

PROTOTIPO DE UN VIDEOJUEGO SERIO SOBRE SISTEMAS FERROVIARIOS EN
COLOMBIA

MIGUEL ÁNGEL PLATA RODRÍGUEZ
MATEO SALAZAR SERRANO

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2026

PROTOTIPO DE UN VIDEOJUEGO SERIO SOBRE SISTEMAS FERROVIARIOS EN
COLOMBIA

MIGUEL ÁNGEL PLATA RODRÍGUEZ
MATEO SALAZAR SERRANO

Trabajo de grado para optar al título de Ingeniero de Sistemas

Director

Urbano Eliécer Gómez Prada
Doctor en Tecnología Educativa

UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA

2026

DEDICATORIA

Dedico principalmente este gran logro en mi vida a mis papás, por toda la paciencia y el apoyo que me brindaron durante mi carrera profesional.

A mis hermanas: Danis, Majo y Gaby, por su compañía y respaldo incondicional desde siempre.

Y a Karencita, mi compañera de vida; gracias por estos más de cinco años caminando juntos, por tu cariño, tu comprensión y tu apoyo constante.

Todo este trabajo es fruto del amor, el acompañamiento y la fuerza que me dieron para seguir avanzando, incluso en mis momentos más difíciles, y para comprender que cada paso y cada conexión que construyo forman parte de lo que soy hoy. ¡Muchas gracias y los quiero mucho!

“¿Qué somos en esta cadena que nunca termina, sino quienes la heredamos y extendemos sin certeza alguna?”

Miguel Ángel Plata Rodríguez

Primero, gracias a mi familia, quienes han sido el pilar fundamental en cada etapa de mi vida. En especial a mis padres, Álvaro Salazar Serrano y Ana Milena Serrano Vesga, por su esfuerzo incansable, su ejemplo de disciplina y su amor incondicional, que me han guiado y motivado a alcanzar este logro académico. A mis hermanos, Daniel, Alejandra y Luciana, por su compañía, apoyo y alegría, que me recordaron siempre la importancia de la unión y la fraternidad, y que con sus palabras y gestos me impulsaron a seguir adelante en los momentos más difíciles. A Stella Mantilla, por su comprensión, respaldo y cariño, que se convirtieron en un soporte invaluable durante este proceso de formación. Cada uno de ustedes ha sido parte esencial no solo en mi desarrollo académico, sino también en mi crecimiento como persona con valores y principios sólidos. Con su amor, paciencia y dedicación, contribuyeron a construir la persona que hoy soy y a darme la fortaleza necesaria para culminar este camino. Este trabajo de grado es también suyo, porque detrás de cada página y cada esfuerzo está el reflejo de su apoyo incondicional y de la confianza que siempre depositaron en mí.

“La diferencia entre el novato y el maestro es que el maestro ha fallado más veces que el novato lo ha intentado”.

– Koro Sensei

Mateo Salazar Serrano

AGRADECIMIENTOS

Queremos expresar nuestro más sincero agradecimiento a todas las personas e instituciones que hicieron posible la realización de este trabajo de grado, el cual representa no solo el cierre de una etapa académica, sino también el fruto de años de esfuerzo, dedicación y aprendizaje.

En primer lugar, agradecemos profundamente a la Universidad Industrial de Santander (UIS), por brindarnos el espacio académico, los recursos y el acompañamiento necesarios durante el desarrollo de nuestra carrera.

A nuestro director, el profesor Urbano Eliécer Gómez Prada, extendemos un reconocimiento especial por su guía, paciencia y atención, y sus reiteradas menciones de su cariño y amor hacia Betulia.

Nuestro agradecimiento también se dirige a la comunidad del software libre, cuyo esfuerzo colectivo nos permitió contar con herramientas indispensables como QGIS y Blender. Sin estas plataformas, el camino habría sido mucho más difícil, y reconocemos el valor de quienes trabajan por mantenerlas accesibles y en constante mejora de manera gratuita.

Finalmente, queremos dar las gracias a todos nuestros testers y amigos, quienes con su apoyo, paciencia y valiosas observaciones contribuyeron significativamente al éxito de este proyecto.

Este trabajo de grado es, en esencia, el resultado de un esfuerzo colectivo. A cada persona que nos brindó su tiempo, conocimiento, compañía o palabras de aliento, les extendemos nuestra gratitud más sincera. Sin ustedes, este logro no habría sido posible.

¡Muchas gracias!

CONTENIDO

	pág.
INTRODUCCIÓN	14
1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA	16
1.1. OBJETIVOS	20
1.1.1. Objetivo General	20
1.1.2. Objetivos Específicos	20
2. MARCO DE REFERENCIA	21
2.1. SISTEMAS FERROVIARIOS EN COLOMBIA	21
2.1.1. Historia	21
2.1.2. Dominio del transporte por carretera	22
2.2. GAME-BASED LEARNING	23
2.3. MOTORES GRÁFICOS Y HERRAMIENTAS DE MODELADO 3D	25
2.3.1. Unity	26
2.3.2. Blender	27
2.4. MODELOS DE DESARROLLO	27
2.4.1. RUP	28
2.4.2. Kanban	30
3. METODOLOGÍA	31
3.1. INICIO	31
3.1.1. Actividades realizadas	32
3.2. ELABORACIÓN	32
3.2.1. Actividades realizadas	32
3.3. IMPLEMENTACIÓN	33
3.4. EVALUACIÓN	34
3.4.1. Actividades realizadas	34

4. RESULTADOS	35
4.1. REQUERIMIENTOS	35
4.1.1. Matriz de Riesgos	35
4.1.2. Definición de Épicas e Historias de Usuario	39
4.2. DISEÑO	45
4.2.1. Arquitectura de Escenas en Unity	45
4.2.2. Diagramas de Proceso	47
4.2.3. Diagrama de Componentes	47
4.2.3.1. Main Menu	51
4.2.3.2. Mapa Menu	51
4.2.3.3. MapaCompleto	55
4.2.3.4. Nivel 1 Tutorial	57
4.2.3.5. Niveles 2 a 5	61
4.2.4. Arquitectura del Sistema y API de Informes	64
4.2.4.1. Tablas de base de datos	65
4.3. IMPLEMENTACIÓN	68
4.3.1. Conción del Espacio de Trabajo en Unity	68
4.3.2. Construcción del Entorno Geográfico	70
4.3.2.1. Automatización y Procesamiento de Datos Geográficos	71
4.3.2.2. Estrategia de Segmentación (Chunks)	72
4.3.2.3. Texturizado	73
4.3.3. Desarrollo del Sistema Ferroviario	75
4.3.3.1. Generación de Vías y Optimización	75
4.3.3.2. Sistemas de Construcción y Estructuras	76
4.3.3.3. Lógica y Datos de los Trenes	77
4.3.4. Zonas de daño y prohibición	77
4.3.5. Managers	79
4.3.5.1. Economía	79
4.3.5.2. Sistema de Guardado	80
4.3.5.3. Nivel	80
4.3.5.4. Telemetría	81

4.3.5.5. Tutorial	82
4.3.6. Menú Principal	84
4.3.7. Menú de Pausa	85
4.3.7.1. Guía de Juego.....	85
4.3.8. Selector de Niveles	86
4.3.9. Contratos.....	86
4.3.10. Puntuación	87
4.3.10.1. Sistema de Estrellas	88
4.3.11. Pruebas y Análisis de Resultados.....	89
4.3.12. Encuesta	90
4.3.13. Análisis de Telemetría (Datos de la API)	92
4.3.13.1. Retención y Progresión: El desafío de la curva de aprendizaje.....	92
4.3.13.2. Balance de Dificultad: Victorias frente a Derrotas.....	93
4.3.14. Resultados de la Encuesta de Percepción	94
4.3.14.1. Usabilidad y Experiencia de Usuario (UX).....	94
4.3.14.2. Concientización y Percepción del Aprendizaje (GBL)	95
4.4. ERRORES DETECTADOS	96
5. CONCLUSIONES.....	98
6. TRABAJO FUTURO.....	100

LISTA DE FIGURAS

	pág.
Figura 1. Movimiento de carga – Modo Carretero.	22
Figura 2. Movimiento de carga – Modo Ferroviario.	22
Figura 3. Tiempo vs recursos utilizados en cada fase en RUP.	29
Figura 4. Escala de evaluación de riesgos utilizada durante el proyecto.	35
Figura 5. Diagrama de la arquitectura basada en escenas aditivas.	46
Figura 6. Diagrama de proceso del menú.	48
Figura 7. Diagrama de proceso de la partida (Figura a).	49
Figura 8. Diagrama de proceso de la partida (Figura b).	50
Figura 9. Diagrama de componentes organizado por escenas de Unity.	52
Figura 10. Diagrama de componentes del Menú Principal	53
Figura 11. Diagrama de componentes del Mapa del Menu	54
Figura 12. Diagrama de componentes del Mapa completo	56
Figura 13. Diagrama de componentes del Nivel 1 Tutorial (Figura a)	58
Figura 14. Diagrama de componentes del Nivel 1 Tutorial (Figura b)	59
Figura 15. Diagrama de componentes de los niveles 2 a 5 (Figura a).	62
Figura 16. Diagrama de componentes de los niveles 2 a 5 (Figura b).	63
Figura 17. Arquitectura del sistema e integración con la API de informes.	65
Figura 18. Colección única de la base de datos FireBase en la API.	66
Figura 19. Interfaz del repositorio Git	68
Figura 20. Interfaz de desarrollo en Unity.	70
Figura 21. Primer intento de generar el terreno usando un modelo 3D en Blender. .	71
Figura 22. Captura del proyecto en QGIS utilizado para el terreno.	72
Figura 23. Fragmento del script encargado de la importación de terrenos.	73
Figura 24. Captura de la clase <i>HeighBandsPainter</i> desde el inspector.	74
Figura 25. Implementación del entorno utilizando el sistema de Terrenos.	75
Figura 26. Modelo de un puente.	76
Figura 27. Modelo de un túnel.	76

Figura 28. Modelado de locomotoras.	77
Figura 29. Sistema de partículas de daño.	77
Figura 30. Ejemplo de una zona de daño en el terreno.	78
Figura 31. Interfaz de retroalimentación tras un evento de derrota.	81
Figura 32. Personaje guía <i>Don Rail</i>	83
Figura 33. Interfaz del Menú Principal.	84
Figura 34. Interfaz del Menú de Pausa superpuesta a la simulación.	85
Figura 35. Panel de la Guía detallando controles, mecánicas y recomendaciones. .	86
Figura 36. Escena de selección de niveles sobre el mapa geográfico de Colombia.	86
Figura 37. Ficha de contrato que establece los objetivos y el presupuesto del nivel.	87
Figura 38. Pantalla de victoria con la calificación obtenida según los cuatro criterios.	89
Figura 39. Formulario de evaluación de la experiencia de usuario y aprendizaje. . .	90
Figura 40. Tasa de retención de jugadores únicos por nivel.	93
Figura 41. Proporción de victorias frente a derrotas por nivel.	94
Figura 42. Nivel de facilidad percibida en los controles.	94
Figura 43. Respuestas a la pregunta 4	95
Figura 44. Percepción de la complejidad constructiva tras la experiencia de juego.	95
Figura 45. Respuestas a la pregunta 29	96
Figura 46. Reflexión final sobre los desafíos del desarrollo ferroviario en Colombia.	96

LISTA DE CUADROS

	pág.
Cuadro 1. Tablero Kanban propuesto para la fase de Construcción.....	33
Cuadro 2. Matriz de riesgos del proyecto.	37
Cuadro 3. Listado de Épicas del proyecto.....	39
Cuadro 4. Tabla final de historias de usuario implementadas.	41

RESUMEN

TÍTULO: PROTOTIPO DE UN VIDEOJUEGO SERIO SOBRE SISTEMAS FERROVIARIOS EN COLOMBIA*

AUTOR: MIGUEL ÁNGEL PLATA RODRÍGUEZ, MATEO SALAZAR SERRANO**

PALABRAS CLAVE: VIDEOJUEGO SERIO, UNITY, GBL, FERROCARRILES, COLOMBIA, SIMULACIÓN

DESCRIPCIÓN: La infraestructura ferroviaria en Colombia presenta un estancamiento crítico con más del 60 % de su red inactiva, lo que motiva la creación de herramientas pedagógicas que fomenten la valoración social del sistema. Mediante el motor gráfico Unity 6, se desarrolló un prototipo de videojuego serio fundamentado en el marco de Game-Based Learning (GBL) para concientizar sobre los desafíos técnicos y económicos del sector. La implementación integró los seis criterios esenciales del GBL: la inmersión mediante un entorno geográfico basado en datos reales; la interacción y el control del aprendiz a través de libertad de control del jugador para dibujar vías en un terreno abierto; el apoyo al aprendizaje mediante mentoría a través de calificaciones; una narrativa orientada al cumplimiento de contratos estatales con un personaje llamado Don Raíl; y una evaluación técnica gestionada por una API de telemetría. Los resultados de la prueba piloto evidenciaron una concientización de la problemática ferroviaria Colombiana, logrando que el 81,8 % de los participantes identificara la dificultad en la infraestructura férrea como una tarea compleja y costosa debido al impacto de la topografía y el uso necesario de túneles y puentes. Adicionalmente, los datos de telemetría validaron la representación de la dificultad real de la ingeniería ferroviaria nacional. Así, el videojuego logra trascender el entretenimiento al transformar la percepción del usuario hacia una visión técnica y estratégica, cumpliendo con el objetivo de favorecer el aprendizaje sobre los retos estructurales del transporte férreo en el país.

*Trabajo de grado.

**Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática. Ingeniería de Sistemas. Director: Urbano Eliécer Gómez Prada. Doctor en Tecnología Educativa.

ABSTRACT

TITLE: PROTOTYPE OF A SERIOUS VIDEO GAME ABOUT RAILWAY SYSTEMS IN COLOMBIA*

AUTHOR: MIGUEL ÁNGEL PLATA RODRÍGUEZ, MATEO SALAZAR SERRANO**

KEY WORDS: SERIOUS GAME, UNITY, GBL, RAILWAYS, COLOMBIA, SIMULATION

DESCRIPTION: Colombian railway infrastructure presents a critical stagnation with over 60 % of its network inactive, which motivates the creation of pedagogical tools that foster the social valuation of the system. Using the Unity 6 game engine, a serious game prototype was developed grounded in the Game-Based Learning (GBL) framework to raise awareness about the technical and economic challenges of the sector. The implementation integrated the six essential criteria of GBL: immersion through a geographic environment based on real data; interaction and learner control through the player's freedom to draw tracks on open terrain; learning support through mentorship via ratings; a narrative oriented towards the fulfillment of state contracts with a character named Don Raíl; and a technical assessment managed by a telemetry API. The results of the pilot test evidenced an awareness of the Colombian railway issues, with 81.8 % of participants identifying the intricacy of rail infrastructure as a complex and costly task due to the impact of topography and the necessary use of tunnels and bridges. Additionally, telemetry data validated the representation of the real difficulty of national railway engineering. Thus, the video game succeeds in transcending entertainment by transforming the user's perception towards a technical and strategic vision, fulfilling the objective of promoting learning about the structural challenges of rail transport in the country.

*Degree work.

**Faculty of Physical-Mechanical Engineering. School of Systems Engineering and Informatics. Systems Engineering. Director: Urbano Eliécer Gómez Prada. PhD in Educational Technology.

INTRODUCCIÓN

El transporte ferroviario ha desempeñado un papel fundamental en el desarrollo económico y logístico de los países industrializados. Colombia, a pesar de haber sido pionera en la introducción del ferrocarril en América Latina, ha experimentado un estancamiento significativo durante las últimas décadas, producto de la priorización de la infraestructura vial o de debilidades institucionales y técnicas en el sector. Según datos del Ministerio de Transporte, más del 60 % de la red ferroviaria nacional se encuentra inactiva, y el modo férreo apenas representa el 0,1 % de la carga nacional movilizada³.

Frente a este escenario, y en un momento de renovado interés evidenciado por proyectos como el corredor de La Dorada–Chiriguaná⁴ y el Regiotram de Bogotá⁵, el país se enfrenta no solo a retos técnicos y económicos, sino también a un desafío cultural y educativo: la limitada comprensión y valoración social del sistema ferroviario.

La falta de conocimiento público sobre su funcionamiento, beneficios y complejidades ha contribuido históricamente a su baja priorización dentro de las políticas nacionales de transporte.

En este contexto, surge la necesidad de desarrollar herramientas innovadoras que no solo transmitan información, sino que logren despertar interés y promover la comprensión profunda de los desafíos que implica el desarrollo ferroviario. Una alternativa prometedora para lograrlo es la aplicación de tecnologías interactivas, en particular los videojuegos educativos, que combinan elementos lúdicos y pedagógicos para facilitar el aprendizaje

³Ministerio de Transporte. *Transporte en Cifras*. 2023. URL: <https://mintransporte.shinyapps.io/transporteencifrasapp/> (visitado 06-06-2025).

⁴Ministerio de Transporte. *Gobierno Nacional firma contrato de concesión de la primera APP férrea de Colombia en el corredor La Dorada–Chiriguaná*. Mayo de 2025. URL: <https://mintransporte.gov.co/publicaciones/12015/gobierno-nacional-firma-contrato-de-concesion-de-la-primera-app-ferrea-de-colombia-en-el-corredor-la-dorada-chiriguana/> (visitado 06-06-2025).

⁵Alcaldía Mayor de Bogotá. *En abril inician las obras del corredor férreo de Regiotram de Occidente*. 2025. URL: <https://bogota.gov.co/mi-ciudad/movilidad/obras-del-regiotram-de-occidente-que-beneficia-bogota-inician-abril> (visitado 06-06-2025).

experiencial.

Dentro de esta categoría, los *videojuegos serios*—desarrollados con fines que trascienden el entretenimiento—⁶, se presentan como un enfoque idóneo para representar de manera didáctica y accesible problemas complejos del mundo real, como los que enfrenta la infraestructura férrea colombiana.

Bajo este paradigma, el Game-Based Learning (GBL) se postula como una guía de referencia para analizar la educatividad de un videojuego, tomando como base las características de diseño identificadas por Nadolny⁷ en su marco de análisis.

Este trabajo de grado tuvo como objetivo el desarrollo de un prototipo de videojuego serio orientado a la comprensión de los desafíos asociados al desarrollo ferroviario en Colombia. Implementado mediante el motor gráfico Unity 6 y el software de modelado Blender, y haciendo uso de la metodología RUP y Kanban, se buscó crear una experiencia educativa que incorporara los criterios esenciales del GBL: inmersión, interacción, control del aprendiz, apoyo al aprendizaje, narrativa y evaluación⁸.

⁶Ute Ritterfeld, Michael Cody y Peter Vorderer. *Serious Games: Mechanisms and Effects*. Taylor & Francis, 2009.

⁷L. Nadolny et al. «Examining the characteristics of game-based learning: A content analysis and design framework». En: *Computers & Education* 156 (2020). DOI: 10.1016/j.compedu.2020.103936.

⁸Ibíd.

1. PLANTEAMIENTO Y JUSTIFICACIÓN DEL PROBLEMA

Colombia, a pesar de haber sido pionera en la introducción del ferrocarril en Latinoamérica, se ha visto estancada en su desarrollo ferroviario. Tras una búsqueda en la plataforma web de la Agencia Nacional de Infraestructuras sobre los proyectos activos⁹, de los 133 proyectos activos, sólo 4 son proyectos ferroviarios, de los cuales 2 llevan desde 1998 y 1999 en operación. Además, según datos del Ministerio de transporte de Colombia¹⁰, el 63,2 % de la red ferroviaria nacional se encuentra inactiva y, excluyendo el transporte de carbón y petróleo, el modo ferroviario apenas representa el 0,1 % de la carga nacional movilizada, lo cual evidencia una falta de interés e iniciativa por parte de los gobiernos que, a su vez, ha derivado en una priorización de la inversión pública en la infraestructura de carreteras¹¹.

Según la monografía *Desafíos del transporte ferroviario de carga en Colombia*, la dificultad en desarrollar efectivamente un transporte ferroviario en Colombia es debido a factores como los conflictos políticos, el manejo de los recursos y de la debilidad ingenieril del sector. Específicamente se menciona: “La institucionalidad del sector presenta hoy debilidades en materia de planeación y política ferroviaria; ingeniería ferroviaria conceptual...”¹².

Actualmente, hay que considerar que recientemente la nación está intentando reiniciar el desarrollo de infraestructuras ferroviarias. Algunos de los proyectos que evidencian este reciente interés son: la firma del contrato de concesión, la primera Asociación Público-Privada (APP) del país, correspondiente al corredor de La Dorada–Chiriguaná¹³; el inicio

⁹Agencia Nacional de Infraestructura. *Ferrocarriles - fichas de proyecto (ANISCOPIO)*. 2025. URL: <https://aniscopio.ani.gov.co/ferrocarriles-public/ficha-proyecto-drupal> (visitado 01-03-2025).

¹⁰Ministerio de Transporte. *Proyecto de Decreto/Resolución: Asignación de surcos ferroviarios en la Red Férrea Nacional*. 15 de mayo de 2024. URL: <https://mintransporte.gov.co/loader.php?IServicio=Tools2&ITipo=descargas&IFuncion=descargar&idFile=47723> (visitado 06-06-2025).

¹¹Íd., *Transporte en Cifras*, óp.cit.

¹²Juan Champin et al. *Desafíos del transporte ferroviario de carga en Colombia*. Inter-American Development Bank, abr. de 2016. URL: <https://publications.iadb.org/publications/spanish/document/Desaf%C3%ADos-del-transporte-ferroviario-de-carga-en-Colombia.pdf>, p. 17.

¹³Ministerio de Transporte, *Gobierno Nacional firma contrato de concesión de la primera APP férrea de Colombia en el corredor La Dorada–Chiriguaná*, óp.cit.

de las obras del Regiotram de Occidente, que beneficiará a Bogotá y a sus municipios aledaños¹⁴; y el megaproyecto del Tren de Cercanías del Valle del Cauca¹⁵, que busca conectar Cali con Jamundí, Yumbo y Palmira.

Aún así, ante la carencia de herramientas que refuercen la planeación y la ingeniería ferroviaria, resulta oportuno desarrollar recursos que impulsen el aprendizaje y la concientización sobre los desafíos de este sector.

Los videojuegos en su mayoría se enfocan en el entretenimiento del jugador, sin embargo, existe la posibilidad de diseñar juegos educativos, tal es el caso de un estudio descrito en el artículo “Examining the characteristics of game-based learning: A content analysis and design framework”¹⁶ en donde, a partir del análisis y la codificación de 194 artículos diferentes sobre juegos educativos, se definió un marco de diseño de un *Game-Based Learning* (GBL), o aprendizaje basado en un juego.

Un GBL se compone por dos grupos de características, las primarias y secundarias, siendo las primarias los aspectos que se consideran esenciales para un juego educativo¹⁷: Criterios de inmersión, interacción, control del aprendiz, apoyo al aprendizaje, narrativa y evaluación.

Además, según el libro *Serious Games: Mechanisms and Effects*, se define un videojuego serio como “cualquier forma de software de juego interactivo basado en computadora (...) y que se ha desarrollado con la intención de ser más que entretenimiento.”¹⁸. Aspecto que aplica para el caso de estudio de este proyecto, ya que se planteó, además de entretener, educar sobre los desafíos del desarrollo ferroviario en Colombia.

¹⁴Alcaldía Mayor de Bogotá, *En abril inician las obras del corredor férreo de Regiotram de Occidente*, óp.cit.

¹⁵Valora Analitik. *Así es el megaproyecto del Tren de Cercanías del Valle*. 23 de oct. de 2024. URL: <https://www.valoraanalitik.com/megaproyecto-tren-de-cercanias-del-valle-asi-es/> (visitado 06-06-2025).

¹⁶Nadolny et al., «Examining the characteristics of game-based learning: A content analysis and design framework», óp.cit.

¹⁷Ibíd.

¹⁸Ritterfeld, Cody y Vorderer, *Serious Games: Mechanisms and Effects*, óp.cit., p. 6.

Al respecto, existen diversos antecedentes de videojuegos de simulación enfocados en diseñar y construir sistemas férreos, como son el *Railway Empire*¹⁹ ambientado en Estados Unidos entre siglo XIX y XX, *Railroad Tycoon 3*²⁰ basado en escenarios para recrear sistemas férreos importantes en la historia de la humanidad y *Train Valley 2*²¹ enfocado en la resolución de puzzles; sin embargo, ninguno de los videojuegos mencionados tiene propósitos más allá del entretenimiento, objetivo que sí se buscó en este proyecto de grado.

Tras una búsqueda con el motor Google Academics y usando la ecuación: (“Colombia” AND “Sistema Férreo” AND (“Videojuego Serio” OR “Videojuego de simulación” OR “Videojuego educativo”)), se encontró solamente una tesis de grado que tenía como objetivo “Investigar sobre interfaces hápticas para aplicaciones 3D, así como analizar su interactividad y usabilidad en una aplicación interactiva para la maqueta de transportes ESPOCH.”²², en el que no es un aspecto suficiente para que sea considerado el desarrollo de un videojuego serio, ya que en ningún momento describe el proceso de desarrollo de un videojuego, solo se menciona como un ejemplo. En ese orden de ideas, se puede concluir que no se encontró ningún videojuego serio de sistemas ferroviarios ambientado en Colombia.

Por lo anterior, se formuló la siguiente pregunta de investigación:

¿Cuál debería ser la estructura de un prototipo de videojuego serio sobre los desafíos del desarrollo del sistema ferroviario en Colombia y cómo se implementaría para que tenga en cuenta los criterios de inmersión, interacción, control del aprendiz, apoyo al aprendizaje, narrativa y evaluación del marco GBL?

¹⁹Gaming Minds Studios. *Railway Empire*. 2018. URL: https://store.steampowered.com/app/503940/Railway_Empire/ (visitado 06-06-2025).

