



TOKIO NEW TECHNOLOGY SCHOOL

**MÁSTER EN PROGRAMACIÓN DE VIDEOJUEGOS Y
REALIDAD VIRTUAL CON UNITY**

PROMETHEUS-9

Videojuego narrativo de ciencia ficción en primera persona

Sandra Ciudad Moreno

Abril, 2025

RESUMEN

Prometheus-9 es un videojuego narrativo de ciencia ficción desarrollado en Unity que combina exploración, puzzles y decisiones morales para contar una historia inmersiva en un entorno espacial. El jugador despierta solo en una nave averiada, sin rastro de la tripulación, y debe restaurar sistemas, descubrir grabaciones del pasado y enfrentarse a una decisión final que determinará el desenlace de la misión. A lo largo de 10 misiones, se desbloquean fragmentos narrativos en forma de cintas de vídeo que revelan una historia de sabotaje, contacto alienígena y dilemas éticos sobre la supervivencia.

El proyecto fue concebido como una experiencia jugable completa con enfoque en la integración entre narrativa y mecánicas. Cada misión representa tanto un reto jugable como una pieza de la historia, construyendo un flujo progresivo donde jugabilidad y narrativa se retroalimentan. El jugador explora distintas zonas de la nave, resuelve desafíos técnicos, recolecta objetos y escucha grabaciones que humanizan a los personajes ausentes, culminando en una toma de decisión con tres posibles finales.

Desde el punto de vista técnico, se aplicaron buenas prácticas de ingeniería de software, incluyendo una arquitectura modular en C#, patrones de diseño como Singleton, controladores independientes por misión y sistemas de inventario, UI y cámaras bien estructurados. El desarrollo siguió metodologías ágiles adaptadas a un entorno unipersonal, lo que permitió iteraciones frecuentes, pruebas constantes y corrección de errores durante todo el proceso.

El juego está acompañado por un sistema de gamificación sutil: iconos de oxígeno y energía, recompensas narrativas, penalizaciones lógicas y una interfaz clara que guían al jugador sin romper la inmersión. La ambientación sonora y visual refuerza la sensación de soledad y misterio, y el ritmo está cuidadosamente medido para mantener la tensión narrativa.

En conclusión, *Prometheus-9* es un proyecto que demuestra cómo, con recursos limitados pero planificación y visión clara, se puede construir una experiencia interactiva rica en contenido narrativo, emocionalmente impactante y técnicamente sólida. El resultado final no solo cumple sus objetivos, sino que deja abierta la puerta a futuras ampliaciones y evoluciones dentro del mismo universo de ficción.

ABSTRACT

Prometheus-9 is a narrative science fiction video game developed in Unity, blending exploration, puzzles, and moral decision-making to tell an immersive story set in a deep-space environment. The player wakes up alone aboard a damaged spaceship, with no sign of the crew, and must restore systems, uncover past events through security recordings, and ultimately face a final decision that will determine the fate of the mission. Across 10 missions, players unlock narrative fragments through video logs that reveal a story of sabotage, alien contact, and ethical dilemmas regarding survival.

The project was conceived as a complete playable experience, with a strong focus on the integration between gameplay mechanics and storytelling. Each mission offers both a mechanical challenge and a narrative piece, building a progressive flow where gameplay and narrative enhance each other. The player explores different areas of the ship, solves technical problems, collects items, and listens to recordings that bring depth to the absent characters, culminating in a meaningful final choice with three different endings.

From a technical perspective, the game applies solid software engineering practices, including a modular architecture in C#, design patterns like Singleton, independent mission controllers, and well-structured systems for inventory, UI, and camera control. The development followed agile methodologies adapted to a solo developer workflow, enabling constant iteration, testing, and debugging throughout the process.

Gamification elements are subtly implemented: energy and oxygen icons, narrative rewards, logical penalties, and a clean user interface guide the player without breaking immersion. The audiovisual atmosphere reinforces a sense of isolation and mystery, with carefully measured pacing to maintain narrative tension.

In conclusion, *Prometheus-9* demonstrates that with limited resources but strong planning and a clear creative vision, it is possible to build an interactive experience that is narratively rich, emotionally impactful, and technically robust. The final result not only fulfills its objectives but also opens the door to future expansions and deeper explorations within the same fictional universe.

ÍNDICE

RESUMEN	3
ABSTRACT	5
CAPÍTULO 1. INTRODUCCIÓN.....	10
1.1. EL AUGE DE LOS VIDEOJUEGOS: CONTEXTO CULTURAL Y TECNOLÓGICO	10
1.2. CONTEXTO DEL PROYECTO: EVOLUCIÓN DEL MEDIO Y TENDENCIA DE LOS JUEGOS NARRATIVOS	11
1.3. ALCANCE DEL PROYECTO	13
1.4. MOTIVACIÓN Y SOLUCIÓN PROPUESTA	13
1.5. ESTRUCTURA DE LA MEMORIA.....	15
CAPÍTULO 2. OBJETIVOS DEL PROYECTO Y HERRAMIENTAS UTILIZADAS	17
2.1. OBJETIVOS TÉCNICOS, NARRATIVOS Y METODOLÓGICOS DEL DESARROLLO	17
2.2. HERRAMIENTAS UTILIZADAS	19
2.2.1. Motor de desarrollo: Unity	19
2.2.2. Lenguaje de programación: C#	19
2.2.3. Entorno de desarrollo: Visual Studio.....	20
2.2.4. Control de versiones: Git y GitHub	21
CAPÍTULO 3. FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE	22
3.1. ¿QUÉ SON LOS JUEGOS NARRATIVOS?	22
3.1.1. Definición y evolución histórica	22
3.1.2. Ventajas de los juegos narrativos	22
3.1.3. Narrativa emergente vs. narrativa estructurada.....	23
3.1.4. Diseño narrativo en videojuegos	23
3.1.5. Elementos clave de un juego narrativo	24
3.2. GAMIFICACIÓN	25
3.2.1. Origen y concepto de gamificación	25
3.2.2. Principales mecánicas de gamificación	25
3.2.3. Tipos de motivación en gamificación	27
3.2.4. Gamificación como diseño consciente en Prometheus-9	28
CAPÍTULO 4. METODOLOGÍA DE TRABAJO	29
4.1. METODOLOGÍAS ÁGILES APLICADAS.....	29
4.1.1. Contexto y necesidad de un enfoque ágil.....	29
4.1.2. Aplicación específica de Scrum en el proyecto	29
4.2. ARQUITECTURA DEL SISTEMA	31
4.2.1. Programación orientada a componentes.....	31
4.3. EVOLUCIÓN DEL PROYECTO	34
4.3.1. Anteproyecto	34

4.3.2. Estudio Teórico	35
4.3.3. Diseño.....	35
4.3.4. Implementación	37
4.3.5. Pruebas.....	54
4.3.6. Documentación	55
4.3.7. Gráfica complejidad-tiempo del proyecto	56
CAPÍTULO 5. RESULTADOS	58
5.1. DISEÑO DEL JUEGO	58
5.1.1. Descripción del juego.....	58
5.1.2. Jugabilidad centrada en la narrativa	59
5.1.3. Historia	62
5.1.4. Interfaz.....	73
5.2. ARQUITECTURA FINAL DEL JUEGO	74
5.2.1. Inventario	74
5.2.2. Grabaciones	75
5.2.3. Misiones.....	75
5.2.4. Jugador	77
5.2.5. Estado del juego	78
CAPÍTULO 6. CONCLUSIONES Y PROPUESTAS DE TRABAJO FUTURO.....	80
6.1. CONCLUSIONES GENERALES.....	80
6.1.1. Unir narrativa y jugabilidad	80
6.1.3. La narrativa como motor emocional.....	80
6.2. OBJETIVOS ALCANZADOS.....	81
6.2.1. Revisión exhaustiva de los objetivos propuestos	81
6.3. COMPETENCIAS ADQUIRIDAS	82
6.3.1. Competencias técnicas avanzadas.....	82
6.3.2. Competencias creativas y de diseño narrativo	82
6.3.3. Competencias organizativas y de gestión de proyecto.....	82
6.4. LIMITACIONES TÉCNICAS Y NARRATIVAS	83
6.4.1. Alcance narrativo limitado.....	83
6.4.2. Recursos visuales genéricos.....	83
6.4.3. Inteligencia Artificial limitada.....	83
6.4.4. Duración corta.....	83
6.5. TRABAJO FUTURO	84
6.5.1. Ampliación narrativa	84
6.5.2. Mejora de la estructura jugable	84
6.5.3. Evolución audiovisual	84
6.6. CONCLUSIÓN FINAL.....	84

CAPÍTULO 1. INTRODUCCIÓN

El propósito de este capítulo es detallar el contexto en el que se va a desarrollar este Trabajo de Fin de Máster, presentando su motivación, objetivos generales y la estructura del documento. Además, se describe el contexto narrativo del juego y se introduce la problemática que motiva su desarrollo.

1.1. EL AUGE DE LOS VIDEOJUEGOS: CONTEXTO CULTURAL Y TECNOLÓGICO

El sector de los videojuegos se ha convertido en una de las industrias de entretenimiento más importantes a nivel global en las últimas décadas. De ser un pasatiempo de nicho ha pasado a superar en ingresos a industrias tradicionales como son el cine y la música. En el año 2023, la industria mundial del videojuego generó cerca de 188 mil millones de dólares, en contraste con los 118 mil millones que pudieron generar en conjunto las industrias del cine y la música ese mismo año.

En términos de alcance, se estima que existen más de 3.300 millones de jugadores activos a nivel mundial en 2024, lo que supone casi la mitad de la población global. Esta base de usuarios ha ido creciendo de manera sostenida en los últimos años, impulsada en gran parte por la accesibilidad de los juegos en dispositivos móviles, la distribución digital global y la creciente relevancia cultural del *gaming*.

En la Figura 1. se puede observar el crecimiento estimado de la cantidad de jugadores de videojuegos activos a nivel mundial entre 2015 y 2024. Es fácil percibir un incremento de más del 60% en menos de una década, pasando de unos 2 mil millones de jugadores en el año 2015 a más de 3.3 mil millones en el año 2024.

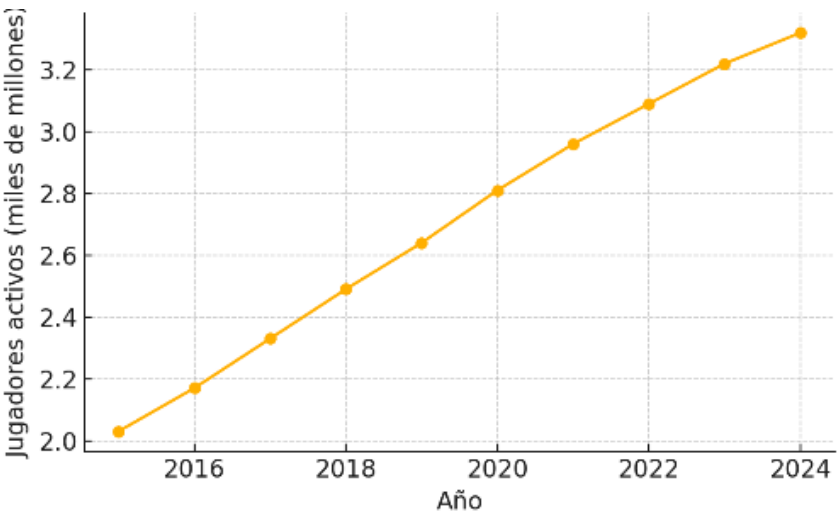


Figura 1. Crecimiento de jugadores a nivel mundial (2015-2024)

Sin embargo, este auge no es meramente económico, sino también cultural. Los videojuegos son una pieza esencial de la cultura popular digital, equiparable a otros medios como el cine o la literatura en cuanto a la influencia y el impacto que suponen. De hecho, la Unión Europea ha reconocido el valor cultural del videojuego, recomendando su salvaguardia como parte del patrimonio cultural digital. Asimismo, el informe de la Comisión Europea sobre la industria del videojuego en 2022 destacó que en Europa hay más de 237 millones de jugadores de videojuegos de entre 6 y 64 años, cifra que creció significativamente durante la pandemia COVID-19.

Los juegos actuales exploran temáticas complejas, estilos artísticos variados y narrativas profundas, llegando a ser considerados por críticos y jugadores como verdaderas obras de arte interactivas. Un ejemplo destacado es *The Witcher 3: Wild Hunt*, al que se le ha reconocido uno de los mejores guiones jamás escritos para un videojuego, demostrando que una buena historia en un videojuego puede lograr tanto o más impacto emocional que en otros medios.

Centrándonos ahora en el plano tecnológico, son varios los factores que han contribuido a la expansión y sofisticación del medio. La disponibilidad de motores de desarrollo asequibles y potentes, como Unity o Unreal Engine, ha democratizado la creación de videojuegos, permitiendo que estudios independientes innoven en jugabilidad y narrativa con recursos antes impensables. Paralelamente, el avance del hardware amplía las fronteras técnicas y ofrece nuevas formas de contar historias cada vez mucho más inmersivas. Al mismo tiempo, la conectividad de Internet y las plataformas de distribución digital han facilitado enormemente la difusión, haciendo posible que juegos de autor o experimentales encuentren su audiencia. Todos estos elementos conforman un contexto en el que desarrollar un videojuego narrativo original es tanto un desafío como una oportunidad, aprovechando la madurez del medio y las herramientas disponibles para ofrecer experiencias novedosas al público actual.

1.2. CONTEXTO DEL PROYECTO: EVOLUCIÓN DEL MEDIO Y TENDENCIA DE LOS JUEGOS NARRATIVOS

La forma en que los videojuegos cuentan historias ha evolucionado drásticamente desde los inicios. En las primeras décadas, las tramas eran simples excusas para la acción: *Pac-Man* (1980) huye de fantasmas en un laberinto sin mayor trasfondo narrativo. *Super Mario Bros* (1985) presenta un héroe rescatando a una princesa sin profundizar en personajes. Con el tiempo, a medida que la tecnología y el diseño avanzaban, surgieron juegos con narrativas más elaboradas y universos ficcionales ricos. En los años 90, géneros como la aventura gráfica (por ejemplo, *Monkey Island* o *Myst*) pusieron la historia en primer plano, requiriendo al jugador resolver puzzles integrados en una trama. En los 2000, franquicias de rol y acción como *Metal Gear Solid*, *Final Fantasy* o *Half-Life* demostraron que se podían ofrecer argumentos complejos y personajes desarrollados, acercando al videojuego a formas narrativas propias del cine o las novelas gráficas. Esta progresión continúa: hoy en día no es extraño que un videojuego presente arcos argumentales profundos y sofisticados, equiparables a una serie de televisión en calidad narrativa.

La **narrativa digital** se ha consolidado, así como un género en sí mismo, capaz de contar historias con recursos propios del medio interactivo. A diferencia de la literatura o el cine, en el videojuego el público participa activamente en el desarrollo de la historia, lo cual transforma completamente la experiencia narrativa. El jugador deja de ser un espectador pasivo y se convierte en un agente dentro del relato, tomando decisiones, explorando a su propio ritmo e influyendo en el curso de los acontecimientos. Esta interacción distingue a los videojuegos de otras aproximaciones artísticas, y hoy por hoy no se concibe un juego sin algún grado de interactividad narrativa. Gracias a esta capacidad de involucrar al jugador, muchos videojuegos modernos logran un nivel de inmersión emocional difícil de alcanzar en otros medios.

En el contexto actual, existe una clara tendencia hacia juegos centrados en la narrativa, en especial dentro de la escena independiente. Durante la última década han ganado reconocimiento numerosas obras donde la historia y los personajes son el eje principal, por encima de mecánicas tradicionales de competencia o acumulación de puntos. Títulos como *Gone Home* (2013), *Firewatch* (2016), *What Remains of Edith Finch* (2017), *Life is Strange* (2015) o *Oxenfree* (2016) son ejemplos aclamados de experiencias narrativas envolventes, cada una con estilos y temáticas distintas pero unidas por el énfasis en contar una buena historia. En *Gone Home*, por ejemplo, el jugador explora una casa familiar para desentrañar un misterio personal, descubriendo la trama a través de cartas, diarios y pistas ambientales en ausencia de cualquier combate o desafío tradicional. Por su parte, *Life is Strange* incorporó mecánicas de retroceso en el tiempo para permitir al jugador influir en una narrativa ramificada, mientras *What Remains of Edith Finch* presentó una serie de viñetas interactivas, cada una con mecánicas únicas, que narran la historia de varios miembros de una familia. El éxito crítico y comercial de estos juegos confirma que las historias interactivas pueden ser tanto o más influyentes que las experiencias centradas únicamente en la acción.

En esta línea, el presente proyecto se enmarca en la corriente de juegos narrativos, aprovechando la evolución del medio y la receptividad del público hacia propuestas de narrativa interactiva. Se trata de un videojuego de ciencia ficción que busca aunar las lecciones aprendidas de las obras antes mencionadas con una visión original propia. *Prometheus-9* aspira a ofrecer una experiencia unipersonal enfocada en la historia, donde el jugador descubrirá progresivamente un relato profundo a medida que interactúa con el entorno y las pistas que este oculta. En un mercado donde conviven superproducciones con gráficos fotorrealistas y humildes juegos indie experimentales, *Prometheus-9* se posiciona deliberadamente en el espacio de estos últimos: un juego desarrollado con recursos limitados, pero con grandes ambiciones narrativas. Al insertarse en la tendencia de juegos narrativos contemporáneos, el proyecto busca demostrar cómo incluso un equipo pequeño puede crear una experiencia significativa, apoyándose en una buena historia y un diseño ingenioso, alineado con las expectativas actuales de los jugadores que valoran la narrativa en el videojuego.

1.3. ALCANCE DEL PROYECTO

El objetivo de este apartado será definir los límites y el alcance de este proyecto.

