



Universidad  
Rey Juan Carlos

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA**

**GRADO EN INGENIERÍA DEL SOFTWARE**

**Curso Académico 2023/2024**

**Trabajo Fin de Grado**

**DESARROLLO DE UNA APLICACIÓN DE JUEGO CON UNITY PARA  
ALUMNOS DE EDUCACIÓN PRIMARIA**

**Autora:** Marina Fernández Suárez

**Directora:** María Zapata Cáceres

# Licencia



Imagen 1. Licencia Creative Commons

Usted es libre de:

- **Compartir:**

Copiar y redistribuir el material en cualquier medio o formato.

- **Adaptar:**

Remezclar, transformar y construir a partir del material. **Bajo los siguientes términos:**

- **Atribución:**

Usted debe dar crédito de manera adecuada, brindar un enlace a la licencia, e indicar si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.

- **No Comercial:**

Usted no puede hacer uso del material con propósitos comerciales.

- **Compartir Igual:**

Si remezcla, transforma o crea a partir del material, debe distribuir su contribución bajo la misma licencia del original.

# Agradecimientos

Quiero expresar mi profunda gratitud a todas las personas que han contribuido a lo largo de este proyecto y de mi trayectoria académica.

A mis profesores, por compartir su tiempo y conocimientos, y por elegir la docencia como vocación. Su dedicación y sabiduría han sido fundamentales en mi formación. A mis tutores, por su inquebrantable apoyo, paciencia, y sobre todo, por creer en mis capacidades. Sus consejos y orientación han sido esenciales para alcanzar este logro.

A mis padres y hermanas, por su amor incondicional y por motivarme a dar lo mejor de mí en cada paso del camino. Su apoyo ha sido mi mayor fortaleza.

A mi abuelita, por escucharme pacientemente hablar sobre informática, aunque no comprendiera los detalles. Su amor y atención han sido un bálsamo para mi espíritu.

Al resto de mi familia, por sus constantes ánimos y por estar siempre presentes en los momentos difíciles. Su cariño y respaldo han sido vitales en mi camino.

A una persona muy especial, por su aliento constante, por el tiempo dedicado a ayudarme y a permitirme trabajar en paz. Gracias por confiar en mí y en mis habilidades, tu apoyo ha sido inestimable.

A mis amigos, por su comprensión y por estar siempre dispuestos a escucharme. Gracias por ser una vía de desconexión y por sus consejos llenos de buenos sentimientos.

A mis compañeros de ambas carreras, por compartir tantos recuerdos, tanto buenos como desafiantes, por las horas de estudio y trabajo conjunto, y por su compañerismo a lo largo de estos años.

A las mujeres de MIRA, a mis compañeros de inglés y a mis vecinos, por darme fuerzas cuando flaqueaba y por sus constantes ánimos. Su apoyo ha sido una fuente de inspiración y motivación.

# Resumen

Este proyecto se centra en la creación de una aplicación educativa diseñada para desarrollar las habilidades de pensamiento computacional (PC) en alumnos de la primera etapa de Educación Primaria. La finalidad de esta aplicación es complementar las enseñanzas impartidas en las escuelas, alineándose con los objetivos educativos establecidos en la legislación vigente.

El desarrollo de la aplicación se lleva a cabo utilizando el motor de videojuegos Unity, una herramienta robusta y versátil que permite crear experiencias interactivas y educativas. El juego propuesto es un Nonograma en 2D, diseñado para entrenar y evaluar habilidades estratégicas y de descomposición de problemas, esenciales para el desarrollo del PC. La interfaz del juego es intuitiva y visualmente atractiva, adaptada a las capacidades cognitivas y motrices de los niños en esta etapa educativa.

El resultado final es una aplicación educativa que permite a los alumnos de Educación Primaria desarrollar sus habilidades de PC de manera efectiva y divertida. La aplicación no solo refuerza los conocimientos académicos de los alumnos, sino que también fomenta su motivación y curiosidad científica. Además, cumple con las directrices educativas establecidas por la normativa vigente, representando una herramienta valiosa para los educadores en la implementación de competencias transversales en el currículo escolar.

En resumen, este proyecto demuestra cómo el uso de tecnologías innovadoras como Unity puede potenciar el aprendizaje y desarrollo de habilidades clave en los estudiantes, alineándose con los objetivos educativos modernos y proporcionando una herramienta eficaz y accesible para la educación primaria.

# Abstract

This project focuses on creating an educational application designed to develop computational thinking (CT) skills in students in the first stage of Primary Education. The purpose of this application is to complement the teachings imparted in schools, aligning with the educational objectives established in current legislation.

The application development is carried out using the Unity game engine, a robust and versatile tool that allows for the creation of interactive and educational experiences. The proposed game is a 2D Nonogram, designed to train and assess strategic and problem-decomposition skills, essential for the development of CT. The game's interface is intuitive and visually appealing, adapted to the cognitive and motor abilities of children at this educational stage.

The final result is an educational application that enables Primary Education students to effectively and enjoyably develop their CT skills. The application not only reinforces the academic knowledge of the students but also fosters their motivation and scientific curiosity. Additionally, it complies with the educational guidelines established by current regulations, representing a valuable tool for educators in the implementation of cross-curricular competencies in the school curriculum.

In summary, this project demonstrates how the use of innovative technologies like Unity can enhance the learning and development of key skills in students, aligning with modern educational objectives and providing an effective and accessible tool for primary education.

# Índices

## Índice de contenidos

<b>Resumen</b>	<b>4</b>
<b>Abstract</b>	<b>5</b>
<b>Capítulo I. Introducción</b>	<b>12</b>
<b>1.2 Definición de Pensamiento computacional</b>	<b>12</b>
<b>1.3 Legislación</b>	<b>13</b>
<b>1.4 Motivación</b>	<b>14</b>
<b>1.5 Objetivos</b>	<b>15</b>
<b>1.6 Herramientas</b>	<b>16</b>
Motor.....	16
Photoshop .....	17
VisualStudio.....	17
<b>1.7 Metodologías y planificación</b>	<b>18</b>
Kanban.....	18
Trello.....	18
Diagrama de Gantt .....	22
<b>1.8 Estructura de la memoria</b>	<b>23</b>
<b>Capítulo II. Marco Teórico</b>	<b>25</b>
<b>2.1 Análisis histórico</b>	<b>25</b>
<b>2.2 Pensamiento Computacional en Primaria</b>	<b>26</b>
<b>2.3 Proyectos relacionados con el PC</b>	<b>26</b>
Proyecto EPCIA .....	26
CodeWeek .....	28
Enfoque educativo STEAM .....	28
<b>2.4 Fundamentos teóricos</b>	<b>29</b>
Evaluación de Pensamiento Computacional.....	30

Computational Thinking test (CTt) .....	30
Beginners Computational Thinking test (BCTt) .....	30
Competent Computational Thinking Test (cCTt).....	31
<b>2.5 Casos de estudio</b>	<b>32</b>
<b>Capítulo III. Diseño de la aplicación</b>	<b>38</b>
<b>3.1 Introducción</b>	<b>38</b>
<b>3.2 Game Design Document (GDD)</b>	<b>38</b>
Resumen Ejecutivo .....	38
Concepto del Juego .....	39
Jugabilidad.....	39
Mecánicas de Juego .....	39
Interfaz de Usuario.....	40
Diseño de Puzzles.....	40
Requisitos Técnicos .....	40
Plan de Pruebas.....	40
Plan de Lanzamiento y Comercialización .....	41
<b>3.3 Diseño de artes</b>	<b>41</b>
Interfaz de usuario y experiencia de usuario (UX).....	41
Diseño de Interfaz de Usuario (UI).....	43
Tipografía.....	43
Sprites.....	46
<b>Capítulo IV. Desarrollo de la aplicación</b>	<b>50</b>
<b>4.1 Desarrollo</b>	<b>50</b>
<b>4.2 Diseño Software</b>	<b>51</b>
<b>4.3 Requisitos</b>	<b>52</b>
Funcionales.....	52
No Funcionales .....	52
<b>4.4 Diagrama UML de Clases</b>	<b>53</b>
<b>4.5 Descripción de las clases</b>	<b>54</b>
<b>4.7 Resultado final de la aplicación</b>	<b>57</b>

<b>Capítulo V. Validación</b>	<b>60</b>
<b>5.1 Pruebas de Interfaz de Usuario (UI)</b>	<b>60</b>
<b>5.2 Pruebas de Lógica del Juego</b>	<b>61</b>
<b>5.3 Pruebas de Manejo de Datos del Usuario</b>	<b>62</b>
<b>5.4 Pruebas de rendimiento</b>	<b>63</b>
Resultados de las Pruebas de Rendimiento .....	65
<b>5.5 Corrección de Errores</b>	<b>66</b>
<b>5.6 Validación de requisitos</b>	<b>67</b>
<b>Capítulo VI. Conclusiones</b>	<b>69</b>
<b>6.1 Cumplimiento de objetivos</b>	<b>69</b>
<b>6.2 Logros y Desafíos</b>	<b>70</b>
<b>6.3 Lecciones Aprendidas</b>	<b>71</b>
<b>6.4 Perspectivas Futuras</b>	<b>72</b>
<b>Glosario</b>	<b>74</b>
<b>Anexo</b>	<b>81</b>
<b>Bibliografía</b>	<b>82</b>



# Índice de Ilustraciones

Imagen 1. Licencia Creative Commons.....	2
Imagen 2. Tablero Trello (1) .....	20
Imagen 3. Tablero Trello (2) .....	21
Imagen 4. Tablero Trello (3) .....	21
Imagen 5. Diagrama de Gantt.....	22
Imagen 6. Code Week .....	28
Imagen 7. STEAM.....	28
Imagen 8. La Torre del Conocimiento .....	32
Imagen 9. Discover Babylon .....	32
Imagen 10. Simple Machines.....	32
Imagen 11. Brain Training Infernal del Dr. Kawashima.....	32
Imagen 12. Flippy's Tesla: Inventemos el futuro .....	32
Imagen 13. Civilization.....	33
Imagen 14. Monument Valley .....	33
Imagen 15. No man's sky .....	33
Imagen 16. Animal Crossing .....	33
Imagen 17. Zoo Tycoon 2 .....	33
Imagen 18. Super Lucky's Tale.....	33
Imagen 19. Minecraft: Education Edition.....	35
Imagen 20. Kahoot.....	35
Imagen 21. DragonBox.....	35
Imagen 22. CodeCombat .....	36
Imagen 23. Quizzit .....	36
Imagen 24. Quizlet.....	36
Imagen 25. Classcraft .....	36

Imagen 26. Duolingo .....	36
Imagen 27. Lightbot .....	36
Imagen 28 Open-Dyslexic Font .....	44
Imagen 29 Comic Neue Font.....	44
Imagen 30 Sassoon Primary Font.....	44
Imagen 31 Jolly Lodger Font .....	45
Imagen 34. Estilo 1.....	48
Imagen 34. Estilo 2.....	48
Imagen 34. Estilo 3.....	48
Imagen 35. Diagrama de Clases.....	53
Imagen 36. Pantallazos App 1.....	57
Imagen 37. Pantallazos App 2.....	58
Imagen 38. Pantallazos App 3.....	58
Imagen 39. Pantallazos App 4.....	59
Imagen 40. Pantallazos App 5.....	59
Imagen 41. Texto auxiliar de verificación de la solución.....	62
Imagen 42. Pruebas Profiler CPU - 1 .....	63
Imagen 43. Pruebas Profiler CPU - 2 .....	64
Imagen 44. Pruebas Profiler Rendering .....	64
Imagen 45. Pruebas Profiler Memory .....	64
Imagen 46. Pruebas Profiler UI.....	65

## Índice de tablas

Tabla 1. Pros y Contras de Unreal Engine y Unity .....	16
Tabla 2. Casos de estudio: Videojuegos para Aprender Jugando .....	34
Tabla 3. Casos de estudio: Juegos para el aprendizaje.....	37
Tabla 4. Estudio de Fuentes.....	45
Tabla 5. Comparativa de estilos para los Sprites .....	47
Tabla 6. Comparativa de Tipografías .....	49

# Capítulo I. Introducción

Se plantea el desarrollo de una aplicación de juego mediante el *motor de videojuegos* Unity para complementar las enseñanzas de la competencia de pensamiento computacional (en adelante PC) en el primer ciclo de Educación Primaria.

El objetivo es diseñar la aplicación de manera que sea accesible para usuarios sin ningún conocimiento previo del tema. Tanto el estilo como la jugabilidad deben ser sencillos y visualmente atractivos para el público objetivo, es decir, alumnos de primaria.

## 1.2 Definición de Pensamiento computacional

El PC ha tenido varias definiciones a lo largo de la historia. La primera definición formal es la de la ingeniera estadounidense Jeannette Wing [1], la cual lo definió como: *“... proceso de pensamiento envuelto en formular un problema y sus soluciones de manera que las soluciones son representadas de una forma en que pueden ser llevadas a un agente de procesamiento de información”* [2].

El desarrollo del PC mejora implica la habilidad de descomponer un problema en partes manejables, abstraer los aspectos esenciales del problema, organizar los datos de manera lógica, y pensar de manera algorítmica para identificar soluciones eficientes. Estas habilidades incluyen la descomposición, la abstracción, el reconocimiento de patrones, y el diseño de algoritmos.[3].

En base a las definiciones anteriores, se extrae la siguiente definición, que se utilizará en este proyecto: *“El PC es una destreza cognitiva a la que se asocian habilidades como la abstracción, paralelización, descomposición de problemas, algoritmia, recursión, etc.”.*

Tal y como se menciona en la tesis *“Enseñanza, evaluación y análisis de habilidades de PC en etapas tempranas”*: *“El aprendizaje y evaluación de habilidades que contribuyen al desarrollo del PC en la educación reglada está cobrando una gran importancia a nivel internacional, habiendo consenso en que este aprendizaje debe comenzar en edades tempranas”* [4].

## 1.3 Legislación

En la Ley Orgánica 3/2020, de 29 de diciembre, por la que se modifica la Ley Orgánica 2/2006, de 3 de mayo, de Educación (*LOMLOE*), se incluye el PC como una competencia esencial en el currículo escolar. Esta inclusión se detalla y amplía en el siguiente Real Decreto (RD):

- Real Decreto 157/2022, de 1 de marzo, por el que se establece la ordenación y las enseñanzas mínimas de la Educación Primaria [1].

En la normativa mencionada, se especifican las competencias relacionadas con la resolución de problemas mediante proyectos de diseño y la aplicación del PC en el contexto de la Educación Primaria.

El proyecto se apoya en esa normativa, que establece las enseñanzas mínimas de la educación, incluyendo el PC como una competencia transversal en diversas asignaturas. También hace hincapié en la elaboración colaborativa de proyectos que aborden desafíos en distintos entornos como el tecnológico. Se destacan las fases esenciales del desarrollo de proyectos, tales como la detección de necesidades, el diseño, la creación de prototipos y la evaluación de resultados.

Además, se menciona el uso del PC para descomponer problemas, reconocer patrones y crear algoritmos. Se resalta la importancia de fomentar la creatividad, la innovación y el trabajo cooperativo, así como el desarrollo de habilidades como la argumentación, la comunicación efectiva y la toma de decisiones compartidas. Estas competencias se relacionan con diversos *descriptores del Perfil de Salida*, como *STEAM*, desarrollo personal y social, y competencia digital.

En resumen, la legislación educativa vigente promueve la integración de herramientas tecnológicas y juegos educativos para el desarrollo de habilidades clave en los estudiantes. Las aplicaciones diseñadas para enseñar programación y PC pueden ayudar a los niños a desarrollar habilidades de resolución de problemas y lógica de una manera lúdica y accesible. Estas herramientas están alineadas con las directrices educativas que fomentan el uso de tecnologías digitales en el aula.

## 1.4 Motivación

En la actualidad, la informática está cada vez más presente en nuestras vidas cotidianas, influyendo en prácticamente todos los aspectos de nuestra existencia. Desde la forma en que nos comunicamos y trabajamos hasta cómo aprendemos y nos entretenemos, la informática desempeña un papel fundamental. Dado este contexto, resulta imperativo desarrollar habilidades relacionadas con la informática en edades tempranas para preparar a las nuevas generaciones para enfrentar los desafíos de la sociedad actual y futura.

La integración de la informática en la educación primaria no solo fortalece el conocimiento técnico de los niños, sino que también les ayuda a desarrollar habilidades esenciales como el pensamiento crítico, la resolución de problemas y la creatividad. Una de las herramientas más efectivas para lograr estos objetivos es el uso de aplicaciones educativas.

Numerosos estudios han demostrado que las aplicaciones, especialmente las que incorporan elementos de *gamificación* [5], pueden ser altamente efectivas para el aprendizaje. Estas aplicaciones no solo hacen que el proceso de aprendizaje sea más divertido y atractivo, sino que también facilitan la comprensión de conceptos complejos a través de la interacción y la práctica. Por ejemplo, aplicaciones diseñadas para enseñar programación y PC pueden ayudar a los niños a desarrollar habilidades de resolución de problemas y lógica de una manera lúdica y accesible.

Sin embargo, a pesar de los beneficios comprobados, existen pocas herramientas que ayuden a los niños a desarrollar habilidades relacionadas con el PC. Esto representa una oportunidad significativa para desarrollar aplicaciones que no solo entretengan, sino que también eduquen y preparen a los niños para un futuro en el que la competencia digital será cada vez más crucial. Por estas razones, se ha decidido crear una aplicación específica que aborde esta necesidad, utilizando un enfoque educativo y lúdico para enseñar habilidades de PC a los niños de educación primaria.

## 1.5 Objetivos

En este apartado, se detallan tanto el objetivo principal, como los secundarios que guiarán el desarrollo de la aplicación, asegurando que se cumplan todas las expectativas y requerimientos establecidos.

### Objetivo Principal (OP)

- **OP:** Crear una aplicación destinada a ayudar a desarrollar el PC en niños de educación primaria.

### Objetivos Secundarios (OS)

- **OS 1:** Realizar una investigación sobre los videojuegos educativos existentes, evaluando su efectividad y metodología.
- **OS 2:** Realizar una investigación sobre los conceptos y metodologías relacionados con el PC.
- **OS 3:** Diseñar y desarrollar una aplicación de juego de puzles que sea educativa y atractiva para los niños.
- **OS 4:** Implementar la aplicación de juego en 2D utilizando el motor de desarrollo Unity, aprovechando sus capacidades para crear experiencias interactivas y visualmente atractivas.
- **OS 5:** Realizar una validación exhaustiva de la aplicación mediante diversos tipos de pruebas, incluyendo pruebas de usabilidad, funcionalidad y retroalimentación de los usuarios.
- **OS 6:** Analizar los resultados obtenidos de las pruebas y sacar conclusiones sobre el trabajo realizado, identificando áreas de mejora y futuros desarrollos.