²⁰PopTop Software. *Railroad Tycoon 3*. 2003. URL: https://www.gog.com/en/game/railroad_tycoon_3 (visitado 06-06-2025).

²¹Alexey Davydov et al. *Train Valley 2*. 2019. URL: https://store.steampowered.com/app/602320/Train_Valley_2/ (visitado 06-06-2025).

²²Luis F. Arévalo Caicho y M. A. Urgiléz Zabala. «Análisis de usabilidad e interactividad en interfaces hápticas para sistemas 3D, aplicación a maqueta interactiva de transporte ESPOCH». Tesis de Grado. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo, 2014.

En el orden de ideas de lo manifestado, se puede considerar viable una iniciativa en el desarrollo de un prototipo de software para la educación del sector ferroviario, por lo tanto, se propuso implementar un primer prototipo de un videojuego serio de sistemas ferroviarios en Colombia, con el fin de concientizar la comprensión de algunos de los desafíos que conllevaría la planeación y el desarrollo de la infraestructura férrea en el país.

Se buscó que el prototipo del videojuego serio sobre los desafíos del sistema ferroviario en Colombia incorporara las características primarias del marco GBL como la evaluación (cómo se mide el aprendizaje), la inmersión (cuán envolvente es el entorno del juego), la interacción (cómo se comunica el jugador con el juego o entre jugadores), el control del aprendiz (el grado de autonomía que tiene el usuario), el apoyo al aprendizaje (tutoriales y ayudas) y la narrativa (la historia o contexto que da sentido a la experiencia). Estas características se usaron para “ayudar a trazar rutas de aprendizaje más personalizadas, motivadoras y potencialmente efectivas para el aprendizaje.”²³.

El prototipo tuvo una prueba piloto dirigida a estudiantes UIS, con el objetivo de evaluar la experiencia del jugador y el desempeño del software. La prueba se orientó a determinar si el prototipo resultaba agradable en su uso y se encontraba libre de errores, permitiendo identificar áreas de mejora para quienes deseen dar continuidad a este proyecto.

²³Jan L. Plass, Richard E. Mayer y Bruce D. Homer. *Handbook of Game-Based Learning*. MIT Press, 2019, p. 10.

1.1. OBJETIVOS

1.1.1. Objetivo General. Desarrollar un prototipo de un videojuego serio para la comprensión de algunos de los desafíos que conllevaría el mejoramiento de la infraestructura férrea en Colombia, usando GBL, Unity 6, Blender y RUP adaptada con elementos de Kanban.

1.1.2. Objetivos Específicos.

1. Definir las historias de usuario del videojuego serio a partir del marco GBL para el apoyo de la comprensión de algunos de los desafíos del desarrollo férreo.
2. Diseñar los componentes que integran el videojuego serio tales como los sistemas de evaluación, inmersión, interacción y control del jugador a partir del contexto de sistemas ferroviarios en Colombia.
3. Implementar el videojuego serio integrando un sistema interactivo de un mapa 3D, así como eventos ambientales, económicos y sociales usando Unity 6 y Blender.
4. Evaluar el videojuego serio realizando pruebas de funcionalidad y usabilidad.

2. MARCO DE REFERENCIA

2.1. SISTEMAS FERROVIARIOS EN COLOMBIA

Aunque Colombia fue pionera en introducir el sistema ferroviario en Latinoamérica, el desarrollo de este presentó diversos declives, abandonos y una mala gestión por parte del estado. En consecuencia, el modo carretero fue ganando poder consolidándose como la principal forma de transporte tanto de carga como de pasajeros en Colombia²⁴. Para entender los desafíos actuales del sistema ferroviario colombiano, es importante analizar su historia, su avance y los retos que hoy en día enfrenta.

2.1.1. Historia. Históricamente, los ferrocarriles en Colombia surgieron a finales del siglo XIX, principalmente como concesiones privadas. En 1954, se intentó unificar las distintas líneas existentes bajo una sola entidad estatal, los Ferrocarriles Nacionales de Colombia (FNC). La red alcanzó su máxima extensión en 1961 con 3,431 km. Sin embargo, el desarrollo de las carreteras y una gestión considerada débil llevaron al abandono progresivo de muchos trazados. Tras la liquidación de la FNC en 1991 y un intento posterior con Ferrovías que tampoco cumplió las expectativas, se optó por concesionar los corredores con mayor potencial: la Red Férrea del Atlántico (1999) y la Red Férrea del Pacífico (1998)²⁵.

Según la monografía *Desafíos del transporte ferroviario de carga en Colombia*²⁶, los ferrocarriles en Colombia se introdujeron a finales del siglo XIX tras acuerdos con concesiones privadas, pero en 1954, como un intento de unificar todas las líneas ferroviarias de trocha angosta (914 mm) en una sola entidad estatal, se fundó los Ferrocarriles Nacionales de Colombia (FNC). Este fue el primero de muchos intentos que, durante 50 años, ninguno llegó a cumplir las expectativas.

²⁴Champin et al., *Desafíos del transporte ferroviario de carga en Colombia*, óp.cit.

²⁵Ibíd.

²⁶Ibíd.

2.1.2. Dominio del transporte por carretera. Hoy en día, la red ferroviaria nacional cuenta con una longitud total estimada en 3,533 km²⁷ y evidencia una importante subutilización, ya que se reveló en un informe de una ponencia para el segundo debate del proyecto de ley número 337 del 2023²⁸ en donde dice que el 63 % (2.262 km) de las vías del país están inactivos. Así mismo, se muestra que en los últimos 10 años el transporte de carga por carretera pasó de un 70.8% en el 2011 a un 85.5% en el 2021. En las figuras 1 y 2 se muestran datos directos del Ministerio de Transporte actualizado al 2023 sobre el transporte de carga en modo carretero vs modo férreo:

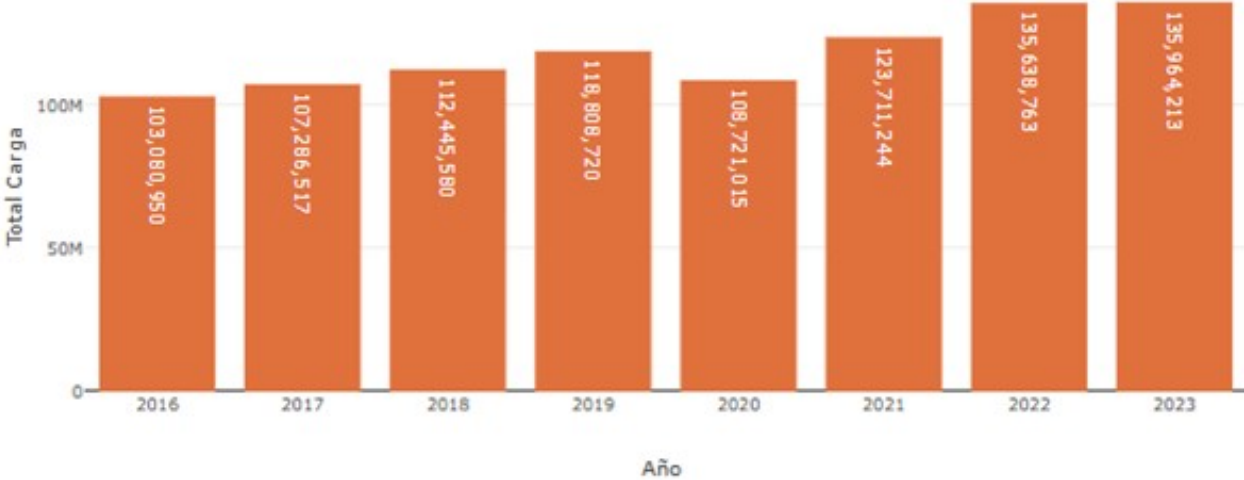


Figura 1: Movimiento de carga – Modo Carretero.

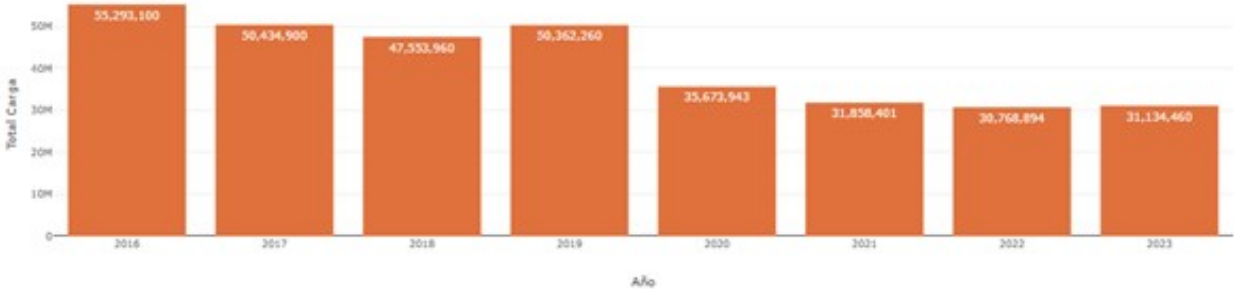


Figura 2: Movimiento de carga – Modo Ferroviario.

Así mismo, el Ministerio de Transporte revela que el transporte de pasajeros en modo

²⁷D. F. Caicedo Navas et al. *Informe de ponencia para segundo debate del Proyecto de Ley número 337 de 2023 Cámara...* 30 de abr. de 2024.

²⁸Ibíd.

férreo es muy despreciable en comparación con el transporte por carretera, por ejemplo, en el 2023 se movilizaron casi 90 millones de pasajeros por carretera versus los casi 500 mil en tren²⁹.

2.2. GAME-BASED LEARNING

El aprendizaje basado en juegos, o Game-Based Learning (GBL) por sus siglas en inglés, es una metodología educativa que hace uso de juegos, tanto digitales como analógicos, como herramienta principal para la enseñanza y el aprendizaje. Dichos juegos están diseñados con el objetivo de mejorar la motivación, la participación y el aprendizaje de quienes buscan aprender un tema específico³⁰. Cabe señalar que el GBL no es lo mismo que la gamificación, ya que este introduce elementos de juegos en contextos donde normalmente estos no son la norma. Por otro lado, el GBL utiliza juegos específicamente creados con el objetivo de facilitar el aprendizaje de contenidos y habilidades.

Múltiples estudios sugieren que los juegos, en sus diversas formas, pueden motivar e interesar a los estudiantes, aumentar la retención del material y mejorar las habilidades de razonamiento y el pensamiento complejo³¹. Además, el GBL ofrece un entorno de aprendizaje donde los errores no tienen consecuencias reales, lo que permite a los estudiantes experimentar, explorar y aprender de sus fallos, y hay que entender que, dentro del contexto de los videojuegos, el fracaso funciona como una forma de retroalimentación, donde el fallo no es el final, sino una enseñanza de los errores cometidos.

Diseñar un GBL requiere la integración de características tales como: Criterios de inmersión, interacción, control del aprendiz, apoyo al aprendizaje, narrativa y evaluación³²:

²⁹Ministerio de Transporte, *Transporte en Cifras*, óp.cit.

³⁰Plass, Mayer y Homer, *Handbook of Game-Based Learning*, óp.cit.

³¹Jan G. Hogle. *Considering Games as Cognitive Tools: In Search of Effective "Edutainment"*. University of Georgia, 1996.

³²Nadolny et al., «Examining the characteristics of game-based learning: A content analysis and design framework», óp.cit.

1. Criterios de inmersión: Estos criterios comprenden el diseño tanto visual como sonoro para ambientar al jugador; el diseño visual incluye elementos como la apariencia general del juego y sus personajes, así como la forma de representación de la información clave; el diseño sonoro proporciona sonidos de fondo que se utilizan a menudo para dirigir la atención del jugador a eventos o momentos importantes del juego, señalar la presencia de peligro u oportunidad, inducir emociones positivas o negativas, y reconocer el éxito o el fracaso de una tarea específica.
2. Criterios de interacción: Los criterios de interacción son aquellos que se encargan de permitir al jugador interactuar con el entorno del juego, permitiéndole tomar decisiones y realizar acciones.
3. Control del aprendiz: El control del aprendiz se trata del nivel de autonomía que se le da al jugador sobre cómo quiere afrontar los desafíos que le presenta el juego, un ejemplo de esto es la libertad de elección en juegos de toma de decisiones, donde la elección del jugador afecta la narrativa y desarrollo de la historia.
4. Apoyo del aprendizaje: Se trata de los sistemas y herramientas que tiene el juego para guiar y reforzar el proceso de aprendizaje del jugador, ya sea con tutoriales, pistas, o recursos adicionales y explicaciones.
5. Narrativa: La narrativa en un juego es la historia que se cuenta mientras el jugador participa. Esta historia puede incluir personajes, misiones, problemas por resolver y un contexto que da sentido a lo que se está haciendo. Una narrativa bien construida motiva al jugador a seguir jugando; una buena historia puede generar sentimientos o emociones, además de poder guiar a los jugadores en su toma de decisiones.
6. Evaluación: La evaluación en los juegos permite registrar el progreso del jugador, medir su desempeño e identificar errores durante el desarrollo de la partida.

2.3. MOTORES GRÁFICOS Y HERRAMIENTAS DE MODELADO 3D

Un motor gráfico es un sistema diseñado para el desarrollo de videojuegos y aplicaciones interactivas que requieren representación visual. Cuenta con herramientas para la creación y gestión de todas las características de un videojuego, como lo son la renderización 2D y 3D, motores de físicas, sistemas de sonido, soporte para compilación de código, entre otros.

Algunos ejemplos bastante conocidos de motores de videojuegos gratuitos son el Unreal Engine³³ y Unity³⁴, los cuales permiten que personas de todo el mundo puedan desarrollar sus propios proyectos digitales.

Unreal Engine “permite crear juegos y experiencias con un alto nivel de detalle geométrico gracias al sistema de geometría de micro poligonal virtualizada de Nanite y a los mapas de sombras virtuales”³⁵ y ha sido utilizado en producciones de alto nivel, no solo en la industria de los videojuegos, sino también en el cine y la arquitectura, un ejemplo de esto es en la serie de *The Mandalorian* donde “parte de esta serie fue rodada en un set de producción virtual con tecnología Unreal Engine”³⁶ gracias a su capacidad para generar entornos visuales realistas. Unity, por otro lado, cuenta con una amplia comunidad de usuarios, una amplia accesibilidad, y tiene la capacidad de exportar proyectos a más de 25 plataformas diferentes³⁷. Además, es especialmente popular en el desarrollo de juegos móviles, simulaciones y experiencias en realidad aumentada (AR) y realidad virtual (VR), siendo utilizado en proyectos como *Hollow Knight*, *Escape From Tarkov*, *Genshin Impact* y diversas aplicaciones educativas y médicas como *Virtual Reality Surgical Simulation Suite*

³³Epic Games. *Unreal Engine*. 2025. URL: <https://www.unrealengine.com/es-ES> (visitado 06-06-2025).

³⁴Unity Technologies. *Unity*. 2025. URL: <https://unity.com> (visitado 06-06-2025).

³⁵Epic Games. *Unreal Engine Features (Nanite, Virtual Shadow Maps)*. 2025. URL: <https://www.unrealengine.com/es-ES/features> (visitado 06-06-2025).

³⁶Meristation. *Cómo Unreal Engine se usa en la industria del cine (The Mandalorian)*. 19 de jun. de 2022. URL: https://as.com/meristation/2022/06/19/reportajes/1655613468_465907.html (visitado 06-06-2025).

³⁷Unity Technologies, *Unity*, óp.cit.

(VR3S)³⁸.

Por otro lado, una herramienta de modelado es un software diseñado para crear, manipular y modificar modelos tanto bidimensionales (2D) como tridimensionales (3D) de objetos y escenas. Entre las herramientas de modelado 3D más populares se encuentran Blender³⁹, el cual es un software gratuito y de código abierto; Autodesk Maya⁴⁰, utilizado mayormente en la industria de los efectos especiales, el cine y la animación; y 3ds Max⁴¹, usado mayormente en diseño arquitectónico y visualización de productos.

2.3.1. Unity. Unity es una plataforma utilizada para “crear y hacer crecer juegos y experiencias interactivas en todas las principales plataformas, desde móviles, PC y consolas, hasta realidad extendida (XR)”⁴², haciendo uso de los lenguajes de programación C# y .NET.

Esto es posible ya que Unity es un motor multiplataforma que permite desarrollar aplicaciones para múltiples dispositivos simultáneamente, sin la necesidad de hacer grandes modificaciones en el código. Cuenta también con una comunidad de usuarios bastante activa donde los usuarios comparten conocimientos, recursos y activos en múltiples redes sociales, así como también en sus foros oficiales. Ciertamente, Unity cuenta con su propia tienda en línea, la Unity Asset Store, donde los desarrolladores pueden comprar y vender activos y herramientas. Entre estos activos se encuentran modelos 3D, plantillas para la creación de juegos, herramientas de desarrollo, entre otros elementos útiles.

Además, cuenta con múltiples licencias de uso, cuyo precio varía dependiendo de necesidades de los estudios, sin embargo, también cuenta con licencias gratuitas para uso

³⁸Unity Technologies. *Virtual Reality Surgical Simulation Suite (VR3S)*. 9 de ene. de 2024. URL: <https://unity.com/resources/virtual-reality-surgical-simulation-suite> (visitado 06-06-2025).

³⁹Blender Foundation. *Blender*. 2025. URL: <https://www.blender.org> (visitado 06-06-2025).

⁴⁰Autodesk. *Autodesk Maya*. 2025. URL: <https://www.autodesk.com/mx/products/maya/overview> (visitado 06-06-2025).

⁴¹Autodesk. *Autodesk 3ds Max*. 2025. URL: <https://www.autodesk.com/latam/products/3ds-max/overview> (visitado 06-06-2025).

⁴²Unity Technologies. *Nuestra compañía*. 2025. URL: <https://unity.com/es/our-company> (visitado 06-06-2025).

personal, estudiantes, y pequeñas organizaciones con ingresos y fondos recaudados en los últimos 12 meses menor a \$200K USD⁴³.

2.3.2. Blender. Blender es una herramienta de modelado de código abierto y gratuito usado para la creación de gráficos y animaciones en 2D y 3D, la cual “cuenta con una amplia variedad de herramientas que lo hacen adecuado para casi cualquier tipo de producción de medios. Las personas y estudios de todo el mundo lo utilizan para proyectos de hobby, comerciales y películas”⁴⁴.

Blender puede ser utilizado para múltiples propósitos, ya sea el modelado de objetos, la animación mediante sus herramientas de *rigging* y líneas de tiempo, el texturizado, el renderizado gracias a los sistemas Eevee y Cycles, la edición de video, la composición digital o incluso los efectos visuales (VFX).

Además, Blender cuenta con la Licencia Pública General GNU (GPL)⁴⁵, lo que permite que el público pueda descargar, modificar y distribuir el programa con total libertad cuantas veces necesite.

2.4. MODELOS DE DESARROLLO

En la ingeniería de software existen diferentes metodologías que fueron diseñadas con el objetivo de establecer un marco de trabajo a la hora de construir un proyecto. En 1970, el ingeniero de software Winston W. Royce estableció por primera vez el modelo en cascada⁴⁶ o también conocido como modelo tradicional, en él se propone un enfoque secuencial en donde cada fase depende del anterior, comenzando desde la fase de análisis hasta ter-

⁴³Unity Technologies. *Planes y precios de Unity*. 2025. URL: <https://unity.com/es/products> (visitado 06-06-2025).

⁴⁴Blender. *Manual de introducción de Blender (v2.82): Sobre Blender*. 14 de feb. de 2020. URL: https://docs.blender.org/manual/es/2.82/getting_started/about/introduction.html (visitado 06-06-2025).

⁴⁵Blender Foundation. *Acerca de Blender (licencia GPL)*. 2025. URL: <https://www.blender.org/about/> (visitado 06-06-2025).

⁴⁶Z. Cataldi et al. *Ingeniería de Software Educativo*. 1999.

minar con la fase de pruebas⁴⁷. No obstante, este modelo es poco utilizado en la actualidad debido a su falta de realismo; plantea un desarrollo de software de forma estrictamente lineal, sin contemplar ajustes ni retroalimentación por parte del cliente durante el proceso.

En consecuencia, en el 2001 un grupo de 17 ingenieros escribieron y firmaron el “Manifiesto por el Desarrollo Ágil de Software”⁴⁸, donde se encuentran los 12 principios que se deben cumplir a la hora de construir software de manera ágil. En resumen, la agilidad en la que se refiere el manifiesto es la capacidad del equipo de adaptarse a cambios, incluso si estos son pedidos en las últimas etapas del desarrollo, ya que el objetivo principal es realizar varias entregas constantes en ciclos de trabajo definidos con una retroalimentación para satisfacer lo más posible al cliente.

2.4.1. RUP. El Proceso Racional Unificado o RUP (por sus siglas en inglés de Rational Unified Process), es una metodología ágil de desarrollo de software creado por la empresa Rational, que más adelante fue adquirido por IBM⁴⁹. RUP no fue descrita como una metodología con pasos estrictamente detallados, sino más bien, como un conjunto de declaraciones sobre lo que se debería reflejar durante el desarrollo del software, esto con el fin de que pueda ser flexible y adaptable según las necesidades del proyecto.

RUP, en realidad, es una extensión de una anterior metodología llamada UP, la cual propuso 4 fases:⁵⁰

1. Inicio: Es una fase corta donde se desarrolla la idea inicial del proyecto; se identifica el modelo de negocio y sus riesgos, y se provee un boceto de la estructura a partir de unos objetivos generales. Es muy importante que los requerimientos que se detallen

⁴⁷Roger S. Pressman. *Ingeniería del Software: Un Enfoque Práctico*. 5a ed. adaptada. Madrid: McGraw-Hill España, 2001.

⁴⁸Kent Beck et al. *Manifiesto por el Desarrollo Ágil de Software*. 2001. URL: <https://agilemanifesto.org/iso/es/manifiesto.html> (visitado 06-06-2025).

⁴⁹eWeek. *IBM Acquires Rational*. 6 de dic. de 2002. URL: <https://www.eweek.com/pc-hardware/ibm-acquires-rational/> (visitado 06-06-2025).

⁵⁰David J. Anderson. *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall Professional, 2003.

aquí no deben restringir la flexibilidad o soluciones que el equipo de trabajo podría encontrar después.

2. **Elaboración:** En esta fase se comienza a detallar los requerimientos, historias de usuario, casos de uso, entre otros diagramas según las necesidades del proyecto. Se sigue teniendo en cuenta la flexibilidad del equipo para mitigar los mayores riesgos. Normalmente, durante el desarrollo de esta fase se requieren más iteraciones que el anterior.
3. **Construcción:** Es la fase más larga del proyecto, ya que es donde se empieza a programar el software; se escribe, prueba y depura el código. En cada iteración se debe cumplir con algún entregable funcional y normalmente se prioriza las funciones más importantes del software.
4. **Transición:** En esta última fase, sucede la entrega del proyecto al cliente y se empieza el mantenimiento a largo plazo, esto también incluye documentación de usuario (manual de uso), preparación del entorno o incluso entrenamiento del usuario. Es muy probable que se tenga que realizar cambios a partir de la retroalimentación de los usuarios finales y así continuamente lanzar nuevas versiones.

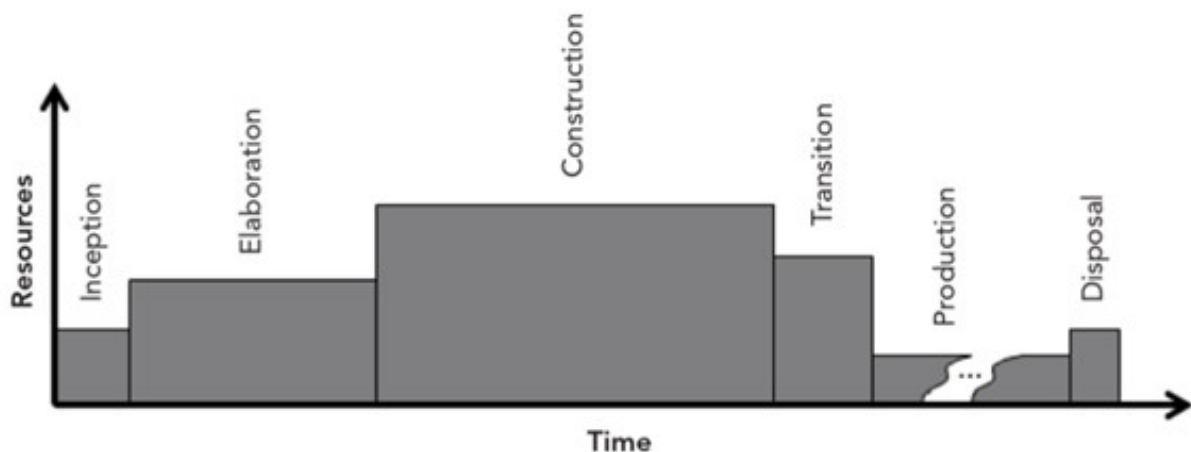


Figura 3: Tiempo vs recursos utilizados en cada fase en RUP.

Junto con las fases, IBM también introduce los *artifacts* que se traduce como “artefactos”⁵¹.

⁵¹Ibíd.

Estos son simplemente los resultados que se generan durante el transcurso del proyecto o también se les considera los entregables y pueden ser entre documentación, modelos o elementos de un modelo.

2.4.2. Kanban. Kanban se traduce del japonés como “tarjeta de señal”, ya que esta metodología fue en un principio creada en la década de los 40 por la empresa Toyota para la organización de inventario. Sin embargo, en el mundo de la ingeniería de software y gracias a David J. Anderson⁵², se empezó a utilizar como una metodología de seguimiento de desarrollo y control de roles.