Prometheus-9 abarcará el diseño completo de la experiencia de juego y la implementación de un prototipo funcional en el motor Unity. Esto cubre todas las fases del desarrollo de un videojuego desde su concepción hasta su implementación y pruebas. Específicamente, se desarrolló:

- **Narrativa:** Una historia original con narrativa fragmentada en 10 grabaciones de audio/vídeo, tres finales distintos y un trasfondo de ciencia ficción cuidadosamente elaborado. Cada grabación revela parte de la trama y juntas conforman un rompecabezas narrativo que el jugador debe recomponer.
- **Misiones:** Un conjunto de 10 misiones interconectadas, cada una con mecánicas únicas (puzles lógicos, minijuegos, desafíos de exploración, secuencias de sigilo, etc.). Las misiones se articulan con la historia, de modo que cada desafío completado desbloquea nueva información o áreas de la nave.
- **Jugabilidad/Interfaz:** Un sistema de interfaz de usuario adaptativa, que incluye gestión de vida y energía del personaje, reproducción de grabaciones encontradas, un inventario básico para objetos clave, e indicadores visuales de objetivos y estado. La interfaz está integrada diegéticamente (pantallas dentro del juego, HUD coherente con la ambientación) para no romper la inmersión.
- **Arquitectura técnica:** Una arquitectura de software modular en C# para Unity, con scripts especializados para misiones, un controlador general de estado, controladores de cámara, scripts de interacción, entre otros. Esta arquitectura soporta la escalabilidad y el fácil mantenimiento del código.
- **Ambientación:** Una ambientación lograda mediante el uso de modelos 3D, efectos de sonido envolventes, música de fondo atmosférica y un sistema de cámaras virtuales para crear momentos cinemáticos. Se diseñó la iluminación de la nave con tonos tenues y parpadeos eléctricos y se incorporaron detalles visuales que cuentan parte de la historia de forma ambiental.

1.4. MOTIVACIÓN Y SOLUCIÓN PROPUESTA

El origen de *Prometheus-9* surge tanto de motivaciones personales como profesionales. En el plano personal, la autora de este proyecto siente un profundo interés por la narrativa interactiva y la ciencia ficción. Los mundos futuristas, las historias de exploración espacial y las distopías tecnológicas despiertan su imaginación, creando un entorno perfecto para desarrollar un juego como *Prometheus-9*. En el ámbito profesional, la motivación radica en afrontar el desafío de integrar narrativa y tecnología de forma efectiva. Desarrollar un videojuego narrativo en Unity permite poner en práctica todas las habilidades adquiridas durante el máster, y al mismo tiempo profundizar en la arquitectura de experiencias interactivas centradas en la historia. Existe además el incentivo de contribuir, aunque modestamente, a la creciente escena de videojuegos

narrativos en español, demostrando las posibilidades del medio para contar historias de ciencia ficción maduras y envolventes.

Se ha optado deliberadamente por la ciencia ficción como género para la historia de este juego dado que ofrece un marco creativo amplio donde es posible explorar ideas sobre tecnología, sociedad y futuro que invitan a la reflexión, a la vez que proporcionan escenarios atractivos y misteriosos para el jugador. Además, este género cuenta con numerosos referentes exitosos en videojuegos narrativos, desde aventuras gráficas espaciales hasta *thrillers* futuristas. Inspiraciones indirectas podrían citarse en títulos como *System Shock*, *SOMA* o *Observation*, que combinan entornos de ciencia ficción con la exploración e investigación guiada por la historia.

La elección del nombre *Prometheus-9* evoca simbólicamente el mito de Prometeo y sugiere temáticas de descubrimiento prohibido o el coste del conocimiento, ideas recurrentes tanto en la ciencia ficción como en el mito original. Estas referencias aportan un gran significado al trasfondo del juego y sirven de guía para el desarrollo narrativo.

Una de las premisas fundamentales de la solución propuesta es la integración armónica entre las mecánicas de juego y la narrativa. En muchos proyectos de videojuegos narrativos se destaca que la historia alcanza su máximo potencial cuando está entrelazada con las acciones del jugador y las reglas del juego. Siguiendo este principio, *Prometheus-9* se diseña de manera que cada mecánica introducida tenga un sentido dentro del relato, y viceversa, que cada elemento narrativo clave requiera del jugador una interacción significativa. El acto de buscar pistas en archivos de la nave, terminales de computadora o notas de personajes no será simplemente un *coleccionable*, sino que constituirá el núcleo de la jugabilidad y a la vez el método de avance en la historia. Del mismo modo, el jugador deberá recopilar y analizar fragmentos de información dispersos para armar el rompecabezas argumental. Cada misión dentro del juego proporcionará nuevas piezas de ese rompecabezas que iluminarán gradualmente la verdad de lo ocurrido en la historia. Esta narrativa fragmentada exige al jugador una participación activa intelectualmente: no se trata solo de superar desafíos de destreza, sino de atar cabos, formular hipótesis y comprender las motivaciones de los personajes a través de lo que las pistas revelan.

La solución propuesta para estructurar la progresión en este juego gira en torno a las grabaciones y culmina en decisiones finales del jugador. En vez de presentar la historia en orden cronológico lineal, el juego sitúa al jugador en media res y le permite descubrir el pasado a través de grabaciones dispersas. Cada grabación desbloqueará información crucial y a la vez abrirá nuevas preguntas. De esta manera, la narrativa se presenta como un rompecabezas no lineal, donde el jugador podría descubrir piezas de la historia en distinto orden según su ruta de exploración, pero necesitará reunir la suficiente información para comprender el panorama completo.

A medida que el jugador progresa y recopila la mayor parte de las grabaciones, el juego se encamina hacia su desenlace. En este punto, la interactividad narrativa alcanzará su máxima expresión al requerir que el jugador tome una decisión final que determinará el cierre de la historia. Estas decisiones estarán fundamentadas en la información que el jugador haya logrado reunir y entender. Idealmente, no habrá una única opción correcta, sino finales alternativos que reflejen distintos juicios morales realizados por el jugador.

De esta forma, *Prometheus-9* se convierte en una historia jugada, donde cada acción tiene un sentido dentro del relato y cada descubrimiento modifica la percepción del jugador sobre lo ocurrido. La solución propuesta equilibra narrativa y jugabilidad de tal modo que el jugador es a la vez protagonista y narrador: interactúa para avanzar y, al hacerlo, revela la trama que da contexto a sus acciones.

1.5. ESTRUCTURA DE LA MEMORIA

La presente documentación está organizada en capítulos que recorren todos los aspectos de este proyecto, desde su concepción hasta su implementación y conclusiones. A continuación, se describe brevemente el contenido de cada uno de esos capítulos:

- **Capítulo 1: Introducción.** El capítulo actual, donde se enmarca el contexto cultural y tecnológico del proyecto, se expone el auge de la industria de los videojuegos, se detalla el contexto en el que nace *Prometheus-9*, el alcance del proyecto, las motivaciones que lo impulsan y un resumen de la solución propuesta. Esta introducción proporciona las bases y justificaciones necesarias antes de profundizar en los detalles técnicos y creativos en posteriores capítulos.
- **Capítulo 2: Estado del Arte y Contexto Teórico.** En este capítulo se recopila información relevante sobre el estado actual de los videojuegos narrativos. Se analizan obras similares a *Prometheus-9* y referentes clave, extrayendo lecciones de sus enfoques de diseño. También se discuten conceptos teóricos importantes apoyándose en fuentes académicas y profesionales. Este marco teórico sirve para fundamentar las decisiones de diseño tomadas en el proyecto, asegurando que se basan en buenas prácticas y conocimientos actuales de la industria.
- **Capítulo 3: Diseño del Juego.** En este apartado se detallará el proceso de diseño de *Prometheus-9*. Incluye la descripción del concepto de juego, el diseño narrativo y el diseño de mecánicas. Se presentarán diagramas de flujo narrativo, bocetos de misiones y prototipos en papel de secuencias importantes para ilustrar la visión antes de la implementación. Además, se abordarán las decisiones tomadas respecto a la estética visual y sonora del juego. Al final del capítulo, el lector tendrá una imagen clara de *qué* se va a construir y *cómo* se espera que funcione la experiencia de juego.
- **Capítulo 4: Implementación.** Este capítulo cubre la realización práctica del proyecto en el motor Unity. Se explica la arquitectura de software del juego, describiendo los módulos principales desarrollados. Se incluyen detalles técnicos relevantes como el uso de ciertos patrones de diseño, optimizaciones realizadas o integración de paquetes/extensiones de Unity utilizados para agilizar el desarrollo. Asimismo, se documenta el pipeline de desarrollo seguido y se describen los retos significativos que se han encontrado. El objetivo es que este capítulo refleje el trabajo de ingeniería detrás de *Prometheus-9*, mostrando cómo las ideas de diseño del capítulo previo tomaron forma en un producto software funcional.
- **Capítulo 5: Validación y Resultados.** En este penúltimo capítulo se presentan los resultados obtenidos y la evaluación del proyecto. Se describe el estado final del prototipo del videojuego, especificando qué características están completamente

implementadas y cuáles de forma parcial o simulada. A continuación, se reportan los resultados de las pruebas realizadas, que permiten discutir hasta qué punto el proyecto alcanzó sus objetivos iniciales. También se mencionan los méritos del producto logrado y se identifican las limitaciones o áreas de mejora detectadas. En conjunto, este capítulo ofrece una visión crítica de los logros del proyecto y sirve de puente hacia las conclusiones finales.

- **Capítulo 6: Conclusiones y Trabajo Futuro.** El capítulo final de la memoria recapitula los puntos más importantes del proyecto. Se reflexiona sobre el cumplimiento de los objetivos planteados, resaltando las conclusiones clave en cuanto al desarrollo de un videojuego narrativo de ciencia ficción en un contexto académico. Se discute el aprendizaje personal obtenido por la autora en términos de habilidades técnicas, de diseño y de gestión de proyecto. Además, se delinean posibles líneas de trabajo futuro que podrían extender o mejorar *Prometheus-9*. En definitiva, las conclusiones cierran el documento consolidando la contribución del proyecto y dejando abiertas puertas para desarrollos posteriores.

En conjunto, esta memoria está estructurada para comprender desde el porqué de *Prometheus-9* (motivaciones, contexto, diseño), pasando por el cómo se realizó (implementación, detalles técnicos), hasta qué se obtuvo (resultados, conclusiones).

CAPÍTULO 2. OBJETIVOS DEL PROYECTO Y HERRAMIENTAS UTILIZADAS

A lo largo de este capítulo se detallarán los objetivos principales que guían el desarrollo de este proyecto, tanto a nivel técnico como narrativo. Además, se presentan todas las herramientas utilizadas, desde motores y lenguajes de programación hasta software auxiliar de diseño y control de versiones.

2.1. OBJETIVOS TÉCNICOS, NARRATIVOS Y METODOLÓGICOS DEL DESARROLLO

Este proyecto se planteó en base a una serie de objetivos generales y específicos que abarcan la narrativa, los aspectos técnicos y la jugabilidad. A continuación, se definen tanto el objetivo principal del juego como los objetivos específicos en distintas áreas.

El objetivo principal es desarrollar un videojuego narrativo de ciencia ficción en primera persona, completo y funcional, que ofrezca una experiencia altamente inmersiva al jugador mediante la integración equilibrada de narrativa interactiva, exploración y puzzles, garantizando múltiples desenlaces según las decisiones tomadas y manteniendo un cierto grado de calidad técnica y usabilidad a lo largo de la experiencia.

Por otro lado, los objetivos específicos son los que se exponen a continuación:

1. Creación de una **narrativa inmersiva con múltiples finales**. Lograr que la historia atrape al jugador y lo haga sentir parte del juego, proporcionando una experiencia envolvente. La inmersión se fomenta construyendo un entorno creíble y empleando técnicas narrativas interactivas donde el jugador influye activamente en la trama. Se busca que las decisiones del jugador tengan consecuencias en la historia, incluyendo múltiples finales basados en dichas decisiones. Este enfoque de narrativa ramificada aumenta la rejugabilidad al invitar a experimentar distintas rutas argumentales. En *Prometheus-9*, esto implica diseñar tres finales diferentes condicionados por una decisión final crítica, cumpliendo con el objetivo de profundidad narrativa y sensación de agencia para el jugador.
2. Implementación de un **sistema de misiones** integrado en la historia. Estructurar la progresión del juego en misiones, de manera que cada misión represente tanto un desafío jugable como un avance en la trama. Se planificó incluir en torno a 10 misiones variadas, coherentes con la narrativa y que combinen exploración y toma de decisiones. Cada misión debe tener una justificación narrativa clara; es decir, las mecánicas y objetivos de la misión deben alinearse con hechos de la historia, logrando una integración efectiva entre gameplay y narrativa. Esto garantiza que la jugabilidad y la historia se retroalimenten, manteniendo la coherencia del mundo y un ritmo narrativo adecuado. Las revelaciones argumentales se distribuyen de forma equilibrada a lo largo del juego para mantener el interés del jugador.

3. Desarrollo del juego sobre una **base técnica sólida**. En el plano técnico, un objetivo central fue realizar el desarrollo en el motor Unity, aprovechando sus capacidades 3D y herramientas integradas. Asimismo, un objetivo más específico fue implementar mecánicas de juego coherentes con la historia: por ejemplo, un sistema de inventario para objetos relevantes, interfaces de terminales dentro del juego, puzzles lógicos y eventos programados que desencadenen escenas narrativas. Igualmente, se buscó utilizar las herramientas de Unity para orquestar momentos clave de la historia y gestionar estados del juego de forma modular.
4. **Aplicar buenas prácticas de ingeniería de software y diseño ágil**. Un objetivo claro era asegurar la calidad del código y la facilidad de mantenimiento, para lo que se utilizaron principios sólidos de ingeniería del software y buenas prácticas durante el desarrollo, como son las siguientes:
 - Separación clara de responsabilidades en los scripts.
 - Uso de patrones de diseño adecuados, como *Singleton* para manejo de estado, y controladores específicos por misión para aislar la lógica de cada nivel.
 - Empleo de prefabs reutilizables para objetos interactivos comunes y sistemas reutilizables.
 - Iteraciones con pruebas frecuentes, integración progresiva de funcionalidades y uso riguroso de control de versiones con Git, de forma de detectar y corregir errores rápidamente.

Además, se decidió utilizar una metodología de desarrollo ágil como Scrum con iteraciones cortas, revisiones semanales y evolución continua del backlog, con el objetivo de garantizar un progreso incremental, reducir riesgos, y permitir ajustes sobre la marcha sin perder el control del proyecto.

5. **Garantizar la usabilidad y una jugabilidad equilibrada**. Se establecieron objetivos centrados en la experiencia de usuario, de manera que el juego debe presentar una interfaz intuitiva y diegética, que ofrezca la información necesaria sin romper la inmersión. En *Prometheus-9* se optó por interfaces integradas en el mundo del juego para mantener la coherencia temática. Asimismo, la progresión por exploración es un objetivo de diseño: el jugador avanza investigando entornos, descubriendo grabaciones y resolviendo misiones, lo que exige que el juego esté diseñado de forma abierta pero guiada sutilmente mediante pistas visuales o sonoras. Se pretende estimular la curiosidad del jugador para que la exploración sea recompensada con avances narrativos. Asimismo, se busca una curva de dificultad equilibrada a lo largo de las misiones, incrementando gradualmente la complejidad de los desafíos y puzzles sin que llegue a ser frustrante. Mantener el equilibrio entre desafío y accesibilidad es crucial para que los jugadores no se aburran ni se frustren, aceptando los nuevos retos porque resultan estimulantes.

En resumen, estos objetivos orientaron todo el desarrollo de este proyecto. Se trata de alcanzar un producto final donde la narrativa inmersiva con múltiples finales, la implementación técnica en Unity y la jugabilidad intuitiva converjan de forma armoniosa. El cumplimiento de estos objetivos asegura que el videojuego resultante brinde una experiencia memorable, coherente y pulida.

2.2. HERRAMIENTAS UTILIZADAS

En esta sección se presentan las principales herramientas utilizadas para llevar a cabo el desarrollo de este proyecto y asegurar el cumplimiento de los objetivos que han sido descritos anteriormente. Asimismo, se proporciona una justificación de su elección en comparación con distintas alternativas.

2.2.1. Motor de desarrollo: Unity

El motor de juego seleccionado fue Unity 3D (versión 2021.3.22f) debido a que ofrece un entorno de desarrollo muy accesible y una interfaz intuitiva, lo que resulta ideal para proyectos independientes. Además, en comparación con otros motores de videojuegos, la curva de aprendizaje de Unity es menos pronunciada, presentando un equilibrio perfecto entre potencia y simplicidad.

Unity emplea C# como lenguaje de programación, siendo este un lenguaje orientado a objetos de alto nivel, con una sintaxis clara y gestión automática de memoria, lo que agiliza el desarrollo frente a otros lenguajes. También es reseñable destacar que Unity cuenta con una de las comunidades más amplias en el ámbito del desarrollo de juegos. Esta gran comunidad activa y la extensa documentación fueron recursos invaluable para resolver dudas, encontrar *plugins* y seguir buenas prácticas, lo que permitió un desarrollo eficiente y de calidad.

De la misma manera, Unity permite desarrollar juegos para múltiples plataformas (PC, web, móviles, consolas, VR/AR, etc.), lo que no limita a este proyecto a un único entorno. Si bien el juego ha sido enfocado únicamente en PC, existe la posibilidad de ampliar a otras plataformas en un futuro. Es más, Unity proporciona un renderizado en tiempo real eficiente, iluminación dinámica, sombreadores personalizables y un sistema de físicas integrado, que fueron aprovechados para crear la ambientación de la nave espacial con efectos visuales y sonoros envolventes.

Por último, cabe destacar que Unity utiliza un enfoque orientado a componentes, lo que facilitó considerablemente la organización del proyecto y ayudó a mantener un código limpio y estructurado, dado que cada componente maneja una única responsabilidad. Adicionalmente, Unity cuenta con herramientas internas como Cinemachine, Timeline o el sistema de UI basado en Canvas, que redujeron el esfuerzo de implementación técnica, permitiendo a la autora de este proyecto centrarse más en el diseño de la experiencia.

En resumen, Unity proporcionó un equilibrio óptimo entre potencia y usabilidad. Su elección está alineada con los objetivos narrativos y con los objetivos técnicos.

2.2.2. Lenguaje de programación: C#

El lenguaje de programación empleado para todos los scripts lógicos fue C#, que es el lenguaje nativo de scripting en Unity. Su elección vino determinada por la selección del motor, pero conviene destacar sus ventajas y la forma en que contribuyó al cumplimiento de los objetivos técnicos de este proyecto:

- **Integración total con Unity.** Unity está diseñado para trabajar con C# de forma óptima, lo que permitió acceder fácilmente a las funcionalidades del motor y conectar comportamientos de juego con los elementos del editor.
- **Facilidad y productividad del lenguaje.** C# es un lenguaje maduro conocido por ser sencillo de manejar debido a su gestión automática de memoria. Además, cuenta con características de programación orientada a objetos que facilitan la implementación de sistemas robustos como clases Singleton para mantener un estado persistente entre escenas, Interfaces, y eventos y delegados para gestionar comunicaciones flexibles entre sistemas.
- **Herramientas de desarrollo asociadas.** El uso de C# permitió aprovechar potentes entornos de desarrollo como Visual Studio con funcionalidades como IntelliSense, depuración enriquecida, análisis estático de código, etc., que aceleraron en gran medida la programación. La familiaridad con C# también permitió incorporar librerías externas para ampliar funcionalidades.