Estos objetivos están diseñados para garantizar que el proyecto no solo cumpla con su propósito educativo, sino que también proporcione una experiencia de usuario de alta calidad y esté respaldado por una sólida base de investigación.

## 1.6 Herramientas

En el contexto de este proyecto, se han seleccionado diversas herramientas que facilitan tanto la creación de la aplicación como su optimización y mantenimiento.

### Motor

La elección del motor de desarrollo es una de las decisiones más importantes al inicio de un proyecto. Los motores más populares son Unreal Engine y Unity, ambos permitiendo el uso de su software de forma gratuita.

A continuación, se detallan las ventajas y desventajas de cada uno:

	Ventajas	Desventajas
<b>Unreal Engine</b>	<ul style="list-style-type: none"><li>Interfaz de usuario más avanzada e intuitiva.</li><li>Posibilidad de crear gráficos de alta calidad y realismo.</li><li>Gran cantidad de recursos y comunidad de desarrolladores disponibles en línea.</li><li>Compatibilidad con una amplia gama de plataformas, incluidos ordenadores, consolas y dispositivos móviles.</li></ul>	<ul style="list-style-type: none"><li><i>Curva de aprendizaje</i> más difícil en comparación con Unity.</li><li>Requiere una mayor cantidad de recursos del sistema.</li><li>Costo elevado de licenciamiento para desarrolladores comerciales.</li></ul>
<b>Unity</b>	<ul style="list-style-type: none"><li><i>Curva de aprendizaje</i> más suave en comparación con Unreal Engine.</li><li>Es más fácil de usar y requiere menos recursos del sistema.</li><li>Costo más bajo de licenciamiento para desarrolladores comerciales.</li><li>Gran cantidad de recursos y comunidad de desarrolladores disponibles en línea.</li></ul>	<ul style="list-style-type: none"><li>Interfaz de usuario menos avanzada en comparación con Unreal Engine.</li><li>Limitaciones en la creación de gráficos de alta calidad y realismo.</li><li>Compatibilidad limitada con algunas plataformas.</li></ul>

Tabla 1. Pros y Contras de Unreal Engine y Unity



La primera opción considerada fue Unreal Engine, pero la elección final fue Unity. Esto se debió a varias razones, siendo una de las más importantes su *curva de aprendizaje* más asequible para un desarrollador sin experiencia en el campo del desarrollo de aplicaciones.

En general, Unity es considerado más fácil de aprender que Unreal Engine, lo que lo hace una opción más apropiada para desarrolladores sin experiencia previa en aplicaciones.

Además, el objetivo general del proyecto no es solo el desarrollo de una aplicación que cumpla los requisitos para ser utilizada en las aulas, sino también hacerlo de manera sencilla, partiendo de la base de estudios en Ingeniería del Software y no específicos para videojuegos. Unity es uno de los motores de desarrollo más populares y con una gran comunidad de usuarios, lo que significa que hay muchos recursos y tutoriales disponibles para ayudar en el proceso de desarrollo.

En conclusión, la elección de Unity como motor para este proyecto se considera una decisión acertada, ya que su curva de aprendizaje es más asequible y su objetivo principal es facilitar el desarrollo de la aplicación. La popularidad de Unity y su amplia comunidad de usuarios aseguran que se dispone de numerosos recursos para apoyar el proceso de desarrollo.

## Photoshop

Adobe Photoshop es una herramienta útil en el desarrollo de aplicaciones debido a sus capacidades avanzadas de edición gráfica. Se ha utilizado para la creación y edición de los *sprites* y otros elementos visuales de la interfaz de usuario. Su amplia gama de funcionalidades permite ajustar detalles visuales con precisión, lo que es crucial para mantener la calidad estética de la aplicación.

La elección de Photoshop se debe a su versatilidad y robustez, características que lo convierten en una herramienta estándar en la industria del diseño gráfico

## VisualStudio

Visual Studio es un entorno de desarrollo integrado (IDE) y que ofrece una amplia gama de herramientas para la programación y depuración de aplicaciones. En este proyecto, Visual Studio ha sido utilizado para la codificación y prueba de la lógica del juego y la interfaz de usuario.

Sus características avanzadas de depuración, junto con su integración con Unity, facilitan el desarrollo y la corrección de errores en el código. Además, Visual Studio proporciona un entorno de trabajo eficiente con soporte para múltiples lenguajes de programación, lo que lo convierte en una opción ideal para el desarrollo de esta aplicación

## 1.7 Metodologías y planificación

El proyecto de desarrollo se ha planteado utilizando la metodología *Agile de Kanban*, la cual es conocida por su flexibilidad y capacidad para adaptarse a los cambios durante el proceso de desarrollo. La metodología *Kanban* se basa en la visualización del flujo de trabajo, permitiendo gestionar y optimizar las tareas de manera efectiva.

Para llevar a cabo la planificación y seguimiento del proyecto, se ha utilizado la herramienta Trello. Trello ha sido fundamental para la gestión de tareas y subtareas, organizándolas en orden de complejidad, tiempo y dependencia de otras tareas.

### Kanban

La metodología *Agile Kanban* es un enfoque para la gestión de proyectos que se basa en los principios del método *Kanban* y se adapta a los principios ágiles de desarrollo de software. El objetivo de *Kanban* es mejorar la eficiencia y la capacidad de respuesta al cambio mediante la visualización del flujo de trabajo y el control de las tareas en curso. En el contexto de *Agile*, *Kanban* se utiliza para limitar el trabajo en curso y enfocarse en entregar valor al cliente de manera continua [11].

### Trello

Trello es una herramienta de gestión de proyectos basada en la metodología *Kanban*. Permite a los usuarios crear tableros virtuales con listas de tareas y tarjetas, las cuales pueden ser movidas a medida que avanzan en el proceso. Esta herramienta es muy visual y fácil de usar, lo que la hace popular entre equipos de trabajo que buscan una manera simple y eficiente de organizar y rastrear el progreso de sus tareas. Trello también cuenta con funciones de colaboración, como la posibilidad de asignar tareas a miembros del equipo, agregar comentarios y adjuntar archivos [12].

## Tablero Kanban de Trello.

En el desarrollo de este proyecto, se ha adoptado la metodología *Kanban* respaldada por un tablero en la plataforma Trello.

Aunque el proyecto sea gestionado por un único individuo, la aplicación de esta metodología proporciona beneficios sustanciales en términos de organización y gestión de tareas.

La metodología *Kanban* se fundamenta en la visualización de las tareas en un tablero dividido en columnas, cada una representando una etapa del flujo de trabajo.

En este caso, las columnas empleadas son: *Backlog, To Do, In Progress* y *Done*.

En el tablero *Kanban*, las tareas se distribuyen en distintas columnas según su estado. Este enfoque permite visualizar de manera clara el estado de cada tarea y proporciona una visión global del progreso del proyecto. Facilita la priorización, gestión efectiva del flujo de trabajo y la identificación temprana de bloqueos o retrasos.

En el caso de proyectos individuales, la metodología *Kanban* se adapta de manera flexible. Para su implementación, se ha utilizado la plataforma Trello, una herramienta digital que brinda un mayor control y organización.

En resumen, aun siendo un proyecto individual, el uso de *Kanban* y un tablero visual en Trello ofrece ventajas en términos de organización, priorización y seguimiento del trabajo. La visualización clara de tareas y la gestión del flujo de trabajo contribuyen a mantener un enfoque efectivo y alcanzar objetivos de manera eficiente.

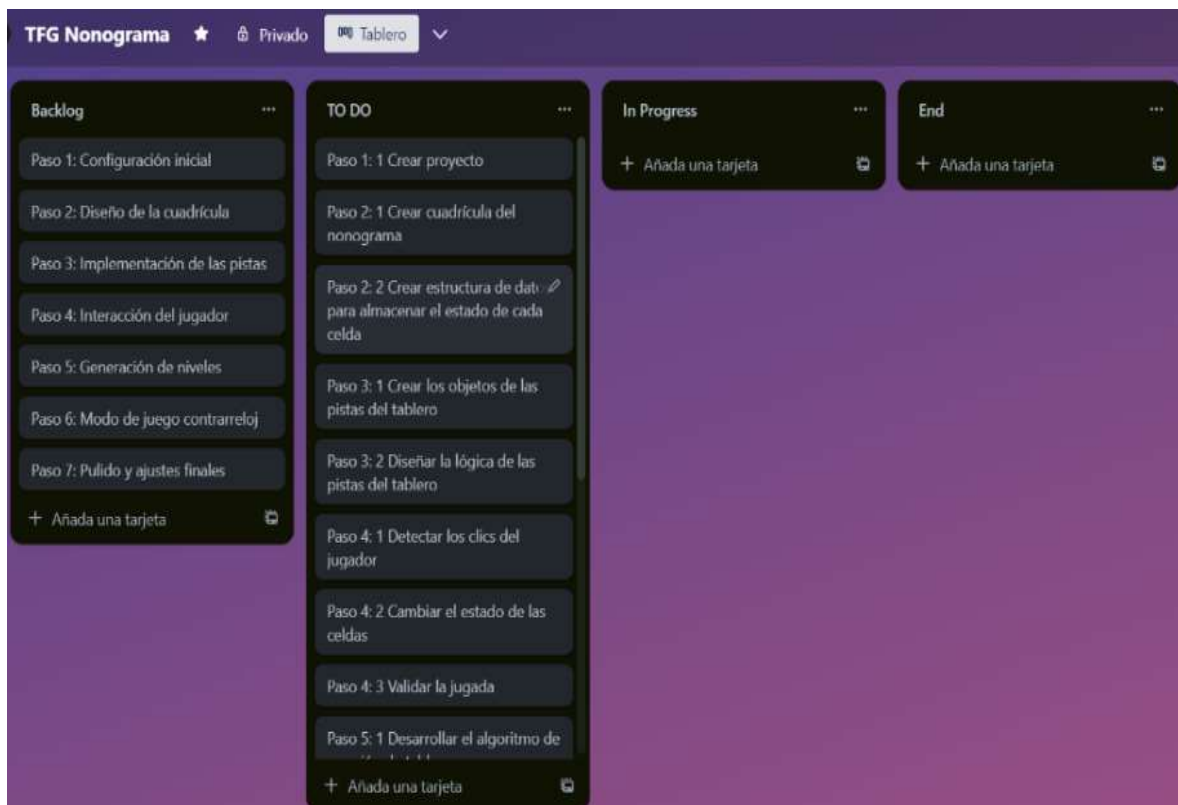


Imagen 2. Tablero Trello (1)

### Descripción de las Columnas:

- **Backlog:** En esta columna se encuentran las tarjetas que representan las tareas generales del proyecto. Estas tareas generales contienen subtareas y requisitos a completar que compondrán las tarjetas del To Do.
- **To Do:** Inicialmente, esta columna alberga las tarjetas que representan las subtareas del *Backlog*, ordenadas según el plan de desarrollo.
- **In Progress:** Las tarjetas asociadas a tareas en desarrollo se mueven a esta columna con forme se van iniciando. Aquí se lleva a cabo el trabajo activo y se registra el progreso.
- **Done:** Una vez completadas, las tarjetas se trasladan a esta columna final, proporcionando una visión clara de las tareas finalizadas.

### Avance del Proyecto:

Las imágenes 3 y 4 muestran cómo el proyecto ha ido avanzado a medida que las tarjetas se han movido a través de las distintas fases del tablero. El uso de este sistema de gestión ha sido fundamental para mantener el control y organizar eficientemente el desarrollo del proyecto.



Imagen 3. Tablero Trello (2)

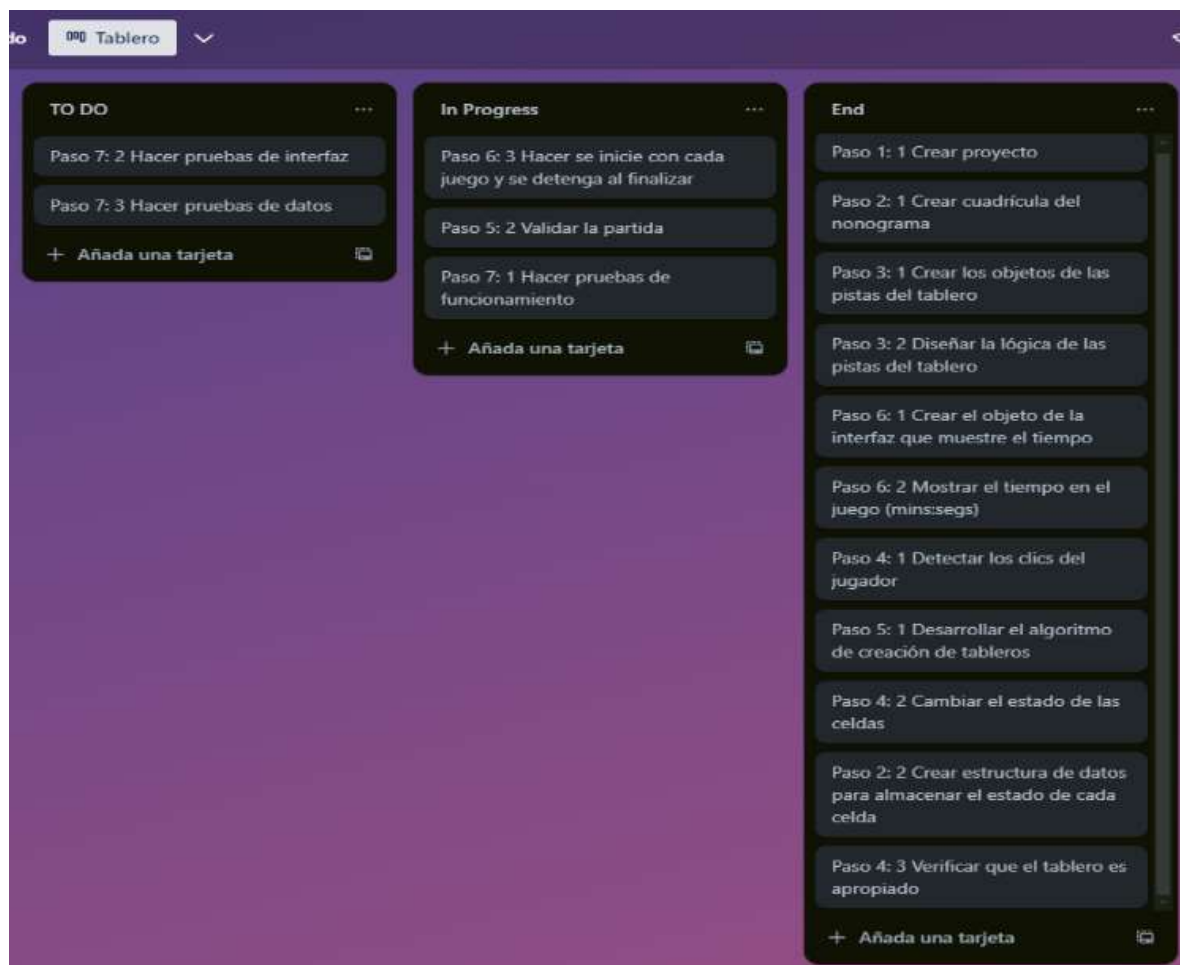


Imagen 4. Tablero Trello (3)

## Diagrama de Gantt

Un diagrama de Gantt se utiliza en la gestión de proyectos se usa para representar las actividades programadas desde el comienzo hasta la conclusión de un proyecto. [10]

El diagrama de Gantt es similar a un *timeline* donde se muestran las tareas y sus dependencias. Cada tarea se etiqueta con el responsable de la misma, la duración aproximada, y la estimación de la duración total del proyecto. Esta representación facilita el seguimiento del desarrollo de una forma rápida y visual [10] .

En la Imagen 5, se muestra el diagrama de Gantt simplificado del desarrollo de la aplicación.

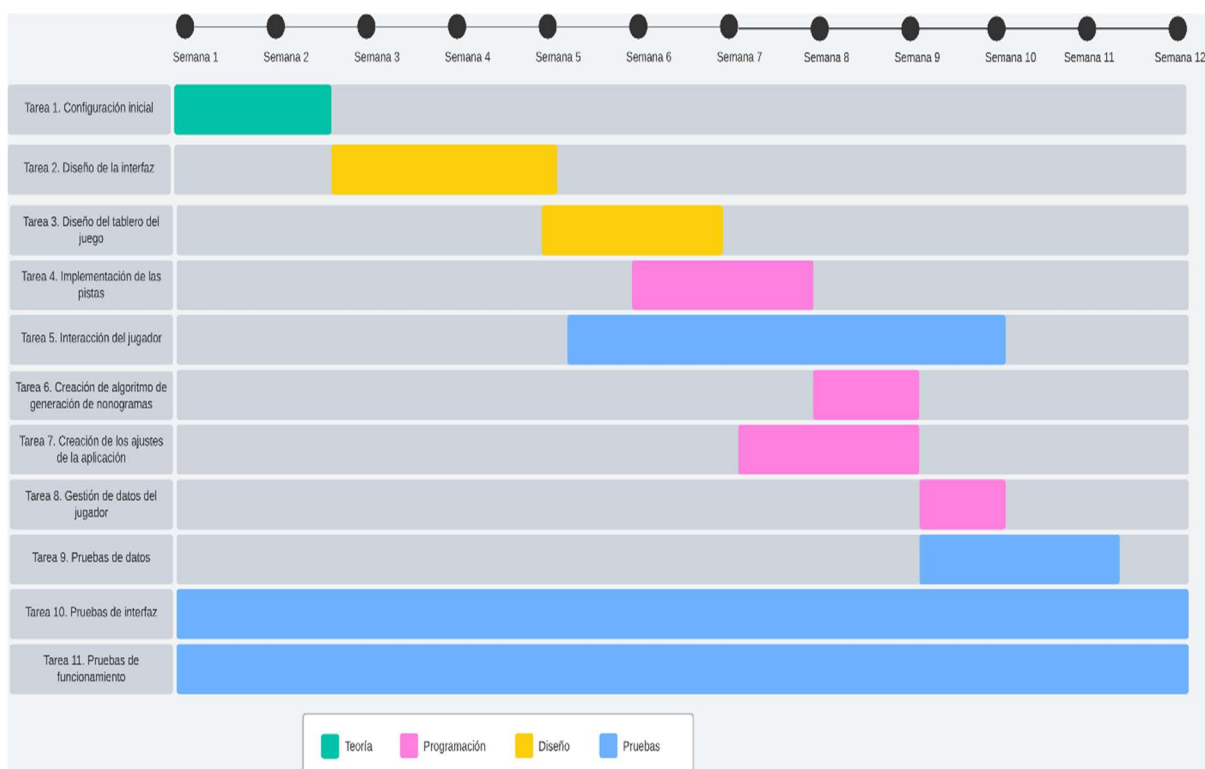


Imagen 5. Diagrama de Gantt

Lo primero que se aprecia, es el peso de las pruebas desde el inicio hasta el final del proyecto. Al hacerse de forma continua, se prevén problemas graves y difíciles de resolver al final del proyecto, y se asegura un mejor desarrollo más limpio y eficiente.