Existe toda una filosofía detallada detrás que no se abordará, sino precisamente, en su aplicación directa. Como bien su nombre dice, se trata de tarjetas virtuales o físicas donde se representa un sistema de gestión de proceso visual en un tablero, donde cada tarjeta representa una tarea asignada a un miembro de un equipo. Estas tarjetas viajan de izquierda a derecha pasando por columnas que indican el estado de la tarea. Es común indicar las columnas como “To do” (Por hacer), “In progress” (En progreso) y “Done” (Terminado), pero esto no es realmente estricto en Kanban y es flexible según las necesidades del proyecto o del hito a alcanzar.

Cada tarjeta forma parte de un objetivo en común y se suele aplicar en iteraciones con fechas de entrega detalladas, es por esto por lo que este sistema de gestión es muy común adaptarlas junto con otras metodologías ágiles para desarrollo de software.

Las tarjetas se componen de un título, una breve descripción, el nombre de la persona asignada, la fecha de creación y la fecha de entrega; estos componentes, como se ha repetido en el documento, nunca son estrictas y se pueden agregar más o menos detalles según sea el caso.

⁵²Ibíd.

3. METODOLOGÍA

La metodología seleccionada para el desarrollo de este proyecto correspondió a una adaptación del Proceso Racional Unificado (RUP) complementado con tableros Kanban, permitiendo combinar un enfoque estructurado con la flexibilidad de los métodos ágiles.

El modelo RUP fue elegido debido a su enfoque iterativo e incremental, que permitió refinar progresivamente el producto a través de fases bien definidas (Inicio, Elaboración, Construcción e Implementación), garantizando la trazabilidad y calidad del proceso.

Por su parte, Kanban proporcionó una herramienta visual de gestión que facilitó la organización de tareas, el control de avances y la asignación de responsabilidades, elementos fundamentales en un equipo de trabajo pequeño que requería adaptabilidad y comunicación constante.

La integración de ambos enfoques permitió mantener la rigurosidad metodológica necesaria para un proyecto académico, sin perder la capacidad de respuesta ante imprevistos técnicos o de diseño durante la construcción del prototipo.

3.1. INICIO

La fase de inicio tuvo como propósito establecer las bases conceptuales, técnicas y pedagógicas del proyecto, en la cual se abordaron todos los conceptos utilizados en el artículo recopilatorio GBL⁵³. En esta etapa se definió la visión general del videojuego serio, sus objetivos formativos y las necesidades que este debía atender dentro del contexto del aprendizaje sobre sistemas ferroviarios en Colombia.

También se identificaron los principales riesgos técnicos y de diseño, así como las limitaciones de tiempo, recursos y experiencia del equipo, con el fin de planificar un alcance

⁵³Nadolny et al., «Examining the characteristics of game-based learning: A content analysis and design framework», óp.cit.

realista del prototipo

3.1.1. Actividades realizadas.

1. Definir las HU: Se definió un primer documento borrador de las historias de usuario iniciales del videojuego para que pudieran cumplir con los requisitos para ser un GBL.
2. Identificar riesgos: Se definieron en una lista las posibles dificultades técnicas encontradas en Unity para cumplir con las historias de usuario planteadas.
3. Planificar la fase de elaboración: Se empezó a elaborar un primer borrador del documento de los requisitos clave funcionales.

3.2. ELABORACIÓN

El objetivo principal de esta fase fue transformar la visión conceptual en un modelo técnico y funcional inicial, detallando los requerimientos, la arquitectura y el diseño general del videojuego. Durante esta etapa se buscó reducir la incertidumbre técnica mediante la creación de maquetas y la priorización de historias de usuario.

3.2.1. Actividades realizadas.

1. Organizar las HU: Se pulió el documento de las historias de usuario previamente encontradas y se organizaron en épicas para después definir un orden de prioridades. Además, se explicó cómo se abordarían los criterios del GBL y los desafíos ferroviarios en las historias de usuario.
2. Elaborar maqueta: Se construyó una primera maqueta en Unity atacando directamente a la lista de identificación de riesgos, se obtuvo un primer “esqueleto” del juego sobre el cual se empezó la fase de construcción.

3. Refinar la lista de riesgos: Se modificaron los riesgos según el resultado de la primera maqueta, para después investigar posibles soluciones.
4. Planificar las iteraciones: Se definieron las iteraciones que se darían en la fase de construcción, priorizando los requerimientos e historias de usuario más importantes.

3.3. IMPLEMENTACIÓN

Durante esta fase se llevó a cabo la programación y construcción del prototipo del videojuego de forma incremental, completando las funcionalidades y asegurando la calidad a través de pruebas continuas. La actividad principal de esta fase fue implementar el videojuego y, a partir de las iteraciones planificadas, se empezó a construir el código en Unity apoyándose en el sistema de gestión de procesos Kanban.

El apoyo del sistema de gestión Kanban permitió mantener una comunicación visual y fluida sobre el avance del proyecto, registrando las tareas en las columnas *Por hacer* y *Haciéndose*. Cada integrante era responsable de una o varias historias de usuario, y, una vez una tarea estaba en la columna *Revisándose* el compañero debía validar su funcionamiento antes de marcar la tarea como *Hecho*.

Cuadro 1: Tablero Kanban propuesto para la fase de Construcción.

Equipo	Por hacer	Haciéndose	Revisándose	Hecho
Miguel				
Mateo				

3.4. EVALUACIÓN

La fase de evaluación tuvo como propósito validar el prototipo desde los aspectos técnicos, pedagógicos y de usabilidad. Para ello se diseñó una prueba piloto dirigida a un grupo de estudiantes universitarios de la Universidad Industrial de Santander (UIS), quienes interactuaron con el videojuego y respondieron a un cuestionario estructurado para medir la experiencia y el aprendizaje percibido

3.4.1. Actividades realizadas.

1. Publicar videojuego serio: Se publicó y publicitó un link durante un periodo de 1 semana donde las personas pudieran instalar y probar fácilmente el videojuego.
2. Obtener los resultados: Los resultados de juego y rendimiento de los participantes se guardaron en una base de datos. Además se les pidió realizar un pequeño cuestionario para calificar la usabilidad del software.
3. Análisis de datos: Los datos recopilados pasaron a ser organizados y estudiados para extraer conclusiones.
4. Preparar informe y presentación: Se recopiló la documentación, el proceso de trabajo y los resultados para agruparlos en un libro entregable junto con una presentación.

4. RESULTADOS

En este capítulo se presentan los productos resultantes del proceso de desarrollo del prototipo del videojuego serio, abarcando desde la consolidación de los requerimientos y el diseño de la arquitectura, hasta la implementación del prototipo y el análisis de los datos obtenidos durante las pruebas individuales.

4.1. REQUERIMIENTOS

Los artefactos presentados a continuación son el producto de un proceso de refinamiento continuo llevado a cabo durante las fases de inicio, elaboración e implementación. Tras múltiples ciclos de iteración y ajuste, se obtuvieron estos documentos que constituyen la versión final y consolidada de los requisitos funcionales y el análisis de riesgos que guiaron la implementación del videojuego serio.

4.1.1. Matriz de Riesgos. Para categorizar el estado de las amenazas identificadas, se aplicó la escala de medición definida en la etapa de planificación, en la cual el nivel de riesgo se determina por el producto del Impacto por la Probabilidad de ocurrencia (ver figura 4).

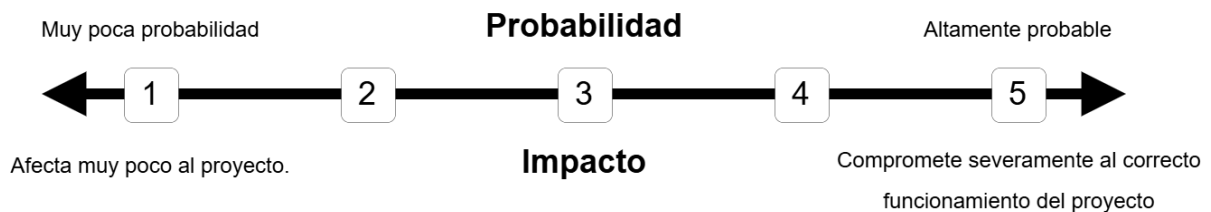


Figura 4: Escala de evaluación de riesgos utilizada durante el proyecto.

La creftab:matriz-riesgos-final presenta la matriz de riesgos, donde las columnas representan lo siguiente:

1. Ref: Identificador único alfanumérico asignado a cada riesgo para facilitar su seguimiento y referencia en el documento.
2. Descripción: Detalle narrativo del evento incierto o problema técnico que podría afectar el desarrollo del proyecto.
3. Causa: Factor, condición o decisión técnica que daría origen a la aparición del riesgo identificado.
4. Impacto: Estimación de la magnitud de las consecuencias negativas sobre los objetivos del proyecto (rendimiento, tiempo o calidad) en una escala del 1 al 5.
5. Probabilidad: Calificación numérica de la posibilidad de que el evento de riesgo ocurra durante el ciclo de vida del desarrollo, evaluada del 1 al 5.
6. Riesgo: Valor cuantitativo obtenido del producto entre el Impacto y la Probabilidad, utilizado para priorizar la criticidad de la amenaza.
7. Plan de Acción: Conjunto de estrategias, tareas o medidas de mitigación diseñadas para reducir o eliminar el efecto del riesgo en el sistema.
8. Estado: Se definieron dos posibles estados para cada riesgo: *Abierto*, que indica que el riesgo persiste y requiere de una resolución, y *Cerrado*, que señala que el riesgo se ha materializado, ha sido mitigado exitosamente o se ha aceptado conscientemente que no se realizarán más acciones sobre él.

Tras las iteraciones de desarrollo y las correcciones implementadas en la fase de implementación, todos los riesgos técnicos críticos fueron mitigados y cerrados exitosamente antes de la fase de evaluación.

Cuadro 2: Matriz de riesgos del proyecto.

Ref	Descripción	Causa	Imp.	Prob.	Riesgo	Plan de acción	Estado
R3	Vías poco naturales e inconexas	Uso de rectas para generar vías	5	4	20	Implementación de curvas Bézier y Splines.	Cerrado
R1	El modelo del terreno está mal optimizado	Sobrecarga en memoria gráfica por modelo 3D denso	4	4	16	Uso de Terrenos nativos de Unity y carga por bloques (LOD).	Cerrado
R7	Dificultad para representar hidrografía completa	Densidad excesiva de ríos en Colombia	4	4	16	Selección de ríos principales y lagos mayores únicamente.	Cerrado
R13	Bajos FPS en equipos de gama baja	Desarrollo en equipos de gama alta	5	3	15	Optimización de draw calls y reducción de polígonos.	Cerrado
R8	Las vías atraviesan montañas	Falta de interpolación de alturas	3	4	12	Generación automática de túneles al detectar colisión con terreno.	Cerrado
R2	La cámara atraviesa el terreno	Falta de colisión en la cámara	2	5	10	Implementación de Clamping y colisionadores para la cámara.	Cerrado

Continúa en la siguiente página

Ref	Descripción	Causa	Imp.	Prob.	Riesgo	Plan de acción	Estado
R12	Incompatibilidad de Shaders y URP	Conción por defecto de Unity	3	3	9	Ajuste manual de materiales para el Universal Render Pipeline.	Cerrado
R9	Estadísticas de locomotoras no leídas	Datos dispersos en hijos del GameObject	4	2	8	Unificar la lógica en el script controlador padre.	Cerrado
R4	Terreno con parches o vacíos	DEM con datos erróneos	2	3	6	Corrección mediante algoritmos de suavizado.	Cerrado
R6	Circuito cerrado no detectado	Spline no cerrado	2	3	6	Detección de proximidad de nodos inicial/final.	Cerrado
R10	HUD no muestra vida tras cambio	Referencias perdidas al cambiar locomotora	2	2	4	Función para reiniciar estadísticas al instanciar.	Cerrado
R11	Modelos 3D con escalas erróneas	Exportación incorrecta desde Blender	1	3	3	Aplicar escalas y rotaciones antes de exportar.	Cerrado
R5	Textura de vías solapada	Vía al mismo nivel del suelo	1	2	2	Mover las vías generadas una pequeña distancia hacia arriba.	Cerrado

4.1.2. Definición de Épicas e Historias de Usuario. Con el objetivo de garantizar que el videojuego cumpliera con los lineamientos del Aprendizaje Basado en Juegos (GBL), el proceso de definición de requisitos comenzó con la elaboración de un conjunto preliminar de Historias de Usuario (HU). Este primer barrido buscó capturar todas las funcionalidades necesarias para cubrir tanto las mecánicas básicas como los objetivos del proyecto.

Posteriormente, para facilitar la gestión del desarrollo y la planificación de las iteraciones, estas historias de usuario fueron analizadas y agrupadas en categorías de alto nivel denominadas *Épicas*. Esta organización permitió clasificar los requisitos de manera lógica según el subsistema o área funcional a la que pertenecían.

En la cuadro 3 se presentan las 8 épicas resultantes de esta categorización.

Cuadro 3: Listado de Épicas del proyecto.

ID	Nombre de la Épica
1	Selección de Nivel y Contratos
2	Interacción con el Mapa 3D
3	Sistema de Construcción de Vías
4	Gestión de Material Rodante
5	Motor de Simulación y Eventos
6	Sistema de Evaluación y Puntuación
7	Progresión y Rejugabilidad
8	Narrativa y Mentoría

La cuadro 4 presenta el formato consolidado de las historias de usuario tras el proceso de refinamiento continuo tras cada iteración durante las fases del proyecto. Se define las columnas de la siguiente manera:

1. Épica: Categoría funcional de alto nivel que agrupa un conjunto de historias de usuario relacionadas con un mismo módulo o subsistema del proyecto definidas anteriormente.
2. Código: Referencia alfanumérica única asignada a cada requisito (ej. HU1) para facilitar su identificación y rastreabilidad técnica.
3. Historia de Usuario: Descripción breve de una funcionalidad o necesidad del sistema, redactada desde la perspectiva del rol respectivo.
4. Prioridad: Nivel de importancia asignado mediante el método MoSCoW, clasificando las tareas en: *Must have* (M) como imprescindibles, *Should have* (S) como importantes, *Could have* (C) como deseables y *Won't have* (W) para futuras implementaciones.
5. Complejidad: Valor numérico en una escala de 1 a 5 que estima el grado de dificultad técnica y el esfuerzo requerido para la implementación de la funcionalidad.

Cuadro 4: Tabla final de historias de usuario implementadas.

ÉPICA	CÓDIGO	HISTORIA DE USUARIO	PRIO.	COMP.
1	HU1	Como jugador, quiero ver una pantalla de selección de niveles para elegir cuál jugar.	M	2
1	HU2	Como jugador, quiero que al empezar un nivel, se me presente una pantalla de contrato con presupuesto y objetivos.	M	1
2	HU3	Como jugador, quiero poder mover la cámara (panorámica) por el mapa para explorar el terreno.	M	1
2	HU4	Como jugador, quiero poder acercar y alejar la vista (zoom) para ver el mapa con diferentes niveles de detalle.	M	1
2	HU5	Como jugador, quiero ver las delimitaciones políticas y ríos en el mapa.	M	3
2	HU6	Como jugador, quiero ver un terreno en 3D del nivel a jugar.	M	4
2	HU7	Como jugador, quiero poder rotar la cámara para observar el terreno desde distintos ángulos.	S	1
2	HU8	Como jugador, quiero poder activar una capa de vista topográfica para entender el relieve.	C	5
3	HU10	Como jugador, quiero poder construir un tramo de vía férrea entre dos puntos.	M	2

Continúa en la siguiente página

ÉPICA	CÓDIGO	HISTORIA DE USUARIO	PRIO.	COMP.
3	HU11	Como jugador, quiero poder eliminar un tramo de vía férrea para rediseñar mis rutas.	M	1
3	HU12	Como jugador, quiero que se construyan puentes automáticamente al cruzar valles.	M	4
3	HU13	Como jugador, quiero que se construya automáticamente un túnel si mi vía férrea atraviesa una montaña.	M	4
3	HU14	Como jugador, quiero poder construir estaciones para definir los puntos de inicio y fin.	M	3
3	HU15	Como jugador, quiero ver mi presupuesto actual en todo momento.	M	2
3	HU16	Como jugador, quiero poder deshacer mi última acción de construcción.	S	2
4	HU17	Como jugador, quiero acceder a un menú para seleccionar y comprar la locomotora.	M	3
4	HU18	Como jugador, quiero añadir vagones específicos (pasajeros, carga) a mi locomotora.	M	2
4	HU19	Como jugador, quiero que el menú me muestre solo las locomotoras aptas para la ruta diseñada.	C	2

Continúa en la siguiente página

ÉPICA	CÓDIGO	HISTORIA DE USUARIO	PRIO.	COMP.
5	HU20	Como jugador, quiero ver una alerta visual cuando el tren sufre daños en zonas de evento.	M	2
5	HU21	Como jugador, quiero que el botón de "Testeo" me informe si falta algún requisito.	S	3
5	HU22	Como jugador, quiero volver a la construcción después de un testeo para corregir la ruta.	S	1
6	HU23	Como jugador, al finalizar un nivel, quiero ver una pantalla de puntuación.	M	3
6	HU24	Como jugador, quiero que la puntuación considere la rentabilidad (costo vs. contrato) y durabilidad.	M	3
6	HU25	Como jugador, quiero que la puntuación considere la eficiencia (tiempo empleado).	S	3
6	HU26	Como jugador, quiero ver un informe detallado que desglose mi puntuación final.	S	3
7	HU27	Como jugador, quiero desbloquear nuevos niveles tras completar el contrato actual.	M	2
8	HU28	Como jugador, quiero que Don Raíl presente el contrato con un diálogo breve y contexto histórico.	S	3
8	HU29	Como nuevo jugador, quiero que Don Raíl me guíe con un tutorial interactivo en el primer nivel.	S	4

Continúa en la siguiente página

ÉPICA	CÓDIGO	HISTORIA DE USUARIO	PRIO.	COMP.
8	HU30	Como jugador, quiero recibir advertencias contextuales sobre pendientes pronunciadas.	S	4
8	HU31	Como jugador, quiero ver datos históricos al seleccionar una ciudad importante.	S	4
8	HU32	Como jugador, quiero comentarios de Don Raíl cuando el tren sufre un evento.	S	3
8	HU33	Como jugador, quiero que Don Raíl comente mi puntuación final (éxito o consejo).	S	2
8	HU34	Como jugador, quiero ver anécdotas ferroviarias en las pantallas de carga.	M	2
6	HU35	Como administrador, quiero guardar metadatos de los jugadores en una base de datos.	M	3

4.2. DISEÑO

El diseño de la arquitectura de software un videojuego serio, se orientó a satisfacer los requisitos fundamentales del marco GBL, priorizando la articulación coherente de las HU y la mitigación efectiva de los riesgos técnicos identificados previamente. Más allá de la estabilidad funcional, esta estructura se concibió para soportar una integración transversal con servicios externos, garantizando que la simulación ferroviaria gestione eficientemente el intercambio de datos vía API para la telemetría del jugador y la generación posterior de informes de auditoría.

4.2.1. Arquitectura de Escenas en Unity. Para optimizar la gestión de recursos y organizar lógicamente los componentes del videojuego, se implementó una arquitectura modular basada en escenas aditivas. A diferencia de los entornos monolíticos tradicionales, este sistema permite fragmentar la aplicación en capas independientes que se gestionan dinámicamente en tiempo de ejecución.

Como se ilustra en la figura 5, el diagrama describe el ciclo de vida de estas capas a través de tres zonas funcionales. En la parte izquierda (zona de carga), el sistema invoca e instancia nuevas escenas de manera aditiva, sumándolas al entorno sin detener la ejecución actual. Estas pasan a situarse en la sección central (zona de apilamiento), donde coexisten y operan simultáneamente como capas superpuestas, manteniendo su independencia lógica aunque compartan el mismo espacio de memoria.

Finalmente, el flujo concluye en la parte derecha (zona de descarga), la cual representa la liberación selectiva de recursos. En esta etapa, el sistema puede retirar una capa específica de la pila —como un nivel completado— sin afectar a las demás escenas persistentes. Esta estrategia de "quitar y poner capas evita la reinicialización redundante de sistemas transversales como la del terreno de juego, garantizando transiciones fluidas y un uso eficiente del procesador.

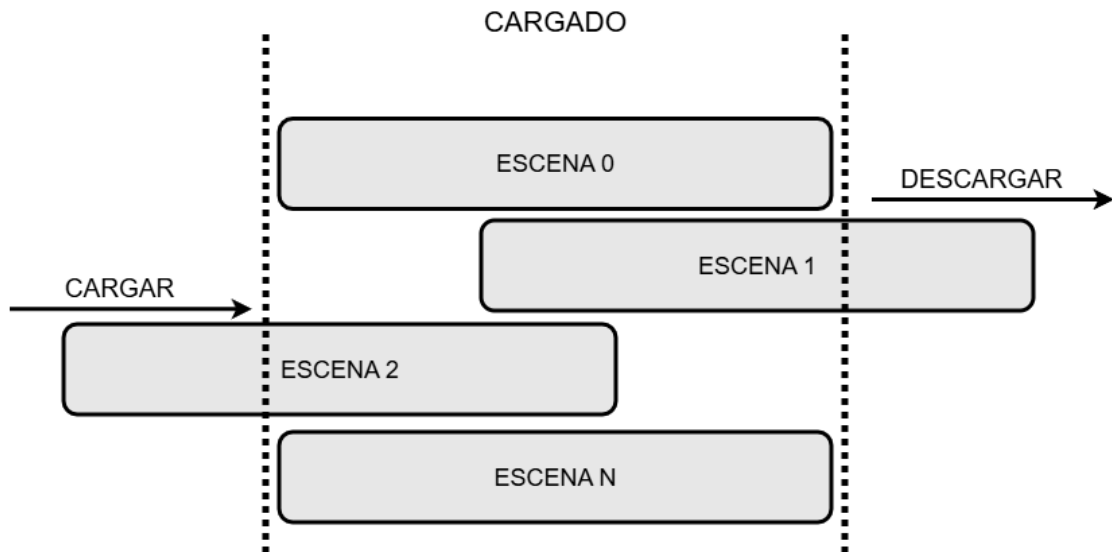


Figura 5: Diagrama de la arquitectura basada en escenas aditivas.

En nuestro caso, el flujo de ejecución comienza con la carga simultánea de las escenas *MainMenu* y *MapaMenu*. Posteriormente, al iniciar la partida, se descarga la escena decorativa y se carga la escena *MapaCompleto* junto con la escena del nivel correspondiente (por ejemplo, *Nivel 1 Tutorial*), manteniendo siempre activa la escena base de gestión.

A continuación, se describen las 8 escenas que componen el videojuego serio dentro de Unity:

1. *MainMenu*: Es la escena persistente que contiene el lienzo (*Canvas*) principal de la interfaz y los gestores globales del sistema. Los objetos instanciados aquí están condos para no destruirse entre cargas (*DontDestroyOnLoad*), asegurando la continuidad de la música, los datos del jugador y la gestión de estados.
2. *MapaMenu*: Funciona como una escena puramente decorativa para el fondo del menú principal. Consiste en un fragmento extraído del mapa completo (exactamente 1 *tile* de terreno) en el cual se ejecutan animaciones de trenes en bucle, brindando contexto visual sin cargar la totalidad de la geografía.
3. *MapaCompleto*: Es la escena más compleja y pesada del sistema, encargada de con-

tener la totalidad de la representación geográfica. Esta escena integra la malla del terreno dividida en una cuadrícula de 16×16 *tiles*, la red hidrográfica completa (ríos lineales y cuerpos de agua), las delimitaciones políticas administrativas y el etiquetado geográfico de todos los municipios posicionados sobre la topografía.

4. Niveles (1 al 5): Corresponden a 5 escenas independientes (*Nivel 1 Tutorial*, *Nivel 2*, etc.) que se cargan de forma aditiva sobre el mapa completo. Estas escenas contienen únicamente los objetos necesarios para la jugabilidad específica de ese contrato: los puntos de inicio y fin, la conción de las mecánicas, los gestores de eventos locales y los *Canvas* específicos de la misión.

4.2.2. Diagramas de Proceso. En el diagrama que se presenta en la figura 6, se muestra la forma en la que la interacción con el menú principal es realizada por parte del jugador, siguiendo todos los casos de cada interacción por parte del jugador con todos los elementos de la interfaz del juego, además, notese como el sistema, una vez detectado la falta de un archivo de guardado, se encargará de generar uno y activar los tutoriales para que el jugador pueda comenzar.

Por su parte, en el diagrama que se presenta en la figura 7 y figura 8 se demuestra la forma en la que una partida es realizada por el jugador, tomando en cuenta la interacción con cada elemento de la interfaz y con los elementos físicos como los son el mouse (clics y arrastre) y el teclado (teclas WASD). De manera general, el proceso abarca desde la aceptación del contrato y la gestión de movimiento del jugador, pasando por la construcción de vías y la elección de los trenes a utilizar, hasta la ejecución de la simulación, donde se evalúan los criterios de victoria o derrota. Finalmente, el sistema automatiza el envío de métricas a la API y la persistencia de datos en la base de datos NoSQL.