En definitiva, C# potenció el cumplimiento de los objetivos técnicos al ser un lenguaje productivo, seguro y compatible con la escala del proyecto.

2.2.3. Entorno de desarrollo: Visual Studio

Para la programación en C# se utilizó el entorno de desarrollo integrado Microsoft Visual Studio 2019 por diversas razones técnicas, especialmente por su excelente integración con Unity y sus potentes herramientas de edición y depuración.

- **Integración.** Unity descarga y utiliza por defecto como editor Visual Studio para asegurar la compatibilidad, por lo que la configuración fue trivial y sin problemas.
- **Productividad.** Visual Studio ofrece características que resultan de gran utilidad para el desarrollo en C#. En particular, la función IntelliSense proporciona un autocompletado inteligente del código, mostrando métodos y propiedades disponibles a medida que se escribe, y resaltado de errores en tiempo real incluso antes de compilar. Esto permite acelerar la programación y reduce el número de fallos, ayudando a mantener un código de calidad.
- **Depuración avanzada.** Una de las mayores ventajas a la hora de utilizar Visual Studio fue su capacidad de depuración integrada con Unity, que permite ejecutar el juego y adjuntar el depurador de Visual Studio para poner *breakpoints*, inspeccionar variables en tiempo real y recorrer el código paso a paso durante la ejecución. Esto, permitió la detección y corrección rápida de errores lógicos.
- **Herramientas de refactorización.** A medida que el proyecto fue creciendo, se hizo necesario refactorizar ciertas partes del código, lo cual fue muy sencillo gracias a las herramientas automáticas de refactorización con las que cuenta Visual Studio, que ayudaron a mantener un código limpio.

En definitiva, el uso de Visual Studio como IDE contribuyó a alcanzar los objetivos técnicos al proporcionar un entorno de programación profesional integrado con las necesidades específicas de *Prometheus-9*.

2.2.4. Control de versiones: Git y GitHub

Durante el desarrollo de este proyecto, se utilizó Git como sistema de control de versiones, con un repositorio remoto alojado en GitHub. Adoptar control de versiones desde el inicio fue una decisión técnica fundamental para asegurar la integridad del proyecto y facilitar un flujo de trabajo ágil. Las razones y beneficios de esta elección son:

- **Historial de cambios y reversión de errores.** Git permite llevar un registro de cada modificación en el código fuente mediante *commits*, lo que significa que, en cualquier momento, se pueden consultar todos los cambios realizados. Es más, Git brinda la posibilidad de volver a versiones anteriores si algo sale mal.
- **Copia de seguridad remota.** Al alojar el repositorio en GitHub, se obtuvo automáticamente una copia de seguridad en la nube de todo el proyecto, lo cual protege contra pérdida de datos en caso de fallo del equipo local y facilitó trabajar desde diferentes entornos.
- **Comparativa con otras soluciones.** Se consideraron brevemente alternativas como SVN (Subversion) o Perforce. No obstante, se optó por utilizar Git debido a su amplia documentación, soporte integrado en múltiples herramientas y su enfoque distribuido. SVN ofrece una estructura más centralizada, pero es menos distribuido y habría dificultado el trabajo offline. Por otro lado, Perforce es utilizado en grandes estudios de videojuegos por el manejo eficiente de archivos binarios de gran tamaño. Sin embargo, dado que el volumen de archivos de este proyecto es relativamente reducido, Git se presentó como la opción más coherente con el tamaño y las necesidades del proyecto.

En conclusión, el uso de Git y GitHub permitió un desarrollo de forma iterativa y segura, soportando la metodología ágil adoptada.

En conjunto, las herramientas seleccionadas conformaron un ecosistema de desarrollo robusto y adecuado a *Prometheus-9*. La elección de Unity y C# proporcionó la base tecnológica para crear rápidamente el juego narrativo 3D, cumpliendo los objetivos técnicos de implementación de misiones, decisiones y ambientación inmersiva. Y, por su parte, Visual Studio y Git aseguraron la calidad y control del código fuente durante todo el proceso, apoyando las buenas prácticas de ingeniería y permitiendo iterar sobre la narrativa y las mecánicas sin comprometer la estabilidad.

CAPÍTULO 3. FUNDAMENTOS TEÓRICOS Y ESTADO DEL ARTE

Este capítulo proporciona un análisis teórico de los videojuegos narrativos y la gamificación, explorando sus características, ventajas, clasificación y casos relevantes. A su vez, se contextualiza el proyecto dentro de las tendencias actuales en diseño de experiencias interactivas.

3.1. ¿QUÉ SON LOS JUEGOS NARRATIVOS?

3.1.1. Definición y evolución histórica

El término juego narrativo se refiere a una categoría de videojuegos en la que el desarrollo de la historia es el elemento central de la experiencia del jugador. A diferencia de otros juegos tradicionales, cuyo foco suele estar en las mecánicas de acción, en los reflejos o en la obtención de puntuaciones, en los juegos narrativos el jugador es partícipe en el desarrollo de una trama estructurada, mediante elecciones, exploración o interacción con el entorno. En esencia, el juego narrativo pasar a ser el núcleo que guía la jugabilidad, convirtiendo al jugador en protagonista de una historia y ofreciendo una experiencia participativa que ninguna otra forma narrativa lineal puede lograr por sí sola.

Históricamente, los primeros intentos de incorporar narrativas complejas en videojuegos se remontan a los años 70 y 80 con títulos pioneros como *Colossal Cave Adventure* (1976) o *Zork* (1980), que eran aventuras donde el jugador leía descripciones y escribía comandos para realizar acciones, viviendo una historia interactiva a través de palabras. Más tarde, empresas como Sierra On-Line y LucasArts popularizaron el género con aventuras gráficas como *Monkey Island* o *King's Quest*, donde el jugador debía resolver puzzles e interactuar con personajes para avanzar en la historia. Estas obras sentaron las bases de cómo contar historias en un medio interactivo.

Con la evolución del hardware y de los motores gráficos en los 90 y 2000, la narrativa en videojuegos fue ganando en complejidad y se fusionó con otros géneros. Surgieron juegos como *Metal Gear Solid*, *Silent Hill*, *Bioshock*, *The Last of Us*, entre muchos otros, que ofrecían tramas dignas de cine o literatura, con alto contenido emocional y cinemáticas elaboradas, pero manteniendo al jugador en control durante buena parte del tiempo. Estos títulos demostraron que era posible equilibrar narrativa y jugabilidad activa, diluyendo la frontera entre juego de acción y juego narrativo.

3.1.2. Ventajas de los juegos narrativos

Al poner la historia en primer plano, los juegos narrativos ofrecen ventajas únicas, siendo la principal de ellas la inmersión, pudiendo generar un nivel de involucración emocional profundo por parte del jugador, al integrarlo dentro de un mundo ficticio. La combinación de narrativa y participación activa también fomenta la empatía del jugador hacia los

personajes de la historia, al permitirle vivir en primera persona determinadas situaciones. Asimismo, este tipo de juegos ofrecen una toma de decisiones significativas que refuerza la sensación del jugador de tener el control sobre el curso de los acontecimientos. Tomar decisiones importantes aumenta el compromiso, pues el jugador siente que su participación importa en el rumbo de la historia; incluso en casos en los que la historia global ya está escrita, las decisiones aportan matices personalizados a la experiencia.

3.1.3. Narrativa emergente vs. narrativa estructurada

Uno de los grandes debates en el diseño narrativo de videojuegos es la contraposición entre narrativa estructurada y narrativa emergente.

- **Narrativa estructurada.** La historia está predefinida, por lo que el jugador avanza a través de una trama planificada por los creadores. Es similar a una novela o a una película, donde los eventos principales ya están escritos y, en este caso, el jugador puede influir en el ritmo o en pequeños detalles.
- **Narrativa emergente.** En este caso, la historia no está definida de antemano, sino que surge de la interacción del jugador con el mundo del juego. Aquí, los desarrolladores establecen escenarios, pero las historias van siendo creadas con las acciones del jugador. Ejemplos clásicos son *Minecraft*, *RimWorld* o *Dwarf Fortress*, donde a partir de mecánicas de simulación resultan anécdotas o aventuras únicas para cada jugador.

Prometheus-9 se alinea con la narrativa estructurada, ya que hay una trama concreta que el jugador debe reconstruir. A pesar de que la historia global está escrita, se incluyen elementos de exploración y fragmentación no lineal: el jugador puede descubrir las grabaciones en distintos órdenes, generando su propia experiencia de descubrimiento. Esto añade una pizca de narrativa emergente en tanto que cada jugador podría recomponer mentalmente la historia en un orden distinto según cómo explore la nave.

3.1.4. Diseño narrativo en videojuegos

Tal como plantea Henry Jenkins en su ensayo “*Game Design as Narrative Architecture*” (2004), los juegos no narran historias de la misma forma que el cine o la literatura, sino que crean espacios narrativos donde el jugador puede descubrir fragmentos de la historia y recomponerla por sí mismo. Jenkins introduce la idea de que el entorno del juego puede ser el narrador, a través de lo que se conoce como *environmental storytelling*. Siguiendo esa línea, existen varias estrategias de diseño narrativo en juegos.

- **Narrativa ambiental.** El mundo y sus objetos cuentan la historia (mensajes escritos en paredes, diarios o notas encontrados, grabaciones de audio, etc.). El jugador deduce la trama observando el entorno, sin necesidad de largas exposiciones verbales.
- **Narrativa modular.** La historia se revela en módulos que el jugador desbloquea a medida que avanza, pero que no necesariamente recibe en orden cronológico, lo que exige al jugador la reconstrucción de la secuencia temporal. Es típico de

juegos donde coleccionas fragmentos de memoria, audios, o donde visitas capítulos en desorden temporal.

Prometheus-9 se apoya fuertemente en ambas estrategias. Por un lado, emplea narrativa ambiental: la nave abandonada está llena de pistas visuales que cuentan parte de lo ocurrido antes de que el jugador despierte. Por otro lado, utiliza narrativa modular a través de las 10 grabaciones de seguridad: cada grabación es un bloque de la historia que el jugador obtiene tras una misión, y aunque están numeradas de 1 a 10, el jugador las experimenta intercaladas con la acción, asimilándolas progresivamente y muchas veces teniendo que deducir conexiones entre ellas. Al conjugar estos métodos, *Prometheus-9* logra que el jugador se sienta como un investigador dentro del juego: explorando la nave y recogiendo evidencia fragmentada para finalmente entender el relato completo.

3.1.5. Elementos clave de un juego narrativo

Los juegos narrativos comparten elementos fundamentales que les permiten construir una experiencia centrada en la historia. Entre estos elementos encontramos:

1. **Historia central.** Una línea argumental clara con planteamiento, conflicto y desenlace. En el caso de *Prometheus-9*, la historia central podría resumirse así: el protagonista despierta solo en una nave espacial aparentemente averiada (planteamiento), descubre que la nave fue sabotada por un miembro de la tripulación (conflicto) y finalmente debe decidir cómo resolver la situación (desenlace, con varias posibilidades).
2. **Personajes con motivaciones creíbles.** Protagonistas y secundarios que tienen deseos, miedos y objetivos propios que impulsan la trama. En *Prometheus-9*, la historia emerge a través de una serie de personajes presentes en las grabaciones, cuyas motivaciones creíbles dan peso a la narrativa.
3. **Decisiones significativas.** El jugador suele poder tomar decisiones que afectan a la historia o la percepción de la misma. No todos los juegos narrativos tienen narrativas ramificadas amplias, pero sí ofrecen decisiones que al menos cambian ciertos matices. En *Prometheus-9*, el recorrido de misiones no es lineal por lo que el jugador podrá desbloquear información en cualquier orden. Además, su decisión final determinará cuál de los tres finales va a tener lugar.
4. **Ambiente inmersivo.** Un mundo de juego consistente y detallado que refuerza la historia. Música, sonidos, estilo visual, todo contribuye a sumergir al jugador en la atmósfera narrativa. *Prometheus-9* cuida mucho este aspecto: la ambientación tenebrosa de la nave, los crujidos metálicos o el sonido ambiental de los sistemas de soporte vital sumergen al jugador en una sensación de soledad en el espacio y de suspense al no saber qué sucedió allí.
5. **Interacción contextualizada.** Las acciones que realiza el jugador tienen sentido dentro de la historia. Esto es crucial para que narrativa y jugabilidad no se sientan desconectadas. Cada misión en *Prometheus-9* está contextualizada (arreglar un reactor, desactivar una alarma, etc.), de modo que el jugador siente que lo que hace forma parte de la historia, no es un simple obstáculo artificial.

En conjunto, *Prometheus-9* es un ejemplo de narrativa interactiva de ciencia ficción espacial, donde el jugador participa activamente en el desarrollo de la historia, ayuda a su reconstrucción mediante el descubrimiento progresivo de las grabaciones, decide el desenlace final e interpreta los hechos a través no solo de los diálogos, sino del entorno, documentos y sonidos, estimulando su imaginación.

3.2. GAMIFICACIÓN

3.2.1. Origen y concepto de gamificación

El termino gamificación fue popularizado a inicios de los 2000 y se refiere al uso de mecánicas de juego en contextos no lúdicos con el objetivo de motivar, enseñar o involucrar al usuario. En otras palabras, consiste en tomar elementos típicos de los juegos y aplicarlos a ámbitos como la educación, la salud, el marketing, la productividad empresarial, entre otros, para hacer estas actividades más atractivas y estimulantes.

La teoría de la gamificación se sustenta en principios de psicología del comportamiento y motivación, apoyándose en conceptos de condicionamiento operante y, especialmente, en teorías de motivación intrínseca y extrínseca. Ha sido usada con éxito en aplicaciones tan variadas como plataformas educativas (por ejemplo, Duolingo gamifica el aprendizaje de idiomas con puntos y rachas diarias), en la salud y el ejercicio (Fitbit convierte hacer ejercicio en un juego de metas y recompensas), o en gestión del tiempo y hábitos (Habitica, que gamifica tareas cotidianas). En todos estos casos, la gamificación busca aumentar el compromiso del usuario y fomentar ciertos comportamientos deseados.

En los videojuegos en sí, aunque pudiera parecer un término redundante, la gamificación se refiere a la aplicación consciente y dirigida de ciertos principios que aumentan la motivación del jugador más allá de la acción básica. Es decir, diseñar deliberadamente sistemas de recompensas, feedback y objetivos internos al juego para potenciar el engagement, la satisfacción y la permanencia del jugador.

3.2.2. Principales mecánicas de gamificación

La gamificación emplea una variedad de mecánicas y elementos de juego para enganchar a los usuarios. Según Sebastián Deterding y otros autores, algunos elementos clave empleados en gamificación son los siguientes:

- **Recompensas.** La mecánica más básica es otorgar puntos u otras recompensas cuantitativas por realizar ciertas acciones, cuya acumulación proporciona una medida de progreso y un refuerzo inmediato por completar tareas. A menudo los puntos se canjean o están asociados a monedas virtuales para comprar ventajas. Otras formas de recompensa incluyen insignias o medallas por hitos alcanzados y objetos coleccionables. En *Prometheus-9*, los puntos se traducen en elementos diegéticos: el jugador *recupera energía* al completar misiones (recompensa de supervivencia) y desbloquea cintas de grabaciones como premio por avanzar.

- **Retroalimentación inmediata.** En los juegos, cada acción suele generar una respuesta inmediata del sistema, que alivia la incertidumbre del jugador y refuerza su comportamiento. En gamificación se implementa mostrando barras que se llenan, *pop-ups* de felicitación, efectos de sonido, etc. *Prometheus-9* aplica este principio al dar indicadores audiovisuales claros cuando el jugador acierta o falla una acción, por ejemplo, cometer un error como cortar un cable incorrecto hace que se dispare una alarma y salgan chispas.
- **Desafíos, misiones y objetivos claros.** Otra mecánica fundamental es plantear retos al usuario, con metas definidas que alcanzar, lo cual da dirección a la participación y activa la motivación por logro. En gamificación, las actividades suelen estructurarse como “misiones” a completar, donde los jugadores saben qué hacer y qué ganarán al lograrlo. En *Prometheus-9*, la propia estructura del juego está dividida en 10 misiones, cada una con objetivos explícitos. Esto refleja una gamificación interna ya que cada misión completada desbloquea más contenido de la historia, dando al jugador una sensación de progreso.
- **Barra de progreso y niveles.** Las barras de progreso, indicadores de porcentaje o conteos regresivos hacen tangible cuánto falta para lograr un objetivo, incentivando a los usuarios a completarlos. Muchos entornos gamificados muestran una barra de experiencia que aumenta conforme el usuario realiza ciertas actividades, hasta permitir alcanzar el siguiente nivel. *Prometheus-9* incorpora un HUD con iconos de vida y energía siempre visibles que informan constantemente al jugador de su estado en términos de supervivencia.
- **Penalizaciones.** En los juegos, las penalizaciones se manifiestan en la posibilidad de perder vidas, fallar una misión, retroceder si algo sale mal, etc., lo cual hace más emocionante una actividad. En gamificación se introduce a veces mediante conteos regresivos, vidas limitadas o consecuencias negativas que añaden presión. El objetivo no es castigar sino agregar un desafío significativo, ya que, si todo son premios sin posibilidad de fallo, la motivación puede decaer. *Prometheus-9* implementa penalizaciones con la pérdida de salud, oxígeno y energía si el jugador comete ciertos errores o pierde mucho tiempo, lo que aumenta la tensión y la urgencia dentro del juego.
- **Desbloqueo de contenidos.** Muchas estrategias de gamificación usan el desbloqueo incremental, que aprovecha la curiosidad y el sentido de logro para mantener el interés a largo plazo. En juegos se ve cuando nuevas habilidades o zonas del mapa se habilitan al superar un nivel. *Prometheus-9* adopta esta mecánica claramente, ya que, al inicio del juego, varias puertas y secciones de la nave están completamente cerradas, y solo pueden abrirse al completar ciertas misiones. También aparecen nuevas herramientas cuando el jugador avanza, lo que incita al jugador a progresar para ver qué hay detrás de la siguiente puerta.
- **Competición, rankings y componentes sociales.** Aunque *Prometheus-9* por ser una experiencia single-player no incluye un rankings ni componentes sociales, vale la pena mencionar que en gamificación es común introducir tablas de clasificación para aprovechar la motivación por competencia o reconocimiento. Ver nuestro nombre en un ranking, superar a amigos, o cooperar en equipo son factores extrínsecos potentes.