Puesto que el desarrollo en Unity tiene un alto componente de programación visual a la hora del diseño de interfaces, se puede observar cómo esta parte pesa más en el inicio del

desarrollo, cediéndole el peso a la parte de la programación de la lógica y de los algoritmos en una segunda fase del desarrollo.

## 1.8 Estructura de la memoria

La memoria se estructura en seis capítulos, agrupados en tres secciones principales, junto con un glosario, un anexo y una bibliografía final.

### Introducción y Marco Teórico (Capítulos I y II)

- **Introducción al problema y planteamiento inicial:** Se describe el problema a abordar y se justifica su relevancia.
- **Investigaciones pertinentes:** Se exponen las investigaciones previas que han sido útiles para el desarrollo de la aplicación.
- **Metodología y planificación:** Se define la metodología empleada para el desarrollo del proyecto y se establecen los principios teóricos que lo sustentan.

### Diseño y Desarrollo de la Aplicación (Teórico y Práctico) (Capítulos III y IV)

- **Diseño de la aplicación:** Se detalla el proceso de diseño, incluyendo decisiones sobre la interfaz de usuario y la experiencia del usuario.
- **Desarrollo y codificación:** Se documenta el proceso de desarrollo y codificación de la aplicación, abarcando tanto los aspectos técnicos como los gráficos.

### Validación y Conclusiones. (Capítulos V, VII)

- **Validación de objetivos y requisitos y resultados:** Se realizan pruebas y validaciones para verificar el cumplimiento de los objetivos y requisitos planteados inicialmente. Se indican las correcciones realizadas y se presentan los resultados finales.
- **Conclusiones y propuestas futuras:** Se extraen conclusiones sobre el proyecto y se realiza una valoración del aprendizaje obtenido. Además, se plantean opciones para futuras extensiones o mejoras del desarrollo.

### Secciones Adicionales

- **Glosario:** Se incluye un glosario de términos técnicos y específicos utilizados a lo largo del documento para facilitar su comprensión.

- **Anexo:** El anexo contiene información complementaria relevante para el desarrollo del proyecto, como diagramas adicionales, enlace de descarga de la aplicación y una guía de usuario.
- **Bibliografía:** Se proporciona una lista exhaustiva de las fuentes y referencias bibliográficas consultadas y citadas en el desarrollo del proyecto.

Esta estructura permite un análisis exhaustivo y ordenado del proyecto, facilitando la comprensión de todas las etapas del desarrollo y asegurando una evaluación integral de los logros alcanzados.



# Capítulo II. Marco Teórico

## 2.1 Análisis histórico

Los Estados miembros de la Unión Europea se enfrentan al desafío de mejorar las habilidades digitales de sus ciudadanos, reconociendo la necesidad de actualizar los sistemas educativos para abordar esta cuestión. Para abordar este desafío, la Comisión Europea ha propuesto una Recomendación del Consejo a la cual se ha llamado *Digital Education Action Plan – Action 10* [13].

Este plan de acción (*Action 10*) busca promover el desarrollo de las competencias digitales en todas las etapas educativas, empezando por las edades más tempranas. Esta iniciativa incluye medidas para invertir en la formación profesional de los docentes y fomentar el intercambio de mejores prácticas en educación informática en todos los niveles educativos.

Dentro de este marco, se han llevado a cabo diversas actividades clave para avanzar en la mejora de las habilidades digitales en la educación. Por ejemplo:

- El informe “*Reviewing Computational Thinking in Compulsory Education*” publicado en marzo de 2022 por el Centro Común de Investigación (JRC) [14], el cual ofrece una revisión exhaustiva del PC en la educación obligatoria.
- Asimismo, el Estudio de Eurydice sobre “*Informatics in School Education*”, publicado en septiembre de 2022, proporciona una visión detallada de la informática en la educación escolar [15].
- Además, en el primer trimestre de 2023 se presentó una propuesta de Recomendación del Consejo sobre la mejora de la provisión de capacidades digitales en la educación y la formación [16].

Estas acciones representan un compromiso conjunto para abordar las necesidades de habilidades digitales en la población europea y mejorar la calidad de la educación en todos los Estados miembros de la Unión Europea, con el objetivo de preparar a los ciudadanos para el mundo digital en constante evolución.

## 2.2 Pensamiento Computacional en Primaria

Como se ha mencionado en apartados anteriores, la normativa española establece que debe desarrollarse la competencia digital y el PC en todas las etapas educativas.

Es decir, se espera que los estudiantes adquieran habilidades en programación y algoritmia, así como en el uso crítico y ético de las tecnologías y la comunicación. Esto se incorporará de forma transversal en diferentes áreas de estudio, como matemáticas, ciencias, lengua y educación física, entre otras.

Además, también se espera que éstos sean capaces de aplicar dichas habilidades en otros contextos y situaciones, fomentando su capacidad de resolución de problemas, el pensamiento creativo y el trabajo colaborativo.

## 2.3 Proyectos relacionados con el PC

El PC es una competencia fundamental en la educación actual, permitiendo a los estudiantes abordar problemas de manera sistemática y lógica, al igual que un profesional en informática. Diversos proyectos y programas han sido implementados globalmente con el objetivo de integrar el PC en el currículo educativo, promoviendo habilidades que son esenciales en la era digital.

En este apartado se examinan algunos de los proyectos más relevantes relacionados con el PC, proporcionando una visión sobre cómo diferentes iniciativas están contribuyendo al desarrollo de estas habilidades en estudiantes de diversas edades.

### Proyecto EPCIA

El proyecto EPCIA (Escuela de Pensamiento Computacional e Inteligencia Artificial) era una iniciativa del INTEF (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado) en colaboración con las Comunidades y Ciudades Autónomas de España, que se realizó anualmente desde 2018 hasta 2022.

Su objetivo era explorar cómo integrar el PC y la Inteligencia Artificial (en adelante IA) en la educación. Esto se logró mediante la creación de recursos educativos, prácticas en las aulas e

investigación. La formación era modular y permitió a los participantes personalizar su aprendizaje.

Cada edición constó de tres fases: Formación, implementación en el aula e investigación. La última (2021-2022), se centró en cinco áreas principales: PC, programación, lenguaje de programación Python, IA, Robótica.

En las ediciones anteriores, se siguieron estructuras similares, ofreciendo formación online y presencial, implementando actividades en las aulas y realizando investigaciones sobre el impacto del proyecto. Se abordaron diferentes niveles educativos, con el fin de promover el desarrollo de habilidades digitales esenciales para el siglo XXI.

### Ediciones:

Las propuesta y temáticas desarrolladas en cada curso escolar han ido variando e incluyendo distintas temáticas [6].

- **Curso 2018-2019:** En su primer año, el proyecto se centró en mejorar el PC y la programación en diferentes niveles educativos. Los maestros recibieron formación presencial y en línea para incorporar estas habilidades en sus clases, sentando las bases para futuras ediciones.
- **Curso 2019-2020:** En su segunda edición, centraron el proyecto en ofrecer recursos educativos, formación e investigación sobre el PC, y la IA. Aunque más de mil maestros se inscribieron, la implementación en las aulas se vio afectada por la pandemia, lo que dificultó completar la investigación.
- **Curso 2020-2021:** La tercera edición se llevó a cabo, a pesar de los problemas que el curso anterior ocasionó la pandemia. Casi 900 maestros recibieron formación y alrededor de 7500 estudiantes se beneficiaron de las actividades en el aula. También se realizó una investigación para entender mejor cómo influye en el aprendizaje.
- **Curso 2021-2022:** En esta última edición, el proyecto EPCIA continuó de forma similar a los cursos anteriores. Los maestros recibieron formación online y diseñaron propuestas didácticas para enseñar el PC, y la IA. Después, se pusieron en práctica estas ideas y se colaboró en la investigación sobre el impacto en los alumnos. Además, se amplió la oferta formativa resumiéndose en cinco áreas: PC Desconectado, Programación por bloques, Lenguajes de Programación: Python, Inteligencia Artificial, y Robótica.

## CodeWeek

Anualmente, se celebra el evento CodeWeek. La primera edición se celebró del 11 al 17 de octubre de 2013: “*Europe CodeWeek, la Semana Europea de la Programación promovida por la Comisión Europea*”. En este evento europeo se celebraron distintos tipos de eventos para dar visibilidad a la programación, y eliminar mitos relacionados con la misma, además de promover su aprendizaje desde edades tempranas. Su última edición fue la del 07 al 22 de octubre de 2023 [7].



Imagen 6. Code Week

## Enfoque educativo STEAM

El significado de *STEAM* proviene de un acrónimo, el cual se refiere a las áreas de conocimiento en las que suelen trabajar los científicos y los ingenieros “*Science, Technology, Engineering, Arts and Mathematics*” (Ciencia, Tecnología, Ingeniería, Artes y Matemáticas) [8].

Desde sus orígenes en las últimas décadas del siglo XX, el enfoque educativo *STEAM* pretende aunar ciencia, tecnología, ingeniería y matemáticas. El concepto es acuñado y promovido por la Comisión Europea en la primera década de este siglo, para impulsar iniciativas que contrarresten el descenso de jóvenes con vocación científico-tecnológica [9].



Imagen 7. STEAM

## 2.4 Fundamentos teóricos

En este apartado se exploran los fundamentos teóricos que sustentan el desarrollo de la aplicación, haciendo énfasis en la didáctica y su aplicación en el PC en la educación primaria. Se analiza cómo estos conceptos teóricos se integran y aplican en la práctica educativa.

### Didáctica

La didáctica es una disciplina que se enfoca en el estudio del proceso de enseñanza-aprendizaje, y actualmente, la mayoría de los autores coinciden en que su objeto de estudio es el proceso mismo. Desde una perspectiva formal, la didáctica busca seleccionar métodos y estrategias eficaces para llevar a cabo el proceso de enseñanza-aprendizaje de manera efectiva. El objetivo principal es mejorar la calidad de la enseñanza y facilitar el aprendizaje de los estudiantes.

La incorporación del PC en la educación primaria es cada vez más relevante debido al creciente papel de la tecnología en la sociedad. La didáctica desempeña un papel crucial en la implementación de esta enseñanza, ya que permite diseñar estrategias eficaces para la enseñanza del PC. Además, la evaluación del PC en la educación primaria es esencial para medir el progreso y el nivel de comprensión en este tema.

### Piaget y Vygotsky

Las enseñanzas de Piaget y Vygotsky subrayan la importancia de la construcción activa del conocimiento en la educación. Estas teorías se han integrado en diversas metodologías pedagógicas contemporáneas, incluyendo el análisis del discurso, la alfabetización, la estructuración, la práctica guiada, el diálogo socrático, el lenguaje integral, la evaluación continua, algunas formas de aprendizaje colaborativo y cooperativo, comunidades de aprendizaje y estrategias enfocadas en el desarrollo de habilidades específicas.

Es fundamental comprender y utilizar estas estrategias para lograr una práctica educativa reflexiva y trabajar en colaboración con estudiantes y profesionales de la enseñanza para alcanzar una educación de excelencia y *pertinencia*[19].

## Evaluación de Pensamiento Computacional

Es importante mencionar que existen distintos test de PC que han sido desarrollados y validados por diferentes investigadores y equipos de investigación en todo el mundo. Estos test suelen evaluar habilidades como la programación, la capacidad de algoritmia, el pensamiento lógico, la resolución de problema, etc. Sin embargo, cada test tiene sus propias metodologías y escalas de medición. Es importante mencionar que existen diferentes metodologías de evaluación y es importante que sean validadas y adaptadas para cada contexto educativo y cultural.

### Computational Thinking test (CTt)

En 2015 Román González, Pérez González, & Jiménez Fernández presentaron un test de PC con el objetivo de evaluar el nivel de aptitud desarrollo del del mismo, instrumento dirigido para estudiantes que cursan la escuela secundaria [20]. Esta herramienta de evaluación ha sido validada psicométricamente en los últimos años. Comprende 28 tareas de opción múltiple diseñadas para abordar siete conceptos de PC. Fue desarrollado para niños de 10 a 16 años [18].

El *CTt* se recomienda especialmente para evaluar a los estudiantes en los últimos grados de Educación Primaria, ya que permite una mejor diferenciación en estos estudiantes, además de tener una capacidad moderada para discriminar todo el rango de habilidades relacionadas con el PC.

### Beginners Computational Thinking test (BCTt)

Esta herramienta propone una evaluación similar a *CTt*, pero orientada a alumnos de edades más tempranas, con tareas más simples y una apariencia más amigable. El objetivo principal del desarrollo de este test fue identificar las relaciones cognitivas del PC mediante la evaluación de su relación con habilidades específicas en niños de Educación Primaria.

Los expertos han considerado el diseño del *BCTt* como adecuado para la evaluación del PC en alumnos de Educación Primaria, especialmente para alumnos de segundo ciclo con dificultades de aprendizaje, y de primer ciclo de Educación Primaria, así como de Educación Infantil [19].

En “Enseñanza, evaluación y análisis de habilidades de PC en etapas tempranas” [4], se concluye que el *BCTt* puede ser utilizado en alumnos de Educación Primaria, particularmente en los primeros cursos (5 a 7 años). Por lo tanto, el *BCTt* puede considerarse una extensión fiable del *CTt* de Román et al. para estudiantes más jóvenes [4].

## Competent Computational Thinking Test (cCTt)

En 2021, se desarrolló y validó el Competent Computational Thinking Test (cCTt) para evaluar el PC en estudiantes de primaria superior (7-9 años).

Esta prueba, basada en el BCTt, incluye 25 preguntas de opción múltiple con dificultad progresiva. Los conceptos evaluados abarcan secuencias, bucles, condicionales y declaraciones *while*.

La validación del cCTt se realizó en dos fases:

- **Primera fase:** 37 expertos evaluaron la validez del contenido, constructo y apariencia del test, proporcionando recomendaciones para su mejora.
- **Segunda fase:** se administró la prueba a 1519 estudiantes en Suiza, demostrando propiedades psicométricas adecuadas como fiabilidad, dificultad y discriminación.

Los análisis factoriales confirmatorios validaron la coherencia de los bloques de preguntas con los conceptos de CT pretendidos. Se evaluó la posibilidad de acortar el test para reducir el tiempo de administración sin comprometer su validez y fiabilidad.

El cCTt se presenta como una herramienta válida y fiable para medir competencias en pensamiento computacional en estudiantes de primaria superior, complementando otras formas de evaluación [20].

## 2.5 Casos de estudio

En este apartado, se presenta una investigación sobre el uso de videojuegos como herramientas educativas diseñadas para enseñar y desarrollar habilidades académicas en diversos contextos. A través de la selección de casos de estudio relevantes, se analiza cómo estos videojuegos se han integrado en el entorno educativo y los resultados obtenidos.

### Videojuegos para Aprender Jugando

La investigación se centra en analizar videojuegos conocidos que han sido desarrollados para que los usuarios aprendan mientras juegan. En este contexto, se han seleccionado varios videojuegos destinados al aprendizaje. Los videojuegos seleccionados para este análisis son:



Imagen 8.

**La Torre del Conocimiento:** Videojuego educativo destinado a niños de Primaria para reforzar Lengua, Matemáticas, Inglés, Sociales y Naturales de forma divertida.



Imagen 9.

**Discover Babylon:** Videojuego para PC adecuado para trabajar conceptos de Historia que traslada al jugador a la Antigua Mesopotamia.



Imagen 10.

**Simple Machines:** El objetivo principal es ayudar a un robot a reunir piezas para hacer funcionar una máquina de tipo simple de forma eficiente, con poleas y palancas. Con él los estudiantes aprenden conceptos relacionados con los principios de la física.

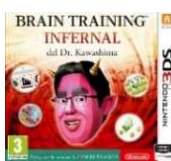


Imagen 11.

**Brain Training Infernal del Dr. Kawashima:** Videojuego clásico del cálculo mental. Además, se puede convertir en una herramienta de gran ayuda para alumnos con trastorno de déficit de atención e hiperactividad (TDAH), al tratarse de un videojuego social y cooperativo, cuyos logros se recompensan, se produce una motivación extra.



Imagen 12.

**Flippy's Tesla: Inventemos el futuro:** Basado en la realidad virtual, tiene como principal objetivo estimular la curiosidad de los jugadores para que se interesen por la ciencia.





Imagen 13.

**Civilization:** Es considerado uno de los mejores videojuegos educativos por su alto contenido histórico y cultural sobre el desarrollo de la humanidad.



Imagen 14.

**Monument Valley:** Donde el jugador debe manipular arquitecturas y geometrías imposibles y guiar a una princesa silenciosa por un mundo de gran belleza.



Imagen 15.

**No man's sky:** Aventura de ciencia ficción en primera persona, donde los jugadores son introducidos en un universo desconocido explorando planetas y estrellas.



Imagen 16.

**Animal Crossing:** Se trata de una aventura donde el jugador puede personalizar su casa, su personaje, decorar el paisaje e ir creando poco a poco su propia isla paradisíaca, además debe hacer misiones y fomentar su relación con los vecinos.



Imagen 17.

**Zoo Tycoon 2:** Permite profundizar en el mundo animal y aprender a gestionar un zoo. Las instalaciones las diseñan y construyen los alumnos desarrollando su creatividad. Uno de sus puntos fuertes es el aprendizaje de las diferencias entre especies, y sus particularidades alimenticias y de desarrollo.



Imagen 18.

**Super Lucky's Tale:** Juego de plataformas que desarrolla la capacidad artística y permite introducir en el mundo de las matemáticas a los más pequeños.

