecho con el videojuego.**

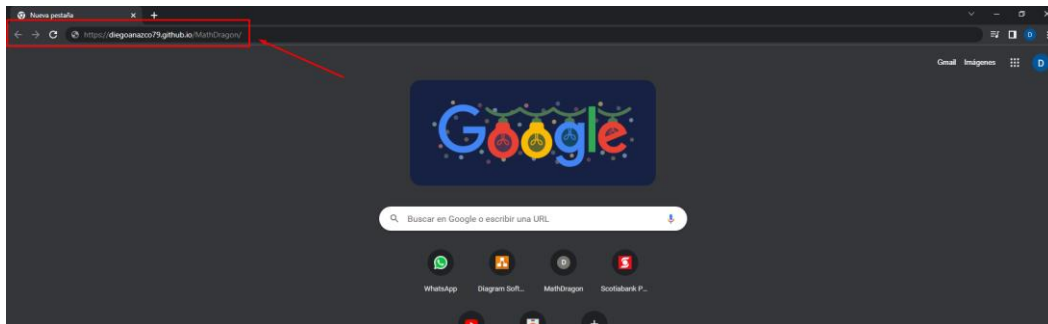
( ) ( ) ( ) ( ) ( )  
Totalmente en desacuerdo En desacuerdo Neutral De acuerdo Totalmente de acuerdo

### 3. Manual de usuario de MathDragon

El siguiente manual tiene como principal objetivo enseñar al usuario como ingresar y jugar MathDragon. De igual forma usted puede encontrar un videotutorial en el siguiente enlace: [https://www.youtube.com/watch?v=hT38Ln\\_dMFA](https://www.youtube.com/watch?v=hT38Ln_dMFA)

#### Inicio del videojuego

Para iniciar el videojuego es necesario tener instalado en su computador/laptop cualquier navegador web (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, etc.) e ingresar al siguiente link <https://diegoanazco79.github.io/MathDragon/> en la barra de búsqueda de su navegador, como se muestra en la siguiente figura (Fig. 41).



*Fig. 41 Manual de Usuario: Ingresar a MathDragon con su enlace*

Fuente: Elaboración propia

#### Escena: Menú principal

Al ingresar al enlace de MathDragon, se encuentra con la escena del menú principal. Para continuar y poder escoger un nivel, se debe hacer clic en el botón de *play* señalado en la Fig. 42



*Fig. 42 Manual de Usuario: Escena del Menú Principal*

Fuente: Elaboración propia

## Selección de niveles

Una vez hecho clic en el botón de *play* en la anterior escena, se tiene la opción de seleccionar dos niveles: Conseguir Comida y Destruir Venenos. Para seleccionar alguno se tiene que hacer clic en su cartel correspondiente (Fig. 43).



*Fig. 43 Manual de Usuario: Escena de Niveles*

Fuente: Elaboración propia

## Niveles: Puntajes, tiempo y controles

El nivel Conseguir Comida y Destruir Veneno comparten los siguientes retos: tiempo y contador de vidas.

Los controles del dragón son las flechas direccionales del teclado o con las teclas A (izquierda), W (arriba), S (abajo), D (derecha).

Si el dragón colisiona con una bomba, el contador de vidas se restará en uno; además, se tiene un temporizador de tiempo si el mismo llega a 0 el nivel finaliza (Fig. 44).



*Fig. 44 Manual de Usuario: Vidas y temporizador*

Fuente: Elaboración propia

## Resolución del problema

Una vez finalizado el nivel, sea por vidas o por tiempo, se presenta un cartel con la operación matemática por resolver. Ver la Fig. 45 y Fig. 46

En las Fig. 45 y Fig. 46 se encuentra señalado el *input* donde el usuario debe ingresar su respuesta y presionar la tecla *enter* o hacer clic en el botón que dice “Enviar”