- **Narrativa contextualizada.** Un elemento distintivo en la gamificación es proveer un marco narrativo a la experiencia, de modo que las tareas tengan un sentido dentro de una historia. Pese a no ser indispensable, puede apelar a la motivación intrínseca. En *Prometheus-9*, la narrativa es el eje, estando su diseño gamificado *al servicio* de la historia: todas las mecánicas mencionadas tienen coherencia temática dentro de la historia. Esto ejemplifica una buena práctica, ya que, integrar la gamificación con la narrativa produce una experiencia más orgánica.

En resumen, las mecánicas anteriores son piezas de diseño que un gamificador puede combinar para crear un sistema atractivo. Cada elemento busca aprovechar algún aspecto de la motivación humana para hacer la experiencia más memorable.

Una implementación exitosa de gamificación no necesariamente usa todos estos componentes, sino los adecuados según el contexto y la audiencia objetivo. Además, la clave está en el equilibrio: suficientes recompensas para incentivar, suficiente desafío para ser interesante, y una progresión bien calibrada para ni frustrar ni aburrir al usuario.

3.2.3. Tipos de motivación en gamificación

Como se ha mencionado anteriormente, la gamificación actúa sobre la motivación humana; siendo útil la distinción de los dos grandes tipos de motivación que intervienen:

- **Motivación intrínseca:** Es aquella que surge del interés o disfrute de la propia actividad. Una persona está intrínsecamente motivada cuando realiza algo porque lo encuentra satisfactorio, divertido o desafiante, y no por recompensas externas. *Prometheus-9* fomenta motivación intrínseca mediante su misterio y exploración: el jugador siente curiosidad por descubrir qué ha sucedido en la nave y cómo terminará la historia, lo que le impulsa a seguir avanzando por deseo propio. Otros factores intrínsecos podrían ser la diversión de resolver un puzle, la sensación de logro al superar un reto difícil o la creatividad al improvisar una solución. Además, la narrativa fragmentada y la ambientación intrigante despiertan esa curiosidad y compromiso interno, de manera que el jugador no explora solo por recompensas, sino porque quiere entender la historia.
- **Motivación extrínseca.** Es la que surge de querer hacer algo para obtener una recompensa o evitar un castigo, y no proviene de la propia actividad en sí misma. En gamificación, la mayoría de los elementos son estímulos extrínsecos, ya que buscan empujar al usuario con recompensas tangibles o reconocimiento social. En *Prometheus-9* también existen componentes extrínsecos ligeros, como las grabaciones de seguridad como recompensa tras cada misión, recuperar recursos como energía, vida u oxígeno al progresar, y evitar consecuencias negativas.

En una gamificación efectiva, lo ideal sería que los elementos extrínsecos estén alineados con y sirvan para potenciar la motivación intrínseca, en lugar de reemplazarla. Un buen diseño buscará que, con el tiempo, el usuario encuentre la satisfacción en la propia actividad más allá de las recompensas iniciales. En *Prometheus-9*, podemos observar un bucle de motivación bien calibrado: el jugador explora y resuelve un reto (motivación intrínseca) → obtiene una recompensa narrativa (motivación extrínseca) que a su vez revela más historia → por lo que el jugador sigue explorando para desvelar el siguiente misterio (incremento del interés intrínseco).

3.2.4. Gamificación como diseño consciente en Prometheus-9

En *Prometheus-9*, la gamificación no es un añadido superficial sobre la experiencia, sino que está profundamente entrelazada con la narrativa y la jugabilidad. Esto significa que se incluyen mecánicas típicas de gamificación, pero siempre justificadas dentro de la historia y la ambientación, de modo que el jugador no las perciba como elementos artificiales externos, sino más bien como una parte natural del juego. Por ejemplo: cortar el cable correcto en un panel eléctrico no otorga “puntos” en un cuadro de puntuación, sino que permite reestablecer un sistema vital de la nave dentro de la ficción y, como consecuencia, el jugador desbloquea una nueva grabación de video que aporta información de la trama. Del mismo modo, fallar una tarea no resulta en una simple pantalla de "Game Over", sino que impacta al jugador reduciendo su energía, aumentando la tensión y el realismo. Esto refuerza la inmersión narrativa, dado que las mecánicas gamificadas se traducen en eventos del mundo del juego en lugar de métricas abstractas.

En resumen, *Prometheus-9* utiliza la gamificación de manera orgánica y un diseño de decisiones que apela a la motivación del jugador por ver las distintas consecuencias. Esta integración consciente hace que el jugador casi no perciba “gamificación” como tal, sino que simplemente se siente inmerso en un juego cuya estructura le resulta estimulante.

CAPÍTULO 4. METODOLOGÍA DE TRABAJO

Este capítulo describe en detalle la metodología de trabajo adoptada, la arquitectura del sistema implementada y la evolución temporal del desarrollo de este proyecto.

Se expone el enfoque ágil utilizado, se presenta la arquitectura técnica del proyecto, y se detalla la evolución del proyecto a través de sus fases describiendo decisiones clave de cada etapa.

4.1. METODOLOGÍAS ÁGILES APLICADAS

4.1.1. Contexto y necesidad de un enfoque ágil

El desarrollo de un videojuego como *Prometheus-9*, que combina narrativa interactiva, mecánicas variadas y sistemas interdependientes, exige un enfoque de gestión que permita flexibilidad, adaptación constante y entrega incremental de resultados.

A diferencia de proyectos con requisitos fijos y bien definidos desde el inicio, un juego es un producto muy iterativo, ya que muchas ideas se prueban sobre la marcha, el feedback durante el desarrollo puede hacer cambiar ciertos aspectos, se descubren requisitos nuevos según se avanza, etc. Por tanto, resulta natural adoptar algún modelo ágil de trabajo en lugar de una metodología rígida en cascada.

Ante esta realidad, se optó por utilizar una metodología de desarrollo ágil, concretamente el marco de trabajo Scrum, adaptado al entorno de desarrollo reducido (un desarrollador). Esta adaptación permitió mantener un ritmo de trabajo constante y supuso un impacto directo en la calidad del proyecto, corrigiendo rápidamente los errores y refinando todo aquello que sí funcionaba.

4.1.2. Aplicación específica de Scrum en el proyecto

Desde el inicio se construyó una lista de tareas (*Product Backlog*) que se dividió en categorías para cubrir todos los frentes del proyecto:

- Funcionalidades básicas (movimiento del jugador, sistema de cámara, interacciones básicas, interfaz mínima HUD).
- Módulos del sistema (gestor de estado global *GameManager*, sistema de inventario, sistema de gestión de misiones).
- Misiones desglosadas en sus pasos clave, requisitos previos y necesidades de pruebas para asegurarse de su correcta implementación.
- Narrativa (escritura de las grabaciones, integración del audio de voz, escenas de cámara para reproducirlas).

- Feedback y QA (testeo en cada iteración, corrección de bugs, balance de dificultad de puzles).
- Finales alternativos y sus condiciones de desbloqueo.

Cada ítem se prioriza, abordando primero las tareas más críticas en *sprints* cortos. Se define que cada sprint tendría objetivos concretos, y durante ese sprint, se trabajaría en orden lógico: primero asegurar la interacción básica, luego integrar la interfaz, más tarde añadir efectos o transiciones para pulir la presentación, etc. Al final del sprint se realiza una revisión y se integra la funcionalidad desarrollada con el resto del juego.

Al finalizar cada sprint, se lleva a cabo una revisión funcional completa de lo añadido o modificado. En caso de detectar algún fallo, ese aspecto queda anotado en el backlog como tarea de mejora para el siguiente sprint. De este modo hay una retroalimentación constante. El desarrollo avanza por iteraciones, pero volviendo a pulir y arreglar detalles de lo que ya se ha desarrollado anteriormente.

La adopción de Scrum aportó múltiples beneficios durante el desarrollo de este proyecto.

- La entrega incremental y frecuente de versiones jugables al final de cada sprint permitió una mejora progresiva del juego con menor riesgo.
- Se pudieron ajustar los niveles de dificultad a mitad del proyecto tras realizar distintas pruebas.
- Se reordenaron ciertas grabaciones narrativas tras detectar que el orden original revelaba demasiado pronto un dato clave de la historia.
- Se detectaron y corrigieron errores de forma continua, evitando la acumulación de errores al final. Cada sprint incluía su propio mini-QA, lo que llevó a una versión final más estable.

A pesar de los múltiples beneficios, la aplicación de metodologías ágiles no estuvo exenta de desafíos, siendo el principal la estimación de la duración de las tareas. En desarrollos software, especialmente en videojuegos, es complejo estimar con precisión, de manera que en los primeros sprints se pueden observar multitud de tareas sin completar debido a estimaciones demasiado optimistas. A través de las retrospectivas, se abordó este punto, ajustando las estimaciones y desglosando tareas grandes en subtareas más manejables.

Otro aspecto de gran relevancia fue la documentación; si bien Agile enfatiza "software funcionando sobre documentación extensiva", al tratarse de un proyecto académico fue necesario mantener cierta documentación. Se encontró un equilibrio manteniendo actualizados al final de cada sprint ciertos documentos (como el control de cambios o el de las misiones), lo que ayudó a evitar que la documentación se convirtiera en un obstáculo añadiendo carga de trabajo extra.

En definitiva, la implementación de metodologías ágiles Scrum en *Prometheus-9* hizo posible que creciera de manera controlada sin comprometer la visión original del juego. Incluso siendo un proyecto llevado por una sola persona, organizar el trabajo en sprints y backlog dio claridad de objetivos a corto plazo y permitió presentar avances tangibles periódicamente, manteniendo la motivación alta y facilitando ajustes a tiempo.

4.2. ARQUITECTURA DEL SISTEMA

Desde la fase de diseño técnico se optó por una arquitectura modular, separando claramente las responsabilidades de cada componente para evitar dependencias innecesarias. Esta decisión de diseño permitió:

- Tener un sistema fácil de depurar, ya que los problemas podían aislarse en módulos específicos.
- Posibilidad de reutilización de código en futuras misiones u otros proyectos.
- Aislamiento de errores; un fallo en la lógica de una misión afecta solo a esa misión y no a todo el juego, puesto que cada una corre en su propio controlador.
- Escalabilidad. Se pueden añadir nuevas misiones, objetos o interacciones sin necesidad de reescribir la base del código, simplemente creando nuevos módulos con la misma estructura.

Con respecto a patrones de diseño, se aplicó principalmente el patrón Singleton para los gestores globales, y una variante del patrón Controlador para cada misión (similar al patrón *State*). También se siguió una arquitectura de escenas separadas para aislar contextos (menú, juego principal, etc.), comunicándose a través del estado persistente.

Los scripts se comunican entre sí de manera controlada mediante varias técnicas como las llamadas a métodos públicos de otros scripts, el uso de eventos/delegados para notificar acciones sin acoplar directamente los componentes o el uso de flags y enumeraciones de estado para que otros scripts comprueben estados globales.

Gracias a esta arquitectura, cada sistema tiene únicamente la información necesaria, estando la mayoría de las interacciones orquestadas a través del GameManager o de eventos. Esto resulta en un código fácilmente sostenible, donde agregar una nueva misión significaría crear un nuevo MissionController, pero sin tocar el resto de los sistemas.

4.2.1. Programación orientada a componentes

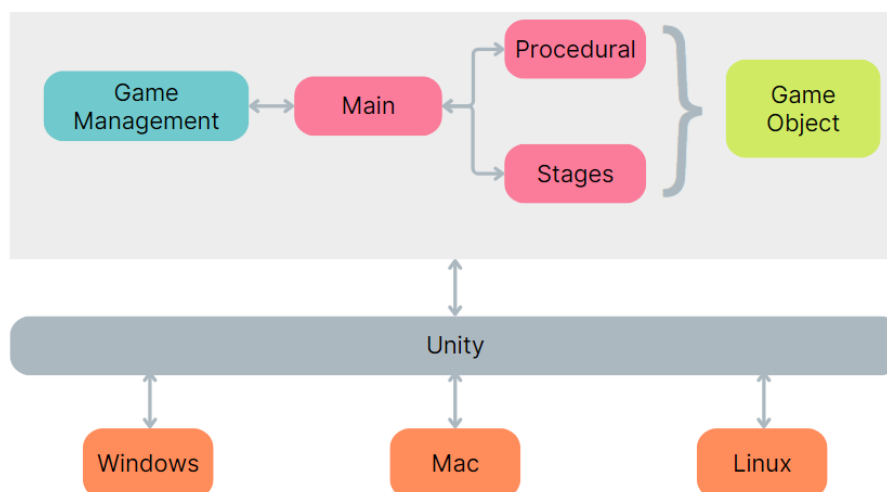


Figura 2. Arquitectura de este proyecto.

Unity se basa en la programación orientada a componentes, ya que todos los objetos del juego están formados por componentes que definen un comportamiento específico, debido al gran peso que tiene la composición frente a la herencia en este motor. Todo esto conforma la arquitectura de este proyecto y puede verse en la Figura 2.

Como paso previo al desarrollo de la evolución del proyecto y con el objetivo de una mejor comprensión de esta, en este apartado se detallan algunos de los elementos principales que componen la arquitectura de un proyecto en Unity.

- **GameObject.** Los GameObjects son todos aquellos objetos que componen un juego, desde los personajes o el escenario, hasta los efectos especiales. La peculiaridad que presentan estos objetos es que no son capaces de realizar ninguna acción por sí mismos, sino que necesitan almacenar una serie de componentes que implementen alguna funcionalidad. Un claro ejemplo de esto es la Figura 3, donde se puede apreciar un GameObject llamado “Cube”, que almacena componentes como ‘Transform’ o ‘Box Collider’, entre otros.

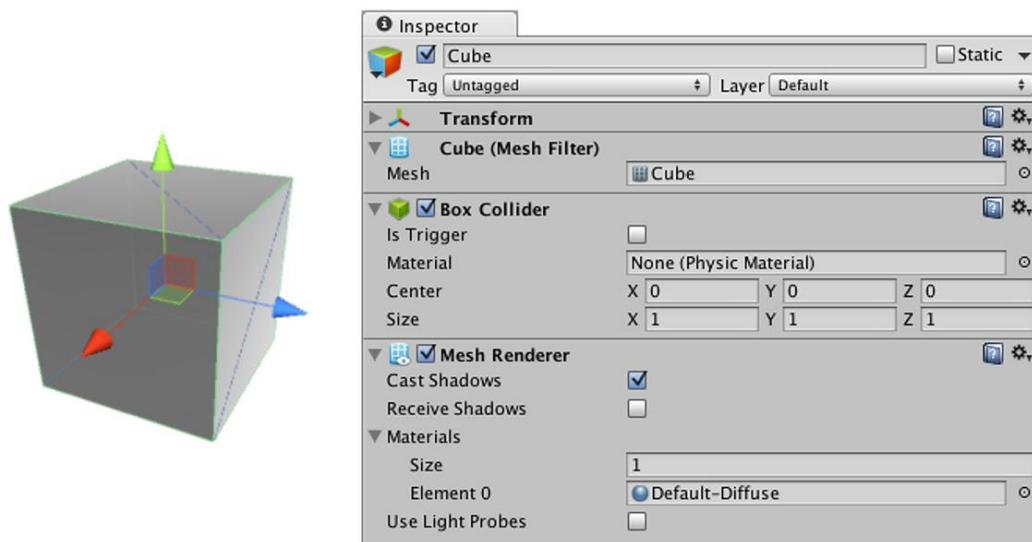


Figura 3. GameObject con varios componentes.

- **Jerarquía.** La ventana de jerarquía contiene todos los GameObjects que conforman la escena actual. Estos objetos pueden ser desde instancias directas de archivos hasta objetos personalizados. Además, la jerarquía utiliza relaciones de parentesco para relacionar objetos entre sí, de manera que se puede crear un objeto “padre” que contenga una serie de objetos “hijo”. En la Figura 4 se puede ver un ejemplo de la ventana de jerarquía y las relaciones de parentesco, donde Environment es el objeto padre, y SpaceShip, videoPlayer, Lights, etc. son los hijos que lo componen.

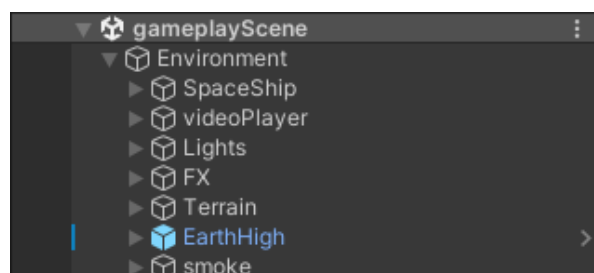


Figura 4. Ventana de jerarquía con relaciones de parentesco entre GameObjects.

- **Escenas.** Las escenas son elementos imprescindibles en Unity que pueden ser utilizadas bien para crear un menú principal o bien para constituir distintos niveles de dificultad en un juego. Cada juego cuenta con, al menos, una escena, y cada una de ellas está formada por una serie de GameObjects. En el caso del presente proyecto, se decidió contar con una única escena.
- **Animaciones.** Unity permite crear y asignar animaciones a los GameObjects de la escena con el objetivo de dotar al juego de un mayor dinamismo. Esto puede llevarse a cabo importando animaciones desde fuentes externas o creándolas desde cero dentro del editor utilizando la ventana de animación. Esta ventana permite animar múltiples propiedades de los objetos, desde la posición, escala y rotación de este, pasando por la intensidad de la luz, hasta el color de un material. Todas las animaciones pueden organizarse gracias al Animator Controller, que es una estructura similar a la del diagrama de flujo que mantiene un seguimiento de la animación que se está reproduciendo en cada momento.
- **Cámara.** La cámara es un objeto que define una determinada vista en una escena. Su posición determina el punto de vista desde el cual se observa la escena, mientras que sus ejes X, Y y Z definen la orientación completa de la cámara: el eje X controla la dirección horizontal, el eje Y define la orientación vertical y el eje Z determina la dirección de la vista. Además, los componentes de la cámara permiten ajustar el tamaño y la forma de visualización, proporcionando control sobre el campo de visión.
Es importante resaltar que la cámara únicamente mostrará todos aquellos objetos que estén en modo visible y dentro del rango que esta misma abarca.
En este caso, al tratarse de un videojuego en primera persona, la cámara ha sido asignada al propio jugador, tal y como puede verse en la Figura 5, de manera que mostrará todo lo que el jugador puede ver mientras se mueve por la escena.