A continuación, se muestra una tabla con los datos analizados de estos juegos:

	Objetivo Educativo	Habilidades Desarrolladas	Público Objetivo
<b>La Torre del Conocimiento</b>	Historia y Cultura.	Conocimiento histórico y cultural.	Primaria y Secundaria.

<b>Discover Babylon</b>	Historia y Cultura de Mesopotamia.	Comprensión histórica.	Primaria y Secundaria.
<b>Simple Machines</b>	Ciencias Físicas.	Principios de la física.	Primaria.
<b>Brain Training Infernal</b>	Agilidad Mental.	Cálculo mental, memoria.	Todas las edades.
<b>Flippy's Tesla</b>	Ciencias y Tecnología.	Principios de la electricidad.	Primaria.
<b>Civilization</b>	Historia y Estrategia.	Planificación estratégica, historia.	Secundaria y superior.
<b>Monument Valley</b>	Razonamiento Espacial.	Resolución de problemas, percepción espacial.	Todas las edades.
<b>No Man's Sky</b>	Exploración y Ciencia.	Curiosidad científica, exploración.	Secundaria y superior.
<b>Animal Crossing</b>	Economía y Gestión de Recursos.	Planificación, gestión de recursos.	Todas las edades.
<b>Zoo Tycoon 2</b>	Biología y Gestión Ambiental.	Gestión de recursos, conocimiento zoológico.	Primaria y Secundaria.
<b>Super Lucky's Tale</b>	Razonamiento y Habilidades Motoras.	Resolución de problemas, coordinación motora.	Primaria.

**Tabla 2. Casos de estudio: Videojuegos para Aprender Jugando**

Los casos de estudio seleccionados demuestran que los videojuegos educativos son herramientas eficaces para reforzar habilidades académicas de manera divertida. Además, muestran aplicaciones en diversas asignaturas y también destacan su utilidad para desarrollar habilidades cognitivas como el cálculo mental y la curiosidad científica.

Estos ejemplos confirman que los videojuegos educativos son valiosos para mejorar el aprendizaje y fomentar la motivación estudiantil.

## Juegos para el aprendizaje

En este apartado, se presenta una investigación sobre el uso de videojuegos como herramientas educativas diseñadas para enseñar y desarrollar habilidades académicas en diversos contextos. La incorporación de elementos lúdicos en el proceso de enseñanza-aprendizaje ha demostrado ser efectiva para captar el interés de los estudiantes y mejorar su compromiso y retención de información. A través de la selección de casos de estudio relevantes, se analiza cómo estos videojuegos se integran en el entorno educativo, sus metodologías de implementación y los resultados obtenidos en términos de desarrollo de habilidades y competencias académicas. La investigación se centra en identificar buenas prácticas y lecciones aprendidas que pueden ser aplicadas en el desarrollo de nuevas herramientas educativas basadas en videojuegos.

En este contexto, se han seleccionado varios videojuegos que destacan por su enfoque pedagógico y su capacidad para fomentar el aprendizaje activo y significativo entre los estudiantes. Estos casos de estudio proporcionan una visión detallada de cómo los videojuegos educativos pueden ser diseñados y utilizados para abordar diferentes áreas del conocimiento y habilidades específicas.

Los videojuegos seleccionados para este análisis incluyen:

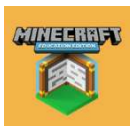


Imagen 19.

**Minecraft: Education Edition:** Juego de construcción y aventura adaptado para el uso en el aula, ofreciendo recursos educativos que permiten explorar conceptos de ciencias, matemáticas, historia y más a través de experiencias de aprendizaje inmersivas y colaborativas.



Imagen 20.

**Kahoot!:** Plataforma de aprendizaje basada en juegos que permite a los educadores crear cuestionarios interactivos y juegos de trivia para evaluar y reforzar el conocimiento de los estudiantes de una manera divertida y competitiva.



Imagen 21.

**DragonBox:** Una serie de juegos diseñados para enseñar matemáticas a los niños de manera intuitiva y entretenida, ayudando a los estudiantes a desarrollar habilidades matemáticas fundamentales a través de la resolución de problemas y la manipulación de objetos virtuales.



Imagen 22.

**CodeCombat:** Juego que enseña a los estudiantes a programar en varios lenguajes de programación mientras participan en misiones de fantasía y resuelven desafíos de codificación en tiempo real.



Imagen 23.

**Quizizz:** Plataforma de aprendizaje gamificada que permite a los educadores crear cuestionarios y juegos interactivos para evaluar y mejorar el conocimiento de los estudiantes. Los estudiantes pueden participar en actividades en tiempo real o completarlas a su propio ritmo.

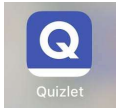


Imagen 24.

**Quizlet:** Herramienta educativa que utiliza tarjetas para ayudar a los estudiantes a aprender y repasar información. Ofrece varios modos de estudio y juegos que hacen el aprendizaje más interactivo y atractivo.



Classcraft

Imagen 25.

**Classcraft:** Juego de rol educativo que transforma el aula en una aventura de fantasía, incentivando a los estudiantes a trabajar juntos, participar en clase y completar sus tareas a cambio de recompensas y poderes en el juego.



Imagen 26.

**Duolingo:** Aplicación de aprendizaje de idiomas donde estudiar una nueva lengua es un juego. Los usuarios avanzan a través de niveles, ganan puntos y compiten con amigos, haciendo que el aprendizaje sea divertido y motivador.



Imagen 27.

**Lightbot:** Juego de rompecabezas que enseña principios básicos de programación y lógica a través de un robot. Los jugadores deben utilizar comandos de programación para guiarlo a través de desafiantes niveles.

A continuación, se muestra una tabla con los datos analizados de estos juegos:

	Objetivo Educativo	Habilidades Desarrolladas	Público Objetivo
<b>Minecraft: Education Edition</b>	Multidisciplinar.	Colaboración, creatividad, pensamiento crítico.	Primaria y Secundaria.
<b>Kahoot!</b>	Evaluación de Conocimientos.	Retención de información, competitividad.	Todas las edades.

<b>DragonBox</b>	Matemáticas.	Conceptos matemáticos fundamentales.	Primaria.
<b>CodeCombat</b>	Programación.	Resolución de problemas, lógica de programación.	Secundaria y superior.
<b>Quizizz</b>	Evaluación de Conocimientos.	Retención de información, autoevaluación.	Todas las edades.
<b>Quizlet</b>	Repaso de Información.	Memoria, retención de información.	Todas las edades.
<b>Classcraft</b>	Gestión del Aula.	Trabajo en equipo, responsabilidad.	Primaria y Secundaria.
<b>Duolingo</b>	Aprendizaje de Idiomas.	Comprensión y producción de idiomas.	Todas las edades.
<b>Lightbot</b>	Programación y Lógica.	Principios de programación, pensamiento lógico.	Primaria.

**Tabla 3. Casos de estudio: Juegos para el aprendizaje**

Estos ejemplos ilustran cómo los videojuegos pueden ser herramientas poderosas para el aprendizaje, proporcionando experiencias educativas que son tanto efectivas como agradables para los estudiantes. Además, se ha observado su efectividad en ayudar a alumnos con discapacidades del aprendizaje, como el TDAH y el TDA.

# Capítulo III. Diseño de la aplicación

En este capítulo se describe y explica el proceso de diseño de la aplicación, incluyendo las decisiones tomadas en el apartado visual y funcional.

## 3.1 Introducción

Se ha decidido realizar un juego de *Nonograma* en 2D para entrenar y evaluar las habilidades estratégicas y de fragmentación de problemas asociadas al PC, además de entretener e incentivar a los alumnos.

La aplicación será un juego de puzzle en 2D basado en la interacción del usuario con una cuadrícula y las pistas asociadas para resolver un tablero. Incluirá tres modos de juego en base a la dificultad y un cuarto modo en formato contrarreloj.

## 3.2 Game Design Document (GDD)

El *Game Design Document* (GDD) es un documento exhaustivo que describe todos los aspectos del diseño y desarrollo de un juego. En este apartado, se presenta un extracto del GDD correspondiente a la aplicación del juego *Nonograma*, destacando los puntos más relevantes para este proyecto.

### Resumen Ejecutivo

La aplicación del juego *Nonograma* es una aplicación de rompecabezas de lógica diseñada para desafiar y entretener a los niños en edad escolar, al tiempo que fomenta su pensamiento lógico y agudeza visual. Basado en el popular concepto de "pintar por números", el *Nonograma* ofrece una experiencia educativa y divertida en dispositivos móviles y tabletas. Dirigido específicamente a alumnos del primer ciclo de primaria, este juego proporciona tiempo de diversión mientras se aprende de manera intuitiva.

## Concepto del Juego

*Nonograma* presenta puzzles de 5x5 que desafían a los niños a completar imágenes simples resolviendo patrones numéricos. Los usuarios seleccionan casillas para llenar o vaciar, siguiendo las pistas numéricas proporcionadas en los bordes del tablero.

El objetivo es desarrollar habilidades de lógica y agudeza visual (ambas relacionadas directamente con el PC) mientras resuelven puzzles y se divierten.

## Jugabilidad

En el juego de *Nonograma*, el usuario debe interactuar con un tablero en el que hay una cuadrícula en 2 dimensiones y números entorno a ella.

Cada una de las celdas de la cuadrícula debe ser coloreada de negro, gris o blanco, y el usuario debe rellenar la cuadrícula correctamente según las cifras que aparecen sobre cada columna y antes de cada fila, también llamadas pistas, que indican el número de bloques negros contiguos que hay en dicha fila/columna.

Para jugar, se deberá usar el dedo o un lápiz inteligente para hacer clic en el tablero y cambiar el color de la celda. Cada clic sobre una celda alterna el color de la casilla entre negro, gris (misma función que el blanco) y blanco.

Se podrá seleccionar entre dificultad fácil, media o difícil, además se incluye una opción de contrarreloj en dificultad media el cual finaliza cuando el usuario quiera verificarlo, o al acabar el tiempo.

## Mecánicas de Juego

Los usuarios seleccionan casillas haciendo clic, alternando entre llenas (negras), vacías (grises) y no seleccionadas (blancas). Los puzzles se verifican al hacer clic en el botón de validar, permitiendo hasta dos errores para ganar la partida.

La cuadrícula será de diferentes tamaños en base a la dificultad elegida.

El usuario puede hacer clic en una casilla para marcarla como negro, gris o blanco.

## Interfaz de Usuario

La interfaz de usuario presenta un diseño intuitivo y colorido, con gráficos atractivos y amigables para los niños. El estilo se centra en ilustraciones simples y llamativas que facilitan la comprensión y la interacción.

## Diseño de Puzles

Los puzles se generan de forma aleatoria utilizando un algoritmo que asigna valores booleanos a una matriz de 5x5, de 7x7 o de 10x10, según el modo de juego. Se garantiza una distribución equilibrada de casillas llenas y vacías para evitar soluciones demasiado simples o complejas.

## Requisitos Técnicos

Los requisitos técnicos son fundamentales para definir las características y las limitaciones que debe cumplir una aplicación. Estos requisitos se dividen en funcionales y no funcionales, donde los requisitos funcionales especifican lo que debe hacer el sistema y los no funcionales definen cómo debe hacerlo, en términos de rendimiento, usabilidad, fiabilidad y otros atributos de calidad.

Los requisitos concretos se detallan en el capítulo de desarrollo de la aplicación.

## Plan de Pruebas

El plan de pruebas se llevará a cabo para garantizar la calidad y funcionalidad de la aplicación del juego de *Nonograma*. Se realizarán pruebas exhaustivas en diferentes áreas para asegurar que el juego cumpla con los estándares de rendimiento esperados y proporcione una experiencia óptima para los usuarios. Las pruebas concretas realizadas se detallan en el capítulo de desarrollo de la aplicación.

El plan de pruebas se ejecutará de manera sistemática y se documentarán los resultados obtenidos en cada fase. Se realizarán ajustes y correcciones según sea necesario para garantizar que el juego cumpla con los estándares de calidad establecidos antes de su lanzamiento.



## Plan de Lanzamiento y Comercialización

Esta aplicación de juego de *Nonograma* no tiene previsto un lanzamiento comercial. En su lugar, se planea ofrecer la aplicación de forma gratuita a las escuelas como una herramienta educativa adicional para ayudar a los niños a aprender y divertirse al mismo tiempo.

### 3.3 Diseño de artes

El diseño de la experiencia de usuario (UX) y la interfaz de usuario (UI) son componentes cruciales en el desarrollo de aplicaciones, especialmente aquellas dirigidas a usuarios jóvenes. Un diseño visual atractivo y funcional no solo mejora la usabilidad de la aplicación, sino que también contribuye a una experiencia de aprendizaje más efectiva y agradable.

Aunque el proyecto no se centra en el diseño gráfico, se reconoce la relevancia de la experiencia visual para que los usuarios puedan aprovechar plenamente el potencial de la aplicación y desarrollar sus *aptitudes* de manera efectiva. En el caso del desarrollo de una aplicación dirigida a niños de los primeros ciclos de primaria, el apartado visual es importante.

Se ha evaluado y seleccionado un estilo de *sprites* adecuado para los niños, manteniendo una interfaz atractiva, visual e intuitiva.

### Interfaz de usuario y experiencia de usuario (UX)

Se considera esencial dedicar atención a la parte visual para garantizar una experiencia satisfactoria y facilitar el desarrollo de las habilidades de los niños a través de la aplicación. Al combinar y reestructurar los textos, se busca lograr una introducción coherente y fluida, destacando la importancia del aspecto visual en el desarrollo de este proyecto. Se resalta la selección de un estilo de *sprites* adecuado y la disposición de la interfaz visual e intuitiva.

Además, el aspecto visual se considera muy relevante para la experiencia de la aplicación y, por tanto, en el desarrollo de habilidades de los niños.

#### Leyes de UX

El diseño centrado en el usuario (UX) se refiere al conjunto de interacciones y percepciones que un usuario experimenta al utilizar una interfaz digital, como una web o una aplicación.

Abarca desde las sensaciones que se transmiten al interactuar con la interfaz, hasta cómo se adapta a las necesidades y expectativas del usuario, pasando por cómo se le incita a realizar ciertas acciones.

La experiencia de usuario abarca todo el proceso, desde el momento en que se accede a la interfaz hasta que se completan los objetivos para los que ha sido diseñada, teniendo en cuenta aspectos como la facilidad de uso, el diseño visual, la velocidad de carga y la satisfacción general del usuario. Para ello el diseño UX se encarga de analizar la mente del usuario final al que se enfoca el diseño y aplica los principios del diseño [21] :

1. *Usa tanto el conocimiento en el mundo como el conocimiento en la cabeza.*
2. *Simplificar la estructura de tareas.*
3. *Hacer visibles las cosas: cerrar los abismos de ejecución y evaluación.*
4. *Obtenga las asignaciones correctas.*
5. *Explotar el poder de las limitaciones, tanto naturales como artificiales.*
6. *Diseño para el error.*
7. *Cuando todo lo demás falla, estandarizar.*

### Principios de usabilidad de Jacob Nielsen

En base a la definición de UX, algunos de los principios de usabilidad acuñados por Jacob Nielsen [22] se pueden aplicar a cualquier tipo de aplicación, incluidos los juegos:

1. [Control y libertad del usuario.](#) Si se elige una opción por error, se debe proporcionar una forma de volver atrás o salir.
2. [Consistencia y estándares.](#) Es importante definir reglas lógicas y adherirse a ellas de manera constante.
3. [Prevención de errores.](#) Se deben tomar medidas para evitar que el usuario cometa errores durante el uso de la aplicación.
4. [Reconocimiento antes que recuerdo.](#) Es importante mostrar de forma clara y concisa las acciones y las opciones disponibles para que el usuario no tenga que recordar información.
5. [Flexibilidad y eficiencia de uso.](#) Las aplicaciones deben adaptarse fácilmente a diferentes tipos de usuarios y permitir hacer las cosas de forma rápida y sencilla.
6. [Estética y diseño minimalista.](#) No se debe incluir información innecesaria, mostrando solo la información relevante para promover su visibilidad.

## 21 Leyes de UX de John Yablonski

El autor Jon Yablonski define 21 leyes para el diseño centrado en el usuario. A continuación, se enumeran algunas de ellas en las que se ha centrado el desarrollo [23]:

*1. Efecto estético - usabilidad. Los usuarios a menudo perciben un diseño estéticamente agradable como un diseño más utilizable.*

*2. Umbral de Doherty. La productividad se dispara cuando una computadora y sus usuarios interactúan a un ritmo (<400 ms) que garantiza que ninguno tenga que esperar al otro.*

*6. Ley de Jakob. Los usuarios pasan la mayor parte del tiempo en otros sitios. Por lo que prefieren que su sitio funcione de la misma forma que todos los demás que conocen.*

*7. Ley de la Región Común. Los elementos tienden a percibirse en grupos si comparten un área con un límite claramente definido.*

*9. Ley de Prägnanz. Las personas percibirán e interpretarán imágenes de la forma más simple posible, porque es la interpretación que requiere el menor esfuerzo cognitivo.*

*10. Ley de Similitud. El ojo humano tiende a percibir elementos similares en un diseño como una imagen, forma o grupo completo, incluso si esos elementos están separados.*

## Diseño de Interfaz de Usuario (UI)

El diseño de Interfaz de Usuario (también llamado “UI” de “user interface”) se refiere a la forma de organizar y diseñar los elementos visuales de la aplicación para que su uso sea sencillo para el usuario final. El objetivo del UI es obtener una aplicación que le dé al usuario una experiencia en su manejo intuitiva, eficiente y sencilla.

Esta parte engloba botones, imágenes, iconos, textos, etc. Además de la tipografía, las proporciones de los componentes y la disposición de los mismos en las diferentes vistas, pantallas o ventanas que compongan la aplicación.

## Tipografía

La legibilidad y la accesibilidad son aspectos cruciales en el diseño de interfaces visuales, incluyendo el desarrollo de juegos. Al crear *sprites* y texto para un juego, es importante considerar la facilidad de lectura y la adaptabilidad a la dislexia, especialmente cuando el público objetivo son niños pequeños en los primeros ciclos de Educación Primaria.

Se han evaluado cuatro tipografías gratuitas y compatibles con Photoshop.

### Open Dyslexic

Open Dyslexic [24] ha sido diseñada especialmente para personas con dislexia. Proporciona una mayor claridad entre las letras y ayuda a reducir la confusión entre caracteres similares.