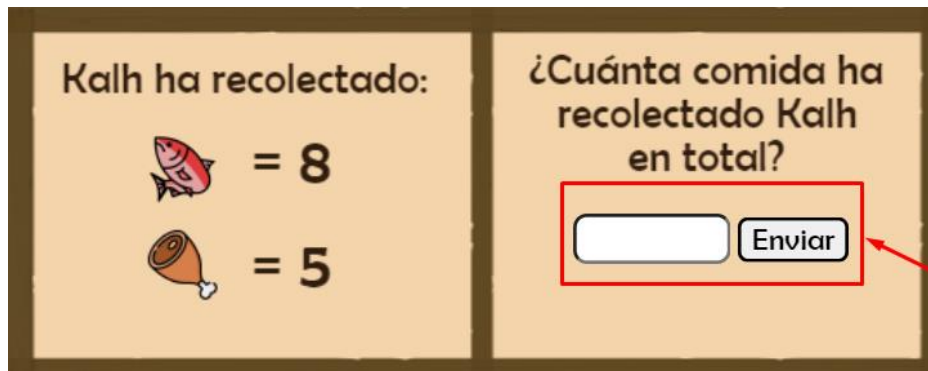


Fig. 45 Manual de Usuario: Cartel nivel Conseguir Comida

Fuente: Elaboración propia

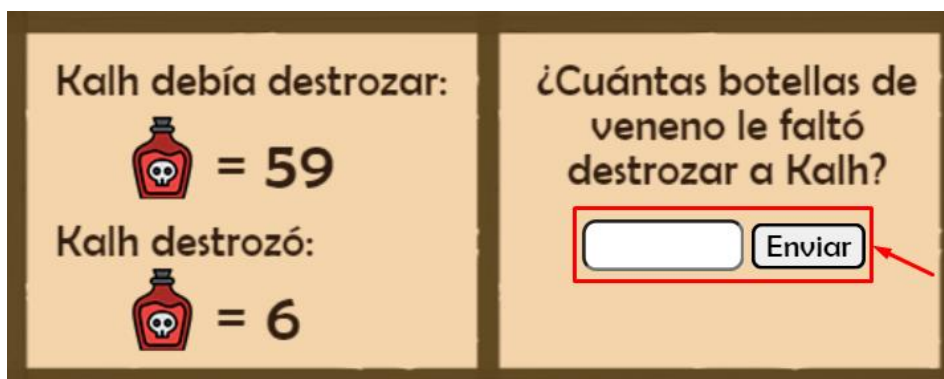


Fig. 46 Manual de Usuario: Cartel Destruir Venenos

Fuente: Elaboración propia

### Reiniciar nivel y volver a selección de niveles

Al responder correctamente el problema matemático o fallar 3 veces seguidas en la respuesta, se muestra el cartel de la Fig. 47. En este cartel se tiene la opción de reiniciar el nivel, el botón del lado derecho y la opción de volver a la escena de selección de niveles, el botón del lado izquierdo.

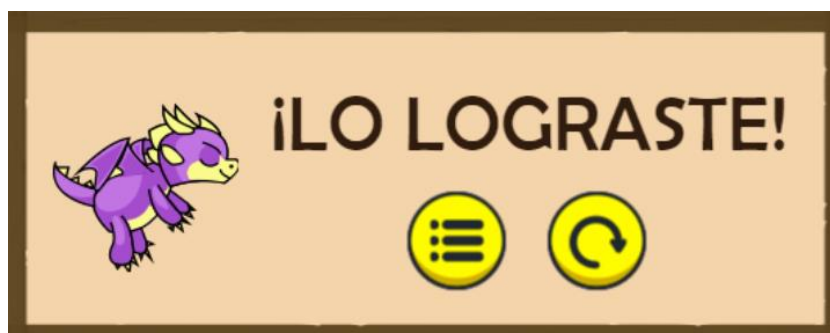


Fig. 47 Manual de Usuario: Cartel de fin del nivel

Fuente: Elaboración propia

#### 4. Ingeniería de Requerimientos: Resumen general de entrevistas a tutores.

Tabla 48. Resumen de respuestas de la entrevista a tutores.