Figura 5. Posición y visión de la cámara en primera persona.

4.3. EVOLUCIÓN DEL PROYECTO

El desarrollo de *Prometheus-9* sigue una serie de fases bien definidas que abarcan desde la concepción inicial del proyecto hasta la obtención de un producto final. A continuación, se describirán detalladamente todas y cada una de estas fases, explicando las actividades realizadas, las herramientas utilizadas y las decisiones clave que han sido tomadas.

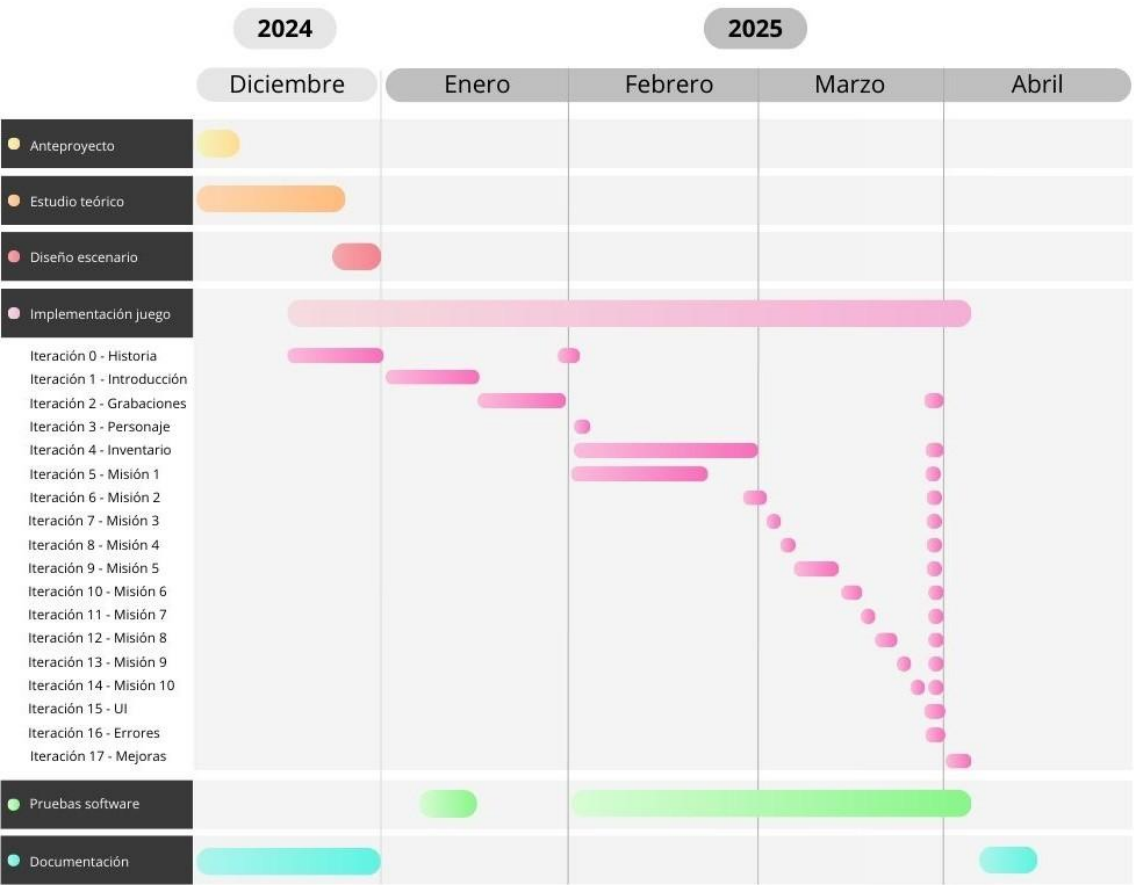


Figura 6. Cronograma de evolución del proyecto.

4.3.1. Anteproyecto

Esta primera fase constituyó el punto de partida del proyecto, abarcando los primeros días del mes de diciembre de 2024. Durante la misma, se definieron las bases iniciales de *Prometheus-9*, conceptualizando el videojuego narrativo y planificando su ejecución.

Principalmente, en esta fase tuvieron lugar múltiples sesiones de *brainstorming* para idear los cimientos de la historia y las mecánicas de juego, de manera que la creatividad jugó un papel fundamental: se exploraron diversas ideas de trama, se plantearon los personajes principales y se imaginó el estilo narrativo. También se delineó el alcance inicial; se definió que el juego contaría con una breve introducción y se identificaron características clave a incluir (sistema de decisiones, sistema de diálogo interactivo, mecánicas de exploración, etc.).

Una decisión clave tomada durante esta fase fue la de utilizar metodologías ágiles. Si bien el anteproyecto en sí mismo se elaboró antes de iniciar los sprints, podríamos considerar esta etapa como un Sprint 0. Seguidamente se elaboró el Product Backlog inicial a partir de las principales características que se querían incluir en el proyecto y se descompusieron en tareas de complejidad moderada, dando lugar así a una hoja de ruta inicial que guiará los sprints.

4.3.2. Estudio Teórico

Una vez asentadas las bases conceptuales, estratégicas y organizativas del proyecto, se lleva a cabo un estudio teórico profundo para respaldar el desarrollo con fundamentos académicos y técnicos. Esta fase, que tuvo lugar durante gran parte del mes de diciembre de 2024, tiene como objetivo principal la investigación del estado del arte en distintas áreas: gamificación, narrativas interactivas, diseño de videojuegos con temáticas similares, etc., de manera que se trató de obtener referencias a buenas prácticas, y aprender de soluciones existentes para evitar errores conocidos. Se investigaron juegos narrativos destacados, publicaciones académicas y numerosos artículos, lo cual fue de gran ayuda a la hora de aportar conceptos de diseño narrativo

En relación con la metodología Scrum, se realizó también un breve estudio teórico acerca de su utilización en el desarrollo de videojuegos, lo que permitió ampliar conocimientos previos y reforzar el plan metodológico inicial.

Durante esta primera etapa, se amplió el Product Backlog con tareas derivadas de la investigación, que, posteriormente, fueron ejecutadas en sprints.

4.3.3. Diseño

Esta etapa fue desarrollada durante la última semana de diciembre de 2024 y en ella se materializó en detalle cómo sería *Prometheus-9*. En lo que a diseño narrativo se refiere, se escribió el guion completo del juego; abarcando la descripción de escenas, los diálogos entre personajes, eventos desencadenantes, los posibles finales, etc.

Se estructuró una narrativa dividida en grabaciones, asegurándose de que cada una de ellas tuviera un arco dramático y alimentara la intriga general. Se identificaron puntos de decisión donde el jugador tuviera la oportunidad de escoger distintos cursos de acción. Para manejar la complejidad, se utilizó un diagrama de flujo narrativo (Figura 7) donde se marcaron los nodos (escenas) y bifurcaciones.

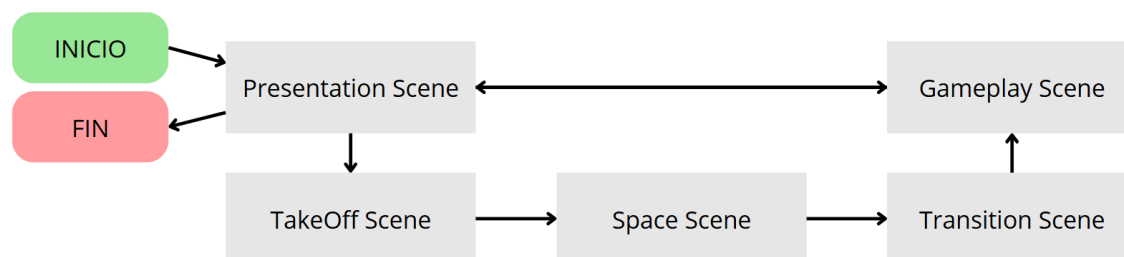


Figura 7. Diagrama de flujo narrativo de escenas.

En cuanto al diseño de mecánicas y gameplay, *Prometheus-9* al ser un juego narrativo no tenía mecánicas de combate o físicas complejas como tal, pero sí se definieron diversas interacciones: movimiento del personaje, interacción con objetos, creación de un inventario, resolución de misiones, etc.

Otro aspecto de gran relevancia fue el diseño de la escena de juego, fuertemente relacionado con el diseño las misiones. Para ello, se consultaron distintos mapas de las secciones principales de una nave espacial, buscando definir áreas concretas como una zona de observación, una sala de comunicaciones, un departamento de tecnología, un laboratorio, etc. definiéndose así el punto exacto donde se encontrarían los elementos interactivos y los puntos de activación de determinados eventos.

También se planificó un listado de los modelos 3D necesarios para conformar las áreas anteriormente descritas y efectos de música o sonidos para mejorar la ambientación.

Las herramientas empleadas en esta fase incluyeron Microsoft Word nuevamente, para escribir los guiones, finales, listados, etc.; Canva, para la realización del diagrama de flujo narrativo y se comenzó a utilizar Unity para conformar las escenas del juego. También se inició un repositorio de código con control de versiones (GitHub).

Es importante resaltar las decisiones clave abarcadas durante la fase de diseño, siendo las principales la definición completa del alcance del proyecto y la priorización de las tareas a realizar, lo cual alimentó en gran medida la planificación de los sprints backlog. Con respecto a la metodología ágil, esta fase no ocurrió enteramente antes de comenzar la implementación, sino que tuvo lugar en paralelo durante los primeros sprints de la misma lo que ayudó a mejorar notablemente la productividad de la autora.

Resta decir, que esta etapa concluye con el guion del juego completamente finalizado, así como el entorno de juego preparado para comenzar la implementación de las mecánicas de juego. En la Figura 8 puede observarse el estado del proyecto tras la fase de diseño.

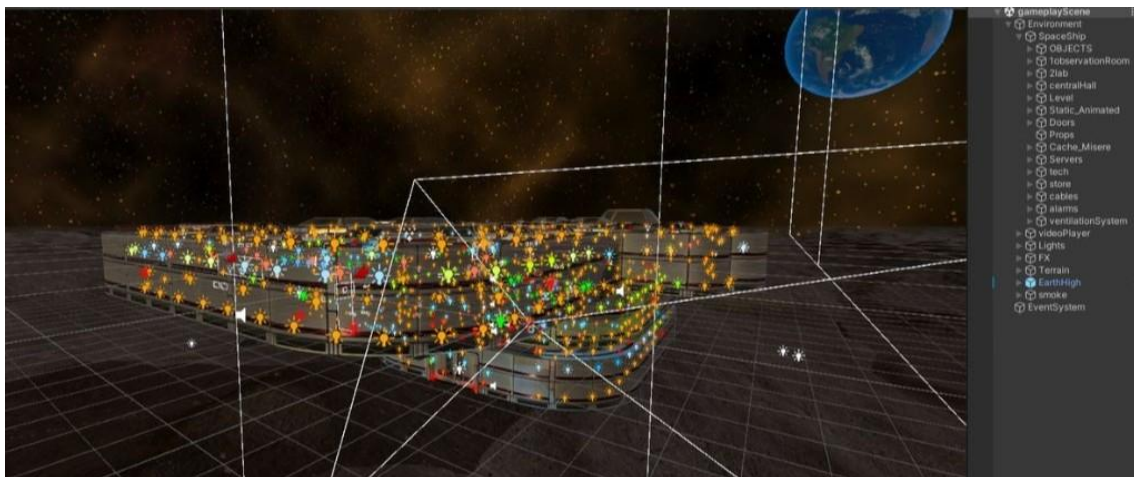


Figura 8. Proyecto tras finalizar la fase de diseño.

4.3.4. Implementación

La fase de implementación representa, sin lugar a duda, la etapa más extensa y compleja de todo el proyecto. Su duración se extiende desde mediados de diciembre de 2024 hasta los primeros días de abril de 2025, abarcando así un amplio período de trabajo intensivo en el que se materializan todas las ideas, diseños y conceptos previos.

Durante esta fase se lleva a cabo el desarrollo técnico del videojuego utilizando el motor Unity, lo cual incluye la programación de las mecánicas de juego, la integración de los distintos assets gráficos y sonoros, la construcción de las misiones, la implementación de escenas narrativas y cinemáticas, así como la definición de los posibles finales.

Es relevante señalar que esta etapa se estructura en múltiples iteraciones, cada una con objetivos bien definidos y temporales específicos, lo que permite un control detallado del avance del proyecto y facilita la adaptación progresiva a posibles cambios o mejoras. A continuación, se describen en detalle cada una de estas iteraciones.

4.3.4.1. Desarrollo de la historia

La primera iteración tuvo una duración aproximada de dos semanas y se centró principalmente en la construcción narrativa del videojuego. Dado que el proyecto se inscribe dentro del género de los videojuegos narrativos, era fundamental definir desde el inicio una historia sólida, coherente y capaz de captar el interés del jugador.

En esta etapa se decidió apostar por una ambientación de ciencia ficción. La historia gira en torno a un personaje que, aparentemente, es el único superviviente tras un aterrizaje forzoso de la nave espacial. A partir de este suceso, el jugador deberá descubrir qué ocurrió realmente a bordo de la nave, explorando sus instalaciones, recolectando pistas y reviviendo fragmentos del pasado mediante grabaciones de seguridad.

Como primer paso en esta iteración, se trabajó en la definición de los personajes que formarían parte de la historia, cada uno de los cuales fue concebido con un trasfondo particular y un rol específico dentro de la misión. Esta diversidad de personajes permite construir una red de relaciones, conflictos y motivaciones que enriquecen la trama y favorecen la aparición de giros argumentales relevantes.

En la Figura 9 puede observarse la información general de cada uno de los personajes.

Madison Vega	Liam Carter	Nicole Bennett	Austin Prescott	Cooper Hartman
				
Comandante de la misión	Ingeniero de Sistemas	Bióloga Espacial	Astrónomo y Explorador	Especialista en IA y Comunicaciones

Figura 9. Personajes de la historia. Información general.

4.3.4.2. Introducción al juego

Esta iteración se desarrolla a lo largo de las dos primeras semanas de enero de 2025 y mantiene una duración similar a la de la fase anterior. El propósito principal de este período de trabajo es la construcción, dentro del motor Unity, de las distintas escenas que componen la secuencia introductoria del videojuego. Esta secuencia resulta fundamental para contextualizar la historia, presentar el universo narrativo y establecer las bases argumentales que guiarán la experiencia del jugador.

Durante esta etapa se diseñan, ambientan y programan las escenas iniciales, que tienen como función sumergir al jugador en la misión espacial que está a punto de comenzar, así como introducir a los personajes principales y establecer el tono narrativo general. Las escenas desarrolladas en esta fase son las siguientes:

- **Presentation Scene.** En esta escena se lleva a cabo la presentación de la misión espacial que partirá rumbo al sistema estelar Helion-3, una expedición científica y de exploración sin precedentes. Además, se presentan uno a uno los distintos tripulantes de la nave espacial *Prometheus-9*, brindando al jugador información relevante sobre sus roles y personalidades. Esta introducción se apoya en recursos visuales y sonoros que refuerzan el carácter trascendental de la misión.
- **TakeOff Scene.** Aquí se muestra el despegue de la nave. A través de una cinemática cuidadosamente elaborada, se puede observar cómo la *Prometheus-9* asciende lentamente, abandonando el suelo terrestre y desapareciendo en el horizonte, transmitiendo una sensación de aventura y expectación.
- **Space Scene.** Esta escena traslada al jugador al espacio exterior, donde la nave ya ha superado la atmósfera terrestre y se encuentra alejándose progresivamente del planeta. Se trata de una secuencia de transición que refuerza la inmersión espacial, mostrando el silencio y la majestuosidad del cosmos.
- **Transition Scene.** Finalmente, esta escena representa el momento crítico en el que la nave atraviesa una zona de turbulencias extremas. El entorno comienza a vibrar violentamente, las alarmas se activan, y el personaje protagonista sufre un desfallecimiento repentino como consecuencia de la inestabilidad del viaje. Esta secuencia marca un punto de inflexión en la narrativa, anticipando que algo fuera de lo común está a punto de suceder y dejando al jugador con una sensación de inquietud e incertidumbre.

Cada una de estas escenas ha sido desarrollada poniendo especial atención en el ritmo narrativo, la ambientación visual y sonora, y la coherencia entre las animaciones y los efectos. De esta manera, se establece una introducción sólida y envolvente que prepara al jugador para sumergirse plenamente en la historia del juego.

4.3.4.3. Grabaciones de seguridad

Durante esta tercera iteración, desarrollada a lo largo de las dos últimas semanas de enero de 2025, el foco principal se centra en la construcción y preparación de los distintos escenarios que conformarán las grabaciones de seguridad de la nave espacial. Estas grabaciones constituyen una pieza clave dentro del sistema narrativo del juego, ya

que permiten al jugador reconstruir los eventos ocurridos a través de fragmentos visuales interactivos.

Para alcanzar este objetivo, se procedió a la definición y ambientación de los espacios más relevantes dentro de la nave, tales como la sala de observación, el laboratorio científico, el departamento de tecnología, la sala de comunicaciones, entre otros. Cada uno de estos entornos fue diseñado cuidadosamente, teniendo en cuenta tanto su funcionalidad dentro del universo del juego como su coherencia visual y narrativa.

Además, se implementaron diversas animaciones específicas y se generaron los diálogos correspondientes a los personajes que interactúan en cada uno de estos escenarios. Estas secuencias permiten dar vida a las escenas y refuerzan la inmersión del jugador, aportando información clave para entender la historia general y los conflictos que se desarrollan a bordo de la nave Prometheus-9.

4.3.4.4. Movimientos del personaje

Esta iteración es la más breve de todo el proceso de desarrollo, ya que su duración se limita a un par de días. A pesar de su extensión, se trata de una etapa clave para garantizar la interacción del jugador con el entorno del juego, ya que su objetivo principal es implementar un sistema de control que permita el desplazamiento libre por el escenario.