4.2.3. Diagrama de Componentes. Dada la división en múltiples escenas y el diseño detallado de los scripts e iteraciones del sistema, se optó por representar esta estructura en un diagrama de componentes. Con el fin de facilitar la lectura, dicho diagrama se

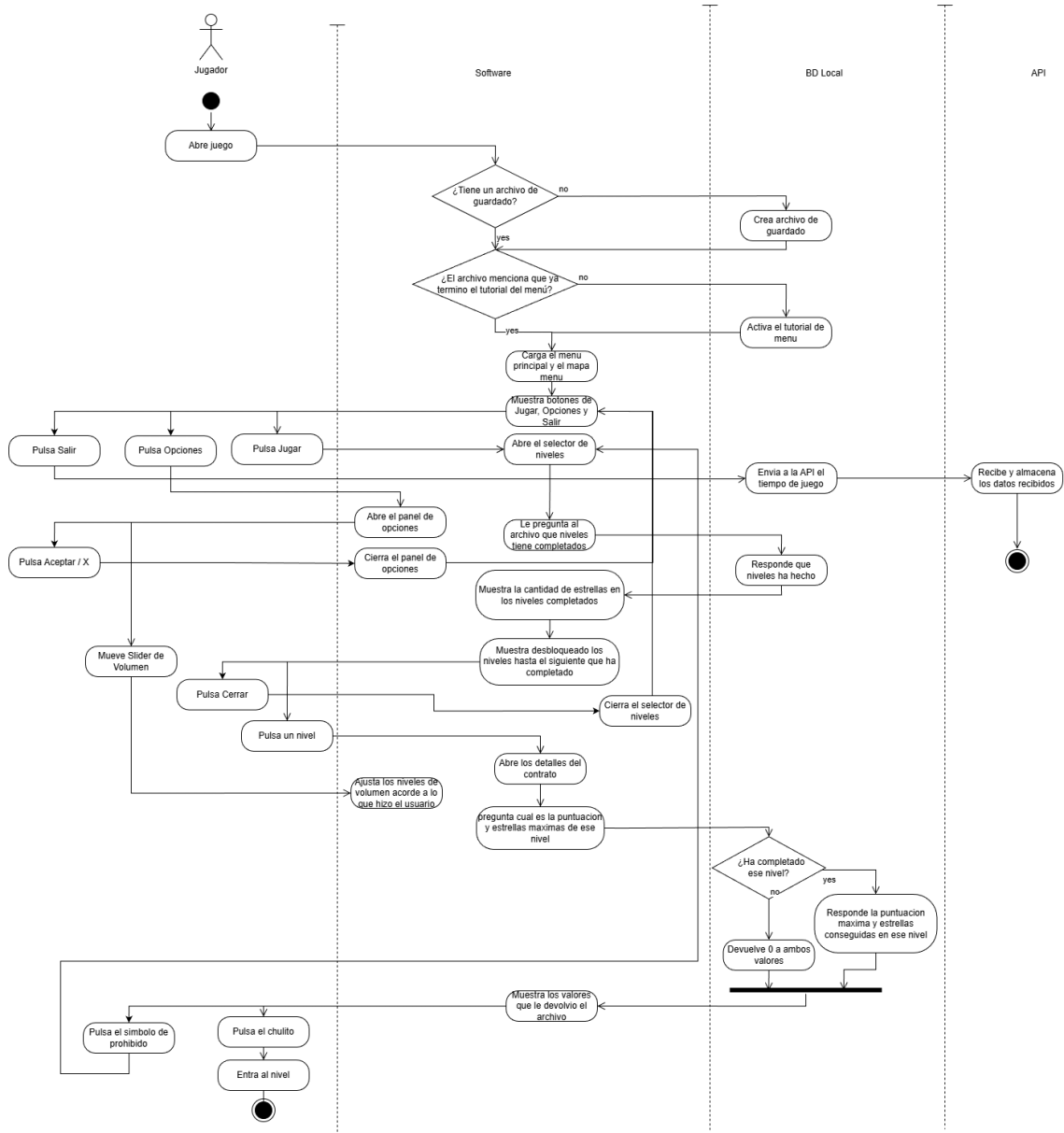


Figura 6: Diagrama de proceso del menú.

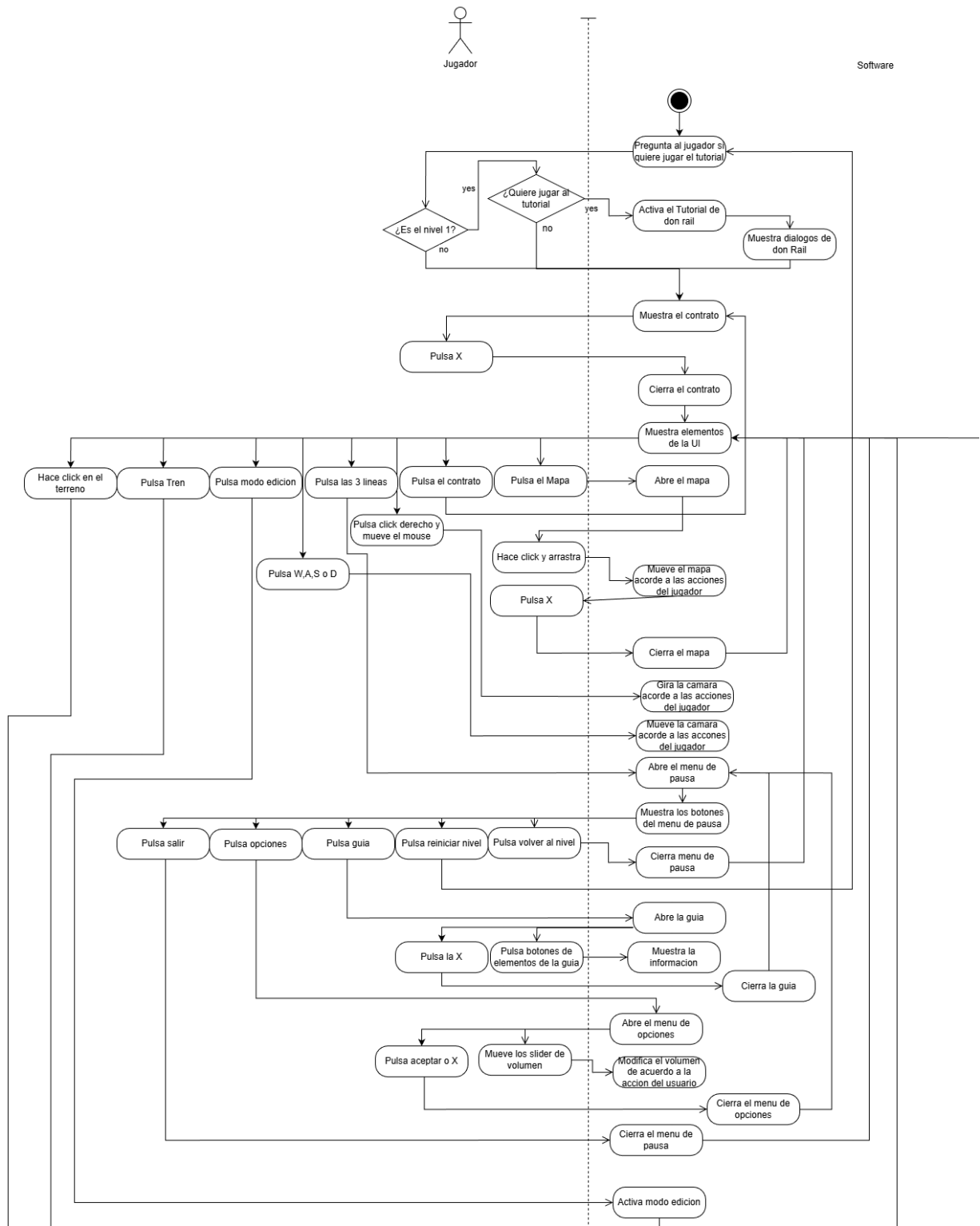


Figura 7: Diagrama de proceso de la partida (Figura a).

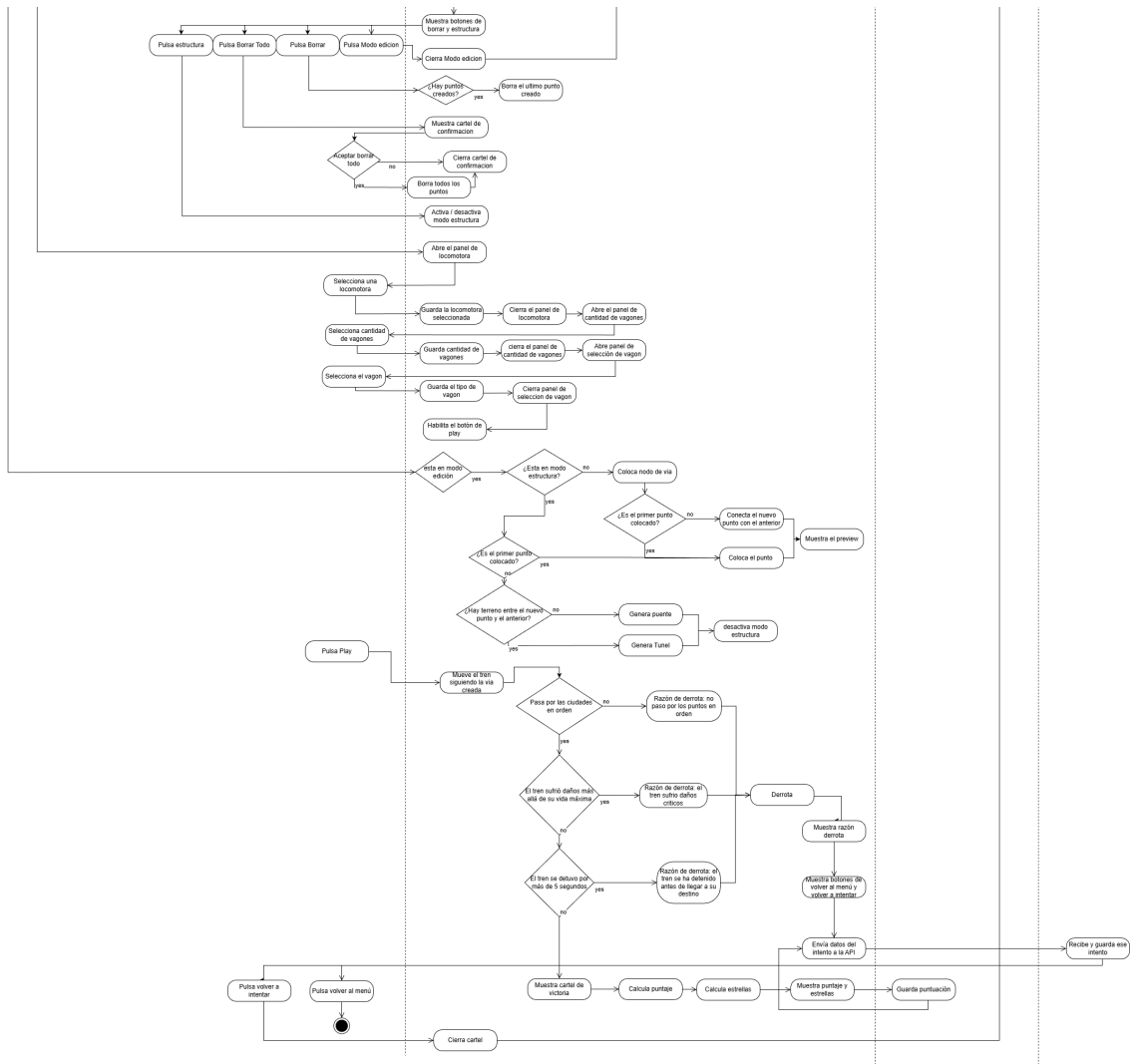


Figura 8: Diagrama de proceso de la partida (Figura b).

fragmentó en módulos que corresponden a cada escena como lo son la gestión del mapa, el diseño de niveles y la interfaz de usuario. Más adelante se presentará la descripción a detalle de cada escena junto con sus componentes principales.

4.2.3.1. Main Menu. El Main Menu es la escena encargada de mostrar la interfaz inicial que verá el jugador al abrir el juego serio, desde esta escena es que se podrán acceder a los niveles y las Conciones.

En esta escena podemos encontrar componentes como:

1. SaveSystem: Es el encargado de manejar el archivo de guardado del progreso del jugador.
2. System_SceneLoader: Se encarga de manejar la carga de las distintas escenas que utilizarán.
3. TutorialManager: Se encarga de los diálogos y controles durante el tutorial la primera vez que se ejecuta el juego.
4. EventSystem: Un componente nativo de unity usado para la detección de interacciones.
5. MúsicaSonido y UISonido: Encargados del manejo del sonido tanto musical como de interacción.
6. Canvas: componente que se encarga de todo lo visual y de interacción en el menú, como son los botones, sliders, etc.

4.2.3.2. Mapa Menu. La escena de Mapa Menu es la encargada de mostrar el mapa que se ve de fondo cuando el jugador esta en el menú principal, conteniendo tanto el mapa como los trenes en movimiento.

En esta escena podemos encontrar componentes como:

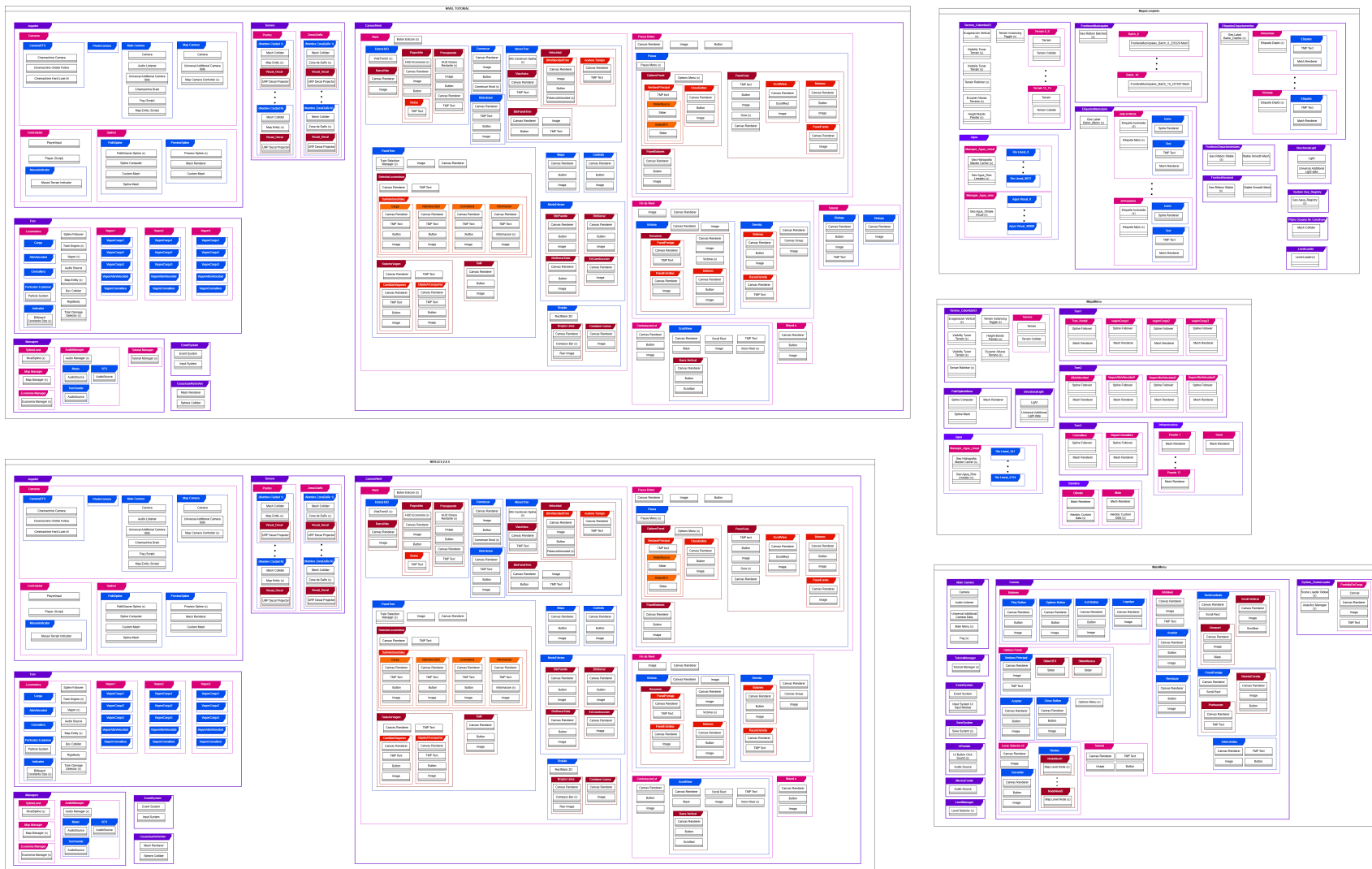


Figura 9: Diagrama de componentes organizado por escenas de Unity.

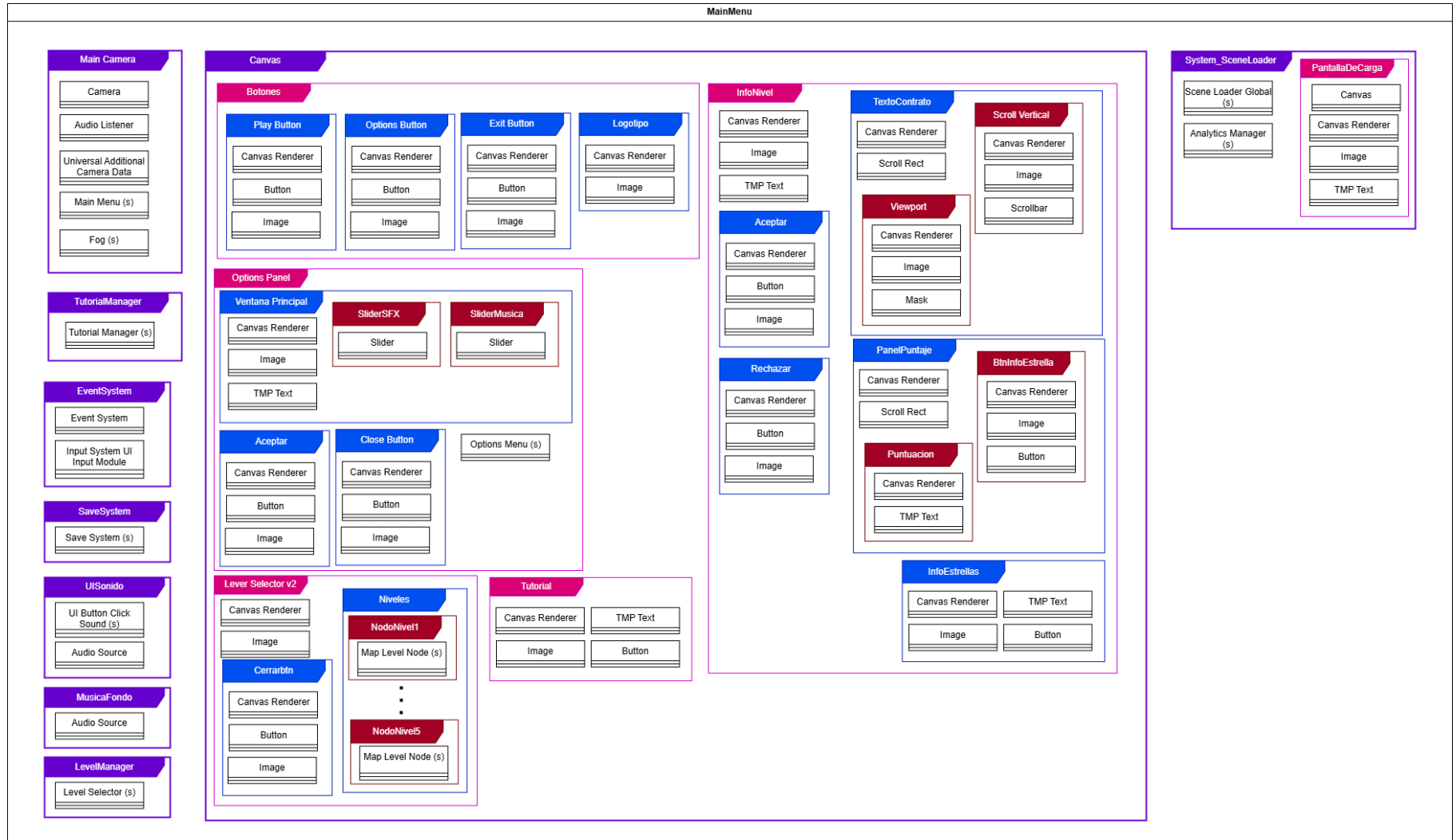


Figura 10: Diagrama de componentes del Menú Principal

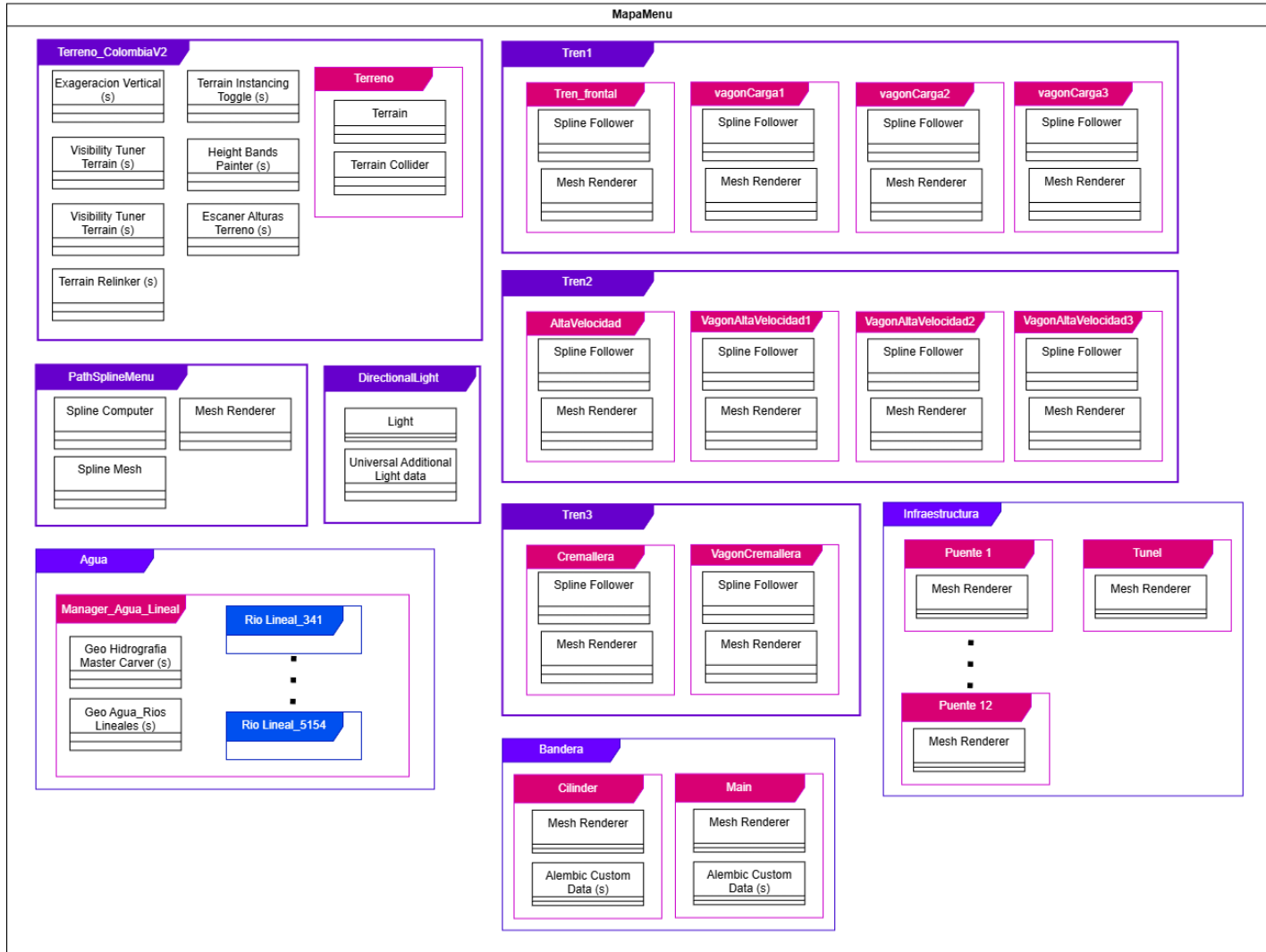


Figura 11: Diagrama de componentes del Mapa del Menu

1. Terreno_ColombiaV2: Contenedor del fragmento del mapa que vemos en el menu.
2. Agua: Se encarga de almacenar los ríos.
3. PathSplineMenu: Se encarga de gestionar los spline de las vías por las que los trenes transitan.
4. Tren1, Tren2 y Tren3: Son las 3 variantes de trenes junto a sus respectivos vagones.
5. Infraestructura: Contiene los puentes y tuneles por los que vemos a los trenes transitar.
6. Bandera: El simbolo patrio, contiene su modelo y su animacion de ser meneada por el viento.

4.2.3.3. MapaCompleto. La escena del Mapa completo se encarga de mostrar el mapa completo de todo el pais, conteniendo tanto el mapa como los cuerpos de agua.

En esta escena podemos encontrar componentes como:

1. Terreno_ColombiaV2: Contenedor del mapa completo del país, contando con 256 fragmentos del mapa que conectados forman toda la extension territorial de Colombia.
2. Agua: Se encarga de almacenar los ríos, lagos y otros cuerpos de agua, contando con mas de 40,000 cuerpos de agua almacenados.
3. Fronteras Municipales: En este estan contenidas todas las fronteras de los municipios.
4. Fronteras Departamentales: En este estan contenidas todas las fronteras de los departamentos.
5. Fronteras Nacional: En este estan contenidas las fronteras nacionales con otros paises como lo son Panamá, Peru, Brasil, Venezuela, y Ecuador.
6. Etiquetas Municipios: Contiene las etiquetas de todos los municipios de Colombia, teniendo etiquetas desde Abejorral hasta Zipaquirá.