<u>Pregunta</u>	<u>Respuesta</u>		
Nombres y apellidos:	Walter Villanueva	Anghela Bobadilla	Lizbeth Robles
¿Qué trabajo realiza y para quienes?	Docente del primer grado de primaria sección A del colegio San Juan Bautista de La Salle.	Docente del primer grado de primaria sección B del colegio San Juan Bautista de La Salle.	Docente del primer grado de primaria sección C del colegio San Juan Bautista de La Salle.
¿Qué dificultades encuentra en su trabajo actualmente?	Actualmente no se presenta dificultades.	Anteriormente los contenidos se trabajaban en equipo, porque la socialización es muy importante en el aprendizaje, con la educación virtual es muy complicado lograrlo.	La falta de interacción física con el niño, no se puede lograr en la educación virtual.
¿Qué actividades/herramientas hacen que su trabajo sea realice con mayor facilidad?	Hace uso de la plataforma de videojuegos matemáticos PIPO, sin embargo, se encuentra desactualizado.	Plataformas de videoconferencias ayudan mucho a la enseñanza. Pizarras virtuales, juegos interactivos.	Los niños al ser nativos digitales ayudo a que puedan aprender fácilmente el uso de las tecnologías utilizadas.

¿Cómo se enseña el tema de unidades y decenas a los estudiantes?	Tiene que reforzarse el concepto de representaciones gráficas. Herramientas: Ábaco, material base 10.	El niño necesita manipular y ver a través de sus sentidos el cambio que existe entre unidades y decenas. Herramientas: Tablero, material base 10, ábaco.	El niño aprende manipulando cosas y realizando analogías con su entorno. Herramientas: Ábaco, material base 10, tablero, materiales cotidianos (chapitas, bolitas)
¿Cuál es el nivel de conocimientos que debe adquirir el estudiante al finalizar el grado?	Resolución de problemas: Hasta el número 99 (unidades y decenas). Conocimiento numérico: Hasta el número 500 (centenas).	Resolución de problemas: Hasta el número 99 (unidades y decenas). Conocimiento numérico: Hasta el número 500 (centenas).	Resolución de problemas: Hasta el número 99 (unidades y decenas). Conocimiento numérico: Hasta el número 500 (centenas).
¿Con qué concepto(s) presenta más dificultades el estudiante? ¿Cuál es la razón de la problemática?	Inicio de los canjes: Restas prestando y sumas llevando. Razón: Falta de base del concepto de unidades y decenas.	Inicio de los canjes: Restas prestando y sumas llevando. Razón: Falta de base del concepto de unidades y decenas, problemas del concepto de cantidad.	Se debe enseñar a contar, no solo reconocer los números representados. Tiene que comprender el concepto de cantidad.
¿Cómo es la interacción del estudiante con las tecnologías utilizadas? ¿Presentan alguna dificultad?	No hay dificultades, la interacción es muy buena debido a que existe un acompañamiento constante por parte de los padres.	No se ha presentado ninguna dificultad, porque es un trabajo conjunto con los padres.	La interacción es buena, no se presenta dificultades ya que existe un acompañamiento por parte de los padres.
¿Todos los estudiantes cuentan con accesibilidad	Sí, todos cuentan con accesibilidad.	Sí, todos cuentan con accesibilidad	Sí, todos cuentan con accesibilidad.

<p>a las tecnologías utilizadas para la enseñanza no presencial?</p>			
<p>¿Qué opina sobre el uso de un videojuego para enseñanza/reforzamiento en la educación?</p>	<p>A favor de los videojuegos, es una buena estrategia, si es divertido y educativo, tendrá mucho beneficio para el niño.</p>	<p>Completamente a favor, porque los videojuegos despiertan el interés y motivación del niño; es como decirle: “Ve practicando, pero no le estas diciendo que practique, le estás diciendo, ve a jugar”</p>	<p>Los videojuegos son muy importantes, sobre todo en edades cortas, si tomamos un videojuego que ayude a reforzar lo enseñado, es un buen material.</p>
<p>¿Con qué problemas se enfrentaría el estudiante al presentar el videojuego ya implementado?</p>	<p>Tiempo de juego. El videojuego debe tener un control por parte de los padres, o un tiempo límite de juego.</p>	<p>El niño no mide su tiempo de juego, es prudente que haya un control por un adulto.</p>	<p>El problema surge si dejamos solos a los chicos, debe haber un control.</p>
<p>¿Con qué características debería contar un videojuego que enseñe el concepto de unidades y decenas?</p>	<p>No respondió.</p>	<p>Tendría que haber una representación analógica con el entorno, y del material de base 10.</p>	<p>Debe existir pasos previos, se deben presentar pasos a seguir para llegar a una meta.</p>



¿Con qué videojuegos están familiarizados los estudiantes?	Minecraft, Fornite, FreeFire, Minecraft, FreeFire, Fornite, Among Us. Pokémon Go.	No he escuchado de videojuegos en los niños de primer grado, pero en grados mayores sí.
--	--	---

Fuente: Elaboración propia.