El primer paso consistió en analizar las posibles mecánicas de movimiento que podrían ser relevantes dentro del contexto del juego. Se contemplaron distintas opciones como correr, agacharse, trepar, deslizarse o impulsarse en gravedad cero. Sin embargo, tras una evaluación basada en la naturaleza del entorno y el estilo narrativo del juego, se descartaron aquellas mecánicas que no aportaban valor significativo a la experiencia del jugador o que resultaban innecesarias en los espacios disponibles.

Finalmente, se optó por un sistema de movimiento básico pero efectivo, que incluye desplazamiento en todas las direcciones (avance, retroceso, y giros laterales hacia la derecha e izquierda), así como la capacidad de saltar. Estas acciones ofrecen la libertad suficiente para explorar las distintas áreas de la nave espacial sin complicar los controles.

4.3.4.5. Inventario

Esta iteración representa la más extensa dentro de la fase de implementación, con una duración estimada de aproximadamente un mes. Tal como se muestra en la Figura 6, su desarrollo transcurre en paralelo a la iteración correspondiente a la Misión 1, explicada en el apartado siguiente. Esta simultaneidad se debe a la necesidad de contar con objetos coleccionables y grabaciones ya presentes en la escena de juego, ya que su existencia es indispensable para poner en funcionamiento el sistema de inventario. De esta manera, se asegura que el jugador pueda interactuar con elementos relevantes desde el principio, recolectarlos y almacenarlos adecuadamente, facilitando la progresión en las misiones.

El objetivo principal de esta interacción es el diseño y la programación del sistema de inventario del juego, una funcionalidad esencial dentro del juego. El inventario fue concebido para ser intuitivo y fácilmente accesible mediante la tecla I, permitiendo al jugador visualizar todos los elementos recolectados, junto a su nombre.

En la Figura 10 pueden observarse los dos apartados principales del inventario, el de las grabaciones de seguridad y el de los objetos recolectados.



Figura 10. Secciones principales del inventario.

Podemos observar también cómo las siguientes figuras (Figura 11 y Figura 12) muestran el inventario de grabaciones y objetos respectivamente.



Figura 11. Inventario de grabaciones de seguridad.



Figura 12. Inventario de objetos recolectables.

4.3.4.6. Misión 1

Esta iteración, como se ha mencionado anteriormente, se desarrolla de manera paralela a la iteración del inventario, abarcando casi la totalidad del mes de febrero. Su objetivo principal es el de introducir al jugador en las mecánicas básicas de interacción con los elementos de la nave.

Esta misión se desarrolla en la sala de observación, que es justo el punto en el que el jugador pierde el conocimiento debido a las fuertes vibraciones de la nave. La misión representa uno de los momentos iniciales de acción dentro del juego, en la que el jugador debe restablecer el suministro eléctrico general de la nave, el cual ha sido dañado tras el aterrizaje forzoso. Para ello, debe buscar en los armarios de dicha sala y encontrar una palanca, con la que podrá forzar una puerta metálica que, tras su apertura, le brindará acceso al panel eléctrico de la nave.

En este punto, el usuario deberá interactuar con el panel utilizando el teclado, de manera que consiga apagar todos los interruptores del mismo y, seguidamente, encenderlos.

En caso de que el jugador complete la misión correctamente y sin demasiadas dificultades, el sistema eléctrico de la nave se restablecerá por completo. Esto provocará el encendido automático de las luces en las distintas áreas de la nave, así como la apertura de algunas puertas de seguridad que habían quedado bloqueadas a raíz del apagón. Además, se le proporcionará al usuario un código con el que podrá desbloquear una caja de seguridad donde se encuentra la primera de las grabaciones.

En cambio, si el jugador falla repetidamente al intentar accionar los interruptores (pulsando un número excesivo de teclas sin lograr reiniciar el sistema), se activarán las alarmas de emergencia. Como consecuencia, el personaje sufrirá una penalización en forma de pérdida de vida y energía, lo que aumentará la tensión del entorno y dificultará el avance en las siguientes tareas.

A continuación, la Figura 13, presenta visualmente esta misión en profundidad.



Figura 13. Misión 1.

4.3.4.7. Misión 2

Esta séptima iteración se desarrolló durante los últimos días de febrero y los primeros días de marzo de 2025. Su duración fue relativamente breve, ya que el alcance de la misión se encuentra bastante acotado.

El objetivo principal de esta misión es que el jugador explore las zonas habilitadas de la nave en busca de cuatro muestras biológicas que deben ser examinadas en el laboratorio. El tiempo con el que cuenta para hacerlo es limitado, por lo que esta tarea requiere gran nivel de atención y observación por parte del usuario, además de rapidez en la ejecución. Una vez recolectadas las muestras, estas deben ser depositadas en el Sistema de Identificación Genética, ubicado en la mesa principal del laboratorio, donde se iniciará de forma automática un proceso de análisis que tomará unos segundos por muestra.

Si el jugador logra completar la misión dentro del tiempo establecido, el sistema revelará que el origen de las muestras es desconocido, lo que constituye una pista clave en el desarrollo narrativo del juego. Además, como recompensa, el jugador podrá obtener acceso a un código que le permitirá desbloquear la caja de seguridad que contiene la segunda grabación, pieza fundamental para avanzar en la reconstrucción de la historia.

Sin embargo, si el jugador no consigue completar la misión a tiempo, una de las muestras liberará una sustancia contaminante que afectará el entorno del laboratorio. Esto tendrá un impacto directo sobre el personaje, reduciendo sus niveles de vida y oxígeno. Como consecuencia, la pantalla comenzará a volverse borrosa y más oscura, y el jugador experimentará dificultades visuales hasta que logre recuperar el oxígeno perdido.

La Figura 14, muestra en mayor detalle la misión explicada anteriormente.

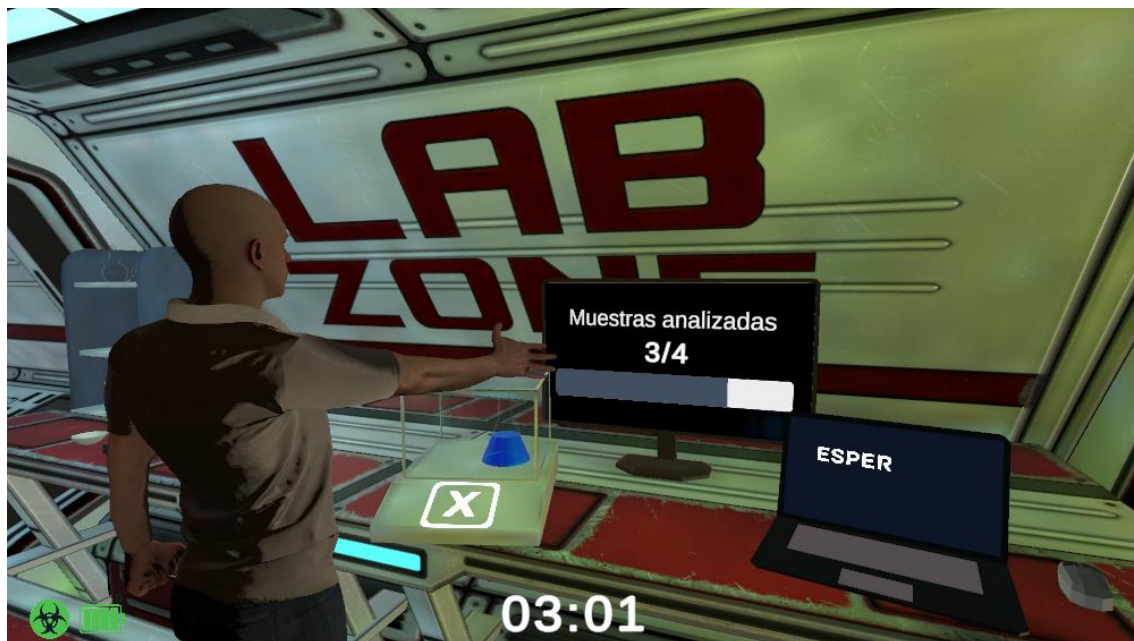


Figura 14. Misión 2.

4.3.4.8. Misión 3

Esta iteración se llevó a cabo durante los primeros días de marzo de 2025 y tuvo una duración muy breve, de apenas un par de días. Esto es debido, en gran parte, al nivel de familiaridad que la autora había alcanzado con el motor Unity y su lenguaje de programación, lo que permitió una implementación ágil y eficiente.

En esta misión, el jugador se enfrenta a una fuga de gas que ha comenzado a expandirse por el laboratorio, poniendo en riesgo su salud y el equilibrio del entorno. Para resolver esta situación, el jugador debe localizar una llave inglesa almacenada en uno de los armarios de la nave. Una vez encontrada y recolectada, podrá utilizarla para ajustar tres válvulas específicas responsables de la fuga.

La reparación de estas válvulas se lleva a cabo mediante un minijuego interactivo que emplea una barra de gradación de colores, representando diferentes niveles de presión. El reto principal consiste en mantener una flecha oscilante dentro de la zona verde (área segura), evitando que se detenga en las zonas naranja o roja, que indican peligro. Esta mecánica pone a prueba la destreza, precisión y coordinación visual del jugador, ya que es la primera vez en el juego que se introduce una interfaz de respuesta inmediata con consecuencias acumulativas.

Si el jugador logra mantener la flecha en el área segura, la fuga se irá reduciendo progresivamente hasta ser sellada por completo. En cambio, si no consigue controlar adecuadamente la presión, la fuga aumentará, provocando una pérdida progresiva de oxígeno y energía, lo que afectará directamente al estado físico del personaje. Como en el resto de las misiones, al completarla exitosamente, el usuario encontrará la clave de desbloqueo de una caja de seguridad, que le permitirá acceder a la siguiente grabación.

La misión se apoya en un entorno altamente inmersivo, con efectos visuales y sonoros que intensifican la sensación de urgencia: el gas comienza a salir con más fuerza de las

válvulas, la pantalla se distorsiona levemente con un efecto de niebla, la respiración del personaje se acelera y sus movimientos se ralentizan a medida que empeora la situación. Esta misión se destaca especialmente por su capacidad para generar tensión.

A continuación, se presenta una representación visual que permite una mejor comprensión de esta misión (Figura 15).



Figura 15. Misión 3.

4.3.4.9. Misión 4

Esta iteración acontece durante los primeros días de marzo de 2025 y su implementación se completó en apenas un par de días. A pesar de su corta duración, se trata de una misión clave, ya que su resolución permite al jugador avanzar a nuevas zonas de la nave.

El objetivo principal de esta misión consiste en reparar un analizador electrónico, que es un dispositivo que controla el acceso a una de las puertas principales de la nave. Para lograrlo, el jugador debe superar un minijuego de secuencias, en el que se le presentan combinaciones de letras que debe repetir correctamente en el mismo orden. La dificultad del desafío incrementa gradualmente, exigiendo concentración, agilidad mental y memoria visual por parte del usuario.

En caso de completar la misión con éxito, el sistema reconocerá la reparación del analizador y se desbloqueará automáticamente una puerta de acceso, permitiendo al jugador explorar nuevas áreas de la nave que hasta ese momento permanecían inaccesibles. Este avance es fundamental tanto para la progresión narrativa como para la exploración y resolución de futuras misiones. Además, al finalizar la misión, el jugador podrá encontrar oculto en el propio analizador el código que le va a permitir el desbloqueo de la siguiente grabación de seguridad.

Sin embargo, si el jugador falla reiteradamente al ingresar las secuencias, se producirá un cortocircuito en el sistema, lo que desencadenará una penalización directa sobre el personaje, reduciendo su nivel de vida disponible.

Esta misión se ejemplifica de forma más clara en la Figura 16.

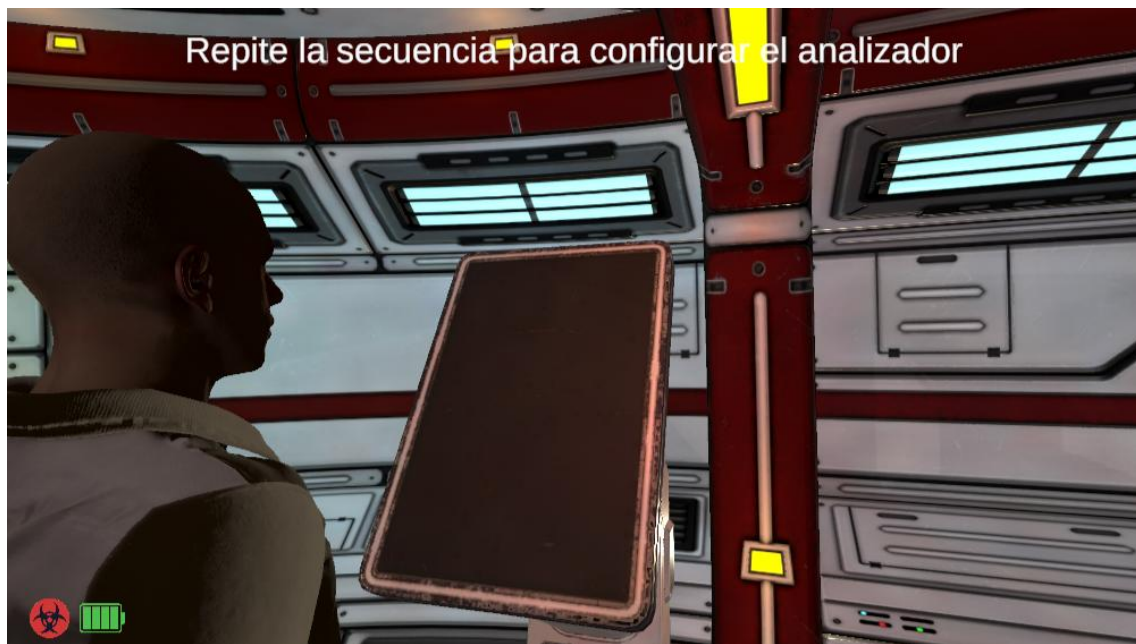


Figura 16. Misión 4.

4.3.4.10. Misión 5

Esta iteración se desarrolla durante la primera mitad de marzo de 2025 y cuenta con una duración aproximada de una semana. El objetivo principal de esta misión es poner a prueba los reflejos, la coordinación y las habilidades de desplazamiento del jugador en un entorno altamente controlado y sensible al error.

Durante esta misión, el jugador debe avanzar cautelosamente por una sala equipada con un sistema de sensores de seguridad. El objetivo es alcanzar uno de los armarios situados al otro extremo de la sala para obtener un cortador de cables, herramienta indispensable para desactivar cuatro conexiones que alimentan el sistema de alarmas. Sin embargo, la dificultad reside en que cualquier contacto con los sensores activará de inmediato las alarmas de la sala, generando consecuencias tanto para el entorno como para el jugador.

Si se completa la misión exitosamente sin ser detectado por el sistema de seguridad, el jugador obtendrá como recompensa un código exclusivo que le permitirá desbloquear la caja de seguridad que contiene la siguiente grabación.

Por el contrario, si el jugador colisiona con los sensores, se activará una alarma sonora y se desencadenarán una serie de penalizaciones: disminuirán sus niveles de vida y energía lo que provocará un ralentizamiento de sus movimientos, aumentando la probabilidad de sufrir nuevas colisiones. Además, tras cada colisión, se produce un corte del suministro eléctrico de la nave, oscureciendo parcialmente los entornos y bloqueando puertas que previamente estaban accesibles, lo cual complica aún más la resolución de la misión.

Esta misión está inspirada en mecánicas clásicas de sigilo, y ha sido cuidadosamente calibrada para ofrecer un equilibrio entre desafío y accesibilidad, incluyendo elementos que incrementan la tensión en el jugador como luces parpadeantes, sensores en movimiento y sirenas sonoras.

En la Figura 17 se muestra la entrada a la zona equipada con sensores de seguridad, fácilmente identificables por los círculos rojos que aparecen en la imagen. Por otro lado, la Figura 18 ofrece una vista ampliada de la misión, en la que el jugador debe cortar los cuatro cables que conforman el sistema de alarma, utilizando el cortador previamente obtenido.



Figura 17. Zona de acceso a la misión 5.



Figura 18. Misión 5.

4.3.4.11. Misión 6

Esta iteración se llevó a cabo a mediados de marzo y tuvo una duración de cuatro días. A diferencia de otras misiones anteriores, esta implicó un mayor nivel de complejidad, principalmente por el componente lógico que introduce, siendo su objetivo principal diseñar una misión basada en deducción, interpretación de patrones y atención al detalle.

El propósito de la misión es que el jugador descifre un mensaje de origen desconocido encontrado en un libro antiguo, utilizando para ello una tabla de traducción de símbolos. Para su desarrollo, primeramente, la autora llevó a cabo una investigación con el fin de seleccionar una colección de 36 símbolos únicos, a los que se asignó individualmente un carácter del alfabeto latino o un número. Más tarde, se diseñó la tabla de traducción, que fue implementada como objeto coleccionable dentro del juego, de manera que el jugador deberá recogerla antes de poder iniciar correctamente la tarea de descifrado.

Una vez comenzada la misión, el jugador deberá observar la tabla de símbolos e ir completando manualmente la traducción del mensaje en un espacio habilitado para ello en la página derecha del libro, como puede apreciarse en la Figura 19.

En caso de que la traducción sea exitosa, el jugador recibirá una doble recompensa. Por un lado, los símbolos destacados en color naranja se traducirán como una secuencia numérica que funciona como código de desbloqueo de la caja de seguridad que contiene la siguiente grabación. Por otro lado, el propio mensaje actúa como clave para desbloquear una puerta que da acceso a la zona de almacenamiento de la nave.

Sin embargo, si el jugador no logra descifrar correctamente el mensaje, no podrá acceder a dicha zona y perderá la oportunidad de recopilar dos grabaciones de seguridad (la de la misión actual, y la que se encuentra dentro de la zona de almacenamiento de la nave), lo que representa una gran pérdida a nivel narrativo. Esto no afecta directamente al estado del personaje, pero tiene un alto impacto emocional, ya que impide al jugador reconstruir una parte crucial de la historia. Esta misión, por tanto, introduce una dimensión más profunda en la narrativa del juego, centrada en la investigación y la lectura del entorno.