Open-Dyslexic  
*Open-Dyslexic*  
**Open-Dyslexic**

Imagen 28 Open-Dyslexic Font

### Comic Neue

Comic Neue [25] es una fuente con un aspecto atractivo y amigable, similar a las letras dibujadas a mano. Es fácil de leer y puede ser una buena opción para niños pequeños.

Comic Neue  
**Comic Neue bold**  
Comic Neue Angular

Imagen 29 Comic Neue Font

### Sassoon Primary

Sassoon Primary [26] ha sido desarrollada específicamente para niños en edad preescolar y en los primeros ciclos de Educación Primaria. Tiene trazos claros y es fácil de leer.

Sassoon Primary Regular	Sassoon Primary Regular
<i>Sassoon Primary Italic</i>	Sassoon Primary Italic
<b>Sassoon Primary Me...</b>	Sassoon Primary Medium

Imagen 30 Sassoon Primary Font

## Jolly Lodger

Jolly Lodger [27] es una fuente con un aspecto lúdico y amigable. Tiene letras con trazos gruesos y claros, lo que la hace adecuada para niños pequeños.

**Jolly Lodger**

**Jolly Lodger**

**Jolly Lodger**

Imagen 31 Jolly Lodger Font

## Estudio de las fuentes

Se ha realizado un estudio con 10 voluntarios de entre 50 y 60 años dedicados a profesiones como la psicología, la pedagogía y la enseñanza a niños de entre 6 y 8 años.

Tras dar puntuaciones del 1 al 5 en las diferentes características de las tipografías presentadas, se ha elaborado una tabla con la media de sus respuestas.

FUENTES	LEGIBILIDAD	ESTILO	APARIENCIA	ADAPTABILIDAD	TOTAL
Open Dyslexic	3/5	3/5	3/5	4/5	13/20
Comic Neue	4/5	4/5	5/5	4/5	17/20
Sassoon Primary	4/5	4/5	4/5	4/5	16/20
Jolly Lodger	2/5	3/5	2/5	2/5	09/20

Tabla 4. Estudio de Fuentes.

Algunos de los comentarios de los encuestados sobre la tipografía de Comic Neue han sido:

- *“Tiene un aspecto amigable y atractivo para los niños”*
- *“Tiene trazos claros y un diseño similar a la escritura a mano por lo que los niños se familiarizarán más fácilmente con ella”*
- *“En base a mi experiencia creo que los niños la encontrarán fácil de leer”*

En resumen, Comic Neue es una tipografía gratuita y compatible con Photoshop, adecuada para niños en los primeros ciclos de Educación Primaria. Ofrece una apariencia atractiva y una legibilidad clara, adaptándose a las necesidades del público objetivo. Al utilizar Comic Neue en este proyecto, se intenta promover la accesibilidad y facilitar la experiencia de los usuarios.

## Sprites

En este apartado, se presentarán diferentes estilos de *sprites* para de la aplicación del juego de *Nonograma* dirigido a niños de 6 a 8 años para después describir los *sprites* realizados.

Cada estilo presenta características únicas que pueden influir en la experiencia de juego. Los estilos que se han barajado son: *dibujos animados*, *bloques o formas*, *temático*, y *acuarela o pintura*.

### Comparativa de estilos

La elección del estilo de *sprites* en un juego de *Nonograma* dirigido a niños de 6 a 8 años es crucial para su atractivo visual y su capacidad para enseñar y entretener.

Estos estilos presentan diferentes ventajas y desafíos, por lo que es esencial considerar las preferencias del público objetivo, la capacidad de reconocimiento visual y la claridad de las pistas al seleccionar el estilo de *sprites* más adecuado. Cada estilo, ofrece ventajas únicas y desafíos propios.

Estilos	Ventajas	Desventajas
<p><b>Dibujos Animados.</b> [28]</p> <p>Se basa en personajes y elementos con apariencia de dibujos animados, recomendado para atraer la atención y el interés de los niños.</p>	<p><b>Atractivo visualmente:</b> Captan la atención con colores vibrantes y personajes adorables, lo que aumenta su interés.</p> <p><b>Reconocibilidad:</b> Los objetos son más reconocibles, lo que ayuda a relacionarlos con los conceptos adecuados.</p> <p><b>Estimula la imaginación:</b> Permite crear entornos fantásticos, fomentando la creatividad e imaginación.</p>	<p><b>Saturación visual:</b> Un exceso de colores y detalles sobrecargaría visualmente a los niños más sensibles.</p>

<p><b>Bloques o Formas. [29]</b></p> <p>Usa formas geométricas o bloques para representar los elementos, dando sencillez visual y claridad.</p>	<p><b>Claridad y simplicidad:</b> Las formas simples son fáciles de identificar y asociar.</p> <p><b>Contraste y legibilidad:</b> Al usar tonos brillantes y contrastantes para las formas, hay mejor legibilidad y menos confusiones.</p>	<p><b>Falta de detalle:</b> Al usar formas simples, puede haber pérdida de detalle visual en los <i>sprites</i>.</p>
<p><b>Temático.</b></p> <p>Relacionado con los intereses y gustos de los usuarios, a través de elementos de un tema concreto para hacer más personal la experiencia.</p>	<p><b>Conexión emocional:</b> Usar <i>sprites</i> de temas que atraen a los niños puede generar una conexión emocional y aumentar su interés.</p> <p><b>Aprendizaje contextualizado:</b> Un estilo temático puede facilitar la asimilación de conceptos y pistas en un contexto familiar, lo que promueve un aprendizaje más significativo.</p>	<p><b>Limitaciones de público:</b> Al usar un tema específico, se excluirían los niños que no estén interesados en el mismo (deportes, animales, ...).</p>
<p><b>Acuarela y Pintura [30]</b></p> <p>Los trazos simples en la acuarela y la pintura brindan una estética artística y agradable, a la par que genera una atmósfera visual atractiva y tranquila.</p>	<p><b>Estética visual atractiva:</b> El uso de estos efectos puede crear un arte que capte la atención de los niños y haga que el juego sea visualmente agradable.</p> <p><b>Sensación de calma:</b> Los efectos texturizados transmiten calma y relajación, lo que ayuda a los niños a concentrarse.</p>	<p><b>Pérdida de nitidez:</b> Es posible que se pierda nitidez y definición en los <i>sprites</i>, lo que podría dificultar el identificar los elementos.</p>

Tabla 5. Comparativa de estilos para los Sprites

Se ha realizado una encuesta a 30 personas con edades entre los 38 y 50 años, y con hijos de entre 5 y 8 años, el estilo más votado para un juego infantil ha sido el de acuarela y pintura.

Según los comentarios aportados por los encuestados, sus opiniones respecto al estilo de acuarela y pintura se pueden resumir con los siguientes 3 comentarios:

- *“Este estilo proporciona una sensación visual agradable y artística para un juego infantil.”*
- *“Es atractivo estéticamente, y puede estimular la imaginación de los niños.”*
- *“Este estilo entretiene, y puede alentar el desarrollo de la imaginación y las habilidades visuales en los niños.”*

A continuación, se describen los diferentes tipos de *sprites* que se utilizarán en el proyecto, los cuales se desarrollarán con Photoshop. Se han dividido en 4 tipos:

- **Sprites de imágenes de victoria o derrota:** Habrá dos *sprites* que acompañen las vistas de victoria y derrota respectivamente.
- **Sprite de fondo:** Habrá 2 fondos común para toda la aplicación pudiendo elegir entre cualquiera de ellos. Serán sencillos y compatibles con una clara visibilidad de los botones y del tablero.
- **Sprites de botones:** Habrá botones para acciones como jugar, verificar la solución, cerrar, pistas e instrucciones. Se usará un *Sprite* para ponerles un fondo.

Con estas consideraciones, se espera desarrollar *sprites* que sean visualmente atractivos, intuitivos y contribuyan a una experiencia de juego satisfactoria para los niños de 6 a 8 años.

Finalmente, Se ha realizado una comparativa de 3 modalidades de *sprites* para la aplicación.

1. Estilo visual suave estilo acuarela.
2. Estilo visual más llamativo estilo pintura.
3. Estilo sugerido de *sprites* de libre uso.

El estudio se ha llevado a cabo con una muestra de 10 niños de edades entre 5 y 8 años. El cual ha concluido con un empate entre los estilos 2 y 3.



Imagen 34. Estilo 1

Imagen 34. Estilo 2

Imagen 34. Estilo 3

Por tanto, se ha realizado una encuesta a un grupo de 15 personas de entre 22 y 30 años que juegan a aplicaciones de juegos de puzzles, en la que han valorado los 3 estilos propuestos.



	Consistencia y Cohesión	Calidad	Atractivo Visual	Inmersión y Experiencia	Satisfacción Personal
<b>Estilo 1</b>	7.5	6.5	6	6.7	7.5
<b>Estilo 2</b>	8.2	8.5	9	8	8.6
<b>Estilo 3</b>	6.8	7	8	7.3	8.2

Tabla 6. Comparativa de Tipografías

### Comentarios y sugerencias de los encuestados

- *"El estilo 1 es demasiado apagado."*
- *"El estilo 3 es común y no aporta nada nuevo."*
- *"El estilo 1 es bueno, pero los estilos 2 y 3 tienen más vida."*
- *"Prefiero el estilo 2 ya que es alegre, llamativo y tiene personalidad."*
- *"Agregar variaciones de colores en el fondo."*
- *"Crear animaciones."*
- *"Hacer más detallados los estilos."*
- *"Asegurarse de que es compatible con distintas resoluciones."*

El resultado de la encuesta es la elección del estilo 2 para la aplicación del juego de *Nonograma* que se está desarrollando.

# Capítulo IV. Desarrollo de la aplicación

En este apartado se describe y explica el proceso de desarrollo e implementación de la aplicación, sus partes, la lógica que sigue, y las decisiones tomadas durante el proceso.

## 4.1 Desarrollo

Para abordar este proyecto, se ha implementado una metodología estructurada que ha permitido la definición de un *Backlog*, donde las tareas se han organizado meticulosamente en función de su importancia y complejidad. El objetivo primordial es crear una experiencia de juego cautivadora y funcional para el usuario final.

### Desarrollo de la interfaz de usuario

Se ha concebido un diseño integral de la interfaz de la aplicación, englobando elementos clave como el tablero de juego, los números y símbolos indicativos, y los controles del juego. Este enfoque busca no solo la funcionalidad sino también la estética para garantizar una experiencia visualmente agradable.

### Generación de puzles

La implementación de un sistema de generación de puzles de *Nonograma* de 5x5, de 7x7 y de 10x10 aleatorios ha sido crucial. Esto permite que los usuarios se enfrenten a desafíos diversos y resolubles, proporcionando una experiencia dinámica y entretenida.

### Integración de puzles

La tarea consiste en incorporar los puzles al juego y garantizar una interacción fluida. Se ha trabajado en permitir al usuario marcar casillas en negro, dejar casillas en blanco o marcarlas en gris. El color gris tiene la misma función que el color blanco, pero se proporciona este tercer

color para facilitar la resolución de los tableros, todo ello asegurando la simplificación del proceso de resolución.

### Control de usuario

La creación de un sistema de control interactivo ha sido esencial. Se han programado las interacciones con el tablero de juego para proporcionar una experiencia intuitiva y accesible.

### Verificación del tablero

Se ha diseñado un sistema robusto de verificación de soluciones para asegurar que los usuarios reciban retroalimentación sobre su intento de resolución de cada puzle.

### Rendimiento y estabilidad

Un enfoque dedicado a optimizar el rendimiento de la aplicación y garantizar su estabilidad, asegurando así una experiencia de usuario sin contratiempos.

### Diseño y estilos

Se ha aplicado un diseño visualmente atractivo basado en el estilo de acuarela, alineado con las preferencias del público objetivo. Esto no solo complementa la funcionalidad de la aplicación, sino que también mejora su atractivo estético.

### Mejoras

Además de abordar las tareas fundamentales, se han identificado posibles áreas de mejora y se han delineado futuras líneas de actuación para continuar optimizando y expandiendo la experiencia de juego.

Este planteamiento inicial ha sido esencial para establecer una base sólida y estructurada en el desarrollo de la aplicación del juego de *Nonograma*. Cada tarea se ha abordado con un enfoque centrado en el usuario y en la calidad de la aplicación, asegurando así una experiencia completa y satisfactoria.

## 4.2 Diseño Software

Los apartados siguientes se componen de las partes necesarias para el correcto diseño *Software* que forma la aplicación.

## 4.3 Requisitos

Este apartado detalla los requisitos que deben cumplirse para el desarrollo y funcionamiento óptimo de la aplicación. Éstos se dividen en funcionales y no funcionales.

### Funcionales

- **RF 1:** La aplicación debe ser compatible con dispositivos móviles y tabletas con sistema operativo Android.
- **RF 2:** Los usuarios deben poder marcar las casillas del tablero de forma precisa y fluida.
- **RF 3:** Se requiere una función de verificación de puzzles que permita a los usuarios validar sus soluciones.
- **RF 4:** La aplicación debe tener una interfaz intuitiva y fácil de entender.
- **RF 5:** La aplicación debe permitir la selección de diferentes niveles de dificultad.
- **RF6:** La aplicación debe incluir un modo contrarreloj.
- **RF 7:** La aplicación debe incluir un tutorial sobre cómo jugar.
- **RF 8:** La aplicación debe registrar y mostrar el progreso del usuario.
- **RF 9:** Se requiere incluir la opción de resetear los datos de usuario y las estadísticas.
- **RF 10:** La aplicación debe permitir la personalización de la misma.

### No Funcionales

- **RNF 1:** La aplicación debe tener un tamaño de archivo reducido para minimizar el uso de recursos y optimizar la velocidad de carga.
- **RNF 2:** Se requiere que la aplicación sea accesible para todos los usuarios, incluidos aquellos con discapacidades del aprendizaje.
- **RNF 3:** La aplicación debe ser estable y libre de errores para garantizar una experiencia de juego fluida y sin interrupciones.
- **RNF 4:** La aplicación debe contar con un diseño atractivo y profesional.
- **RNF 5:** La aplicación debe tener tiempos de respuesta rápidos.
- **RNF 6:** La aplicación debe permitir actualizaciones y mantenimiento sin interrumpir la experiencia del usuario
- **RNF 7:** La aplicación debe estar diseñada para un uso intuitivo incluso por usuarios sin experiencia previa en videojuegos.

## 4.4 Diagrama UML de Clases

En el diagrama de *clases* simplificado, se han incluido las *clases* de la aplicación, y sus relaciones, pero sin indicar la totalidad de los atributos y funciones de la *clase UIManager* y de la *clase StatisticsScript*, por tener muchos de estos elementos (indicado con "..."), lo que haría su visualización en la memoria difícil de leer y comprender.

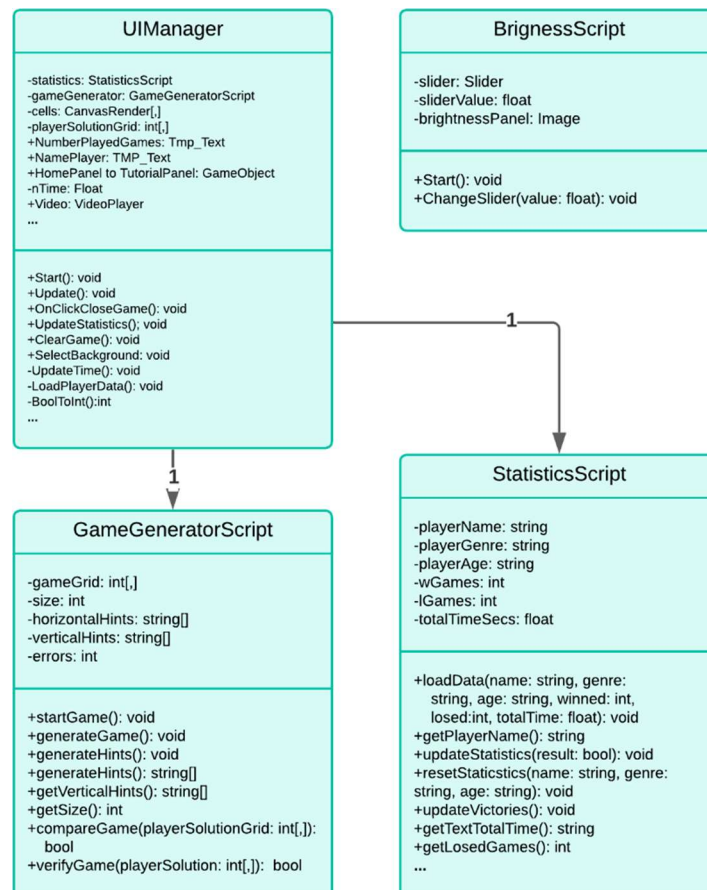


Imagen 35. Diagrama de Clases

Los puntos principales y relevantes de la aplicación se han mantenido, haciendo del diagrama una herramienta muy útil para la comprensión de la aplicación.

En particular, la *clase UIManager* tiene una relación direccional de 1 a 1 con las *clases GameGeneratorScript* y *StatisticsScript*, lo que indica que cada instancia de *UIManager* contiene una instancia de cada una de estas dos *clases*. Por otro lado, la *clase BrignessScript* no se relaciona directamente con las *clases* mencionadas anteriormente.

## Descripción del Diagrama

- **Clase UIManager:** Esta *clase* gestiona la interfaz de usuario del juego. Dentro de su estructura, contiene dos atributos destacables:
  - **Un objeto de la clase GameGeneratorScript:** Esta relación direccional de 1 a 1 indica que *UIManager* utiliza *GameGeneratorScript* para generar y gestionar los elementos del juego.
  - **Un objeto de la clase StatisticsScript:** Similarmente, esta relación direccional de 1 a 1 señala que *UIManager* utiliza *StatisticsScript* para manejar y mostrar las estadísticas del juego.
- **Clase GameGeneratorScript:** Encargada de generar y gestionar los elementos del juego. Su relación con *UIManager* permite que las funciones de generación del juego sean accesibles y controladas a través de la interfaz de usuario.
- **Clase StatisticsScript:** Responsable de calcular, almacenar y mostrar las estadísticas del juego. La relación con *UIManager* asegura que las estadísticas se actualicen y se presenten de manera coherente en la interfaz de usuario.
- **Clase BrightnessScript:** Esta *clase* no tiene una relación directa con las *clases* anteriores. Su función es gestionar el brillo del juego, pero opera de manera independiente en el contexto del diagrama presentado.