## 5. Matriz de Trazabilidad de Pruebas de Sistema

Tabla 49. Matriz de trazabilidad de pruebas de Sistema

	<u>Título de Prueba</u>	<u>Descripción</u>	<u>Módulo/Escena</u>	<u>Datos de Entrada</u>	<u>Resultado Esperado</u>	<u>Resultado Obtenido</u>	<u>Error SI/NO</u>	<u>Comentarios</u>
1	Inicialización del videojuego	Se ingresará al URL del videojuego para inicializarlo	Menú Principal	Ingresar a la URL	Juego incializado.	Inicialización del juego sin errores	NO	
2	Empezar el videojuego.	Se hace click en el botón de play en la escena inicial	Menú principal	Click en el botón de play.	Cambio de escena para escoger nivel	Cambio de escena a “Niveles”	NO	

3	Iniciar nivel de “Conseguir Comida”	Se busca iniciar el nivel “Conseguir comida” correctamente	Niveles	Click en el cartel del nivel “Conseguir comida”	Nivel “Conseguir comida” iniciado correctamente	Nivel “Conseguir comida” iniciado	NO
4	Iniciar nivel de “Destruir venenos”	Se busca iniciar el nivel “Destruir venenos” correctamente	Niveles	Click en el cartel del nivel “Destruir venenos”	Nivel “Destruir venenos” iniciado correctamente	Nivel “Destruir venenos” iniciado	NO
5	Conseguir un “pescado” como puntaje	En el nivel “Conseguir comida” al hacer colisión con un pescado, desparezca	Nivel Conseguir Comida	Colisionar el dragón con el pescado	El objeto del pescado desaparezca al colisionar	El pescado desaparece al colisionar con el dragón	NO
6	Conseguir un “muslo” como puntaje	En el nivel “Conseguir comida” al hacer colisión con un	Nivel Conseguir Comida	Colisionar el dragón con el muslo	El objeto del muslo desaparezca al colisionar	El muslo desaparece al colisionar con el dragón	NO

		muslo, desparezca					
7	Conseguir un “veneno” como puntaje	En el nivel “Destruir venenos” al hacer colisión con un veneno, desparezca	Nivel Destruir Venenos	Colisionar el dragón con el veneno	El objeto del veneno desparezca al colisionar	El veneno desaparece al colisionar con el dragón	NO
8	Verificar la suma de puntaje en el table de los niveles	Cuando se consigue un punto de muslo, pescado o veneno, se refleje en el tablero principal	Nivel Conseguir Comida y Destruir Venenos	Colisionar con los diferentes objetos: muslo, pescado o veneno.	El tablero se actualizará según las colisiones del dragón con los objetos.	El tablero se actualiza correctamente	NO
9	Verificar el temporizador en los niveles	El temporizador es aleatorio y además se va restando	Nivel Conseguir Comida y Destruir Venenos	Inicialización del nivel	El temporizador del table disminuirá por cada segundo de	El temporizador disminuye en segundos en los niveles	NO

		conforme pasa el tiempo			juego en el nivel.		
10	Verificar el contador de vidas los niveles	El contador de vidas del tablero se inicializa en tres vidas al ingresar al nivel	Nivel Conseguir Comida y Destruir Venenos	Inicialización del nivel	El contador de vidas se encontrará en 3 corazones	El contador de vidas iniciado con 3 corazones	NO
11	Restar vida al momento de chocar en una “bomba”	El contador de vidas del tablero disminuye al colisionar el dragón con una bomba	Nivel Conseguir Comida y Destruir Venenos	Colisionar el dragón con una bomba	El contador de vidas se restará un corazón.	El contador de vidas se resta un corazón	NO
12	Evitar que el dragón se salga de los bordes del videojuego	Mover al dragón por todo el escenario evitando que exista superposición en el tablero o	Nivel Conseguir Comida y Destruir Venenos	Flechas direccionales del teclado	El dragón no saldrá de los límites del escenario principal del nivel	El dragón se mantiene en los límites del escenario del nivel respectivo	NO