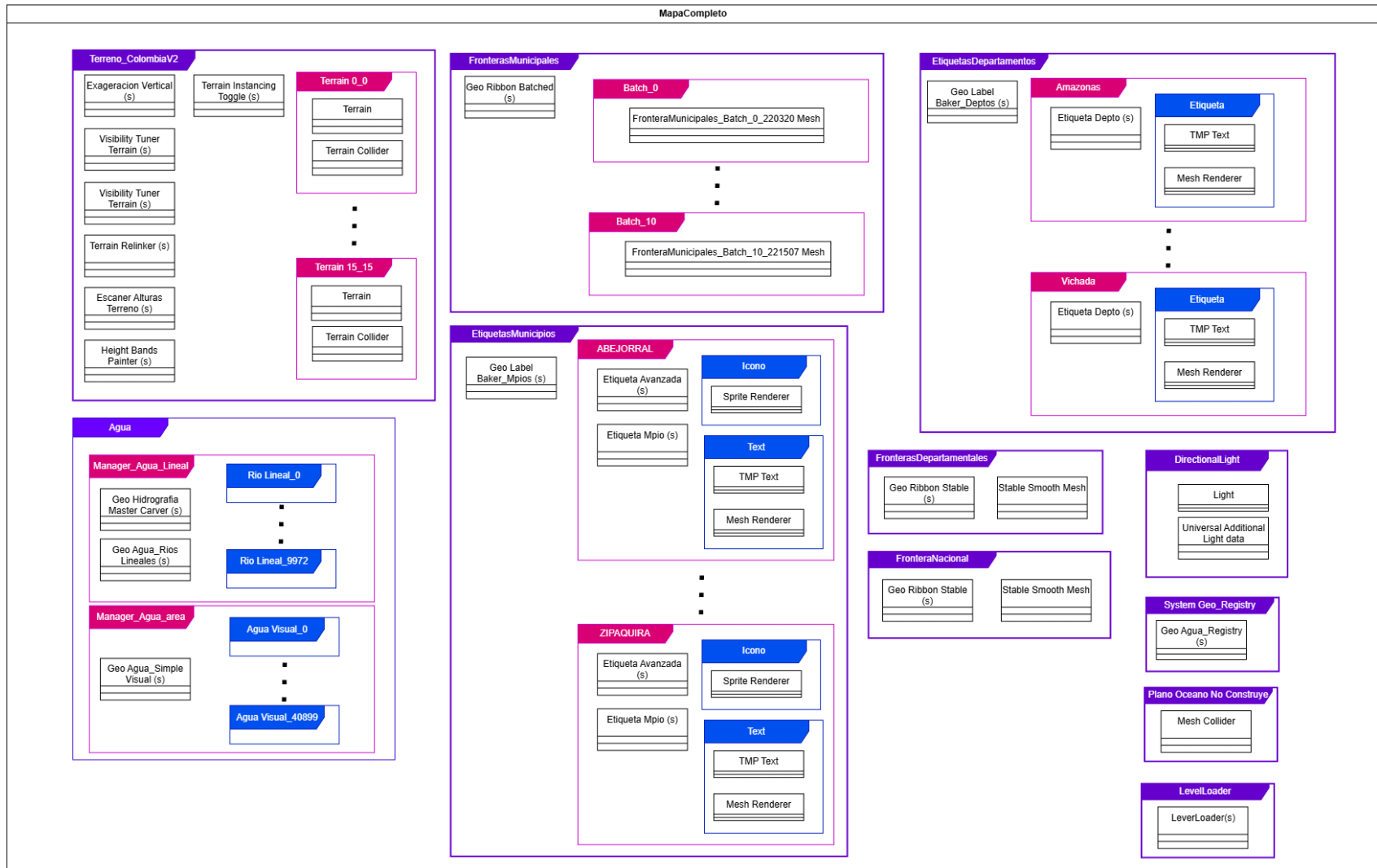


Figura 12: Diagrama de componentes del Mapa completo

7. Etiquetas Departamentales: Contiene las etiquetas de todos los departamentos de Colombia, teniendo etiquetas desde Amazonas hasta Vichada.
8. PlanoOceanoNoConstruye: Un plano 3D gigante, con textura transparente, que cubre el area de todo el pais, asignado a la altura 0, es usado como freno para que los jugadores no construyan en el oceano.
9. LevelLoader: Este elemento esta encargado de manejar la carga de diferentes escenas.

4.2.3.4. Nivel 1 Tutorial. En esta escena se encuentran objetos tanto de visibilidad del jugador como de analisis y calculos de resultados, asi como también manejar el tutorial para guiar al jugador durante su primera experiencia con el videojuego serio

1. Jugador: En este ítem se encuentra todo lo referente al jugador, tales como el controlador de movimiento, la cámara y las funciones de deteccion y construccion de vías.
 - a) Controlador de movimiento: Encargado de recibir y ejecutar los movimientos que el jugador haga por medio del mouse y el teclado
 - b) Cámaras: Encargadas de ser los ojos del jugador, también cuenta con una cámara secundaria que actua como vista aerea encima del país
 - c) Splines: Este componente contiene el script encargado de la creacion de las vías férreas, la previsualizacion de construccion y la construccion de estructuras como tuneles y puentes.
2. Terreno: En este componente se encuentran almacenados los elementos unicos referentes al nivel, tales como lo son las zonas de daño y los puntos por los que debe pasar el tren.
3. Canvas Nivel: En este canvas estan contenidos todos los elementos de interfaz con los que puede interactuar el jugador, tales como botones y sliders.

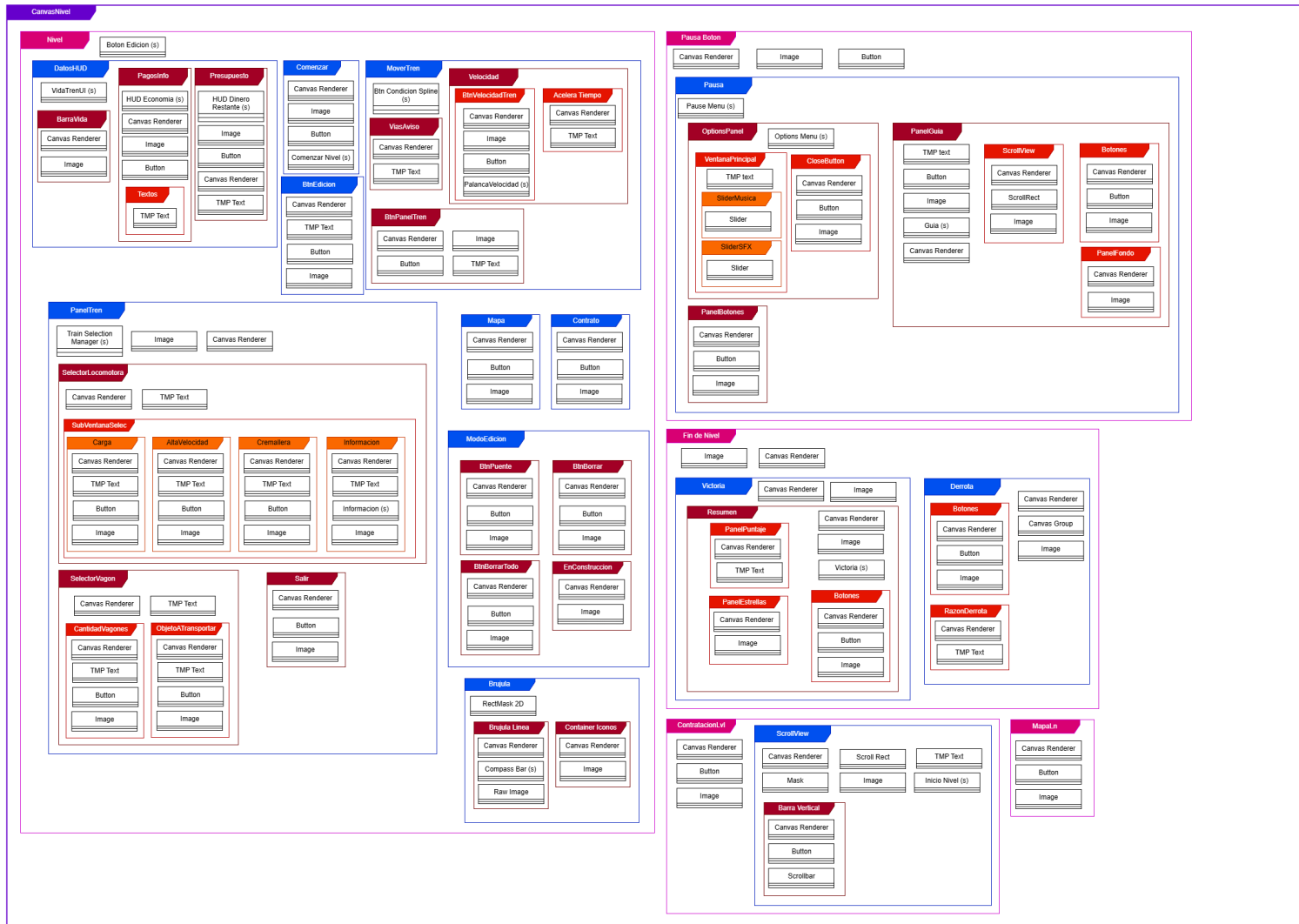


Figura 14: Diagrama de componentes del Nivel 1 Tutorial (Figura b)

- a) Nivel: Este componente contiene todos los elementos que el jugador puede ver durante el desarrollo normal del nivel, entre estos elementos se encuentran las interfaces de vida y dinero, los botones para la activación de modo de construcción y la selección de trenes.
 - b) Pausa: Encargado de almacenar los componentes del menú de pausa, tales como los botones de reinicio, guía y volver al menú principal.
 - c) Fin de Nivel: Aquí se encuentran las interfaces de victoria y derrota.
 - d) Tutorial: Encargado de almacenar los elementos de la interfaz del tutorial, como lo son los cuadros de diálogo y los sprites de "Don Raúl".
 - e) ContrataciónLvl: Encargado de almacenar los elementos del contrato del nivel, como los detalles de paradas y dinero disponible.
4. Tren: Este componente contiene los objetos de las distintas locomotoras y vagones a los que se tienen acceso durante el desarrollo de los niveles.
5. Managers: Aquí está almacenada la lógica interna y de gestión del nivel, cuenta con los siguientes componentes
- a) SplineLevel: Este componente es el corazón del nivel, en él se encuentran los parámetros necesarios para gestionar el nivel, como lo son las estaciones por las que hay que pasar, las zonas de daño que se encontrarán en el nivel y es el encargado de decidir si un intento cuenta como una victoria o una derrota.
 - b) Map Manager: Encargado de almacenar los iconos de los distintos puntos del mapa, como lo son los puertos, las ciudades e incluso el jugador.
 - c) AudioManager: Encargado de la gestión de sonido del juego, comprende elementos como la música y los efectos de sonido del tren en movimiento.
 - d) TutorialManager: En él se encuentran almacenados los diálogos y acciones que se espera que el jugador realice durante el desarrollo del tutorial

- e) **EconomiaManager:** Encargado de la gestión económica del nivel, se encarga de analizar los gastos y gestionar los calculos monetarios del nivel.

4.2.3.5. Niveles 2 a 5. Similar a la escena 1 tutorial, los niveles 2 a 5 comparten una estructura similar, los cambios más destacables son la remoción del tutorial y cambios individuales a cada nivel para determinar los objetivos del mismo

1. **Jugador:** En este ítem se encuentra todo lo referente al jugador, tales como el controlador de movimiento, la cámara y las funciones de detección y construcción de vías.
 - a) **Controlador de movimiento:** Encargado de recibir y ejecutar los movimientos que el jugador haga por medio del mouse y el teclado
 - b) **Cámaras:** Encargadas de ser los ojos del jugador, también cuenta con una cámara secundaria que actúa como vista aérea encima del país
 - c) **Splines:** Este componente contiene el script encargado de la creación de las vías férreas, la previsualización de construcción y la construcción de estructuras como túneles y puentes.
2. **Terreno:** En este componente se encuentran almacenados los elementos únicos referentes al nivel, tales como lo son las zonas de daño y los puntos por los que debe pasar el tren.
3. **Canvas Nivel:** En este canvas están contenidos todos los elementos de interfaz con los que puede interactuar el jugador, tales como botones y sliders.
 - a) **Nivel:** Este componente contiene todos los elementos que el jugador puede ver durante el desarrollo normal del nivel, entre estos elementos se encuentran las interfaces de vida y dinero, los botones para la activación de modo de construcción y la selección de trenes.
 - b) **Pausa:** Encargado de almacenar los componentes del menú de pausa, tales como los botones de reinicio, guía y volver al menú principal.

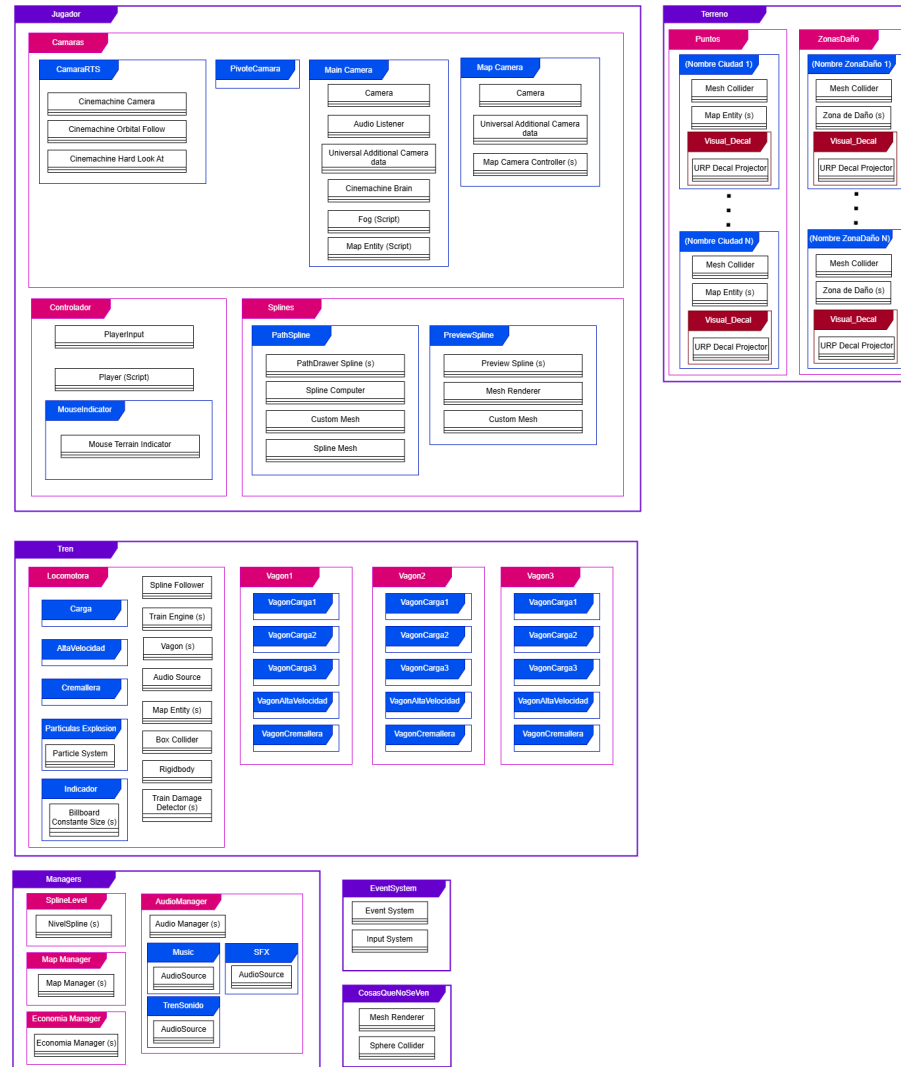


Figura 15: Diagrama de componentes de los niveles 2 a 5 (Figura a).

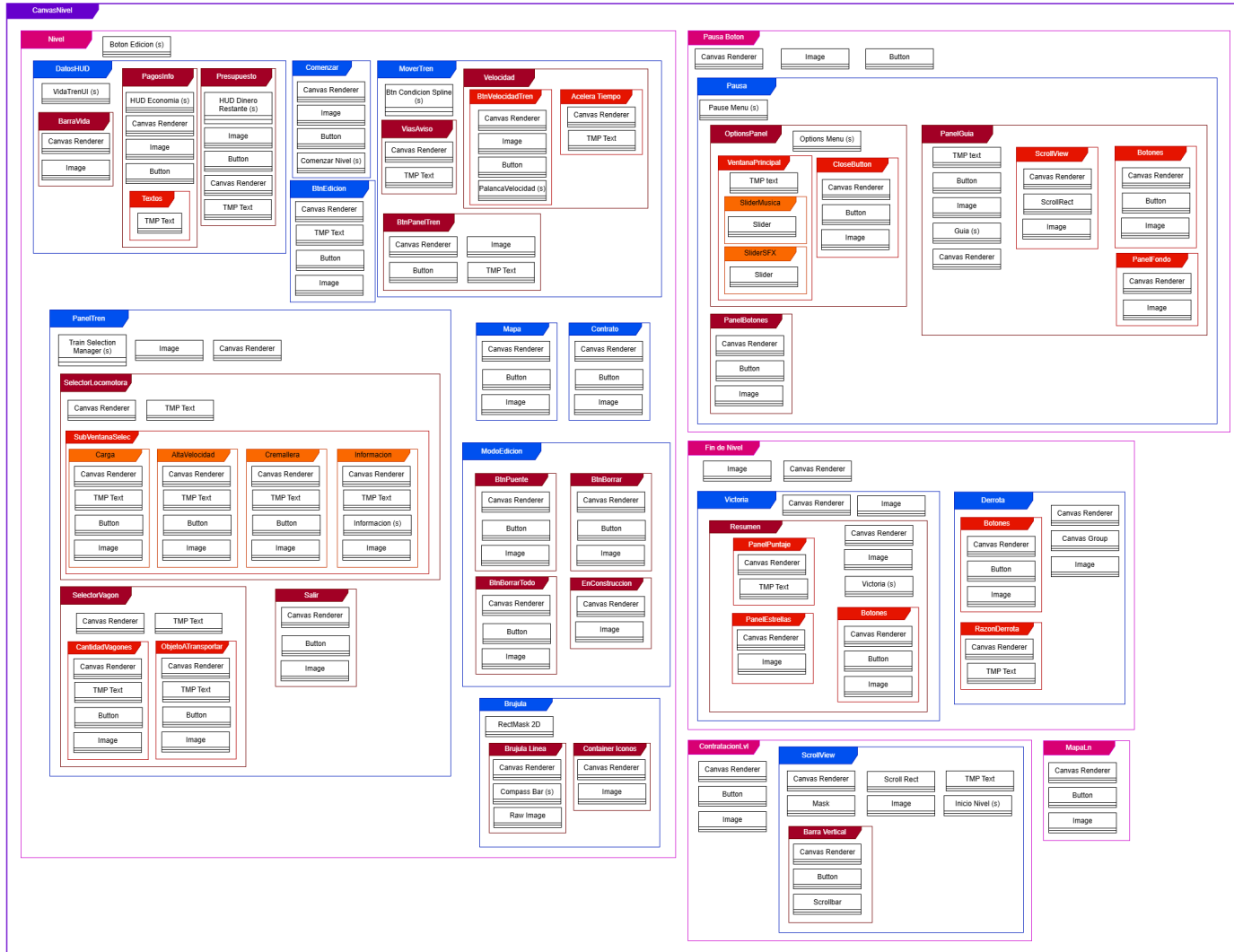


Figura 16: Diagrama de componentes de los niveles 2 a 5 (Figura b).

- c) Fin de Nivel: Aquí se encuentran las interfaces de victoria y derrota.
 - d) ContrataciónLvl: Encargado de almacenar los elementos del contrato del nivel, como los detalles de paradas y dinero disponible.
4. Tren: Este componente contiene los objetos de las distintas locomotoras y vagones a los que se tienen acceso durante el desarrollo de los niveles.
5. Managers: Aquí esta almacenada la lógica interna y de gestión del nivel, cuenta con los siguientes componentes
- a) SplineLevel: Este componente es el corazón del nivel, en él se encuentran los parámetros necesarios para gestionar el nivel, como lo son las estaciones por las que hay que pasar, las zonas de daño que se encontrarán en el nivel y es el encargado de decidir si un intento cuenta como una victoria o una derrota.
 - b) Map Manager: Encargado de almacenar los iconos de los distintos puntos del mapa, como lo son los puertos, las ciudades e incluso el jugador.
 - c) AudioManager: Encargado de la gestión de sonido del juego, comprende elementos como la música y los efectos de sonido del tren en movimiento.
 - d) EconomiaManager: Encargado de la gestión económica del nivel, se encarga de analizar los gastos y gestionar los cálculos monetarios del nivel.

4.2.4. Arquitectura del Sistema y API de Informes. Con el propósito de fortalecer el componente de *Apoyo al Aprendizaje*, requisito fundamental en el diseño GBL, y trascender el sistema básico de puntuación numérica y por estrellas, se implementó una arquitectura Cliente-Servidor para la gestión de datos telemétricos. En este esquema, el cliente (videojuego en Unity) se comunica con un servidor *backend* responsable de procesar y almacenar la actividad de las sesiones. Esto faculta al administrador para analizar los resultados y potenciar la realimentación hacia el jugador mediante la generación de informes detallados.

la figura 17 ilustra el flujo de información:

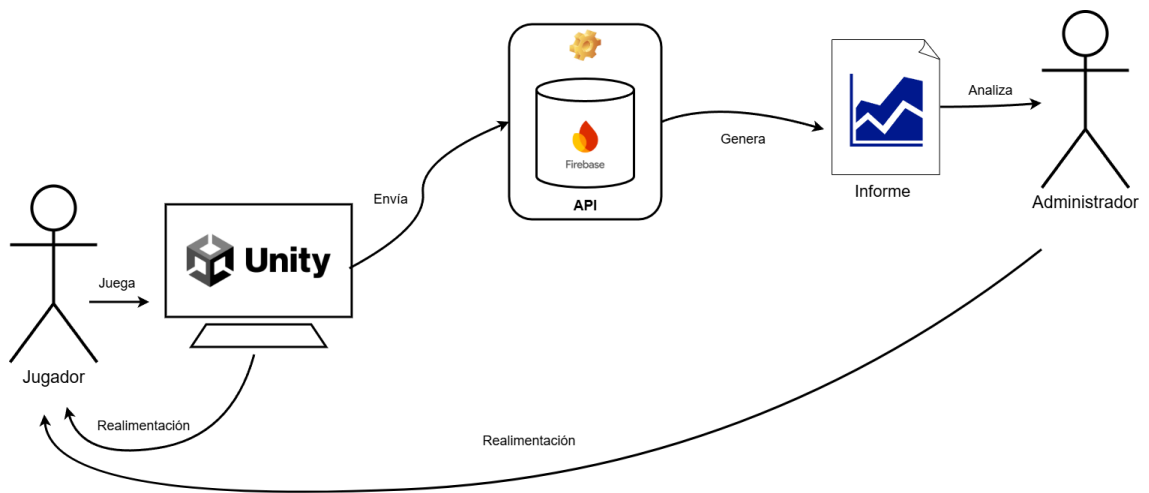


Figura 17: Arquitectura del sistema e integración con la API de informes.

1. Cliente Unity (C#): Al finalizar un nivel, el cliente recopila métricas clave y empaqueta estos datos en un objeto JSON.
2. API REST: Se desarrolló una API intermedia que recibe las peticiones POST provenientes del videojuego. Esta API valida la integridad de los datos y gestiona la seguridad de las transacciones.
3. Base de Datos: Los registros se almacenan en una base de datos No-SQL (FireBase), permitiendo la persistencia histórica de los intentos de los jugadores.
4. Módulo de Generación de Informes: Un servicio independiente consulta la base de datos para generar visualizaciones estadísticas y reportes de rendimiento, los cuales son utilizados para la evaluación pedagógica presentada más adelante en este capítulo.

4.2.4.1. Tablas de base de datos. La razón en usar una base de datos No-SQL vs SQL se debe a la naturaleza de la arquitectura de datos en el desarrollo de un videojuego; durante sus ciclos de vida un videojuego a comparación de otros productos software, sufren muchos cambios a nivel de estructura de datos. Por ejemplo, grandes empresas

de videojuegos como *RiotGames* o *Electronics Arts* hoy en día utilizan DynamoDB, una base de datos No-SQL para sus videojuegos.⁵⁴

Para garantizar la persistencia y el análisis posterior de la telemetría, se diseñó un esquema de datos documental alojado en la infraestructura de Firebase. la figura 18 ilustra la estructura de la colección principal denominada *level_logs*, la cual actúa como un registro histórico de cada sesión de juego de un nivel terminada por el usuario. Este modelo utiliza una relación de composición para almacenar los datos de interacción espacial directamente dentro del documento de la sesión, optimizando así las consultas de lectura.

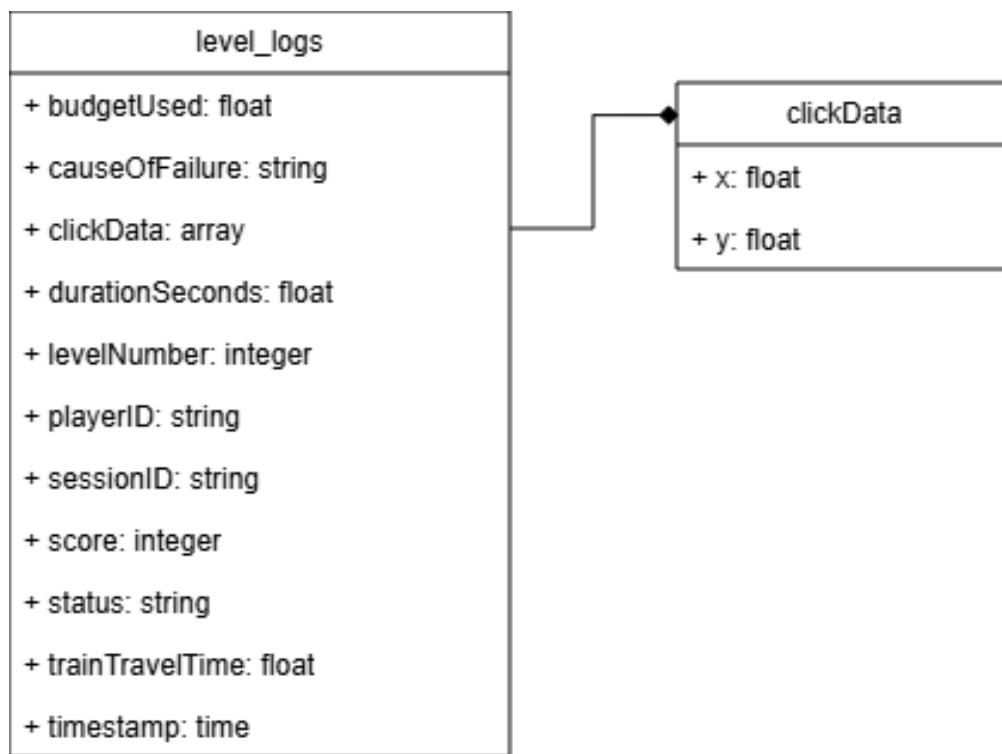


Figura 18: Colección única de la base de datos FireBase en la API.