## 4.5 Descripción de las clases

A la hora de crear una aplicación de un juego en Unity, la programación se basa en varias *clases* o *Scripts* que se van ejecutando según avanzan el juego y el tiempo.

En esta sección, se da una visión de las *clases* clave que conforman la estructura de la aplicación. Cada *clase* desempeña un papel esencial en la implementación y funcionamiento del juego, contribuyendo a un desarrollo coherente y eficiente.

### UIManager

La *clase UIManager* juega un papel central en la gestión de la interfaz de usuario y la interacción con el usuario. Aquí se presenta un resumen de sus componentes y funciones más relevantes:

## Componentes Principales

- **StatisticsScript**: Objeto que maneja la gestión de estadísticas del usuario.
- **GameGeneratorScript**: Objeto encargado de generar y gestionar la lógica del juego.
- **CanvasRenderer[, ] cells**: Matriz que representa las celdas del tablero de juego.
- **int[, ] playerSolutionGrid**: Matriz que registra la solución del usuario.

## Funciones Clave

- **Start()**: Inicializa componentes, carga datos guardados, y configura el estado inicial del juego.
- **Update()**: Gestiona la actualización del tiempo de juego.

## Otras funciones destacables de UIManager

Esta sección detalla algunas funciones de la *clase UIManager* que desempeñan un papel importante en la ejecución de la aplicación:

- **UpdateTime()**: Actualiza y muestra el tiempo de transcurrido durante la partida.
- **InitCanvasRenderGroup()**: Inicializa el grupo de renderizado del lienzo, asignando las referencias de las celdas y textos de pistas en el tablero *Nonograma*.
- **SaveAllData()**: Guarda todos los datos del usuario (nombre, género, edad), sus estadísticas y los ajustes visuales seleccionados por el usuario.

## StatisticsScript

Esta *clase* gestiona y almacena los datos estadísticos del usuario, e incluye funciones que son fundamentales para el seguimiento y actualización de las estadísticas del usuario.

- **LoadData(string name, string genre, string age, int wonned, int losed, float totalTime)**: Permite cargar los datos del usuario, incluyendo nombre, género, edad, partidas ganadas, partidas perdidas y tiempo total de juego.
- **UpdateStatistics(bool result)**: Actualiza las estadísticas en función del resultado de la partida (victoria o derrota).
- **ResetStatistics(string name, string genre, string age)**: Reinicia todas las estadísticas y establece un nuevo nombre, género y edad.

Esta *clase* desempeña un papel crucial en el seguimiento y registro de las estadísticas del usuario, contribuyendo así a una experiencia de juego más completa y personalizada.

## GameGeneratorScript

La *clase GameGeneratorScript* desempeña un papel crucial en la generación dinámica de juegos, facilitando una experiencia de juego variada y desafiante para el usuario. La lógica implementada contribuye a la coherencia y eficiencia del juego de *Nonograma*. Incluye funciones clave que son fundamentales para la generación y verificación de juegos de *Nonograma*:

- **StartGame():** Inicializa los componentes esenciales del juego, estableciendo el tamaño del tablero (5x5) y creando las matrices necesarias.
- **GenerateGame():** Genera un nuevo juego de *Nonograma*, llenando aleatoriamente el tablero y generando las pistas correspondientes.
- **CompareGame(int[,] playerSolutionGrid):** Compara la solución del usuario con la solución generada, registrando la cantidad de errores.

## BrightnessScript

La *clase BrightnessScript* es responsable de gestionar el brillo en la aplicación del juego de *Nonograma*, permitiendo a los usuarios ajustar la visibilidad según sus preferencias. Incluye funciones clave que son fundamentales para la gestión del brillo en la aplicación:

- **Start():** Inicializa el *slider* con el valor almacenado en *PlayerPrefs* y ajusta el color del panel de brillo en consecuencia.
- **ChangeSlider(float value):** Actualiza el valor del *slider* y aplica el cambio de brillo al panel, almacenando también el nuevo valor en *PlayerPrefs*.

La *clase BrightnessScript* proporciona una funcionalidad extra de personalización para el usuario, permitiéndole ajustar el brillo según sus preferencias. La implementación contribuye a una experiencia de usuario más personalizada y cómoda.



## 4.7 Resultado final de la aplicación

El resultado final de la aplicación se muestra a través de diversas capturas de pantalla que ilustran las diferentes pantallas y funcionalidades. Estas imágenes proporcionan una visión clara de la interfaz y las características implementadas, destacando el diseño y la funcionalidad de la aplicación.

En las imágenes de la 36 a la 40, se muestran capturas de distintas pantallas de la aplicación.



Imagen 36. Pantallazos App 1

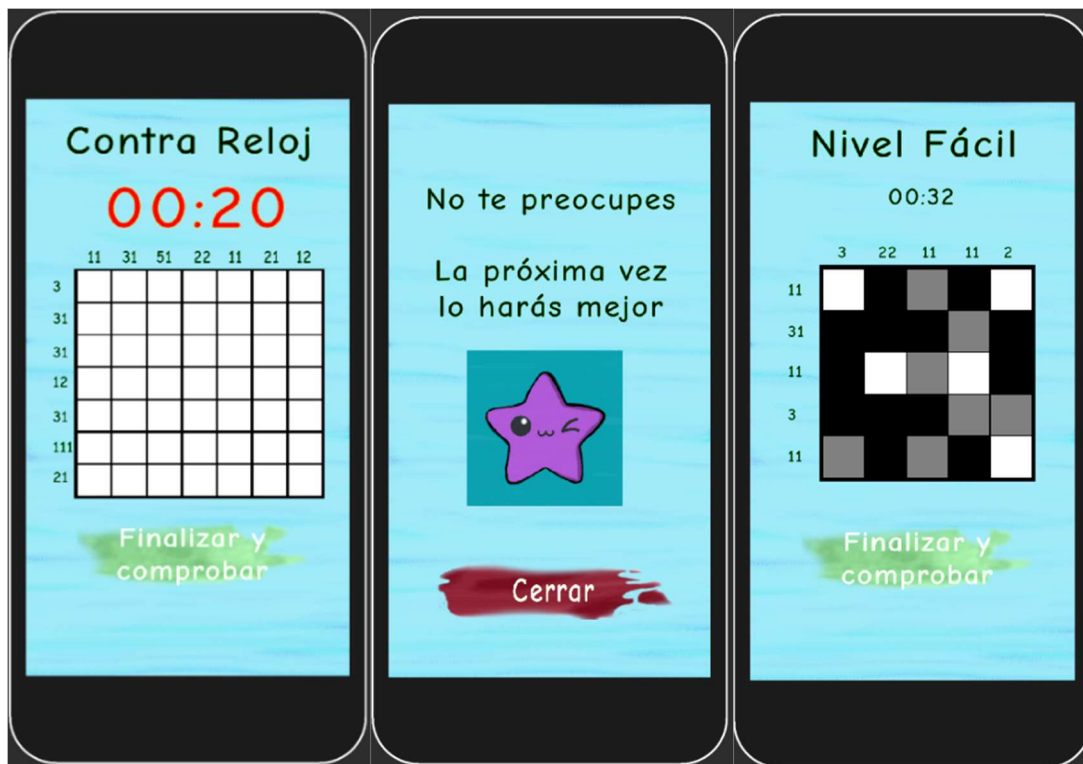


Imagen 37. Pantallazos App 2

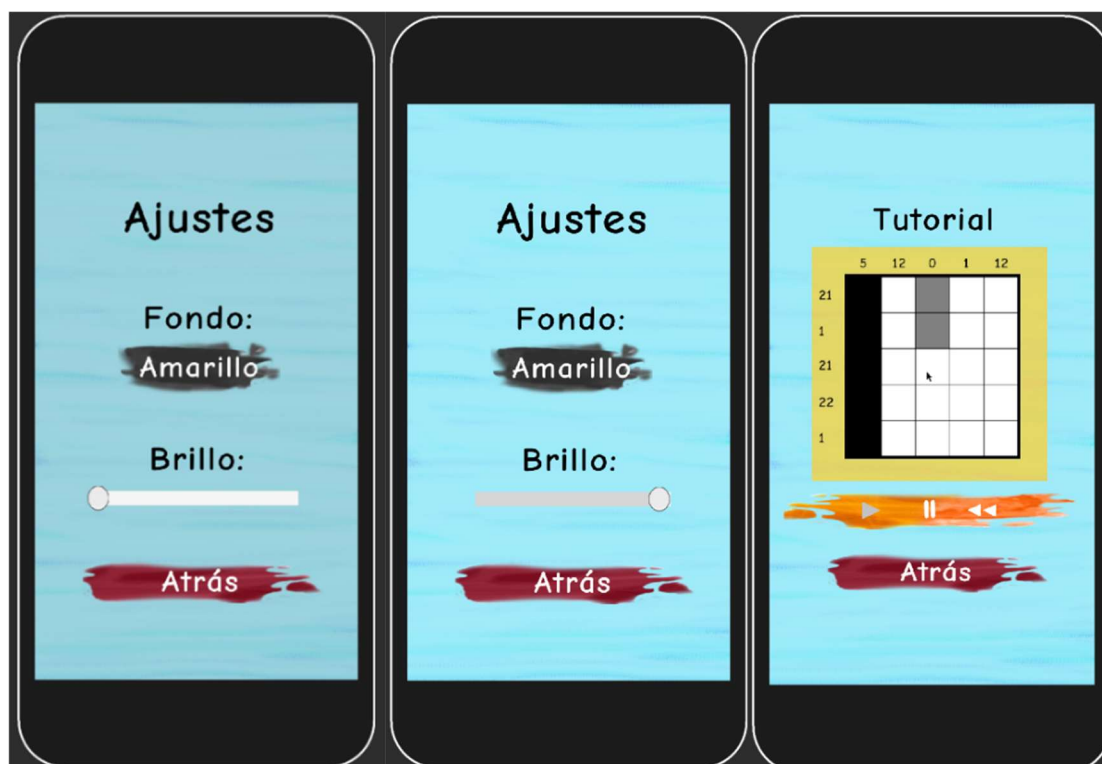


Imagen 38. Pantallazos App 3



Imagen 39. Pantallazos App 4



Imagen 40. Pantallazos App 5

# Capítulo V. Validación

En esta fase del desarrollo, se han llevado a cabo distintas pruebas con el objetivo de identificar posibles errores y asegurar la estabilidad y funcionamiento óptimo de la aplicación. Se abordaron diversas áreas, destacando los aspectos más comunes que suelen ser propensos a problemas durante el desarrollo de aplicaciones.

## 5.1 Pruebas de Interfaz de Usuario (UI)

Las pruebas de interfaz de usuario son fundamentales para asegurar que los usuarios interactúen con la aplicación de manera eficiente y sin contratiempos.

Este apartado se enfoca en las pruebas realizadas para garantizar que la interfaz de usuario cumple con los principios de diseño de UX (Experiencia de Usuario), proporcionando una experiencia intuitiva y agradable.

Además, se describen las pruebas específicas realizadas con un grupo de usuarios para validar la efectividad de la interfaz.

Aplicando las leyes de UX para este apartado, se prestó especial atención a:

- 1. Efecto estético - usabilidad:** Se realizó un diseño sencillo y agradable, asegurando que la estética de la aplicación no comprometiera su usabilidad.
- 2. Umbral de Doherty:** Se buscó obtener eficiencia y tiempos de espera mínimos, optimizando la respuesta de la aplicación para mantener la atención del usuario.
- 6. Ley de Jakob:** Se diseñó una interfaz similar a la de otras aplicaciones de juegos de puzzles, facilitando la familiaridad y la facilidad de uso.
- 7. Ley de la Región Común:** Se agruparon visualmente elementos similares, como los botones de “Atrás” que comparten color, mejorando la organización visual.
- 9. Ley de Prägnanz:** Se emplearon imágenes sencillas en la interfaz, evitando el exceso de elementos visuales complejos.

**10. Ley de Similitud:** Se colocaron los elementos con funcionalidades similares, como los botones de "Atrás", en posiciones similares, ayudando a los usuarios a entender rápidamente su propósito.

Además, se realizaron pruebas intensivas en la interfaz de usuario con un grupo de 6 niños de 6 años en tres sesiones de 10 minutos para verificar que tuvieran una experiencia fluida y sin inconvenientes. Se prestó especial atención a:

- **Botones y Controles:** Se verificó el correcto funcionamiento de botones como "Juego", "Estadísticas", "Ajustes", "Tutorial", "Finalizar y Comprobar", "Reset", "Atrás" (en los diferentes paneles), "Rosa/Azul" (como botón para seleccionar el color del fondo), "Cerrar" (en los diferentes paneles), botones de control del video tutorial, "Aceptar" y "Salir", asegurando que respondieran de manera adecuada a las interacciones del usuario.
- **Formulario y slider:** Se verificó el correcto funcionamiento de los componentes del formulario de datos de usuario (nombre, edad y género) y del *slider* selector del brillo, asegurando que todo funcionara de la forma esperada.
- **Indicador de Tiempo:** Se realizaron pruebas para confirmar que el temporizador de tiempo del juego se actualizara correctamente, se reiniciara al comenzar la partida, y que su formato fuera legible para los usuarios.

## 5.2 Pruebas de Lógica del Juego

Las pruebas de lógica del juego son esenciales para asegurar que la mecánica del juego funcione correctamente y que los usuarios puedan disfrutar de una experiencia sin errores. Este apartado describe las pruebas realizadas en la lógica del juego, enfocándose en la generación de tableros y la validación de soluciones de los usuarios.

- **Generación de Tableros:** Se verificó que los tableros generados cumplieran con las reglas del *Nonograma*, asegurando la presencia de pistas coherentes y soluciones adecuadas. Para solventar el dilema de las posibles soluciones múltiples, se permiten 2 errores en las soluciones, contando como victoria.
- **Validación de Soluciones:** Se realizaron pruebas con soluciones válidas e inválidas para garantizar que el sistema de verificación devolviera la respuesta adecuada, llamando al panel de victoria o derrota de manera correcta.

Para facilitar estas pruebas, se usó de forma temporal un componente de texto (TMP) que mostraba la solución al *Nonograma* generado, el cual se eliminó una vez completada y verificada esta etapa.

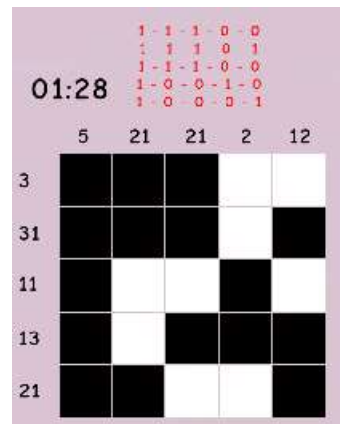


Imagen 41. Texto auxiliar de verificación de la solución.

## 5.3 Pruebas de Manejo de Datos del Usuario

La correcta gestión de los datos del usuario es fundamental para asegurar que la aplicación almacene y recupere la información de manera eficiente. Este apartado detalla las pruebas realizadas para verificar la actualización de estadísticas, la carga y guardado de datos, y el reseteo de los mismos.

- **Actualización de Estadísticas:** Se verificó que las estadísticas del usuario se actualizarán de manera adecuada al finalizar cada juego, reflejando victorias, derrotas y tiempo total jugado.
- **Carga/Guardado de Datos:** Se realizaron pruebas para asegurar que los datos del usuario, y las características elegidas en ajustes (fondo y brillo) se cargaran correctamente al iniciar la aplicación y se guardaran de manera precisa al salir.
- **Reseteo de Estadísticas:** Se verificó que las estadísticas del usuario se reiniciarán de manera adecuada, y se solicitarán los nuevos datos del usuario tales como nombre, edad y género.

## 5.4 Pruebas de rendimiento

La evaluación del rendimiento es una parte crucial del desarrollo de software, donde la eficiencia y la estabilidad son esenciales para asegurar una experiencia de usuario satisfactoria.

En este apartado, se detallan las pruebas de rendimiento realizadas en la aplicación desarrollada, utilizando una herramienta específica de análisis de rendimiento para identificar y mitigar posibles cuellos de botella.

### Profiler de Unity

El *Profiler* de Unity es una herramienta integrada en el entorno de desarrollo de Unity que permite analizar y entender el rendimiento del juego en tiempo real. Proporciona información detallada sobre el uso de la *CPU*, la memoria, la *GPU*, y otros aspectos críticos del rendimiento. Esta herramienta es fundamental para optimizar el juego, ya que permite identificar exactamente dónde y cuándo ocurren los problemas de rendimiento.

El *Profiler* ofrece múltiples módulos de análisis, cada uno enfocado en un aspecto diferente del rendimiento:

- **CPU Usage:** Muestra cómo se distribuye la carga de trabajo entre los diferentes procesos del juego y permite identificar los procesos que consumen más recursos.

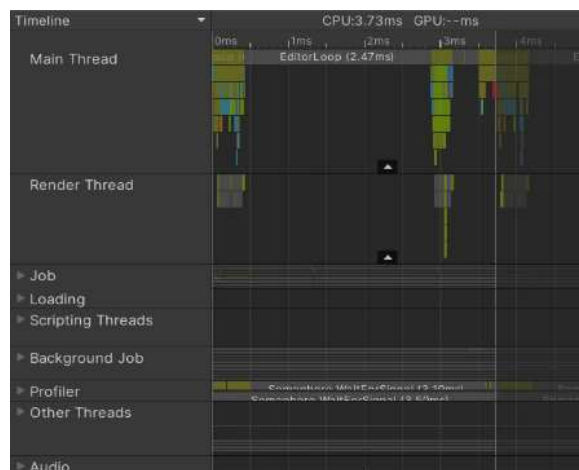


Imagen 42. Pruebas Profiler CPU - 1



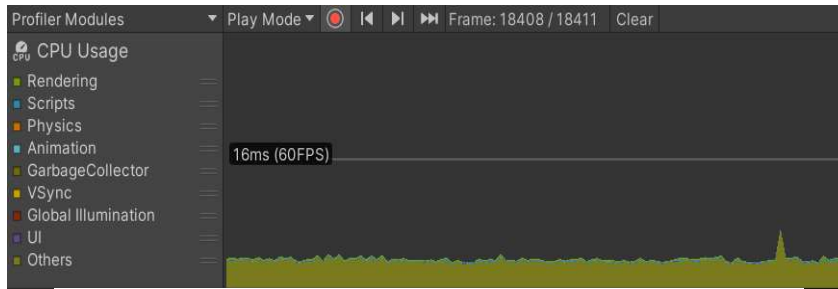


Imagen 43. Pruebas Profiler CPU - 2

- **Rendering:** Analiza el rendimiento del renderizado, proporcionando información sobre el tiempo de renderizado de los objetos y el consumo de recursos gráficos.