		fuera del escenario					
13	Verificar que las flechas direccionales sean funcionales en los niveles	Se presionan las teclas direccionales para mover al dragón	Nivel Conseguir Comida y Destruir Venenos	Flechas direccionales del teclado	El dragón se moverá en todas las direcciones correctamente,	El dragón se mueve en todas las direcciones según la tecla presionada.	NO
14	Verificar que el nivel termine cuando el temporizador del tiempo llegue a cero	Se evitan las bombas para que el temporizador del nivel llegue a cero	Nivel Conseguir Comida y Destruir Venenos	Flechas direccionales del teclado	El nivel terminará cuando el temporizador llegue a cero	El nivel terminó cuando el temporizador llegó a cero	NO
15	Verificar que el nivel termine cuando las vidas lleguen a cero	Se colisionan con las bombas para que el dragón pierda vidas	Nivel Conseguir Comida y Destruir Venenos	Flechas direccionales del teclado	El nivel terminará cuando el contador de vidas llegue a cero	El nivel terminó cuando el contador de vidas llegó a cero	NO

16	Animación de daño del dragón	La animación del daño debe inicializarse cuando se colisiona con una bomba	Nivel Conseguir Comida y Destruir Venenos	Colisión del dragón con una bomba	La animación de daño se activará cuando colisionen el dragón con una bomba	La animación de daño se activó cuando hubo colisión del dragón con una bomba	NO
17	Animación de movimiento del dragón	La animación de movimiento tiene que estar constantemente activa durante el nivel “Conseguir comida”	Nivel Conseguir Comida	Inicialización de la escena “Conseguir comida”	La animación de movimiento se inicializará cuando inicie el nivel de “Conseguir comida”	La animación del movimiento se inició al ingresar al nivel de “Conseguir comida”	NO
18	Animación muerte del dragón	La animación de muerte debe inicializarse cuando el contador de	Nivel Conseguir Comida y Destruir Venenos	Colisión del dragón con bombas tres veces.	La animación de muerte se iniciará cuando el contador de vidas llegue a cero	La animación de muerte se inició cuando el contador de vidas llegó a cero	NO

		vidas llegue a cero					
19	Animación destrucción del dragón	La animación de destrucción tiene que estar constantemente activa durante el nivel “Destruir venenos”	Nivel Destruir Venenos	Inicialización de la escena “Destruir venenos”	La animación de destrucción se inicializará cuando inicie el nivel de “Destruir venenos”	La animación de destrucción se inició al ingresar al nivel de “Destruir venenos”	NO
20	Objetos renderizados aleatoriamente	Los objetos: carnes, muslos, venenos y bombas se renderizan durante la escena.	Nivel Conseguir Comida y Destruir Venenos	Inicialización de la escena	Los objetos se renderizarán aleatoriamente en el escenario	Los objetos se renderizaron aleatoriamente en el escenario	NO
21	Tablero de resultados inicializado correctamente	Cuando finaliza el nivel, se debe mostrar un table con los	Nivel Conseguir Comida y Destruir Venenos	Finalización del nivel	El tablero de resultados se mostrará correctamente	El tablero de resultados se muestra correctamente.	NO

		resultados y la pregunta del nivel					
22	Datos recolectados mostrados correctamente en el tablero de resultados	En el tablero de resultados, los datos del problema corresponden a los puntajes obtenidos en el nivel	Nivel Conseguir Comida y Destruir Venenos	Finalización del nivel	Los datos del tablero de resultados concordarán con los puntajes del nivel	Los datos del tablero de resultados concuerdan con los puntajes	NO
23	Input de respuesta funcional	El input de respuesta funciona correctamente al ingresar un valor	Nivel Conseguir Comida y Destruir Venenos	Ingresar números en el input de respuesta	El input de respuesta enviará y evaluará si la respuesta es correcta o no	El input de respuesta envía y evalúa la respuesta	NO
24	Input solo acepte números	Ingresar un valor invalido al	Nivel Conseguir Comida y	Ingresar cualquier valor	El input de respuesta no	El input de respuesta recibe	NO