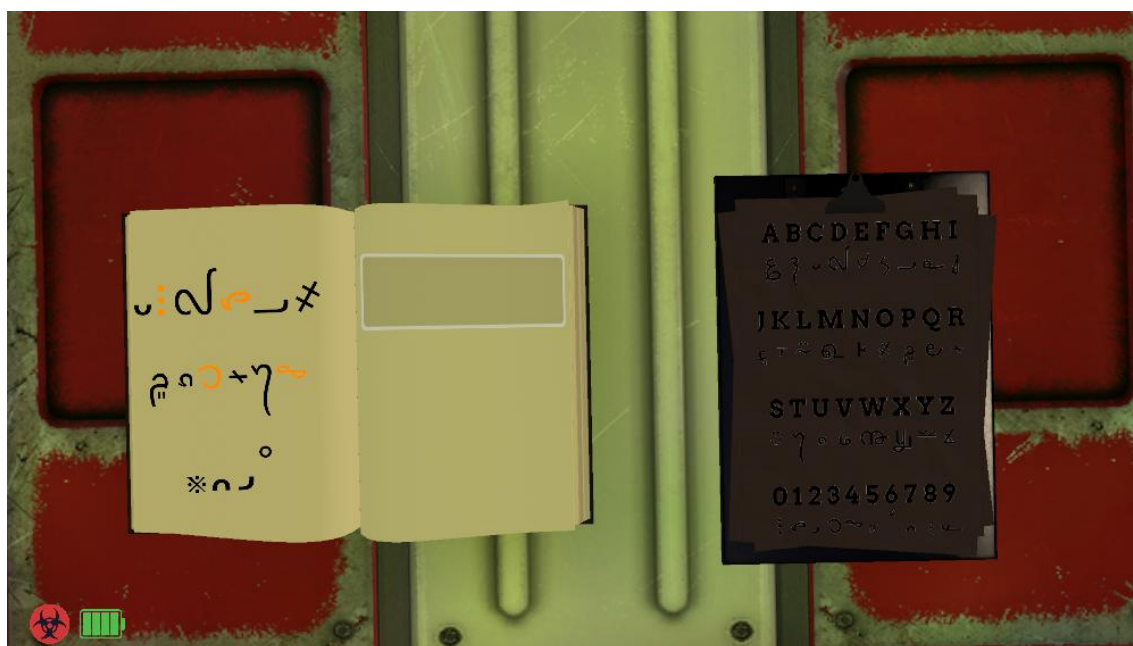


Figura 19. Misión 6.

4.3.4.12. Misión 7

Esta iteración tiene lugar durante la segunda mitad del mes de marzo y se extiende a lo largo de tres días aproximadamente. Su diseño busca ofrecer al jugador libertad de exploración, aunque bajo una presión constante marcada por el tiempo y la gestión de recursos, lo que refuerza la necesidad de actuar con eficacia y rapidez.

Una particularidad de esta misión es que su acceso está condicionado al éxito de la misión anterior. Como se explicó previamente, el almacén donde tiene lugar esta prueba permanece cerrado tras una puerta de seguridad que solo puede abrirse mediante la clave obtenida al traducir el mensaje alienígena. Esta conexión entre misiones promueve una estructura narrativa encadenada y recompensa la progresión lógica del jugador.

El objetivo específico de esta misión consiste en recorrer el almacén en busca de los elementos necesarios para conformar un kit médico de emergencia. El jugador dispone de un tiempo limitado para recolectar todos los objetos requeridos, los cuales están dispersos por el escenario: en las estanterías, sobre las cajas de embalaje, repartidos por el suelo, etc. Esto exige al jugador prestar especial atención al entorno y moverse con agilidad para no perder tiempo.

Si el jugador logra reunir todos los componentes antes de que se agote el tiempo, recibirá como recompensa un código que le permitirá desbloquear una caja de seguridad con una nueva grabación, clave para avanzar en la reconstrucción de los sucesos. En caso de no completar la tarea a tiempo, la misión se reiniciará con un temporizador reducido como penalización. Además, el personaje sufrirá una pérdida significativa de energía, lo que ralentizará sus movimientos e interacciones, haciendo más desafiante el segundo intento.

Esta misión combina exploración, presión temporal y penalizaciones progresivas, y actúa como una prueba de reacción bajo estrés que vincula el rendimiento del jugador con la recompensa narrativa, consolidando así la coherencia entre jugabilidad y relato.

La misión se representa con mayor claridad en la Figura 20, mostrada a continuación.



Figura 20. Misión 7.

4.3.4.13. Misión 8

Esta iteración tiene una duración aproximada de seis días, que transcurren durante la segunda mitad del mes de marzo. A diferencia de misiones anteriores, esta fase requirió un mayor nivel de precisión tanto en su diseño como en su implementación, lo que justificó un tiempo de desarrollo más prolongado. El principal objetivo de esta misión es reforzar el control de los movimientos del jugador, exigiendo un cierto nivel de concentración y coordinación.

Durante la misión, el jugador debe localizar un aspirador en uno de los compartimentos de limpieza de la nave. Una vez obtenido, deberá utilizarlo para limpiar completamente las rejillas del sistema de ventilación, desplazándolo cuidadosamente de un lado a otro mediante el uso de las teclas de dirección del teclado. El reto consiste en pasar el aspirador sobre las zonas cubiertas de suciedad hasta eliminarlas por completo antes de que expire el temporizador que marca el límite de tiempo para completar la tarea.

El fallo en esta misión se produce si el jugador no consigue limpiar todas las rejillas dentro del límite de tiempo. En ese caso, las partículas de suciedad se liberarán, provocando una contaminación del aire en el interior de la nave. Esto repercute directamente sobre el estado de salud del personaje, reduciendo sus niveles de energía y oxígeno, lo que origina como consecuencia una pérdida progresiva de visión, así como una disminución en la agilidad de sus movimientos, complicando aún más la misión.

Esta misión destaca por su enfoque en la precisión y la atención al detalle, y por introducir una mecánica distinta a las vistas anteriormente, basada en el mantenimiento del entorno, lo cual contribuye a la credibilidad y realismo del ecosistema de la nave.

La misión correspondiente puede observarse en detalle en la Figura 21.

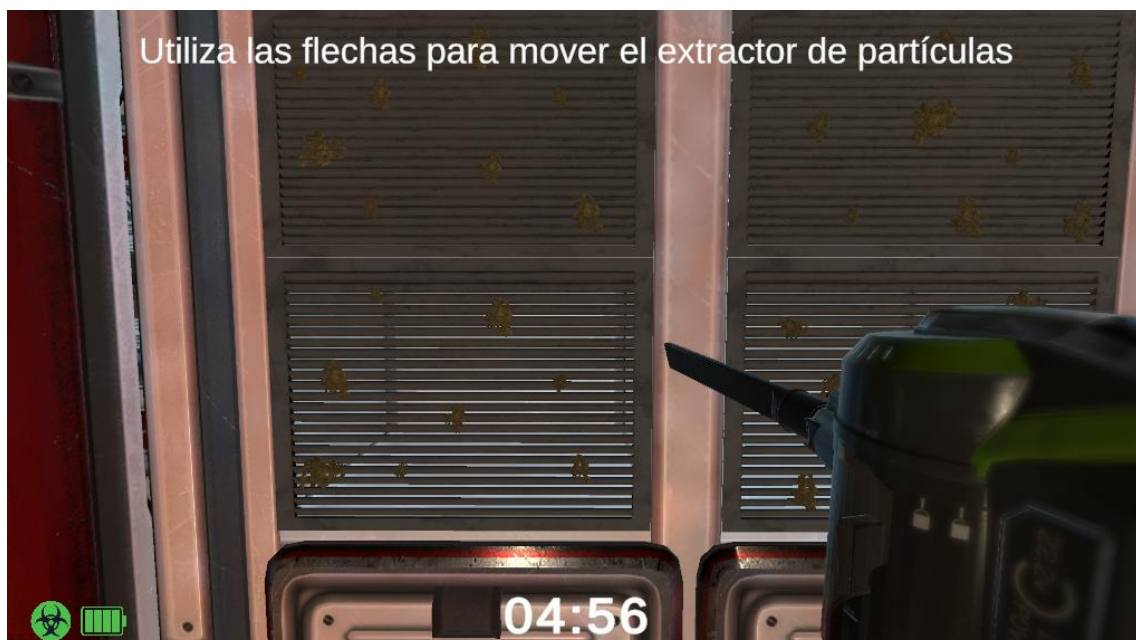


Figura 21. Misión 8.

4.3.4.14. Misión 9

Esta iteración se lleva a cabo durante los últimos días de marzo de 2025 y tiene una duración breve, de aproximadamente dos días. Aunque su desarrollo fue relativamente rápido, esta misión introduce un componente estratégico y lógico, enfocado en la planificación de movimientos.

El objetivo principal de esta misión es que el jugador recalibre el sistema de navegación de la nave para evitar un posible accidente. Para ello, primero deberá localizar una brújula especial guardada en uno de los armarios de la nave. Una vez con ella en su posesión, se activará un rompecabezas tipo 8 puzzle, en el que el jugador deberá reorganizar una serie de piezas que representan el panel de navegación de la nave. El desafío consiste en resolver el puzzle con un número limitado de movimientos.

Si el jugador logra completar el rompecabezas respetando el límite de movimientos, obtendrá como recompensa el acceso a la penúltima grabación de seguridad, almacenada en una caja protegida mediante un sistema de código. Por el contrario, si el jugador no consigue resolver el puzzle en los movimientos estipulados, será penalizado con una reducción progresiva de su vida por cada movimiento adicional realizado. Este diseño obliga al jugador a planificar sus acciones cuidadosamente, fomentando el pensamiento estratégico y evitando decisiones impulsivas.

Esta misión se distingue por su carácter puramente lógico y mental, aportando variedad mecánica al conjunto del juego y ofreciendo una pausa respecto a las misiones más centradas en acción o exploración. A su vez, refuerza la importancia de la anticipación y la precisión en un entorno donde cada decisión puede tener consecuencias relevantes.

Esta misión se ejemplifica de forma más clara en la Figura 22.



Figura 22. Misión 9.

4.3.4.15. Misión 10

Esta iteración se desarrolla durante los últimos días de marzo de 2025, con una duración aproximada de tres días. Su objetivo principal es reforzar la importancia de la observación y la toma de decisiones conscientes antes de actuar, poniendo a prueba la atención del jugador en un entorno de alta precisión.

La misión final tiene como propósito desbloquear la última puerta de la nave, la cual únicamente puede abrirse mediante un sistema de reconocimiento de huellas dactilares. Para ello, el jugador deberá encontrar una linterna de luz ultravioleta escondida en una sala, que utilizará para iluminar la superficie de la puerta y revelar tres huellas distintas ocultas a simple vista. Al interactuar con cada una de estas huellas, el jugador accederá a una interfaz tipo puzle, donde deberá recomponer la huella a partir de cuatro fragmentos desordenados. Cada huella desbloqueará progresivamente la puerta. La dificultad de esta misión radica en identificar patrones con rapidez sin cometer errores reiterados.

Si el jugador consigue resolver correctamente los tres puzles sin errores significativos, la puerta se desbloqueará automáticamente, dando acceso a la última grabación de seguridad. Además, como evento vinculado, se abrirá también el acceso a la sala de comunicaciones de la nave, donde el jugador podrá visualizar todas las grabaciones recolectadas y establecer un contacto final con la estación terrestre.

En caso de que el jugador falle reiteradamente al verificar las huellas (pulsando múltiples veces el botón de comprobación sin éxito), se activarán las alarmas de seguridad, provocando una reducción de la energía del personaje y alterando el entorno visual y sonoro de la escena. Esta penalización intensifica la tensión y enfatiza la necesidad de actuar con precaución y precisión.

A continuación, se presenta la Figura 23, que muestra esta misión en profundidad.



Figura 23. Misión 10.

4.3.4.16. Interfaz de Usuario

Esta iteración se desarrolla durante los últimos cuatro días del mes de marzo de 2025 y tiene como objetivo principal el diseño e implementación de la interfaz de usuario (UI) del juego, un componente fundamental para garantizar una experiencia clara, inmersiva y coherente para el jugador.

Durante esta fase, se trabajó en la creación de los indicadores principales del estado del jugador, concretamente los medidores de energía y oxígeno. Cada uno fue representado visualmente mediante un icono, como puede observarse en la esquina inferior izquierda de la Figura 24. Además, se integraron efectos audiovisuales complementarios para reforzar la inmersión y aumentar el sentido de urgencia en situaciones críticas: por ejemplo, cuando la vida disminuye, la pantalla comienza a oscurecerse progresivamente o cuando disminuye el oxígeno se produce una leve distorsión visual tipo visión borrosa, y se reproducen sonidos de respiración agitada.

Paralelamente, se implementaron indicadores del estado general de la nave, como alertas visuales y efectos ambientales que responden a cortes de electricidad, contaminación por gas o explosiones localizadas. Estos eventos tienen repercusiones tanto en el entorno (apagones, cierre de puertas, activación de alarmas) como en los recursos del personaje, generando una conexión directa entre la interfaz de usuario y la narrativa emergente.

Asimismo, se integró el acceso visual al inventario, se añadieron ayudas visuales en pantalla (como iconos, textos explicativos y marcadores que orientan al jugador) y feedback sonoro contextual (avisos auditivos).

El enfoque del diseño de la interfaz fue mantener una estética minimalista, funcional y coherente con la ambientación de la nave, evitando la saturación visual y asegurando que el jugador se mantenga inmerso en la experiencia sin distracciones innecesarias.



Figura 24. Interfaz de Usuario.

4.3.4.16. Detección de errores

Esta iteración se desarrolla de forma simultánea a la implementación de la interfaz de usuario, y está dedicada por completo a la fase de depuración del juego. Su objetivo principal es la identificación y corrección de errores técnicos y de jugabilidad, asegurando que todos los sistemas funcionen correctamente y que la experiencia del jugador sea fluida, estable y coherente.

La duración de esta iteración fue de tres días, durante los cuales se llevaron a cabo sesiones intensivas de testeo en distintos entornos del juego. Estas pruebas permitieron localizar fallos relacionados con la lógica de las misiones, errores en el funcionamiento del inventario, problemas en la interfaz, y comportamientos inesperados durante las transiciones de escena o la interacción con objetos.

Tal como se puede observar en el cronograma del proyecto (Figura 6), en este punto del desarrollo todas las misiones, así como el sistema de inventario, pasan por una segunda iteración. Esto se debe a que, una vez localizados los errores durante el testeo, fue necesario volver a trabajar sobre esas funcionalidades para aplicar las correcciones pertinentes y verificar su correcto funcionamiento tras los ajustes.

Esta fase fue fundamental para garantizar la consistencia interna del proyecto, así como para mejorar la usabilidad y estabilidad general antes de su cierre. El proceso de depuración no solo resolvió fallos, sino que también permitió realizar pequeños ajustes de equilibrio y optimización que mejoraron la experiencia global del jugador.

4.3.4.16. Implementación de mejoras

La última iteración correspondiente a la fase de implementación tiene lugar durante los tres primeros días del mes de abril de 2025. Esta etapa marca el cierre del desarrollo técnico del proyecto y su objetivo principal es aplicar ajustes y mejoras finales, centrados especialmente en la optimización el rendimiento general del juego.

Durante esta iteración se realizó una revisión de cada una de las escenas, mecánicas y sistemas implementados en el proyecto, tratando identificar posibles puntos de mejora. A partir de esta revisión, se aplicaron diversos ajustes que permitieron refinar la experiencia del jugador y garantizar una ejecución más fluida. Entre las tareas realizadas destacan:

- Optimización del rendimiento: se revisaron y ajustaron scripts, animaciones y efectos visuales para reducir el uso de recursos innecesarios, mejorar la tasa de fotogramas por segundo (FPS) y minimizar los tiempos de carga entre escenas.
- Ajustes en el equilibrio de dificultad: se reajustaron tiempos límite en algunas misiones y se refinaron las penalizaciones para asegurar un reto justo y progresivo.
- Mejoras visuales: se añadieron efectos de postprocesado y se corrigieron errores en la iluminación de ciertas áreas.
- Pulido de interfaz de usuario (UI): se ajustaron elementos visuales y se verificó que todas las ayudas visuales aparecieran correctamente en el momento adecuado.

Estas mejoras fueron clave para elevar la calidad del juego, asegurando que tanto la narrativa como la jugabilidad ofrecieran una experiencia inmersiva y sin interferencias.

4.3.5. Pruebas

La fase de pruebas constituye una etapa esencial dentro del proceso de desarrollo, ya que permite verificar que el software cumple con los requisitos establecidos y que su comportamiento es fiable, estable y coherente con los objetivos del proyecto. Esta fase se ha desarrollado en dos periodos temporales bien diferenciados.

El primer periodo de pruebas tuvo lugar durante la segunda semana de enero de 2025, desarrollándose de manera paralela a la implementación de la introducción. En esta fase, las pruebas estuvieron orientadas principalmente a verificar el correcto funcionamiento de los elementos que conforman la secuencia introductoria, tales como la carga de escenas, la transición entre ellas, y las animaciones narrativas diseñadas para presentar el contexto de la misión espacial y a los personajes involucrados. Se prestó especial atención a la sincronización de efectos visuales y sonoros, garantizando que el jugador iniciara la experiencia con una inmersión narrativa adecuada.

El segundo periodo, de mayor duración y alcance, se extendió a lo largo de los meses de febrero y marzo, finalizando los primeros días de abril. Esta fase transcurrió de manera paralela al desarrollo de elementos fundamentales del juego, como los movimientos del personaje, el sistema de inventario, y las distintas misiones. En total, esta etapa abarcó aproximadamente dos meses y medio de trabajo, donde se llevaron a cabo distintos tipos de pruebas, cada una con un propósito específico.

A continuación, se describen los principales tipos de pruebas realizados:

- **Pruebas unitarias.** El objetivo de este tipo de pruebas es comprobar el funcionamiento de los componentes individuales de forma aislada. Constituyen la primera línea de validación y se aplican a módulos como las misiones individuales, el sistema de inventario, las interacciones iniciales del jugador, etc. el objetivo fue detectar errores locales antes de que estos pudieran llegar a afectar al funcionamiento del sistema completo.
- **Pruebas de integración.** Verifican que los diferentes módulos o unidades de software se integran correctamente entre sí y funcionan como un sistema completo. Estas pruebas pudieron llevarse a cabo una vez se contó con el desarrollo de todo el software al completo. Esto implicó recorrer el juego múltiples veces, tomando diferentes decisiones para verificar que todas las ramas narrativas funcionaban sin errores.
- **Pruebas de aceptación.** En este caso, y debido a que el desarrollo ha sido llevado a cabo únicamente por una desarrolladora, las pruebas de aceptación han sido realizadas internamente, con un enfoque orientado a la experiencia del usuario final. Dichas pruebas buscan validar que el juego sea intuitivo, jugable y accesible, y que cumpla las expectativas planteadas inicialmente. Se evalúan aspectos como la facilidad de uso de la interfaz, la claridad de los objetivos de cada misión, y la retroalimentación ofrecida al jugador en cada etapa.
- **Pruebas de regresión.** Estas pruebas son llevadas a cabo para garantizar que los cambios realizados en el software no hayan introducido errores o afectado negativamente a las funcionalidades existentes. Estas pruebas fueron realizadas tras las mejoras introducidas en la iteración 17. Se verificó que el comportamiento de las mecánicas principales del juego se mantuviera estable y sin fallos.