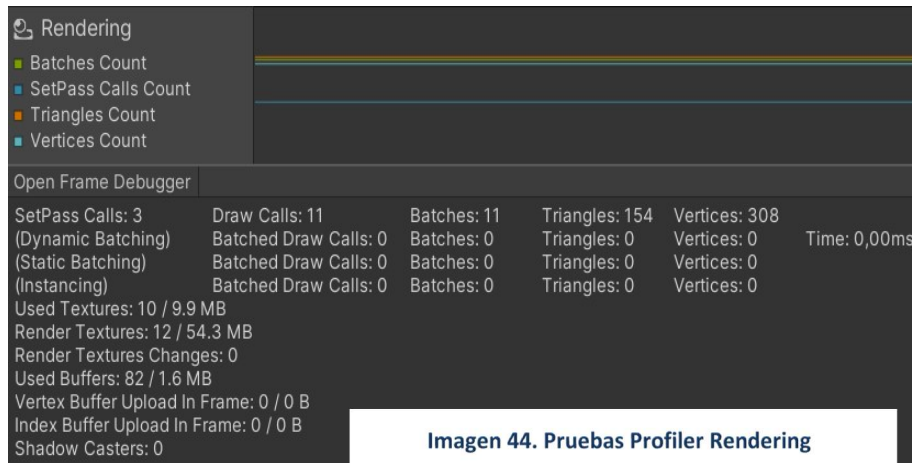


Imagen 44. Pruebas Profiler Rendering

- **Memory:** Monitorea el uso de la memoria para detectar posibles fugas de memoria y gestionar eficientemente los recursos.

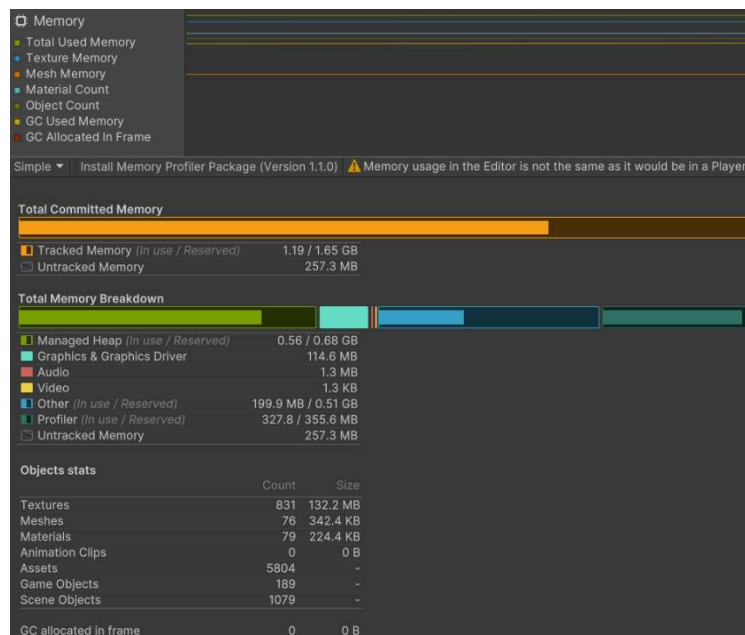


Imagen 45. Pruebas Profiler Memory



- **UI:** Evalúa el rendimiento de la interfaz de usuario, crucial para mantener una experiencia de usuario fluida.

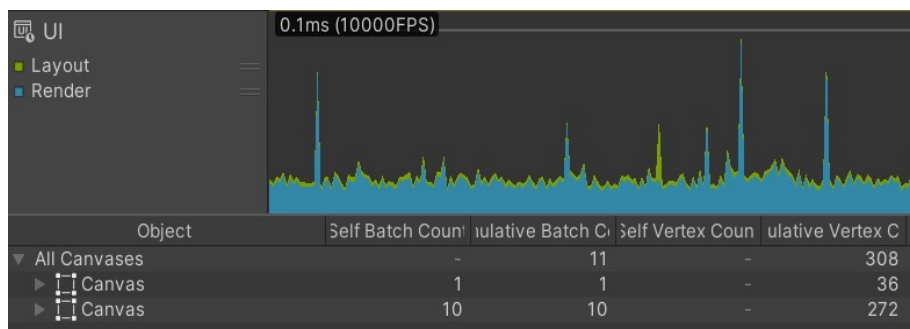


Imagen 46. Pruebas Profiler UI

## Resultados de las Pruebas de Rendimiento

Durante las pruebas de rendimiento, se observó que los picos en el uso de la CPU se agrupan principalmente durante los cambios de ventana durante la navegación.

Sin embargo, el pico más significativo se produce al generar y validar el *Nonograma*. A pesar de estos picos, se logró mantener un rendimiento muy estable en general, con tiempos de respuesta adecuados y sin ralentizaciones perceptibles para el usuario.

El análisis detallado con el Profiler mostró que:

- **CPU Usage:** Los picos más altos se registraron durante la generación y validación del *Nonograma*, pero estos picos fueron mínimos y no afectaron significativamente el rendimiento global.
- **Rendering:** El tiempo de renderizado se mantuvo dentro de los límites aceptables, asegurando una experiencia gráfica fluida.
- **Memory:** No se detectaron fugas de memoria significativas, y el uso de la memoria se mantuvo estable durante las pruebas.
- **UI:** La interfaz de usuario operó de manera eficiente, sin causar picos significativos en el uso de la CPU ni en la memoria.

Las pruebas de rendimiento realizadas han proporcionado una visión clara del comportamiento de la aplicación bajo diferentes condiciones. La identificación de los picos de rendimiento y su análisis han permitido realizar los ajustes necesarios para asegurar un buen rendimiento. Tales como:

- **Optimización de Algoritmos:** Se optimizaron los algoritmos de generación y validación del Nonograma para reducir el consumo de CPU.
- **Mejora en la Gestión de Memoria:** Se implementaron técnicas de gestión de memoria para evitar fragmentación y liberar memoria de manera más eficiente.
- **Ajustes en la Interfaz de Usuario:** Se ajustaron elementos de la interfaz para reducir la complejidad de renderizado y mejorar la eficiencia general.

## 5.5 Corrección de Errores

El proceso de pruebas reveló varios errores que fueron identificados y corregidos para asegurar el funcionamiento óptimo de la aplicación. Este apartado describe los tipos de errores encontrados y las medidas tomadas para solucionarlos.

- **Errores de Interfaz:** Se solucionaron problemas visuales, como la superposición incorrecta de elementos en la interfaz. Por ejemplo, se ajustaron los márgenes y el posicionamiento de los botones para evitar solapamientos.
- **Errores de Lógica:** Se corrigieron situaciones donde la generación de tableros no cumplía con las reglas del juego. Uno de esos casos la aparición de Nonogramas sin solución válida, lo cual se abordó revisando y ajustando los algoritmos de generación. Otro de esos casos fue la confusión a la hora de escribir las pistas horizontales y las verticales, cambiando unas por otras, lo cual se corrigió y verificó tras un par de pruebas.
- **Errores de Datos:** Se abordaron problemas relacionados con la carga/guardado incorrecto de datos del usuario y los ajustes escogidos por el mismo. Por ejemplo, se corrigieron fallos en la serialización de datos que causaban la pérdida de la configuración personalizada al reiniciar la aplicación.

En resumen, las pruebas y correcciones de errores han desempeñado un papel crucial para asegurar una aplicación sólida y libre de problemas. Este proceso continuo de mejora garantiza una experiencia de usuario satisfactoria y contribuye al éxito general del proyecto.

## 5.6 Validación de requisitos

Se han cumplido todos los requisitos funcionales y no funcionales establecidos para el proyecto:

- **RF 1:** La aplicación debe ser compatible con dispositivos móviles y tabletas con sistema operativo Android.

**Validación:** La aplicación ha sido desarrollada y probada para asegurar su compatibilidad con varios tipos de dispositivos Android.

- **RF 2:** Los usuarios deben poder marcar las casillas del tablero de forma precisa y fluida.

**Validación:** La interacción con el tablero ha sido optimizada para asegurar una experiencia de usuario precisa y fluida.

- **RF 3:** Se requiere una función de verificación de puzzles que permita a los usuarios validar sus soluciones y recibir retroalimentación sobre su uso de la aplicación.

**Validación:** La función de verificación ha sido implementada y permite a los usuarios recibir retroalimentación sobre sus soluciones.

- **RF 4:** La aplicación debe tener una interfaz intuitiva y fácil de entender.

**Validación:** La interfaz ha sido diseñada con principios de usabilidad, asegurando que sea accesible y comprensible para los niños.

- **RF 5:** La aplicación debe permitir la selección de diferentes niveles de dificultad.

**Validación:** Se han implementado múltiples niveles de dificultad y se realizaron pruebas para garantizar que funcionen correctamente.

- **RF6:** La aplicación debe incluir un modo contrarreloj.

**Validación:** El modo contrarreloj ha sido implementado y probado, asegurando su correcto funcionamiento y atractivo para los usuarios.

- **RF 7:** La aplicación debe incluir un tutorial sobre cómo jugar.

**Validación:** Se ha incluido un tutorial que guía a los nuevos usuarios a través de las mecánicas del juego.

- **RF 8:** La aplicación debe registrar y mostrar el progreso del usuario.

**Validación:** La funcionalidad de seguimiento del progreso ha sido implementada y probada, mostrando correctamente los avances de los usuarios en diferentes sesiones.

- **RF 9:** Se requiere incluir la opción de resetear los datos de usuario y las estadísticas.

**Validación:** La opción de resetear datos y estadísticas ha sido implementada y probada para asegurar su correcto funcionamiento.

- **RF 10:** La aplicación debe permitir la personalización de la misma.  
**Validación:** Se han implementado opciones de personalización, como el ajuste de brillo y el cambio de color de fondo, permitiendo a los usuarios ajustar estos aspectos del juego según sus preferencias. Estas opciones fueron probadas para asegurar su efectividad.
- **RNF 1:** La aplicación debe tener un tamaño de archivo reducido para minimizar el uso de recursos y optimizar la velocidad de carga.  
**Validación:** La optimización del tamaño del archivo ha sido una prioridad, resultando en tiempos de carga rápidos y un uso eficiente de los recursos.
- **RNF 2:** Se requiere que la aplicación sea accesible para todos los usuarios, incluidos aquellos con discapacidades del aprendizaje.  
**Validación:** Se han implementado características de accesibilidad para asegurar que todos los niños, incluidos aquellos con discapacidades del aprendizaje, puedan utilizar la aplicación efectivamente.
- **RNF 3:** La aplicación debe ser estable y libre de errores para garantizar una experiencia de juego fluida y sin interrupciones.  
**Validación:** Se han realizado pruebas rigurosas de estabilidad han garantizado que la aplicación sea robusta y ofrezca una experiencia de usuario sin interrupciones.
- **RNF 4:** La aplicación debe contar con un diseño atractivo y profesional.  
**Validación:** Se ha cuidado el diseño visual de la aplicación para que sea atractivo y profesional, realizándose pruebas con usuarios para confirmar la percepción positiva del diseño.
- **RNF 5:** La aplicación debe tener tiempos de respuesta rápidos.  
**Validación:** La aplicación ha sido optimizada para asegurar tiempos de respuesta rápidos, mejorando la experiencia del usuario.
- **RNF 6:** La aplicación debe permitir actualizaciones y mantenimiento sin interrumpir la experiencia del usuario.  
**Validación:** Se ha validado que al introducir actualizaciones y mejoras no afectar negativamente la experiencia del usuario.
- **RNF 7:** La aplicación debe estar diseñada para un uso intuitivo incluso por usuarios sin experiencia previa en videojuegos.  
**Validación:** La interfaz y la jugabilidad han sido diseñadas y probadas para asegurar que sean intuitivas y fáciles de entender para usuarios sin experiencia previa en videojuegos.

# Capítulo VI. Conclusiones

El desarrollo de la aplicación del juego de *Nonograma* ha sido un proceso desafiante que ha abarcado diversos aspectos, desde la investigación de los temas relacionados, pasando por la planificación inicial hasta la implementación y las pruebas. Este capítulo presenta las conclusiones clave derivadas de este proyecto, organizadas en objetivos alcanzados, requisitos cumplidos, logros y desafíos, lecciones aprendidas y perspectivas futuras.

## 6.1 Cumplimiento de objetivos

El proyecto ha cumplido con éxito todos los objetivos planteados inicialmente. A continuación, se detallan estos objetivos y se valida su cumplimiento:

- **OP:** Crear una aplicación destinada a ayudar a desarrollar el PC en niños de educación primaria.

**Validación:** La aplicación ha sido diseñada específicamente para niños de educación primaria, proporcionando una herramienta educativa que facilita el desarrollo del PC a través de un enfoque lúdico y atractivo.

- **OS 1:** Realizar una investigación sobre los videojuegos educativos existentes, evaluando su efectividad y metodología.

**Validación:** Se ha llevado a cabo una investigación exhaustiva sobre los videojuegos educativos y los diseñados para el aprendizaje, identificando las mejores prácticas y metodologías que informaron el diseño de la aplicación.

- **OS 2:** Realizar una investigación sobre los conceptos y metodologías relacionadas con el PC.

**Validación:** Se ha completado una investigación profunda sobre el PC, incluyendo sus fundamentos teóricos y aplicaciones prácticas, que ha sido crucial para el desarrollo del contenido educativo de la aplicación.

- **OS 3:** Diseñar y desarrollar una aplicación de juego de puzzles que sea educativa y atractiva para los niños.

**Validación:** La aplicación ha sido diseñada con un enfoque en la usabilidad y la estética, haciendo que el aprendizaje sea una experiencia agradable y motivadora para los niños.

- **OS 4:** Implementar la aplicación de juego en 2D utilizando el motor de desarrollo Unity, aprovechando sus capacidades para crear experiencias interactivas y visualmente atractivas.

**Validación:** La aplicación ha sido desarrollada en Unity, lo que ha permitido crear gráficos atractivos y una experiencia interactiva fluida.

- **OS 5:** Realizar una validación exhaustiva de la aplicación mediante diversos tipos de pruebas, incluyendo pruebas de usabilidad, funcionalidad y retroalimentación de los usuarios.

**Validación:** Se han llevado a cabo múltiples rondas de pruebas de usabilidad y funcionalidad, con la participación de varios usuarios, cuyos comentarios han sido integrados en el proceso de mejora continua.

- **OS 6:** Analizar los resultados obtenidos de las pruebas y sacar conclusiones sobre el trabajo realizado, identificando áreas de mejora y futuros desarrollos.

**Validación:** Los resultados de las pruebas han sido analizados exhaustivamente, proporcionando información valiosa sobre el rendimiento de la aplicación y posibles mejoras futuras.

## 6.2 Logros y Desafíos

El desarrollo de la aplicación ha traído consigo diversos logros significativos y desafíos que se han superado a lo largo del proceso.

### Implementación Efectiva:

La aplicación ha logrado implementar efectivamente las reglas del *Nonograma*, ofreciendo a los usuarios una experiencia auténtica y desafiante que fomenta el desarrollo del PC.

### Interfaz de Usuario Intuitiva

Se ha diseñado una interfaz de usuario intuitiva y amigable, facilitando la navegación y mejorando la experiencia general del usuario, lo cual es crucial para mantener el interés y la motivación de los niños. Sin embargo, esta interfaz está sujeta a mejoras en las artes.

### Gestión de Datos del Usuario

La aplicación incorpora una robusta gestión de datos del usuario, incluyendo estadísticas actualizadas y opciones de carga/guardado, permitiendo una experiencia personalizada que enriquece el aprendizaje.

## Corrección de Errores

A través de diversas pruebas, se han identificado y corregido diversos errores, mejorando significativamente la estabilidad y la calidad de la aplicación.

## 6.3 Lecciones Aprendidas

El desarrollo de la aplicación ha proporcionado una serie de valiosas lecciones que abarcan desde la importancia de las pruebas continuas hasta la necesidad de una planificación meticulosa y la flexibilidad en el desarrollo. Estas lecciones no solo han mejorado la calidad del producto final, sino que también han enriquecido significativamente el proceso de desarrollo y la gestión del proyecto.

En primer lugar, las pruebas continuas han demostrado ser esenciales para la detección temprana de errores y la mejora constante de la aplicación. La implementación de un ciclo iterativo de pruebas y retroalimentación ha permitido identificar y corregir problemas de manera oportuna, evitando que errores críticos llegaran a las fases finales del desarrollo.

Esta práctica ha asegurado que la aplicación mantenga un alto estándar de calidad y funcionalidad. Las pruebas de usabilidad han sido particularmente reveladoras, proporcionando información directa de los usuarios sobre aspectos de la interfaz y la experiencia general del usuario. Esto ha permitido realizar ajustes precisos que han mejorado la accesibilidad y la facilidad de uso de la aplicación.

Además, la planificación meticulosa ha sido un pilar fundamental para el éxito del proyecto. El uso de metodologías ágiles, como *Kanban*, ha facilitado la organización y priorización de tareas, permitiendo un enfoque estructurado pero flexible. La planificación detallada ha ayudado a anticipar posibles desafíos y a establecer hitos claros, lo que ha sido crucial para mantener el proyecto en curso y dentro de los plazos previstos.

La organización de cada etapa del desarrollo ha proporcionado una base sólida para la toma de decisiones y ha permitido una revisión crítica y constructiva del progreso, asegurando que todos los objetivos se cumplieran de manera eficiente.

Finalmente, la capacidad de adaptarse a los cambios ha sido una lección crucial aprendida durante el desarrollo. A lo largo del proyecto, surgieron imprevistos y cambios que requirieron una respuesta ágil y efectiva.

La flexibilidad en la gestión del proyecto ha permitido ajustar las prioridades y reasignar recursos de manera eficiente, garantizando que el desarrollo no se viera comprometido por factores externos. Este enfoque adaptable ha sido clave para integrar nuevas ideas y mejoras que surgieron durante el proceso, enriqueciendo el producto final y asegurando su relevancia y eficacia en el contexto educativo.

En resumen, las pruebas continuas, una planificación adecuada y la flexibilidad para adaptarse a los cambios han sido lecciones fundamentales aprendidas en este proyecto.