		input de respuesta	Destruir Venenos	que no se numérico	recibirá un valor no numérico	solo valores numéricos	
25	Renderizado de la primera pista	Al fallar una primera vez en la respuesta se renderiza correctamente la primera pista	Nivel Conseguir Comida y Destruir Venenos	Ingresar una respuesta errónea en el input de respuesta	Se renderizaría una pista al tener un primer fallo en la respuesta final del usuario	Se renderiza una pista al fallar por primera vez en nuestra respuesta	NO
26	Renderizado de la segunda pista	Al fallar por segunda vez en la respuesta se renderiza correctamente la segunda pista	Nivel Conseguir Comida y Destruir Venenos	Ingresar una segunda respuesta errónea en el input de respuesta	Se renderizaría una pista al tener un segundo fallo en la respuesta final del usuario	Se renderiza una pista al fallar por segunda vez en nuestra respuesta	NO
27	Renderizado del cartel de nivel perdido	Al fallar por tercera vez en la respuesta se renderiza correctamente el	Nivel Conseguir Comida y Destruir Venenos	Ingresar una tercera respuesta errónea en el input de respuesta	Se renderizaría un cartel de nivel perdido al fallar tres veces consecutivas	Se renderiza el cartel de nivel perdido al fallar tres veces	NO

		cartel de nivel perdido					
28	Renderizado del cartel de nivel superado	Al ingresar la respuesta correcta sin importar el intento, se debe renderizar el cartel de nivel superado	Nivel Conseguir Comida y Destruir Venenos	Ingresar una respuesta correcta en el input de respuesta	Se renderizaría el cartel de nivel superado al ingresar una respuesta correcta	Se renderizó el cartel de nivel superado al ingresar una respuesta correcta	NO
29	Botón de volver a los niveles funcional	Presionar el botón de volver a los niveles.	Nivel Conseguir Comida y Destruir Venenos	Click en el botón de volver a los niveles	Se renderizaría la escena de “Niveles”	Se renderizó la escena “Niveles” correctamente con todas sus funcionalidades	NO
30	Botón de repetir nivel actual funcional	Presionar el botón de volver a repetir el nivel actual.	Nivel Conseguir Comida y Destruir Venenos	Click en el botón de repetir nivel actual	Se renderizaría nuevamente la escena actual	Se renderiza el nivel actual nuevamente con	NO

---

todas sus  
funcionalidades

---

Fuente: Elaboración propia

## 6. Matriz de Riesgos

	<u>Riesgo</u>	<u>Probabilidad</u>	<u>Gravedad</u>	<u>Valor</u>	<u>Nivel</u>	<u>Estrategia</u>
1	El videojuego puede generar un uso excesivo de los dispositivos móviles a los estudiantes.	3	3	9	Importante	El videojuego contará con un temporizador que medirá el tiempo de juego. Cuando el temporizador termine, el videojuego finaliza.
2	El estudiante no cuenta con computadora o laptop para poder jugar el videojuego.	1	5	5	Apreciable	El videojuego será implementado de manera responsiva para su correcto funcionamiento desde el dispositivo móvil utilizado.

---

3	Un diseño demasiado complejo exige tener en cuenta complicaciones innecesarias e improductivas en la implementación.	3	4	12	Importante	Asesorarse con expertos en la tecnología que se utilizará, para realizar un diseño adecuado para los requerimientos de los docentes.
4	La falta de conocimiento de una metodología o herramienta técnica necesaria para la ejecución del proyecto.	4	4	16	Muy grave	Estudiar a fondo la metodología o herramienta a utilizar mediante cursos online, documentación o persona experta.
5	Los docentes no participan en los ciclos de revisión de los planes, prototipos y especificaciones.	2	5	10	Importante	Reunirse con el director encargado para que incentive la participación de los docentes en el proyecto.
6	Poca participación por parte de los estudiantes.	2	5	10	Importante	Reunirse con los docentes para que incentive la participación de los estudiantes.

Fuente: Elaboración propia.