A continuación, se detalla la función de cada atributo definido en el esquema:

1. *sessionID* (string): Identificador único generado por la API para diferenciar cada intento de juego, permitiendo trazabilidad individual incluso si un mismo jugador repite el nivel

⁵⁴Edin Zulich. *Amazon DynamoDB: Gaming use cases and design patterns*. 2 de abr. de 2019. URL: <https://aws.amazon.com/es/blogs/database/amazon-dynamodb-gaming-use-cases-and-design-patterns/> (visitado 14-01-2026).

múltiples veces.

2. `playerID` (string): Referencia al identificador del usuario autenticado. Funciona como una clave foránea lógica que vincula la sesión con los metadatos del perfil del jugador.
3. `levelNumber` (integer): Indica el nivel específico que se está jugando (ej. 1 a 5), contextualizando la dificultad y los objetivos activos durante la sesión.
4. `status` (string): Describe el estado final de la partida. Puede tomar valores categóricos como *Victoria*, *Derrota* o *Abandonado*, siendo fundamental para calcular tasas de éxito.
5. `causeOfFailure` (string): Campo opcional que almacena la razón específica de la derrota (ej. *Descarrilamiento*, *Presupuesto Agotado*). Si el estado es de victoria, este campo permanece nulo o vacío.
6. `budgetUsed` (float): Variable económica que registra la cantidad total de dinero virtual gastado por el jugador en la construcción de vías y compra de maquinaria.
7. `score` (integer): Puntuación final obtenida basada en la eficiencia, economía y cumplimiento del contrato del nivel.
8. `durationSeconds` (float): Tiempo total de juego desde el inicio del nivel hasta la finalización o fallo, utilizado para medir la curva de aprendizaje.
9. `trainTravelTime` (float): Métrica específica que contabiliza únicamente el tiempo que el tren estuvo en movimiento sobre las vías, separando el tiempo de construcción del tiempo de simulación.
10. `timestamp` (time): Marca temporal exacta del servidor al momento de guardar el registro, esencial para el ordenamiento cronológico de los datos.
11. `clickData` (Array de Objetos): Estructura anidada (sub-documento) que almacena una lista de coordenadas vectoriales. Cada elemento contiene:
 - a) `x` (float): Posición horizontal del clic en el mundo del juego.
 - b) `y` (float): Posición vertical del clic.

4.3. IMPLEMENTACIÓN

La fase de implementación se llevó a cabo utilizando el motor Unity (versión 6.0), haciendo uso de Git (ver figura 19) para el control de versiones y LFS para la gestión de activos pesados. A continuación, se detalla el proceso técnico de construcción del entorno y las mecánicas ferroviarias.

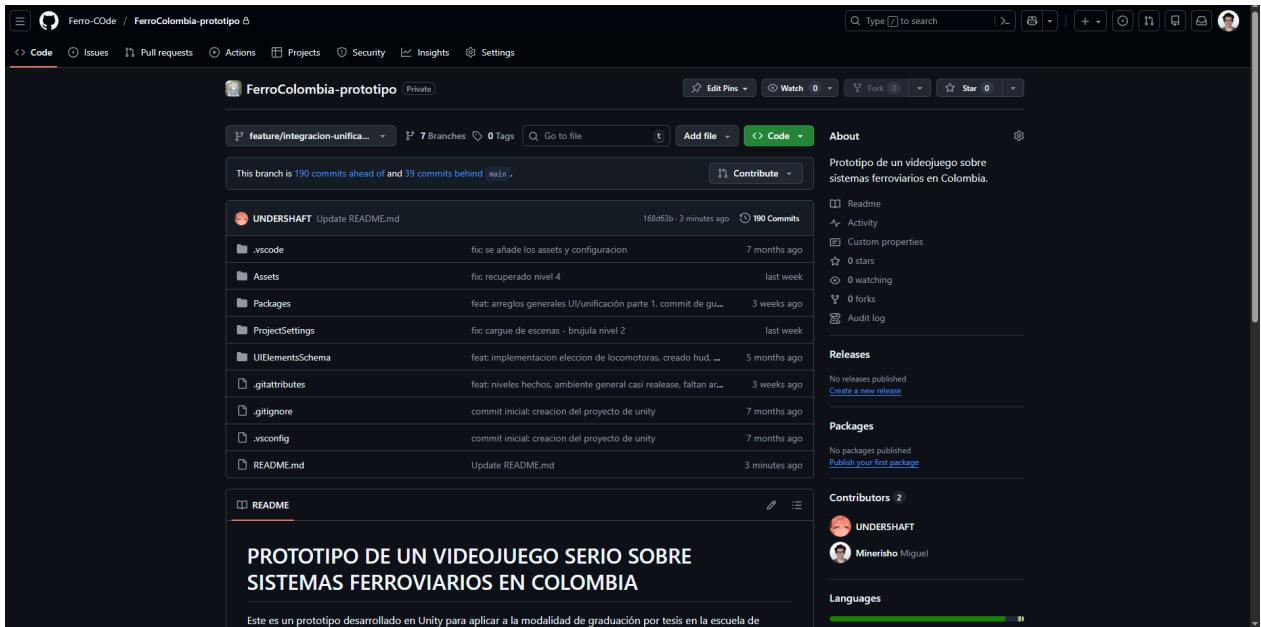


Figura 19: Interfaz del repositorio Git

4.3.1. Conción del Espacio de Trabajo en Unity. Para el desarrollo del proyecto se configuró un entorno de trabajo optimizado en Unity 6.0, diseñado para gestionar la complejidad del mapa de Colombia y la lógica de los sistemas ferroviarios. En la figura 20 se observa la disposición de las herramientas y la interacción entre las ventanas del editor:

1. Jerarquía (Hierarchy): En esta sección se evidencia el uso de un flujo de trabajo basado en múltiples escenas cargadas simultáneamente (*Multi-Scene Editing*), como *MapaCompleto*, *Nivel 1* y *Nivel 2*. Cada escena agrupa sus propios componentes y lógica de forma independiente. En este caso, se encuentra desplegado el *Nivel 2*, donde se ha seleccionado el objeto *Locomotora*, permitiendo una edición focalizada sin alterar

el resto del entorno global.

2. Vista de Escena y Juego (Scene & Game View): El área central permite alternar entre la edición y la visualización final. La pestaña activa (*Scene*) muestra el espacio tridimensional donde se realiza el ensamblaje del relieve y la ubicación de objetos. Adyacente a esta, se encuentra la pestaña *Game*, la cual permite previsualizar la cámara del jugador y la interfaz de usuario final para realizar pruebas de ejecución inmediatas.
3. Inspector: Esta ventana actúa de forma reactiva a la selección realizada en la Jerarquía. Al estar señalada la *Locomotora*, el Inspector despliega todos los componentes asignados a dicho objeto. Desde aquí se gestionan propiedades como el *Transform*, el sistema de audio y las variables físicas del motor del tren, permitiendo ajustar el comportamiento del vehículo en tiempo real.
4. Explorador de Proyecto (Project Window): Contiene la biblioteca de archivos del juego organizada mediante una estructura de carpetas estandarizada (*Meshes*, *Prefabs*, *Scripts*, *Settings*). Esta organización es fundamental para el control de versiones, asegurando que todos los activos del mapa de Colombia y los modelos ferroviarios estén correctamente indexados.

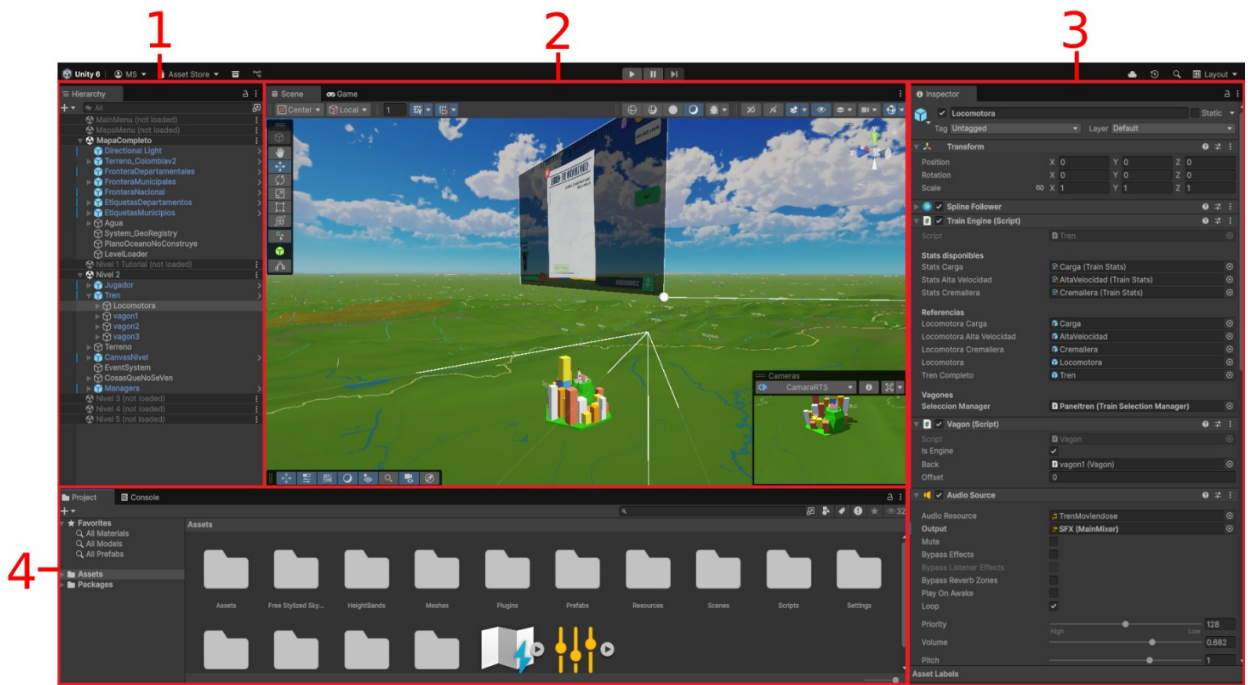


Figura 20: Interfaz de desarrollo en Unity

4.3.2. Construcción del Entorno Geográfico. El mapa de Colombia representa el tablero de juego principal. Su implementación pasó por múltiples iteraciones para equilibrar el realismo geográfico con el rendimiento técnico.

Inicialmente, se importó un modelo 3D de alta resolución generado en Blender a partir de mapas de elevación (DEM). Como se observa en la figura 21, aunque visualmente detallado, este enfoque resultó inviable debido a la carga de 24.5 millones de vértices, lo que redujo la tasa de refresco a menos de 26 FPS (Fotogramas por segundo).



Figura 21: Primer intento de generar el terreno usando un modelo 3D en Blender.

Tras experimentar con versiones de baja cantidad de polígonos (*low-poly*) y fragmentadas que no solucionaron los problemas de latencia en las llamadas de renderizado (*draw-calls*), se optó por la solución de usar el sistema de terrenos nativo de Unity.

4.3.2.1. Automatización y Procesamiento de Datos Geográficos. Para la implementación definitiva, no se esculpió el terreno manualmente. En su lugar, se desarrolló un flujo de trabajo automatizado mediante *scripts* en C# que generaban la topografía a partir de datos reales.

Una herramienta fundamental en esta etapa fue el software QGIS (Quantum Geographic Information System). Esta herramienta permitió procesar y alinear todas las capas de información —fronteras municipales, hidrografía (ríos y lagos) y elevación— bajo un Sistema de Referencia de Coordenadas unificado, específicamente el *EPSG:9377* (MAGNA-

SIRGAS Origen Nacional). Esto garantizó que todos los elementos geográficos mantuvieran su posición relativa correcta al ser importados al motor de videojuego. En la figura 22 se muestra todos los datos recopilados y correctamente ubicados como son el mapa de alturas, los cuerpos de agua, las fronteras nacional, departamentales y municipales de Colombia.

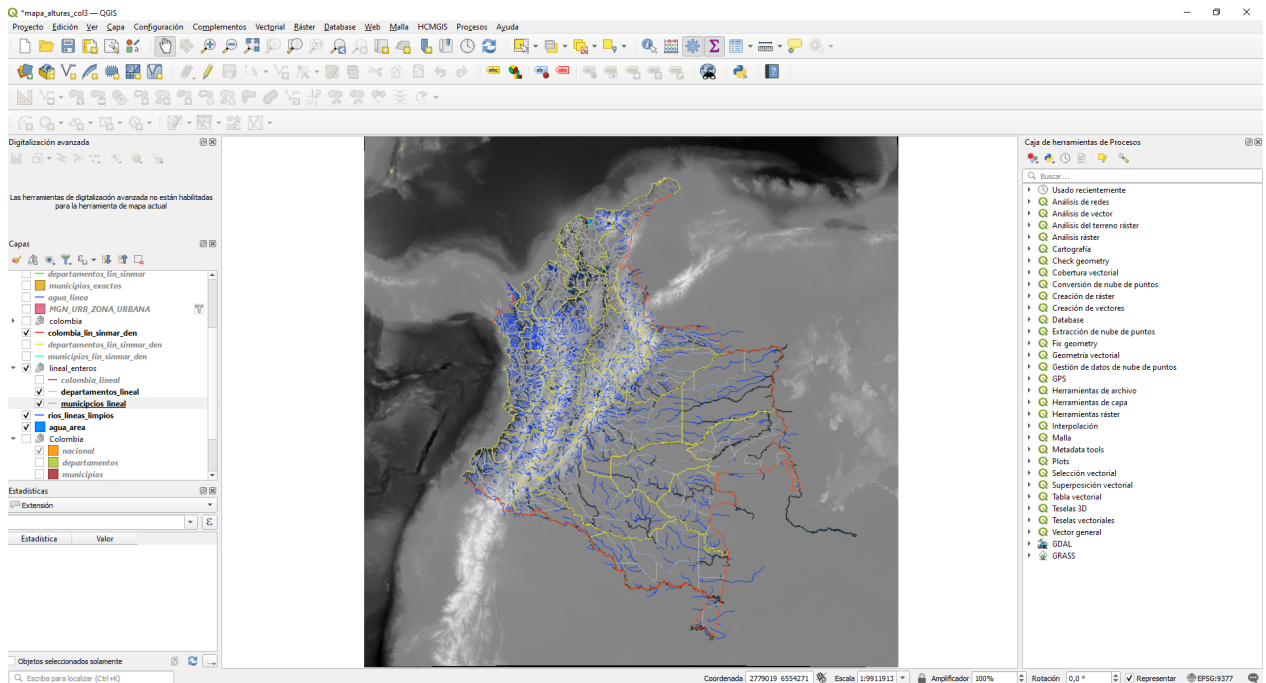


Figura 22: Captura del proyecto en QGIS utilizado para el terreno.

El Modelo Digital de Elevación (DEM) utilizado como base fue obtenido a través del *General Bathymetric Chart of the Oceans*.⁵⁵ Dada la extensión del territorio, se decidió segmentar este DEM en una cuadrícula de 16×16 secciones, generando terrenos interconectados dentro de Unity a una escala de 1:128.

4.3.2.2. Estrategia de Segmentación (Chunks). Esta división en 256 *tiles* o baldosas respondía a una estrategia inicial de optimización basada en la carga dinámica de *chunks*. En este contexto, un *chunk* se definió como una unidad lógica compuesta por un fragmento de terreno y la totalidad de los objetos (etiquetas, cuerpos de agua, divisiones políticas)

⁵⁵GEBCO. *General Bathymetric Chart of the Oceans*. 2025. URL: <https://www.gebco.net/data-products/gridded-bathymetry-data> (visitado 14-01-2026).

situados sobre él. La intención original era cargar y descargar estos bloques de memoria en tiempo real según la posición del jugador.

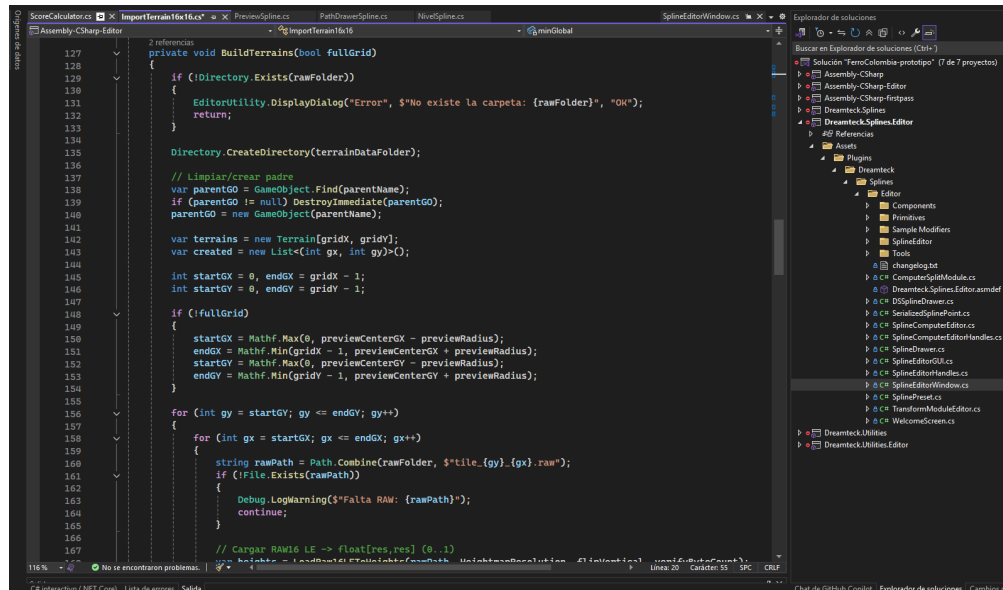


Figura 23: Fragmento del script encargado de la importación de terrenos.

Sin embargo, durante las pruebas de integración, se observó que el rendimiento del sistema era estable incluso con la totalidad del mapa cargado en memoria. Por consiguiente, se descartó el sistema de carga dinámica para reducir la complejidad del código, aunque se mantuvo la estructura modular de 16×16 para trabajo a futuro.

4.3.2.3. Texturizado. Finalmente, el terreno fue texturizado automáticamente aplicando un material autogenerado y semi-autocoloreado según la altura (mar, pasto, nieve). En la figura 24 se muestra el script desde la vista del inspector dentro del objeto *Terreno_Colombiav2* en donde se evidencia la interfaz hecha para manejar las capas por color y alturas.

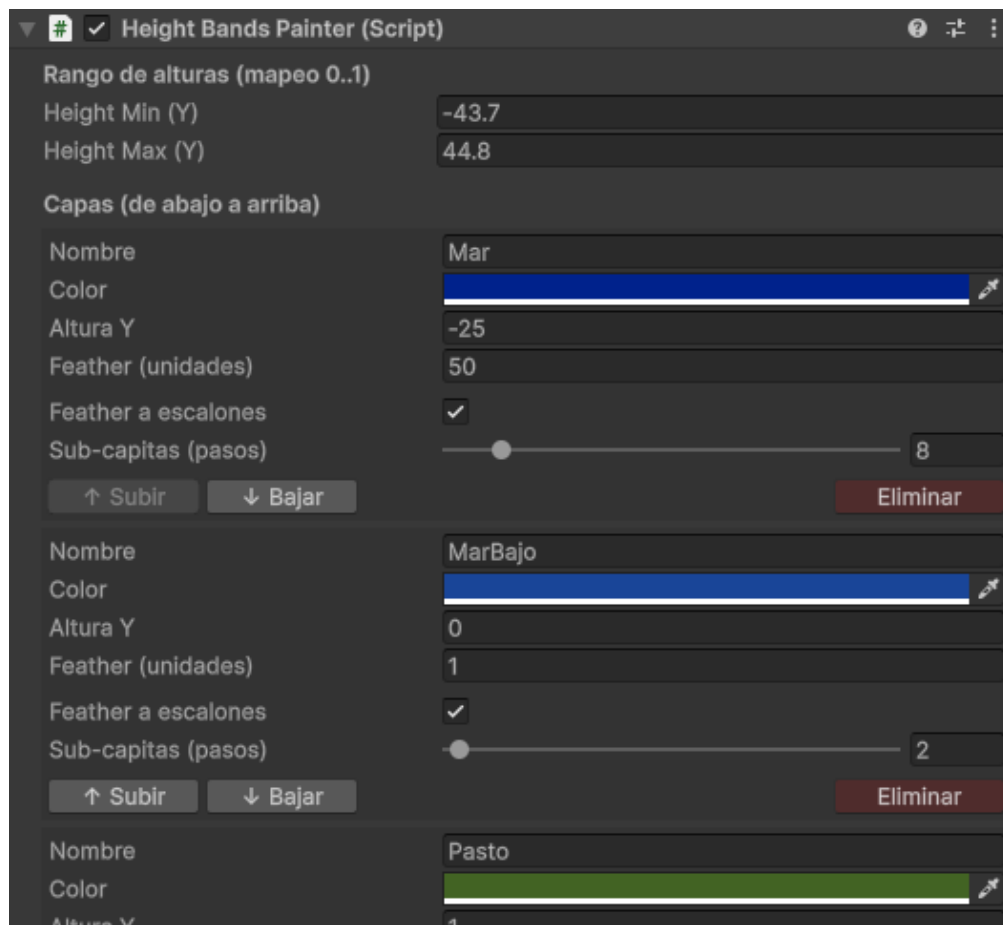


Figura 24: Captura de la clase *HeighBandsPainter* desde el inspector.

La figura 25 muestra el resultado final del terreno implementado.



Figura 25: Implementación del entorno utilizando el sistema de Terrenos.

4.3.3. Desarrollo del Sistema Ferroviario. La implementación del sistema ferroviario se subdividió en tres retos técnicos principales: la generación procedural de las vías, la gestión de estructuras complejas y la sincronización del movimiento de los trenes. A continuación se detalla la solución técnica para cada componente.

4.3.3.1. Generación de Vías y Optimización. Para resolver la lógica matemática del trazado y el seguimiento de los vagones, se utilizó la librería *Dreamteck Splines* como ba-

se. Dado que esta herramienta ofrecía funcionalidades preexistentes, el trabajo se centró en modificar y adaptar su código fuente para que fuera compatible con los modelos de vagones propios del proyecto. Sin embargo, por razones de optimización y rendimiento, se decidió prescindir del sistema de curvado suave de mallas, optando en su lugar por una generación de vías con ángulos agresivos y tramos rectos, visualizados mediante planos texturizados que simulan el balasto y los rieles.

4.3.3.2. Sistemas de Construcción y Estructuras. La interacción para la construcción opera mediante un sistema de dos puntos: el primer clic sobre el terreno establece el nodo de inicio del spline y el segundo define el destino, momento en el cual el algoritmo interpola puntos intermedios para ajustar la vía a la topografía del suelo. El desafío mayor surgió con la implementación de estructuras para atravesar accidentes geográficos como valles, cuerpos de agua o montañas. Para solucionar esto, se desarrolló un modo de construcción estructural que ignora la adaptación al terreno intermedia, forzando una conexión recta entre los puntos. Mediante el uso de *Raycasting*, el sistema analiza el entorno de este vector: si detecta terreno por encima de la vía, instancia automáticamente un túnel; si detecta vacío por debajo, genera los pilares correspondientes para formar un puente.



Figura 26: Modelo de un puente.



Figura 27: Modelo de un túnel.

Además, se implementó un sistema de previsualización al cual nombramos *PreviewSpline*, con el fin de poder mostrar al jugador una previsualización del trazo o camino que va a

realizar, además de ponerse de colores verde o rojo dependiendo si pueden construir en esa zona o no. Esta ayuda visual mejoró drásticamente la claridad e intuición a la hora de construir vías férreas sobre el terreno.

4.3.3.3. Lógica y Datos de los Trenes. Una vez definida la infraestructura, el sistema permite la selección de la locomotora y los vagones, cuyo costo en recursos varía según su tipología. La arquitectura de datos de estas unidades se gestionó mediante *Scriptable Objects*, lo que permitió modularizar estadísticas inmutables como el precio, la salud, la velocidad máxima y el tipo de carga. Finalmente, toda esta lógica se vinculó a la interfaz de usuario (Canvas) del nivel para permitir la interacción en tiempo real, añadiendo detalles de retroalimentación visual como sistemas de partículas que se activan cuando el tren recibe daño durante el recorrido.

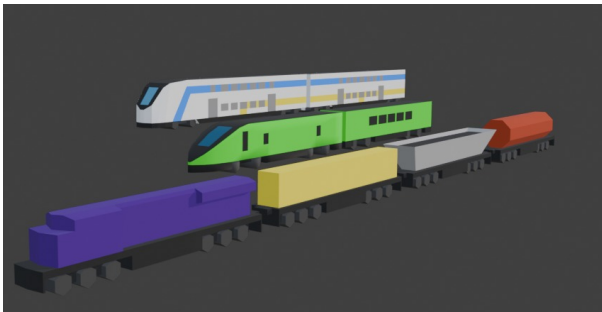


Figura 28: Modelado de locomotoras.



Figura 29: Sistema de partículas de daño.

4.3.4. Zonas de daño y prohibición. Para garantizar la coherencia geográfica y aumentar la dificultad, se implementaron volúmenes de control invisibles en el mapa. La zona de prohibición impiden la construcción directa del jugador, es un plano gigante que cubre todo el terreno pero está al nivel del mar, esto con el objetivo de que no pudieran construir sobre el mar. Mientras que las zonas de daño detectan colisiones físicas del tren

y le reducen la vida periódicamente.