## 6.4 Perspectivas Futuras

El desarrollo de la aplicación no termina con su lanzamiento, si no que a lo largo de la vida de la aplicación necesitará mantenimiento, actualizaciones, cambios, etc. Se han identificado varias áreas para futuras mejoras y expansiones:

### Actualizaciones y Mejoras

Considerando la naturaleza dinámica de los juegos, se planean futuras actualizaciones para enriquecer la experiencia del usuario y agregar nuevas funcionalidades. Estas mejoras incluyen:

- **Incorporación de IA:** Implementación de una inteligencia artificial que genere un sistema de dificultad creciente y acompañe al usuario en su aprendizaje.
- **Informes de Rendimiento:** Posibilidad de que los usuarios puedan descargarse un informe detallado de su rendimiento, facilitando un seguimiento más preciso de su progreso.
- **Personalización de Estadísticas:** Mayor personalización de las estadísticas, permitiendo a los usuarios adaptar la visualización de sus datos según sus preferencias y necesidades.
- **Mejora de Artes:** Actualización y mejora de los elementos visuales del juego, incluyendo gráficos, animaciones y diseño de interfaz. Esto no solo mejorará la estética del juego, sino que también puede contribuir a una experiencia de usuario más inmersiva y atractiva.



- **Exploración de Plataformas:** Se podría expandir la presencia de la aplicación a otras plataformas como aquellas con sistema operativo iOS, brindando a más usuarios la oportunidad de disfrutar del *Nonograma* y beneficiarse de su enfoque educativo.

En resumen, el proyecto ha sido exitoso en alcanzar sus objetivos y proporcionar una herramienta educativa valiosa. Las perspectivas futuras prometen seguir mejorando y expandiendo el impacto positivo de la aplicación en la educación primaria.

# Glosario

## Backlog

Lista de tareas o requisitos que deben ser completados o implementados en un proyecto. Representa la planificación y priorización de tareas a largo plazo y permite a los equipos tener una visión clara de las tareas pendientes y de las metas a alcanzar.

## Clase

Una clase es una plantilla que define las propiedades y comportamientos de un objeto. Incluye variables que representan sus características y funciones que determinan cómo interactúa y se comporta dicho objeto. Las clases son esenciales en la programación orientada a objetos, ya que permiten crear múltiples instancias de un mismo tipo de objeto con características similares.

## CPU (Unidad Central de Procesamiento)

La CPU, o procesador, es el componente principal de una computadora que realiza las instrucciones de un programa mediante operaciones aritméticas, lógicas, de control y de entrada/salida (E/S). Es considerado el "cerebro" de la computadora, ya que ejecuta las tareas y procesos necesarios para el funcionamiento del sistema.

## Curva de aprendizaje

Se refiere a la cantidad de tiempo y esfuerzo que se requiere para aprender a usar un software o tecnología.

## Descriptorios del perfil de salida

Son objetivos educativos que los alumnos deben lograr al final de cada etapa. Funcionan como vínculo entre las competencias generales y las específicas, asegurando la coherencia y continuidad del aprendizaje. Su cumplimiento garantiza que se adquieran las habilidades necesarias para el desarrollo académico, profesional y personal [31].

## Done

En la metodología *Kanban*, "*Done*" representa las tareas completadas y listas para su revisión o entrega final. Estas tareas han pasado por todas las etapas necesarias y cumplen con los criterios de aceptación establecidos.

## Firmware

Es un tipo específico de software que está permanentemente incrustado en el hardware de un dispositivo. Proporciona las instrucciones básicas necesarias para que el hardware funcione y se comuniquen con otros componentes del sistema. A diferencia del software regular, el *firmware* reside en memoria no volátil como la *ROM* o la *memoria flash*, lo que le permite persistir a través de apagados y reinicios.

## Gamificación

La *Gamificación* es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo-profesional con el fin de conseguir mejores resultados, ya sea para absorber mejor algunos conocimientos, mejorar alguna habilidad, o bien recompensar acciones concretas, entre otros muchos objetivos.

## GPU

La *GPU* es la unidad de procesamiento gráfico que maneja la visualización de imágenes y animaciones en dispositivos electrónicos.

## In Progress

En la metodología *Kanban*, "*In Progress*" indica las tareas en las que el equipo o usuario está trabajando actualmente. Estas son las actividades en proceso de realización.

## LOMLOE

LOMLOE es el acrónimo de Ley Orgánica de Modificación de la Ley Orgánica de Educación, una legislación que establece los principios y normas para la educación en España.

## Memoria Flash

Es un tipo de memoria no volátil que se utiliza para el almacenamiento y transferencia de datos en dispositivos electrónicos. A diferencia de la memoria *RAM*, ésta retiene los datos almacenados incluso cuando el dispositivo está apagado.

## Memory (Memoria)

En informática, se refiere a los componentes de hardware que almacenan datos y programas temporales o permanentes. Existen varios tipos, como la *RAM* que se utiliza para almacenar datos temporales mientras un programa se está ejecutando, y la *ROM* que contiene datos permanentes necesarios para el funcionamiento básico del sistema.

## Metodologías Agile

Las metodologías Agile son enfoques de desarrollo de software que promueven la flexibilidad, la colaboración y la adaptabilidad en el proceso de trabajo. Algunas metodologías ágiles populares incluyen *Scrum* y *Kanban*.

## Monitoreo

En informática, el *monitoreo* se refiere a la observación continua y sistemática de sistemas, redes y aplicaciones para asegurar su rendimiento, disponibilidad y seguridad. Esto incluye la recopilación y análisis de datos en tiempo real para identificar problemas, realizar diagnósticos y optimizar el funcionamiento de los recursos tecnológicos.

## Motor de videojuegos

Un motor de juego es un software que proporciona una serie de herramientas y tecnologías que facilitan el proceso de desarrollo.

## Nonograma

Un *Nonograma* es un juego de lógica donde se rellenan celdas en una cuadrícula para revelar una imagen o patrón. Las filas y columnas están etiquetadas con números que indican cuántas celdas consecutivas deben ser llenadas o cambiadas de color. Se requiere lógica y deducción para resolver el juego. Son conocidos también como crucigramas de píxeles o japoneses.

## Patrones de Uso

Son los comportamientos y tendencias observados en la manera en que los usuarios interactúan con sistemas, aplicaciones o recursos tecnológicos. Analizar los *patrones de uso* permite a los desarrolladores y administradores comprender mejor las necesidades de los usuarios, optimizar el diseño del software y mejorar la eficiencia del sistema.

## Pertinencia

Se refiere a la relación de una cosa con otra o con un contexto en particular.

## PlayerPrefs

*PlayerPrefs* es una clase en Unity para guardar y recuperar datos del usuario, como configuraciones o puntajes.

## Producto mínimo viable (PMV)

Producto mínimo viable. Término empleado en las metodologías ágiles que identifica a una versión que, aunque no esté terminada, puede usarse completamente.

## RAM (Memoria de Acceso Aleatorio):

Es un tipo de memoria volátil que se utiliza para almacenar datos y programas que están siendo utilizados o procesados activamente por la CPU. Se caracteriza por su alta velocidad de lectura y escritura, lo que permite un acceso rápido a los datos. Sin embargo, los datos almacenados en la *RAM* se pierden cuando se apaga el dispositivo.

## Reconocibilidad

1. Cualidad de reconocible.
2. Cualidad por la que puede ser conocido algo o a alguien.

## Rendering (Renderizado)

Es el proceso de generar una imagen a partir de un modelo utilizando software. En gráficos por computadora, renderizar implica convertir datos de una representación abstracta en una imagen visual. Este proceso es crucial en el desarrollo de aplicaciones, videojuegos y películas de animación, ya que determina la apariencia final de los gráficos en la pantalla.

## Reset

Se refiere a "restablecer" o "reiniciar". Se utiliza comúnmente en el contexto de la tecnología y los dispositivos electrónicos para indicar la acción de volver a configurar o reiniciar un dispositivo a su estado original o predeterminado.

## ROM (Memoria de Solo Lectura)

Es un tipo de memoria no volátil que almacena datos que no pueden ser modificados fácilmente o que solo se pueden leer. Esta memoria es utilizada principalmente para almacenar *firmware* o software que es esencial para el arranque y el funcionamiento básico de un dispositivo. A diferencia de la *RAM*, los datos en la *ROM* se conservan incluso cuando el dispositivo se apaga.

## Script

Un *script* es un conjunto de instrucciones o comandos escritos en un lenguaje de programación que se utilizan para automatizar tareas o realizar acciones específicas dentro de un software o sistema.

## Scrum

*Scrum* es una metodología ágil ampliamente utilizada en la gestión de proyectos. Se basa en iteraciones cortas de trabajo llamadas *Sprint* y fomenta la colaboración, la comunicación y la adaptación continua.

## Slider

Un Slider es un elemento de UI que permite al usuario seleccionar un valor moviéndolo horizontalmente.

## Sprint

Un *sprint* es un período de trabajo corto y definido, típico de metodologías ágiles como *Scrum*, que dura entre una y cuatro semanas. Durante este tiempo, el equipo de desarrollo se concentra en completar un conjunto específico de tareas. Al final de cada *sprint*, se realiza una revisión para evaluar el progreso y planificar el siguiente ciclo. Este enfoque permite mejoras continuas y adaptabilidad en el desarrollo de proyectos.

## Sprite

Referido a un elemento gráfico, generalmente una imagen o un personaje, utilizado en el campo del diseño y desarrollo de videojuegos y gráficos por ordenador. Los *sprites* son objetos visuales que pueden moverse, cambiar de forma o realizar otras acciones en la pantalla, y son parte importante del diseño de juegos y animaciones.

## STEAM

Acrónimo que se refiere a las áreas de conocimiento con las que suelen trabajar los científicos e ingenieros, *Science – Technology – Engineering – Arts - Mathematics* (Ciencia – tecnología – Ingeniería – Artes - Matemáticas).

## Testing

Se refiere al proceso de evaluar y verificar que un software funcione correctamente. Esto incluye la identificación de defectos, errores o inconsistencias respecto a los requisitos.

## Timeline

*Timeline* (o línea de tiempo) es una herramienta visual que representa una secuencia cronológica de eventos o acciones a lo largo de un período. Su uso es muy habitual en programas de gestión de proyectos y seguimiento de tareas para planificar, organizar y visualizar el progreso y la duración de tareas y actividades.

## To Do

En la metodología *Kanban*, "*To Do*" es la lista de tareas pendientes que aún no han comenzado. Aquí se registran las actividades que el equipo o usuario planea realizar en el futuro cercano.

## Test de Pensamiento Computacional

Evaluación para medir la capacidad de un individuo para resolver problemas mediante métodos y técnicas de PC, tales como abstracción, descomposición, generalización y evaluación sistemática.

## Usage (Uso)

En el contexto de la informática, el uso se refiere a cómo se emplea un recurso, sistema o aplicación. Puede involucrar el *monitoreo* del consumo de recursos como CPU, memoria y almacenamiento, así como el análisis de *patrones de uso* para optimizar el rendimiento y la eficiencia del sistema.

## User experience (UX)

Se refiere a la experiencia de usuario (UX) que se refiere a la percepción general de una persona al interactuar con un producto o servicio, como una aplicación. Incluye todos los aspectos de la interacción del usuario, desde la navegación hasta el diseño visual y la facilidad de uso, con el objetivo de proporcionar una experiencia satisfactoria y significativa.

## User Interface (UI)

Se refiere a la Interfaz de Usuario. Es el medio a través del cual un usuario interactúa con un dispositivo o aplicación, incluyendo elementos como botones, menús y otros controles visuales. Es decir, la parte visual y tangible de un producto o sistema digital con el que el usuario interactúa.

Definiciones obtenidas de la RAE (Real Academia de la lengua Española) [32], y de las referencias consultadas.



# Anexo

El anexo contiene información complementaria relevante para el desarrollo del proyecto, incluyendo:

- Diagrama UML de clases completo.
- Diagrama de Gantt ampliado.
- Archivo ejecutable de la aplicación para Android y para IOS.
- Guía de usuario.

El enlace de descarga del anexo es:

<https://mega.nz/folder/ljoUnR5J#UAQGR-XRcTtY75fKz8u4w>

# Bibliografía

- [1] Unir, «¿Qué es el pensamiento computacional?,» 27 01 2021. [En línea]. Available: <https://www.unir.net/educacion/revista/pensamiento-computacional/>.
- [2] J. M. Wing, «Computational Thinking,» *COMMUNICATIONS OF THE ACM*, vol. 43, nº 3, p. 33–35, 01 Marzo 2006.
- [3] S. Smith, «micaferin,» [En línea]. Available: <http://micaferin.weebly.com/pensamiento-computacional1.html>.
- [4] M. Zapata Cáceres, «Enseñanza, evaluación y análisis de habilidades de pensamiento computacional en etapas tempranas,» *Tesis Doctoral leída en la Universidad Rey Juan Carlos de Madrid en 2022. Directora de la Tesis: Estefanía Martín Barroso, 2022.*
- [5] V. Gaitán, «Gamificación: el aprendizaje divertido,» 2013. [En línea]. Available: <https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>.
- [6] L. Gilibets, «Qué es la metodología Kanban y cómo utilizarla,» *iebschool*, 12 enero 2023.
- [7] J. Mesh, «Metodología Kanban: revoluciona tu manera de trabajar más ágil,» *Trello*, 29 marzo 2020.
- [8] Lucidchart, «Lucidchart,» [En línea]. Available: <https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-gantt>.
- [9] D. Bonjean, «Digital Education Action Plan-Action 10 Artificial intelligence and analytics,» Sept 2018. [En línea]. Available: <https://education.ec.europa.eu/focus-topics/digital-education/action-plan/action-10>.
- [10] S. Bocconi, A. Chiocciariello, P. Kampylis, V. Dagiené, P. Wastiau, K. Engelhardt, J. Earp, M. Horvath, E. Jasuté, C. Malagoli y V. Masiulionytė-Dagienė, «Reviewing computational thinking in compulsory education: state of play and practices from computing education,» marzo 2022.
- [11] J. Monnet, «Informatics education at school in Europe,» september 2022.

- [12] Unión Europea, «Propuesta de Recomendación del Consejo sobre la mejora de la provisión de capacidades digitales en la educación y la formación (primer trimestre de 2023),» vol. 1, 2021.
- [13] «Code Intef,» [En línea]. Available: <https://code.intef.es/epcia/>.
- [14] Instituto nacional de Tecnologías Educativas y de Formación del Profesorado; Ministerio de Educación y Formación Profesional, «Jornada "Programación y Pensamiento Computacional en la Educación",» *Code Intef*, 28 septiembre 2017.
- [15] fi Group, «fi Group,» 10 07 2020. [En línea]. Available: <https://es.fi-group.com/el-concepto-stem-que-significa/#:~:text=El%20significado%20de%20STEM%20proviene,m%C3%A1s%20relevancia%20en%20el%20mundo.>
- [16] EducarEnAcción, «Enfoque educativo STEM en la LOMLOE,» *EducarEnAcción*, 02 06 2021.
- [17] W. C. R. Arocho, «l legado de Vygotski y de Piaget a la educación,» *Revista latinoamericana de psicología*, vol. 31, nº 3, pp. 477-489, 1999.
- [18] A. J. Ogaz Vasquez y B. I. Sánchez Luján, «Evaluación del nivel de aptitud desarrollo de pensamiento computacional en jóvenes de nivel básico secundaria,» *Revista Electrónica Científica De Investigación Educativa*, vol. 4, nº 2, pp. 1151-1163, 07 enero 2019.
- [19] M. F. Garrido, «Formación basada en las Tecnologías de la Información y Comunicación: Análisis didáctico del proceso de enseñanza-aprendizaje,» *Tesis Doctoral. Universitat Rovira i Virgili*, 18 diciembre 2005.
- [20] L. El-Hamamsy, M. Zapata-Cáceres, E. M. Barroso, F. Mondada, J. D. Zufferey y B. Bruno, «The competent computational thinking test: Development and validation of an unplugged computational thinking test for upper primary school,» *ournal of Educational Computing Research*, vol. 60, nº 7, pp. 1818-1866, 2022.
- [21] D. Norman., «Design principles for human-computer interfaces,» de *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*.

- [22] B. A. Miguelsanz, «Los 10 principios de usabilidad web de Jakob Nielsen,» 29 08 2017. [En línea]. Available: <https://profile.es/blog/los-10-principios-de-usabilidad-web-de-jakob-nielsen/>.
- [23] J. Yablosnski, «Laws of UX,» [En línea]. Available: <https://lawsofux.com/>.
- [24] J. J. J. A. D. Wery, «The effect of a specialized dyslexia font, OpenDyslexic, on reading rate and accuracy,» *Annals of dyslexia*, vol. 67, pp. 114-127, 2017.
- [25] N. a. o. Bru Rol, «Alba: un proyecto de literatura infantil con realidad aumentada integrada,» 2021.
- [26] R. a. o. García, «Estudio comparativo sobre la influencia de los tipos de letra utilizados en el material de lectura en niños con dislexia,» pp. 72-73, 2016.
- [27] F. Diner, «Google Fonts - Jolly Lodger,» [En línea]. Available: <https://fonts.google.com/specimen/Jolly+Lodger/about>.
- [28] I. M. Cruz Lucas, «El dibujo animado como aprendizaje en los niños de infantil y primaria,» 2020.
- [29] J. D. F. R. Godino, «Geometría y su didáctica para maestros,» 2002.
- [30] arsdidas, «FICHAS DIDCTICAS MODELO ACUARELA,» 31 03 2017. [En línea]. Available: [http://www.arsdidas.org/content/campus/archivos/FICHAS\\_DIDCTICAS\\_MODELO\\_ACUARELA.pdf](http://www.arsdidas.org/content/campus/archivos/FICHAS_DIDCTICAS_MODELO_ACUARELA.pdf).
- [31] [En línea]. Available: BOE-A-2022-12066.
- [32] EDUCACIÓN 3.0, «Los mejores videojuegos educativos,» [En línea]. Available: <https://www.educaciontrespuntocero.com/recursos/los-mejores-videojuegos-educativos/>.
- [33] «Real Academia Española,» [En línea]. Available: <https://www.rae.es/>.
- [34] Á. d. F. e. L. y. C. D. E. d. INTEF, «formacion intef,» Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado, [En línea]. Available: <https://formacion.intef.es/mod/book/view.php?id=2412&chapterid=1829>.