La implementación de estas zonas de daño se diseñó bajo un principio de modularidad y separación de datos. El componente lógico principal, denominado *ZonaDeDaño*, actúa únicamente como un contenedor referencial que no almacena valores directos. En su lugar, consume un activo de datos externo de tipo *Scriptable Object* llamado *TipoZonasDaño*. Este activo encapsula las propiedades inmutables de la amenaza, definiendo atributos como el nombre identificador y la magnitud del daño flotante.

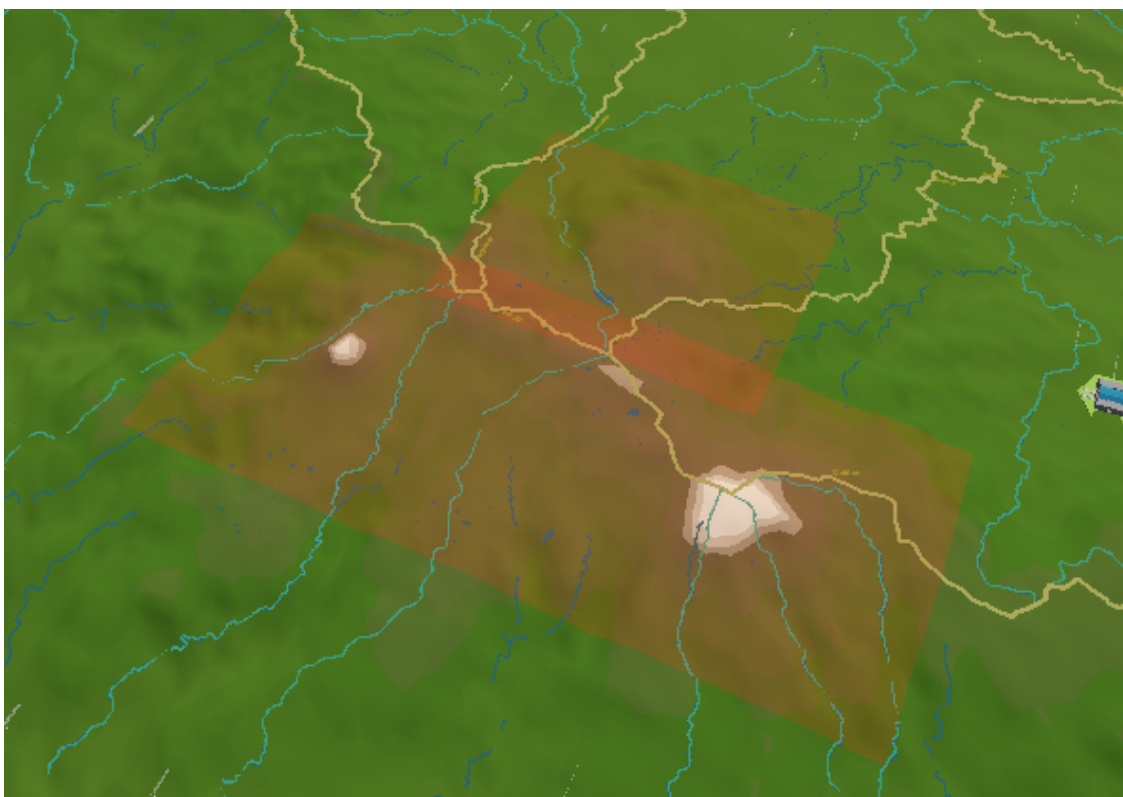


Figura 30: Ejemplo de una zona de daño en el terreno.

Esta estructura permite al equipo de diseño equilibrar la dificultad creando variaciones de peligros (ej. “Derrumbe Ligero” vs “Zona Volcánica”) editando un solo archivo de conción, el cual se propaga automáticamente a todas las instancias en el nivel. Aunque cabe mencionar, que se decidió no usar más de 1 tipo de zona de daño. Esta decisión fue en consecuencia del temor a aumentarle demasiado la dificultad al juego.

Finalmente, la materialización de estas zonas en la escena se logra mediante la composición de dos elementos: un *Mesh Collider* como disparador (*Trigger*) para la detección física, y un proyector *Decal* de color rojo semitransparente que renderiza una advertencia visual sobre la topografía irregular del terreno, permitiendo al jugador identificar el peligro intuitivamente.

4.3.5. Managers. La arquitectura del juego se orquesta a través de una serie de clases controladoras (*Managers*) que centralizan la lógica de negocio y mantienen el estado global de la aplicación, estas son las principales:

4.3.5.1. Economía. La gestión financiera es administrada por la clase `EconomiaManager`, la cual opera en tiempo real durante el ciclo `Update()` para recalcular el costo total de la infraestructura desplegada. Siguiendo el patrón de diseño orientado a datos, este componente no almacena los precios de los activos internamente; en su lugar, consume una referencia externa al *Scriptable Object* `CondicionesGlobales`, lo que permite ajustar la inflación o el costo de los materiales (vías por metro, precio por puente/túnel) en un solo archivo para todo el juego.

El cálculo del saldo disponible se realiza mediante la siguiente sumatoria:

$$S_{remanente} = P_{inicial} - (C_{vas} + C_{estructuras} + C_{tren}) \quad (4.1)$$

Un aspecto fundamental de la experiencia de usuario (UX) en esta implementación es la ausencia de bloqueos restrictivos. A diferencia de los simuladores tradicionales que impiden la construcción al llegar a cero, el sistema permite que el jugador continúe extendiendo la vía incluso si el saldo entra en valores negativos. Esta decisión de diseño prioriza la finalización del nivel y la libertad creativa; sin embargo, incurrir en deuda tiene una consecuencia punitiva en la fase de evaluación, resultando en la pérdida automática de la

Estrella de Economía (más adelante se detallará sobre esto) y reduciendo drásticamente el puntaje final.

4.3.5.2. Sistema de Guardado. La persistencia local de datos es gestionada por la clase `SaveSystem`, un componente que implementa el patrón *Singleton* para garantizar una única instancia de acceso a archivos durante la ejecución. El sistema serializa el estado del juego en formato JSON y lo almacena en el directorio persistente del dispositivo (`Application.persistentDataPath`), asegurando que el progreso sobreviva al cierre de la aplicación.

La lógica de guardado no es destructiva; utiliza un algoritmo de actualización condicional detallado a continuación:

1. **Progresión Lineal:** El sistema incluye el método `IsLevelUnlocked`, el cual verifica la existencia de datos completados del nivel $N - 1$ para permitir el acceso al nivel N .
2. **Validación de Mejor Puntuación:** Al finalizar un nivel, el sistema compara el puntaje obtenido con el almacenado previamente. El archivo solo se actualiza si el jugador ha superado su propio récord (*High Score*).
3. **Fusión de Estrellas:** Las condiciones de victoria (estrellas) se guardan de manera acumulativa. Si el jugador obtuvo la “Estrella de Tiempo” en un intento anterior y la “Estrella de Economía” en el actual, el sistema fusiona ambos resultados mediante una operación lógica OR, permitiendo completar el nivel en múltiples intentos.

4.3.5.3. Nivel. La clase `NivelSpline` actúa como el orquestador central o máquina de estados finitos de la sesión de juego. Su diseño sigue un patrón orientado a datos (*Data-Driven*), delegando la conción de reglas (tiempo límite, presupuesto inicial) y fórmulas matemáticas a dependencias externas inyectadas mediante *Scriptable Objects* (`CondicionesNivel` y `CondicionesGlobales`). Esto permite ajustar el balance del juego sin modificar el código fuente.

El ciclo de vida gestionado por este componente se divide en tres etapas críticas:

1. **Gestión Temporal:** Distingue entre el tiempo de “Planeación” (desde la carga de la escena) y el tiempo de “Simulación” (activación del tren), lo cual es vital para no penalizar al jugador mientras diseña la vía.
2. **Validación de Ruta:** Implementa un sistema de control de puntos de paso (*Check-points*). Mediante la detección de colisiones, verifica que el tren recorra la ruta geográfica obligatoria en el orden correcto, impidiendo atajos no válidos.
3. **Resolución de Estado:** Al finalizar el recorrido, centraliza la información de los demás gestores (economía, integridad del tren y tiempo), ejecuta el algoritmo de puntuación descrito anteriormente y orquesta la persistencia de datos, enviando los resultados tanto al sistema de guardado local como al servicio de telemetría en la nube.



Figura 31: Interfaz de retroalimentación tras un evento de derrota.

4.3.5.4. Telemetría. El componente *AnalyticsManager* es el responsable de la comunicación Cliente-Servidor. A diferencia del guardado local, este sistema no almacena datos en el dispositivo, sino que los empaqueta y envía a una API REST alojada en *Google Cloud Functions*.

Su funcionamiento se divide en tres procesos concurrentes:

1. **Captura de Interacción (Heatmaps):** Durante el ciclo de juego (`Update`), el script registra las coordenadas (x, y) de cada clic realizado por el usuario en la interfaz o el terreno. Estos datos se almacenan temporalmente en una lista de objetos `ClickPos` y se adjuntan al reporte final, lo que permite la futura generación de mapas de calor para analizar el comportamiento del jugador.
2. **Registro de Intentos:** Al finalizar una partida (ya sea por victoria, derrota o abandono), se instancia un objeto `LevelLogPayload`. Este paquete de datos contiene métricas precisas como la duración exacta de la sesión, el presupuesto consumido, la causa específica del fallo y la marca temporal (*timestamp*).
3. **Transmisión Asíncrona:** Para evitar congelamientos en la interfaz de usuario debido a la latencia de red, el envío de datos se realiza mediante Corrutinas de Unity (`IEnumerator`). Se utiliza la clase `UnityWebRequest` con el método `POST` para transmitir el JSON serializado, manejando internamente los encabezados y la confirmación de recepción del servidor.

4.3.5.5. Tutorial. El sistema de aprendizaje no se implementó como una escena aislada, sino como una capa lógica transversal gestionada por la clase `TutorialManager`. Este componente funciona como una máquina de estados secuencial que se superpone tanto en el Menú Principal como en el primer nivel de juego, adaptando su comportamiento mediante la bandera lógica `esTutorialDeMenu`.

La arquitectura del sistema se fundamenta en tres pilares técnicos:

1. **Estructura de Datos Narrativa:** El contenido del tutorial se define mediante una lista de objetos serializables de la clase `DialogoTutorial`. Cada elemento contiene el texto explicativo, la expresión emocional del personaje guía (“Don Rail”), una referencia opcional a un objeto de la interfaz que debe resaltarse (activando indicadores visuales) y, lo más importante, la `TutorialAction` requerida para avanzar.



Figura 32: Personaje guía *Don Rail*.

2. Sistema de Eventos y Polling: Para validar el progreso del jugador, el gestor utiliza un enfoque híbrido.
 - a) *Detección Activa*: En el ciclo `Update`, el script monitorea entradas de hardware específicas, como el movimiento de los ejes del ratón o las teclas WASD, para validar lecciones de control de cámara.
 - b) *Detección Pasiva*: Para interacciones de interfaz (UI) o mecánicas complejas (como conectar vías), otros scripts notifican que una tarea ha sido completada invocando el método público `ReportarAccion(ID)`.
3. Gestión de Tiempos (UX): Para evitar que el tutorial avance abruptamente apenas el usuario toca un control, se implementó una corrutina de espera (`RutinaEsperarYAvanzar`). Esta lógica introduce un retraso intencional (conble, por defecto 3 segundos) después de acciones continuas como mover la cámara, permitiendo que el jugador asimile la mecánica antes de presentar el siguiente cuadro de diálogo.

Finalmente, el sistema se integra con el módulo de persistencia (`SaveSystem`) para verificar si el tutorial introductorio ya fue completado, evitando repeticiones innecesarias en sesiones futuras.

4.3.6. Menú Principal. El punto de entrada a la experiencia es una escena ligera diseñada para ofrecer una navegación intuitiva y rápida. La interfaz gráfica, ilustrada en la figura 33, conserva la identidad visual *low-poly* del proyecto mediante una vista previa animada del entorno ferroviario.

Funcionalmente, el menú dispone de tres puntos de interacción principales:

1. **Jugar:** Inicia la transición hacia la escena del *Selector de Niveles*, cambiando el contexto hacia el mapa geográfico de Colombia para que el usuario elija su contrato.
2. **Opciones:** Despliega una ventana emergente (*pop-up*) dedicada a la configuración del sistema. En la versión actual del prototipo, este módulo se centra exclusivamente en la gestión del *AudioManager*, permitiendo al usuario ajustar de manera independiente el volumen de la música de fondo y de los efectos de sonido (SFX) mediante deslizadores (*sliders*).
3. **Salir:** Ejecuta la instrucción de finalización del ciclo de vida de la aplicación, cerrando el juego y devolviendo el control al sistema operativo.



Figura 33: Interfaz del Menú Principal.

4.3.7. Menú de Pausa. Durante la ejecución de cualquier nivel, el jugador dispone de la facultad de interrumpir el ciclo de simulación pulsando la tecla *Escape*. Esta acción invoca al *PauseManager*, el cual detiene el tiempo de juego (estableciendo `Time.timeScale = 0`) y despliega una interfaz superpuesta que permite gestionar la sesión sin perder el progreso actual.

Como se observa en la figura 34, este menú ofrece cuatro opciones funcionales:

1. Reanudar: Oculta la interfaz y reactiva el cronómetro de la simulación inmediatamente.
2. Reiniciar: Recarga la escena actual desde cero, restableciendo todas las variables de estado (presupuesto y tiempo) para permitir un nuevo intento limpio.
3. Guía: Abre la ventana del menú de guía del juego.
4. Opciones: Habilita una ventana emergente para ajustar el volumen de la música y los efectos de sonido en tiempo real.
5. Salir: Aborta la misión en curso y redirige al usuario a la escena del Selector de Niveles.



Figura 34: Interfaz del Menú de Pausa superpuesta a la simulación.

4.3.7.1. Guía de Juego. Integrado dentro de las opciones del menú de pausa, se encuentra el módulo de consulta denominado *Guía* (figura 35). Este componente actúa

como una enciclopedia interactiva y estática, diseñada para reforzar el aprendizaje de las mecánicas sin obligar al jugador a abandonar la experiencia.



Figura 35: Panel de la Guía detallando controles, mecánicas y recomendaciones.

4.3.8. Selector de Niveles. Para contextualizar geográficamente al jugador, la selección de misiones no se realiza mediante una lista tradicional, sino a través de un mapa interactivo de Colombia (figura 36). Los marcadores sobre el territorio indican la ubicación real de los contratos ferroviarios, reforzando el componente educativo de geografía nacional.

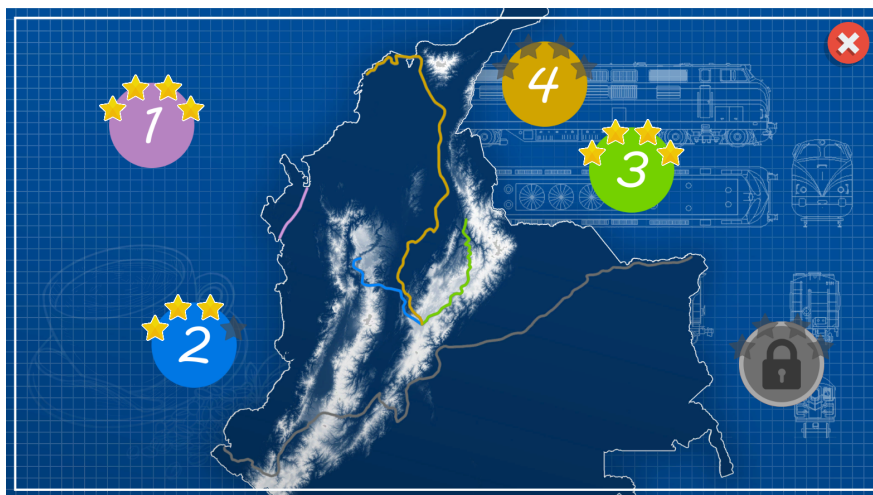


Figura 36: Escena de selección de niveles sobre el mapa geográfico de Colombia.

4.3.9. Contratos. Cada nivel se presenta bajo la metáfora de un ‘Contrato Ferroviario’. Antes de iniciar, el sistema presenta una ficha técnica (figura 37) que detalla el origen,

los destinos, el presupuesto asignado y el tiempo límite acordado del trayecto del tren.

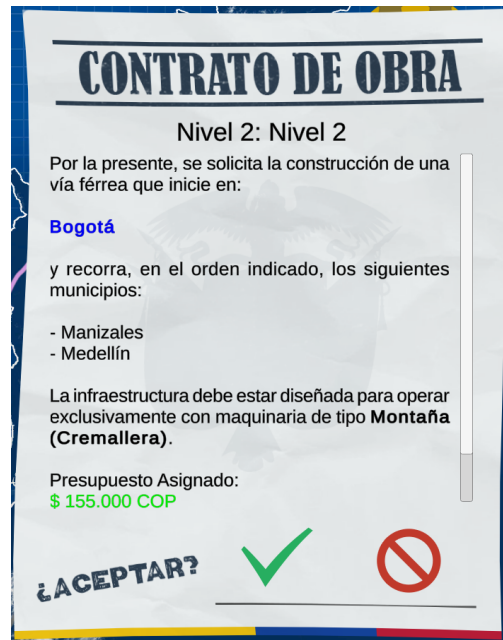


Figura 37: Ficha de contrato que establece los objetivos y el presupuesto del nivel.

4.3.10. Puntuación. El sistema de evaluación calcula el desempeño del jugador mediante un algoritmo de suma ponderada que integra cuatro variables críticas: tiempo, capacidad de carga (vagones), eficiencia presupuestal e integridad del tren. La puntuación total P_{total} se obtiene mediante la Ecuación ecuación (4.2), la cual normaliza el resultado y aplica un factor de escala de 10^6 para presentar el resultado como un número entero de alto valor.

$$P_{total} = [w_t \cdot S_{tiempo} + w_v \cdot S_{vagones} + w_p \cdot S_{presupuesto} + w_s \cdot S_{salud}] \times 10^6 \quad (4.2)$$

Donde w_x representan los pesos o coeficientes de importancia asignados a cada factor según el diseño del nivel, y S_x son los sub-puntajes normalizados calculados de la siguiente manera:

1. Factor Tiempo (S_{tiempo}) Penaliza el exceso de tiempo respecto al objetivo mediante una

función inversamente proporcional:

$$S_{tiempo} = \frac{1}{1 + \frac{T_{usado}}{T_{objetivo}}} \quad (4.3)$$

2. Factor Vagones ($S_{vagones}$) Premia el transporte de mayor carga, normalizado sobre una base de 3 vagones estándar:

$$S_{vagones} = \text{clamp}_0^1 \left(\frac{N_{vagones}}{3} \right) \quad (4.4)$$

3. Factor Presupuesto ($S_{presupuesto}$) Evalúa la precisión en el gasto, penalizando la desviación porcentual respecto al presupuesto asignado (error relativo):

$$S_{presupuesto} = \text{máx} \left(0, 1 - \frac{|D_{gastado} - D_{asignado}|}{D_{asignado}} \right) \quad (4.5)$$

4. Factor Salud (S_{salud}) Representa el porcentaje de integridad restante del tren al llegar a la meta:

$$S_{salud} = \frac{V_{actual}}{V_{maxima}} \quad (4.6)$$

4.3.10.1. Sistema de Estrellas. La puntuación numérica P_{total} se complementa con un sistema de calificación cualitativa visualizado mediante iconos de estrellas. Este esquema evalúa cuatro dimensiones independientes del desempeño del jugador, permitiendo identificar áreas específicas de mejora en la gestión del proyecto ferroviario.

Los criterios de obtención para cada insignia son:

1. Estrella por Completado (Cumplimiento): Es la distinción base que se otorga automáticamente al superar los requisitos mínimos del contrato, es decir, conectar el origen con los destinos correspondientes.
2. Estrella de Presupuesto (Eficiencia Económica): Evalúa la gestión financiera. Se obtie-

ne únicamente si el jugador logra finalizar la obra manteniendo un saldo positivo en la billetera virtual ($Presupuesto_{final} > 0$).

3. Estrella de Vida (Seguridad Operativa): Premia la integridad de la maquinaria. Para conseguirla, el tren debe llegar a la meta sin haber perdido vida por atravesar zonas de daño durante el trayecto.
4. Estrella de Tiempo (Eficiencia Temporal): Se otorga al cumplir el objetivo de tiempo ideal estipulado en el contrato, lo cual exige una planificación de ruta óptima y una velocidad de locomotora adecuada.

Finalmente, la figura 38 muestra la pantalla de consolidación de éxito, donde se visualiza el puntaje numérico global junto con la cantidad de estrellas obtuvidas.



Figura 38: Pantalla de victoria con la calificación obtenida según los cuatro criterios.

4.3.11. Pruebas y Análisis de Resultados. Para validar el prototipo como herramienta de aprendizaje y software funcional, se realizó una prueba unitaria con 28 voluntarios de la comunidad universitaria. El proceso de recolección de datos siguió un enfoque mixto: los datos cuantitativos fueron extraídos directamente de la API de informes (telemetría), mientras que los cualitativos se obtuvieron mediante una encuesta de percepción aplicada tras la sesión de juego.

4.3.12. Encuesta. Se aplicó una encuesta desarrollada en Google Forms a los 28 voluntarios que participaron en las pruebas. Este instrumento permitió contrastar el comportamiento registrado por el sistema con la experiencia subjetiva del usuario, evaluando la efectividad del videojuego como herramienta de aprendizaje GBL. La encuesta, compuesta por 30 preguntas, se enfocó en la claridad de las mecánicas y la comprensión de los retos de la ingeniería ferroviaria en Colombia.

En la figura 39 se observa el formato del instrumento utilizado para la recolección de datos cualitativos.

Evaluación de experiencia de usuario - Ferrocolombia

Esta encuesta ha sido elaborada por los autores del proyecto "Prototipo de un videojuego serio sobre sistemas ferroviarios en Colombia", desarrollado en la Escuela de Ingeniería de Sistemas de la Universidad Industrial de Santander. Su propósito es obtener una noción sobre el nivel de aceptación del videojuego serio desarrollado y evaluar su aporte al aprendizaje acerca de las dificultades y desafíos que enfrentan las vías férreas en el país.

La encuesta es completamente anónima y no se solicitarán datos personales. Se invita a los participantes a responder con total sinceridad las preguntas que se presentan a continuación, ya que sus respuestas son fundamentales para el análisis y mejoramiento del proyecto.

Es importante tener en cuenta que la encuesta debe diligenciarse después de haber participado en al menos una sesión del videojuego serio, de manera que las respuestas reflejen la experiencia directa con la herramienta.

Agradecemos de antemano su colaboración y disposición para contribuir al desarrollo de este proyecto.

* Indica que la pregunta es obligatoria

¿En qué rango de edad se encuentra? *

10 - 14

15 - 17

Figura 39: Formulario de evaluación de la experiencia de usuario y aprendizaje.

Las preguntas que integraron el instrumento de evaluación se presentan a continuación:

1. ¿En qué rango de edad se encuentra?
2. ¿Ha jugado videojuegos antes?
3. Antes de jugar ¿Conocía algo sobre trenes o vías férreas?
4. ¿Qué tan clara fue su primera experiencia al empezar el juego?
5. ¿El estilo visual (colores, íconos) le pareció atractivo?
6. ¿Le pareció que el mapa refleja la complejidad geográfica de Colombia (montañas, valles, ríos)?
7. ¿Le pareció que entendió los conceptos básicos gracias al tutorial?
8. Después de terminar el tutorial, ¿se sintió seguro/a para jugar solo/a sin necesidad de volver a jugarlo para entender?
9. ¿Qué tan fáciles fueron de entender los controles?
10. ¿Qué tan intuitivo le resultó el sistema de trazado de vías?
11. ¿Tuvo problemas con la cámara o el movimiento?
12. Al construir sobre terrenos montañosos, ¿le pareció que el sistema de puentes y túneles se generó de forma lógica?
13. ¿Cómo calificaría la respuesta de los vagones al seguir la locomotora en curvas pronunciadas?
14. ¿Le pareció que el objetivo del nivel estaba claro?
15. ¿Qué tan fácil es identificar el presupuesto y los recursos disponibles mientras construye?
16. ¿Le pareció justa la manera en la que el juego decide si gana o pierde?
17. ¿Entendió el sistema de puntuación?

18. ¿Qué tan divertido le pareció el juego?
19. ¿Qué parte fue su favorita?
20. ¿Qué aspectos del juego le generaron frustración?
21. ¿Tuvo alguna dificultad para leer los textos o distinguir los elementos del mapa debido al tamaño o los colores?
22. ¿Notó caídas de fotogramas durante el juego (imagen congelada, cortes de imagen)?
23. ¿Encontró algún error que detuviera su juego o algo que se viera roto?
24. ¿Qué cambiaría para que la experiencia sea mejor?
25. ¿Encontró algún fallo durante su experiencia? por favor, cuéntenos
26. ¿Le pareció que tenía que planear antes de construir, y no solo colocar vías al azar?
27. ¿Cómo calificaría el impacto de la planificación en sus resultados?
28. Después de jugar, ¿cree que construir un tren entre ciudades lejanas en Colombia es:
29. ¿En qué le pareció que se iba más dinero y esfuerzo?
30. Tras la experiencia de juego, ¿qué reflexiones tuvo adquirió respecto a la problemática del desarrollo de infraestructura férrea en el país?

4.3.13. Análisis de Telemetría (Datos de la API). El uso de telemetría permitió registrar el comportamiento objetivo de los usuarios, eliminando los sesgos de la autopercepción y proporcionando una visión clara sobre la jugabilidad y la curva de dificultad.

4.3.13.1. Retención y Progresión: El desafío de la curva de aprendizaje. La figura 40 muestra el volumen de jugadores únicos que accedieron a cada nivel. El registro indica que 28 usuarios iniciaron el Nivel 1. En el Nivel 2 la participación fue de 9 usuarios, en el

Nivel 3 de 8 usuarios, en el Nivel 4 de 5 usuarios, y finalmente, un único jugador alcanzó el Nivel 5.

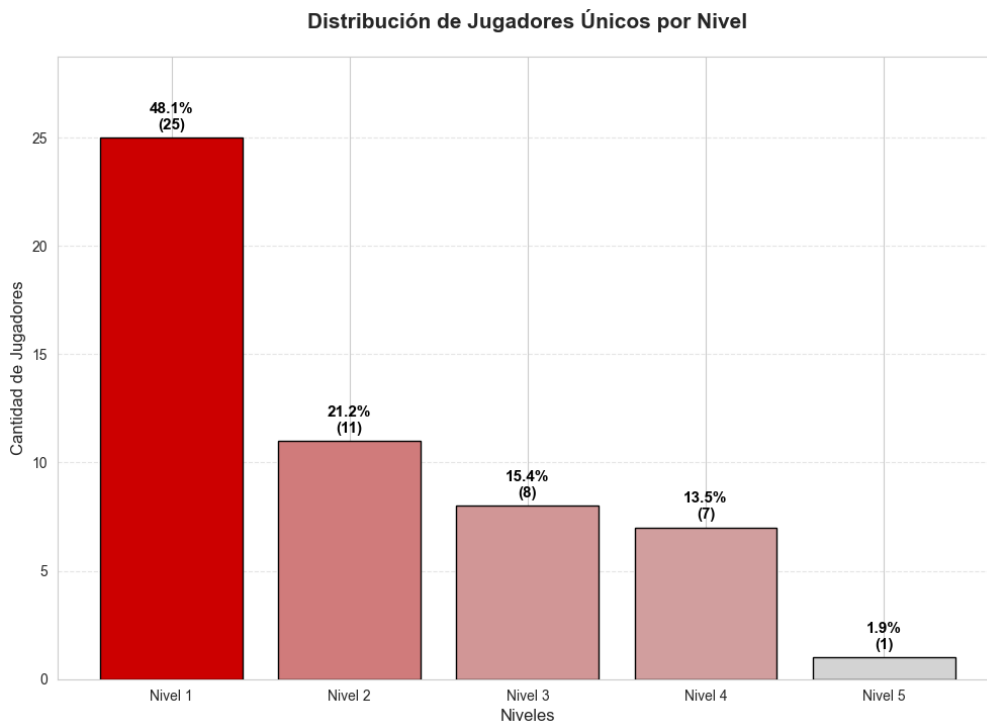


Figura 40: Tasa de retención de jugadores únicos por nivel.

4.3.13.2. Balance de Dificultad: Victorias frente a Derrotas. la figura 41 detalla la proporción de intentos exitosos y fallidos. El Nivel 1 presenta una tasa de derrota del 71.4 %. En el Nivel 2, el porcentaje de éxito fue del 66.7 %, mientras que en el Nivel 3 las derrotas representaron el 69.2 %. Los niveles finales (4 y 5) registraron tasas de fracaso del 89.3 % y 87.5 % respectivamente sobre el total de intentos.

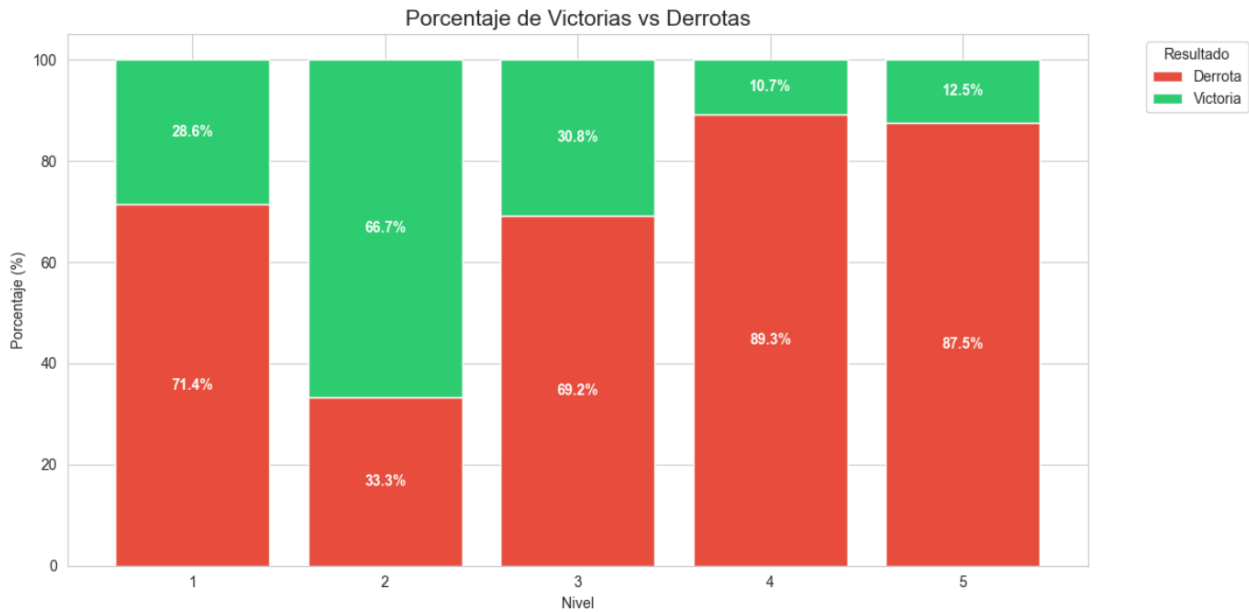


Figura 41: Proporción de victorias frente a derrotas por nivel.

4.3.14. Resultados de la Encuesta de Percepción. La encuesta permitió evaluar si los objetivos de concientización y usabilidad definidos en el diseño se cumplieron satisfactoriamente desde la perspectiva del usuario.

4.3.14.1. Usabilidad y Experiencia de Usuario (UX). La percepción sobre los controles fue mayoritariamente positiva, con un 81.9% de aprobación (figura 42). No obstante, existe un 9% de usuarios que reportaron dificultades.

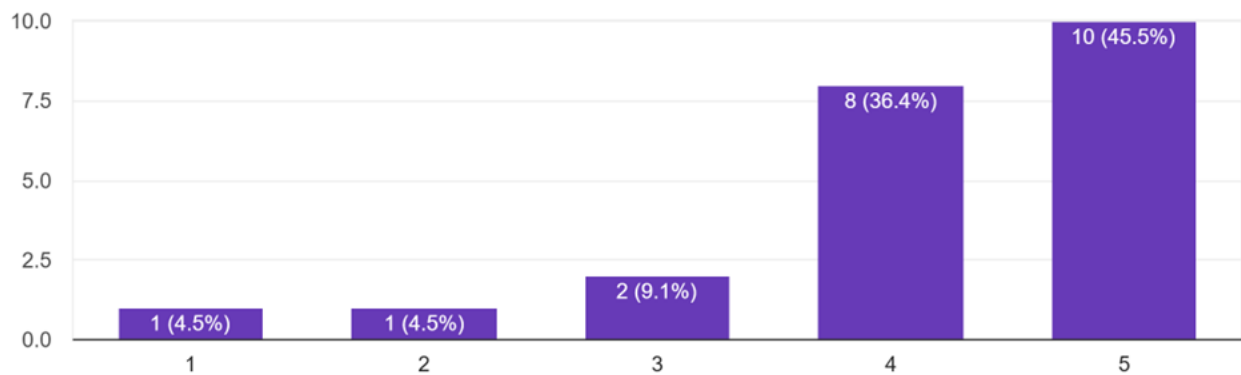


Figura 42: Nivel de facilidad percibida en los controles.

La claridad sobre la primera experiencia de los jugadores fue mayormente favorable, con un 54.5 % de los encuestados reportando que fue clara (figura 43). No obstante existe un 9 % de usuarios que reportaron que era confusa o muy confusa.

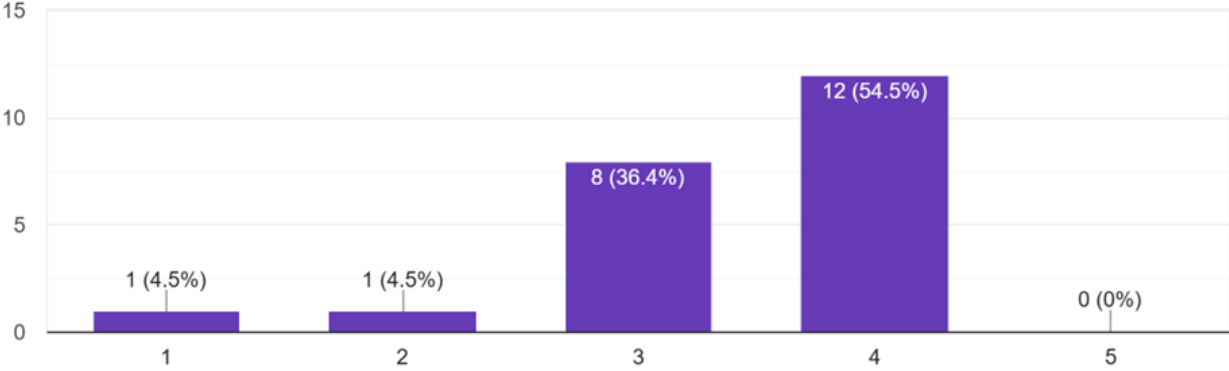


Figura 43: Respuestas a la pregunta 4

4.3.14.2. Concientización y Percepción del Aprendizaje (GBL). Esta sección constituye el núcleo de la validación del aprendizaje. Como indica la figura 44, un abrumador 81.8 % de los participantes concluyó que la construcción ferroviaria en Colombia es *complicada y costosa*.

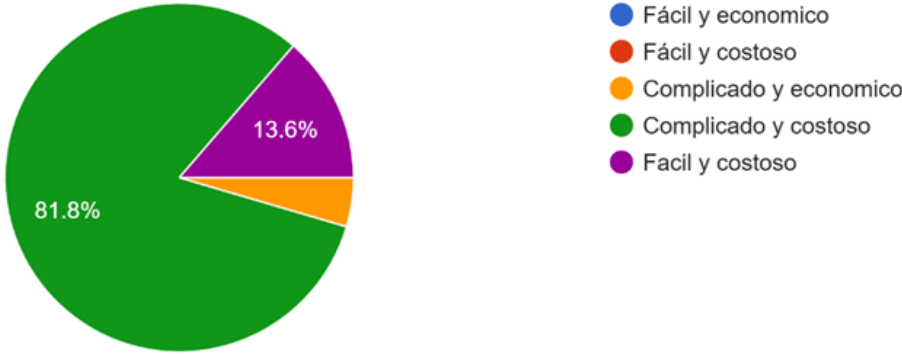


Figura 44: Percepción de la complejidad constructiva tras la experiencia de juego.

En las figuras 45 y 46 se muestran algunas de las respuestas de la preguntas 29 y 30. Esta pregunta fue hecha con énfasis en responder el objetivo del proyecto, el cual involucra saber el impacto que tuvo en los jugadores el videojuego serio.

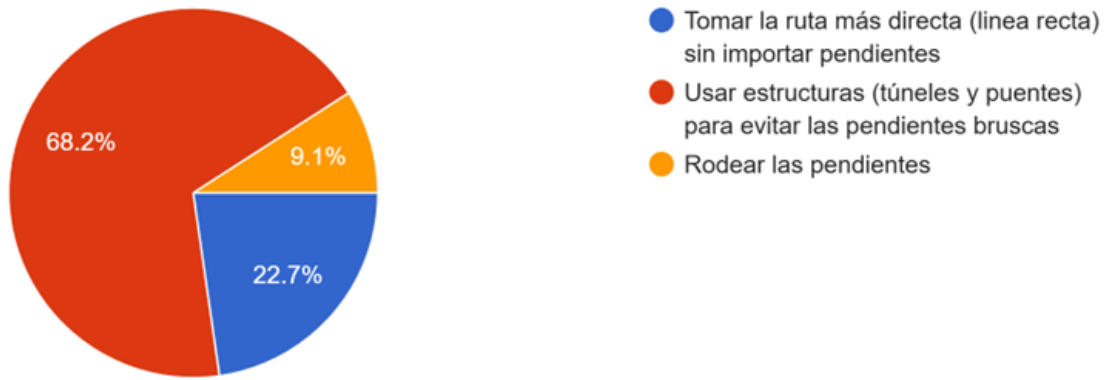


Figura 45: Respuestas a la pregunta 29

Tras la experiencia de juego, ¿qué reflexiones adquirió respecto al desarrollo de infraestructura férrea en el país?

20 respuestas

La dificultad de construcción de este medio
He estudiado un poco del tema y me anima mucho que se vuelva a tocar la importancia de las vías férreas y su respectivo uso en Colombia. Si algún día llega a resurgir, se economizarán muchos trayectos.
Es un tema de mucho conocimiento. Verdaderos maestros en ese ámbito son los que logran estas vías en un país tan complicado geográficamente
Por lo que veo la mayor dificultad sería planear el camino que va a tomar el ferrocarril, esto sumado al tiempo que llevaría construir las vías me hace pensar que más que un trabajo difícil o costoso, es uno que lleva MUCHO tiempo
Sí bien hay formas de hacerlo, hay muchas cosas que tener en cuenta... Presupuesto, terreno, hay muchas cosas que influyen; se puede, sí, pero es necesario una buena planeación antes de tirar pico y pala
La gran complejidad geográfica del territorio colombiano y como este afecta de forma directa las ferrovías del país

Figura 46: Reflexión final sobre los desafíos del desarrollo ferroviario en Colombia.

4.4. ERRORES DETECTADOS

Durante la prueba unitaria, se reportaron los siguientes fallos que requieren corrección:

1. Colisiones fantasma: El tren detectó daño en el 2 % de las partidas sin colisión visual

clara. Se deben refinar los *colliders*.

2. Renderización a larga distancia: Los ríos y el terreno no se visualizan adecuadamente al alejar la cámara; se recomienda aumentar la distancia de renderizado.
3. Don Raúl en el menú: El personaje reaparece tras completar el tutorial, indicando que la variable `tutorialMenuCompletado` no cambia su estado a `True`.
4. Cámara en nivel 5: La cámara inicia erróneamente en el punto final del recorrido en lugar del inicial, generando frustración.

5. CONCLUSIONES

Tras la realización del prototipo del videojuego serio, y de las pruebas, se pueden determinar las siguientes conclusiones:

1. Los criterios GBL fueron una buena estandarización de lo que es realmente un juego educativo, y ayudó mucho a fijar objetivos y artefactos claros como fue el caso de las historias de usuario.
2. La estructura y la metodología usadas durante todo el proyecto ayudaron favorablemente al ritmo del ciclo de vida del videojuego serio. Ya que, a pesar de haber dificultades técnicas y retrasos durante el ciclo, los artefactos generados ayudaron a recuperar el trayecto fácilmente.
3. La arquitectura de escenas aditivas para el desarrollo del videojuego serio en Unity fue una decisión diferenciadora ya que permitió una mejora del rendimiento de carga de escenas y proporcionó una mejor organización de las mismas.
4. Durante las pruebas de usuarios se observó que los jugadores tienden a jugar con la cámara alejada del terreno, esto puede indicar muchas cosas como, la falta de poder ubicar más fácilmente los municipios, la molestia por tener que moverse tanto para hacer la vía, o se les hace más cómodo tener una vista amplia del terreno.
5. Según la figura 40, 28 usuarios que probaron el videojuego serio, pasaron por el nivel 1, 9 personas pasaron por el nivel 2, 8 personas pasaron por el nivel 3, 5 personas pasaron por el nivel 4 y finalmente 1 persona pasó por el nivel 5, esto demuestra una disminución en la retención de los jugadores.
6. Como se puede observar en la figura 41 la proporción de victorias frente a derrotas por nivel demuestra la curva de dificultad, siendo el nivel 1 un nivel con un alto índice de derrota al ser un nivel introductorio, mientras que el nivel 2 demuestra una reducción

considerable de dicho índice y a medida que pasan los niveles la dificultad aumenta, hasta normalizarse en el nivel 4.

7. En la pregunta de la figura 46, las respuestas demostraron que resultados de la prueba piloto evidenciaron una concientización de la problemática ferroviaria Colombiana, además, tal como se vio en la figura 44, el 81,8 % de los participantes identificó la dificultad en la infraestructura férrea como una tarea compleja y costosa debido al impacto de la topografía y el uso necesario de túneles y puentes.
8. La identificación de los túneles y puentes como los mayores focos de gasto (figura 45) confirma que el usuario comprendió el impacto de la topografía. Finalmente, las respuestas abiertas (figura 46) reflejan que los jugadores ahora valoran el desarrollo férreo no solo como un deseo político, sino como un reto de ingeniería de gran magnitud, cumpliendo así el objetivo principal de este proyecto de grado.
9. En las respuestas de la pregunta 9 de la figura 42 (¿Que tan fáciles fueron de entender los controles?) se observó que el porcentaje de jugadores que manifestaron haber tenido problemas con los controles, son el mismo porcentaje que manifestaron haber tenido una confusa primera experiencia de juego en la pregunta 4 de la figura 43.

6. TRABAJO FUTURO

Para expandir el alcance del videojuego serio, se proponen las siguientes líneas de trabajo:

1. Expansión del contenido histórico: Incluir más narrativa sobre la historia real de los Ferrocarriles Nacionales de Colombia, desbloqueable mediante coleccionables en el mapa y mediante diálogos proporcionados por Don Raíl.
2. Expansión del contenido jugable: Incorporar niveles adicionales con una progresión de dificultad más gradual, de modo que la transición entre etapas sea más equilibrada y se eviten saltos bruscos en el reto para el jugador.
3. Modo "Sandbox": Habilitar un modo sin restricciones económicas para fomentar la creatividad libre del usuario.
4. Portabilidad WebGL: Optimizar los recursos gráficos para permitir la ejecución del juego directamente en navegadores web, facilitando su acceso en entornos educativos sin necesidad de instalación.
5. Mejoras de calidad de vida: Implementar diversas optimizaciones orientadas a la experiencia del usuario, tales como la inclusión de un cronómetro al inicio del recorrido del tren, una mayor visibilidad en las zonas de daño, opciones de configuración de pantalla que permitan elegir entre modo ventana o pantalla completa, así como la incorporación de atajos de teclado para agilizar acciones frecuentes (por ejemplo, utilizar `Ctrl+Z` para borrar el último tramo creado).
6. Mejora de tutorial: Aumentar y mejorar la calidad de los tutoriales impartidos por Don Raíl, con el objetivo de que más gente comprenda correctamente el funcionamiento y manejo del sistema.

7. Descarga desde plataformas confiables: Publicar el juego en sitios reconocidos de distribución digital, como *Steam* o *itch.io*, con el fin de facilitar el acceso a las descargas y permitir que un mayor número de usuarios lo pruebe.
8. IP e identificador en la base de datos: Incorporar en la base de datos los campos correspondientes a la dirección IP y al nombre de usuario, con el propósito de optimizar la retroalimentación hacia el jugador y facilitar la consulta de su información.
9. Informes desde la web: Implementar un sistema de elaboración de informes a través de una página web, con el fin de facilitar su generación y acceso.

BIBLIOGRAFÍA

- Agencia Nacional de Infraestructura. *Ferrocarriles - fichas de proyecto (ANISCOPIO)*. 2025. URL: <https://aniscopio.ani.gov.co/ferrocarriles-public/ficha-proyecto-drupal> (visitado 01-03-2025).
- Alcaldía Mayor de Bogotá. *En abril inician las obras del corredor férreo de Regiotram de Occidente*. 2025. URL: <https://bogota.gov.co/mi-ciudad/movilidad/obras-del-regiotram-de-occidente-que-beneficia-bogota-inician-abril> (visitado 06-06-2025).
- Anderson, David J. *Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results*. Prentice Hall Professional, 2003.
- Arévalo Caicho, Luis F. y M. A. Urgiléz Zabala. «Análisis de usabilidad e interactividad en interfaces hápticas para sistemas 3D, aplicación a maqueta interactiva de transporte ESPOCH». Tesis de Grado. Riobamba, Ecuador: Escuela Superior Politécnica de Chimborazo, 2014.
- Autodesk. *Autodesk 3ds Max*. 2025. URL: <https://www.autodesk.com/latam/products/3ds-max/overview> (visitado 06-06-2025).
- *Autodesk Maya*. 2025. URL: <https://www.autodesk.com/mx/products/maya/overview> (visitado 06-06-2025).
- Beck, Kent et al. *Manifiesto por el Desarrollo Ágil de Software*. 2001. URL: <https://agilemanifesto.org/iso/es/manifiesto.html> (visitado 06-06-2025).
- Blender. *Manual de introducción de Blender (v2.82): Sobre Blender*. 14 de feb. de 2020. URL: https://docs.blender.org/manual/es/2.82/getting_started/about/introduction.html (visitado 06-06-2025).
- Blender Foundation. *Acerca de Blender (licencia GPL)*. 2025. URL: <https://www.blender.org/about/> (visitado 06-06-2025).
- *Blender*. 2025. URL: <https://www.blender.org> (visitado 06-06-2025).
- Caicedo Navas, D. F. et al. *Informe de ponencia para segundo debate del Proyecto de Ley número 337 de 2023 Cámara...* 30 de abr. de 2024.

Cataldi, Z. et al. *Ingeniería de Software Educativo*. 1999.

Champin, Juan et al. *Desafíos del transporte ferroviario de carga en Colombia*. Inter-American Development Bank, abr. de 2016. URL: <https://publications.iadb.org/publications/spanish/document/Desaf%C3%ADos-del-transporte-ferroviario-de-carga-en-Colombia.pdf>.

Davydov, Alexey et al. *Train Valley 2*. 2019. URL: https://store.steampowered.com/app/602320/Train_Valley_2/ (visitado 06-06-2025).

Edin Zulich. *Amazon DynamoDB: Gaming use cases and design patterns*. 2 de abr. de 2019. URL: <https://aws.amazon.com/es/blogs/database/amazon-dynamodb-gaming-use-cases-and-design-patterns/> (visitado 14-01-2026).

Epic Games. *Unreal Engine*. 2025. URL: <https://www.unrealengine.com/es-ES> (visitado 06-06-2025).

– *Unreal Engine Features (Nanite, Virtual Shadow Maps)*. 2025. URL: <https://www.unrealengine.com/es-ES/features> (visitado 06-06-2025).

eWeek. *IBM Acquires Rational*. 6 de dic. de 2002. URL: <https://www.eweek.com/pc-hardware/ibm-acquires-rational/> (visitado 06-06-2025).

Gaming Minds Studios. *Railway Empire*. 2018. URL: https://store.steampowered.com/app/503940/Railway_Empire/ (visitado 06-06-2025).

GEBCO. *General Bathymetric Chart of the Oceans*. 2025. URL: <https://www.gebco.net/data-products/gridded-bathymetry-data> (visitado 14-01-2026).

Hogle, Jan G. *Considering Games as Cognitive Tools: In Search of Effective “Edutainment”*. University of Georgia, 1996.

Meristation. *Cómo Unreal Engine se usa en la industria del cine (The Mandalorian)*. 19 de jun. de 2022. URL: https://as.com/meristation/2022/06/19/reportajes/1655613468_465907.html (visitado 06-06-2025).

Ministerio de Transporte. *Gobierno Nacional firma contrato de concesión de la primera APP férrea de Colombia en el corredor La Dorada–Chiriguaná*. Mayo de 2025. URL: <https://mintransporte.gov.co/publicaciones/12015/gobierno-nacional-firma-contrato-de->

- concesion-de-la-primera-app-ferrea-de-colombia-en-el-corredor-la-dorada-chiriguana/
(visitado 06-06-2025).
- Ministerio de Transporte. *Proyecto de Decreto/Resolución: Asignación de surcos ferroviarios en la Red Férrea Nacional*. 15 de mayo de 2024. URL: <https://mintransporte.gov.co/loader.php?IServicio=Tools2&ITipo=descargas&IFuncion=descargar&idFile=47723>
(visitado 06-06-2025).
- *Transporte en Cifras*. 2023. URL: <https://mintransporte.shinyapps.io/transporteencifrasapp/>
(visitado 06-06-2025).
- Nadolny, L. et al. «Examining the characteristics of game-based learning: A content analysis and design framework». En: *Computers & Education* 156 (2020). DOI: 10.1016/j.compedu.2020.103936.
- Plass, Jan L., Richard E. Mayer y Bruce D. Homer. *Handbook of Game-Based Learning*. MIT Press, 2019.
- PopTop Software. *Railroad Tycoon 3*. 2003. URL: https://www.gog.com/en/game/railroad_tycoon_3 (visitado 06-06-2025).
- Pressman, Roger S. *Ingeniería del Software: Un Enfoque Práctico*. 5a ed. adaptada. Madrid: McGraw-Hill España, 2001.
- Ritterfeld, Ute, Michael Cody y Peter Vorderer. *Serious Games: Mechanisms and Effects*. Taylor & Francis, 2009.
- Unity Technologies. *Nuestra compañía*. 2025. URL: <https://unity.com/es/our-company>
(visitado 06-06-2025).
- *Planes y precios de Unity*. 2025. URL: <https://unity.com/es/products> (visitado 06-06-2025).
- *Unity*. 2025. URL: <https://unity.com> (visitado 06-06-2025).
- *Virtual Reality Surgical Simulation Suite (VR3S)*. 9 de ene. de 2024. URL: <https://unity.com/resources/virtual-reality-surgical-simulation-suite> (visitado 06-06-2025).
- Valora Analitik. *Así es el megaproyecto del Tren de Cercanías del Valle*. 23 de oct. de 2024. URL: <https://www.valoraanalitik.com/megaproyecto-tren-de-cercanias-del-valle-asi-es/>
(visitado 06-06-2025).