- **Pruebas de rendimiento.** Finalmente, se llevaron a cabo pruebas orientadas a evaluar el rendimiento técnico del juego, midiendo su eficiencia y capacidad de respuesta bajo diferentes cargas de trabajo. Se buscaron cuellos de botella, caídas de rendimiento o problemas en la carga de escenas. Estas pruebas se realizaron en dos momentos clave: la primera, durante la iteración de detección de errores, y la segunda, tras la implementación de mejoras finales en abril. Se utilizaron métricas como el consumo de memoria, la estabilidad de los FPS, y los tiempos de carga, ajustando aspectos técnicos cuando fue necesario.

En conjunto, la fase de pruebas ha sido determinante para consolidar la calidad del proyecto, asegurando que tanto las funcionalidades individuales como la experiencia global del juego respondan de forma correcta, coherente y satisfactoria para el jugador.

4.3.6. Documentación

Esta etapa, al igual que la fase de pruebas, se ha desarrollado en dos periodos temporales bien diferenciados, respondiendo a distintas necesidades del proyecto según su estado de avance. La fase de documentación ha sido fundamental no solo para dejar constancia del proceso de desarrollo, sino también para estructurar y presentar de forma clara tanto los fundamentos teóricos del videojuego como sus resultados prácticos.

El primer periodo de trabajo tuvo lugar durante todo el mes de diciembre de 2024. En esta etapa se elaboró el anteproyecto completo, incluyendo el planteamiento del problema, los objetivos del proyecto, el contexto narrativo, las decisiones de diseño iniciales y un estudio teórico previo que fundamentó la elección del género, las mecánicas de juego, y la orientación narrativa del proyecto. Esta documentación sirvió como hoja de ruta para el resto de las fases, facilitando una visión clara de la estructura general del videojuego.

El segundo periodo se desarrolló durante las dos primeras semanas del mes de abril de 2025, una vez finalizada la fase de implementación y superada la etapa de pruebas. En este momento, la documentación se centró en recoger y analizar todo el trabajo realizado durante el desarrollo. Se describió la metodología de trabajo seguida, la organización del proyecto, las técnicas utilizadas, las dificultades encontradas a lo largo del proceso, y las soluciones aplicadas. También se redactaron los resultados obtenidos, tanto a nivel técnico como narrativo, y se elaboraron las conclusiones finales, valorando el cumplimiento de los objetivos iniciales y reflexionando sobre las posibilidades de mejora o ampliación del proyecto en el futuro.

En cuanto a las herramientas empleadas durante esta fase, destaca principalmente el uso de Microsoft Word, que se utilizó como base para la redacción y organización de todos los documentos. A través de esta herramienta se estructuró la historia principal del videojuego, se detallaron las misiones, se definieron los personajes y sus roles, y se redactaron los posibles finales narrativos. Word también se utilizó para documentar los aspectos técnicos, como la arquitectura del sistema, los diagramas funcionales, los esquemas de interacción y la cronología de iteraciones implementadas.

Esta fase permitió consolidar todo el conocimiento adquirido a lo largo del proyecto, sirviendo tanto de soporte para la presentación final como de registro completo del desarrollo, clave para futuras mejoras, revisiones o ampliaciones del videojuego.

4.3.7. Gráfica complejidad-tiempo del proyecto

En esta sección se podrá visualizar de manera gráfica el esfuerzo realizado para sacar adelante este proyecto con una comparativa entre la complejidad y el tiempo invertido en cada una de las tareas que se han realizado, tal y como se puede apreciar en la Figura 25. Cabe destacar que este gráfico es completamente subjetivo, basado en las percepciones personales de la autora de este proyecto.

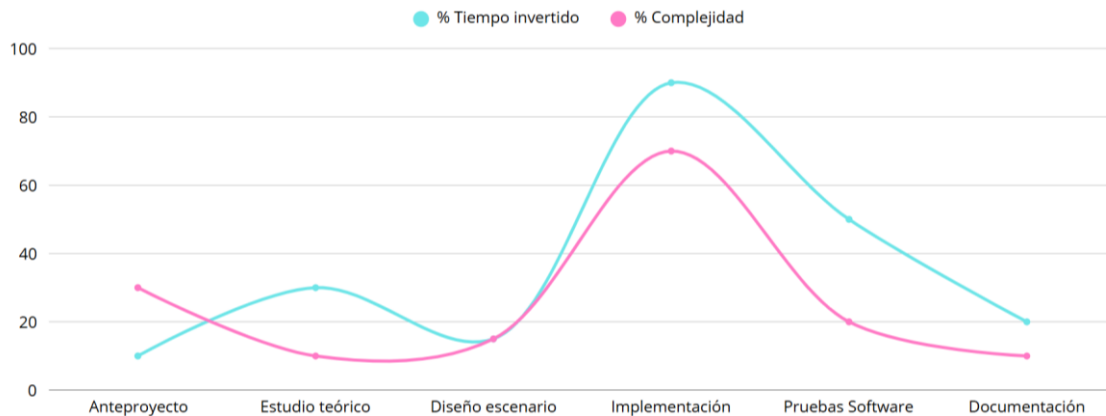


Figura 25. Gráfica complejidad-tiempo.

Para realizar una estimación de la complejidad del gráfico anterior, se ha adoptado una metodología sistemática que implica la descomposición de cada una de las fases del proyecto en tareas más pequeñas y manejables. A cada una de estas tareas se le ha asignado una calificación del 1 al 10 en función de una serie de criterios específicos diseñados para evaluar diversos aspectos de complejidad. Los criterios para la evaluación de cada una de las tareas incluyen:

1. Tamaño de la tarea. Se refiere a la magnitud y alcance de la tarea, considerando la cantidad de trabajo y los elementos involucrados.
2. Novedad. Evalúa el grado de familiaridad con la tarea, es decir, si se trata de un aspecto nuevo o conocido dentro del proyecto.
3. Tecnología utilizada. Considera la complejidad de las herramientas y tecnologías empleadas para llevar a cabo la tarea.
4. Complejidad de uso de la tecnología. Analiza la dificultad inherente en el uso de las tecnologías seleccionadas, incluyendo la curva de aprendizaje.
5. Interdependencias. Mide el grado de conexión de la tarea con otras tareas dentro del proyecto, lo cual puede influir en la dificultad de su realización debido a la necesidad de coordinar y gestionar estas interdependencias.

Cada uno de estos factores se ha evaluado individualmente y ha asignado una puntuación para cada tarea. Posteriormente, las puntuaciones de todas las tareas dentro de una fase se han sumado para obtener una puntuación total de la complejidad. La normalización de esta puntuación resulta en un porcentaje del 0% al 100%, proporcionando así una medida clara y cuantitativa de la complejidad relativa de cada fase del proyecto.

Por otro lado, para evaluar el tiempo, se ha considerado la cantidad de tiempo dedicada a cada fase en específico y, posteriormente, se ha realizado un análisis equilibrando el tiempo global invertido en la totalidad del proyecto con el tiempo específico asignado a cada fase individual. La normalización de esos tiempos ha proporcionado una medida clara y cuantitativa del tiempo relativo dedicado a cada fase en forma de porcentaje.

Este enfoque permite una evaluación detallada y objetiva tanto de la complejidad como del tiempo, facilitando la comparación entre las distintas fases del proyecto y permitiendo la visualización y comprensión del esfuerzo y tiempo requerido en cada una de las etapas.

Observando la gráfica, se puede analizar el esfuerzo realizado en cada una de las fases, haciendo un balance entre el tiempo invertido y el nivel de complejidad de esta.

- Anteproyecto. Representa una fase con un nivel de esfuerzo bajo, ya que, aunque conceptualmente fue un poco exigente, consiguió resolverse en un tiempo muy reducido. Esto indica que, si bien fue una etapa crucial para establecer las bases del trabajo la autora tenía claridad sobre el proyecto desde el principio, lo cual facilitó su rápida elaboración.
- Estudio teórico. En esta fase podemos ver un leve desequilibrio, ya que la cantidad de tiempo empleada mayor que la complejidad. Esto se debe a que fue una fase extensa y meticulosa, centrada en la búsqueda de referencias y análisis conceptual. Aunque no planteó dificultades técnicas ni creativas, sí que requirió constancia y esfuerzo para recopilar, seleccionar y sintetizar toda la información.
- Diseño del escenario. Por el contrario, esta fase sí que presenta un equilibrio perfecto entre tiempo empleado y nivel de complejidad de la tarea, lo que indica que el diseño del escenario del juego fue una tarea de esfuerzo estable y bien gestionado, gracias a una idea clara del estilo de los entornos desde el principio.
- Implementación. Esta fase es, sin duda, el núcleo del proyecto, ya que, no solo es la fase más extensa, sino también la más exigente. Aquí se concentran tareas críticas como la programación, integración de sistemas, diseño de mecánicas, lógica de juego, efectos visuales y sonoros, etc., lo que requiere de habilidades técnicas avanzadas, resolución de problemas constantes y un enfoque detallado. Representa el mayor nivel de esfuerzo de todo el proyecto.
- Pruebas. Aunque menos compleja que la implementación, esta fase supuso una alta inversión de tiempo. Esto se debe a la necesidad de realizar pruebas repetitivas, recorrer el juego en distintos escenarios, validar flujos narrativos, y detectar errores. La carga técnica no fue especialmente alta, pero sí implicó paciencia, meticulosidad y documentación sistemática de errores y correcciones.
- Documentación. Esta fase presenta una baja complejidad y un nivel de tiempo moderado. No supuso un gran desafío técnico o creativo, pero sí requirió de organización, redacción cuidadosa y revisión constante. La autora estructuró bien sus ideas durante el proceso, lo que facilitó la escritura de informes, análisis y conclusiones. Fue una fase ordenada y efectiva.

CAPÍTULO 5. RESULTADOS

A lo largo de este capítulo se detallarán los resultados obtenidos tras la etapa de implementación de este proyecto, analizando tanto el diseño del juego como la implementación técnica y la arquitectura final, así como el impacto que tuvieron la planificación y metodología utilizadas

5.1. DISEÑO DEL JUEGO

Durante este apartado se pretenden detallar diversos aspectos que se consideran esenciales en *Prometheus-9*, como son la jugabilidad, su historia o la interfaz, entre otros.

5.1.1. Descripción del juego

Sinopsis.

Prometheus-9 es una experiencia narrativa en primera persona que sitúa al jugador en una nave espacial averiada, tras un misterioso incidente que ha dejado al protagonista completamente solo a bordo. El jugador despierta sin información clara sobre lo sucedido, rodeado de sistemas fallidos y sin rastro alguno del resto de la tripulación.

A medida que avanza por los distintos módulos de la nave, el jugador descubre grabaciones de vídeo y pistas visuales que comienzan a revelar la verdad detrás del incidente. La narrativa se desvela de forma fragmentada y atmosférica, construyendo un relato de sabotaje, decisiones morales extremas y contacto alienígena.

El eje central de la historia gira en torno a Cooper, el especialista en comunicaciones de la nave, quien, impulsado por un deseo desesperado de contactar con una inteligencia alienígena detectada en las cercanías, sabotegó deliberadamente los sistemas de la nave. Esta decisión, tomada sin el consentimiento del resto de la tripulación, puso en peligro a todos los miembros de la misión y desató una cadena de eventos que el jugador debe tratar de revertir mientras reconstruye los hechos.

El desarrollo del juego se basa en la exploración, la resolución de misiones interactivas y el descubrimiento progresivo de una historia que pone al jugador frente a cuestiones éticas, existenciales y emocionales.

Principales características.

Las principales características de *Prometheus-9* son las que se detallan a continuación.

- Narrativa inmersiva y ramificada. La historia se descubre mediante la exploración activa del entorno, grabaciones de vídeo y decisiones del jugador. Cada elemento recogido aporta una nueva capa de información al misterio principal.

- Mecánicas centradas en la resolución de misiones. El jugador debe reparar sistemas, recopilar pruebas y superar minijuegos que simulan tareas técnicas dentro de la nave. Estas mecánicas están integradas de forma natural en la narrativa con el objetivo de mantener la inmersión del jugador en todo momento.
- Ambiente sonoro detallado y envolvente. El juego apuesta por una atmósfera silenciosa e inquietante, donde cada sonido ambiental o alarma influye en la experiencia emocional del jugador.
- Tono maduro y reflexivo. Lejos de ofrecer acción frenética, *Prometheus-9* apuesta por un ritmo pausado y contenido emocional intenso, fomentando la toma de decisiones con peso narrativo.
- Sistema de grabaciones. El progreso del jugador desbloquea nuevas grabaciones que reconstruyen lo sucedido en la nave, revelando traiciones, sacrificios y secretos cruciales para comprender la historia completa.
- Finales múltiples. Dependiendo de las decisiones tomadas por el jugador, este podrá experimentar diferentes desenlaces para el juego.

Género.

Prometheus-9 se enmarca dentro del género de los videojuegos narrativos de ciencia ficción, con un enfoque particular en la exploración. Combina elementos de aventura en primera persona, thriller psicológico, y suspense espacial, acercándose a la narrativa inmersiva de títulos como *SOMA* (2015) o *Firewatch* (2016), y a la estética atmosférica de películas como *Alien* (1979) e *Interstellar* (2014).

No es un juego de acción tradicional, ya que su propuesta gira en torno a la inmersión, la ambientación y el peso emocional de la historia, donde el jugador asume un rol activo en el descubrimiento de los hechos y la reconstrucción del relato. El diseño evita los combates y se centra en misiones interactivas, recolección de información y elecciones significativas que influyen en el desarrollo y final del juego.

5.1.2. Jugabilidad centrada en la narrativa

La jugabilidad en *Prometheus-9* ha sido diseñada con un enfoque completamente narrativo, dado que cada interacción, movimiento y misión está pensada como parte del desarrollo argumental. El objetivo principal es que el jugador sienta que cada acción que realiza tiene un propósito narrativo claro y aporta sentido al progreso de la historia.

A continuación, se describirán aquellos aspectos que caracterizan la jugabilidad, como son el personaje, las cámaras y los controles de juego.

Personaje.

El jugador asume el rol de Austin, el único superviviente a bordo de la nave espacial *Prometheus-9*. Este personaje funciona como un intermediario entre la historia y el usuario. Su perspectiva, emociones y decisiones están definidas por el entorno y las pistas que va descubriendo, pero al mismo tiempo, deja espacio para que el jugador proyecte sus propias interpretaciones sobre lo que está ocurriendo. Puede observarse una imagen del personaje principal en la Figura 26.



Figura 26. Personaje principal.

Cámara.

Con respecto a la cámara que vamos a encontrar en *Prometheus-9*, cabe mencionar que la perspectiva predominante es la primera persona, lo que favorece una mayor inmersión en el entorno. Esta elección permite al jugador vivir la experiencia directamente desde los ojos del protagonista, incrementando la tensión y el aislamiento característicos del juego.

No obstante, en determinados momentos (al resolver misiones) se activa un sistema de cámaras virtuales, las cuales ofrecen una vista externa y controlada, que permite al jugador observar la acción desde un mejor ángulo. Durante estos segmentos, la cámara en primera persona queda desactivada temporalmente, cediendo el protagonismo a las cámaras secundarias hasta que la misión finaliza. Esta técnica enriquece la narrativa visual y aporta variedad al desarrollo del juego, sin romper la inmersión general.

En la Figura 27 puede verse como se encuentra colocada la cámara sobre el personaje principal. Por otro lado, en la Figura 28, se observa lo que estaría viendo el personaje.

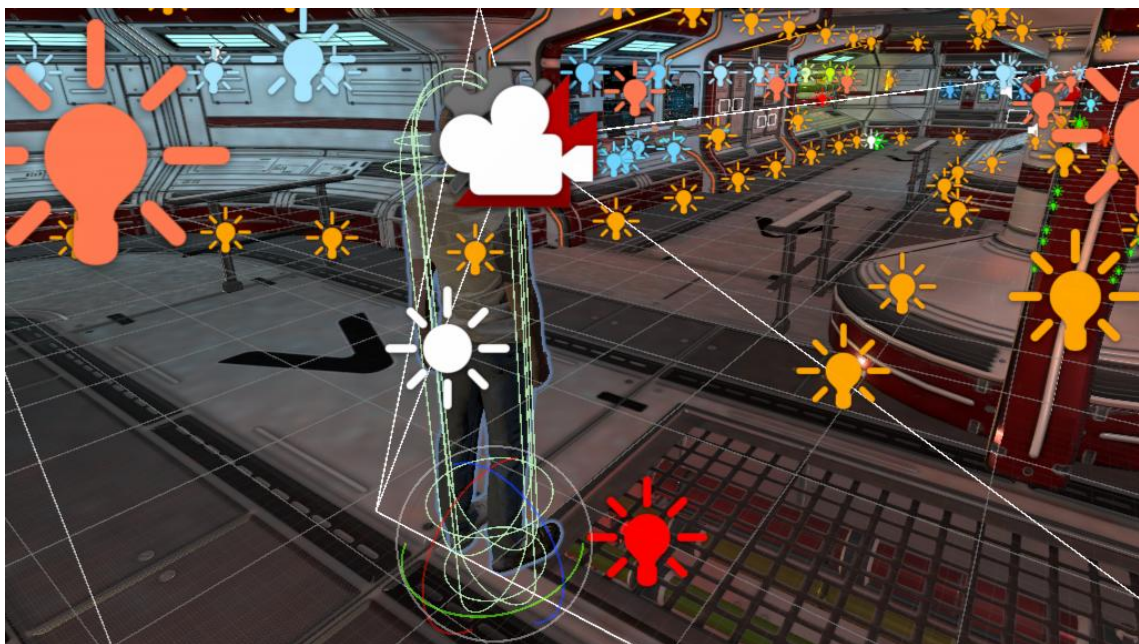


Figura 27. Posición de la cámara sobre el personaje principal.

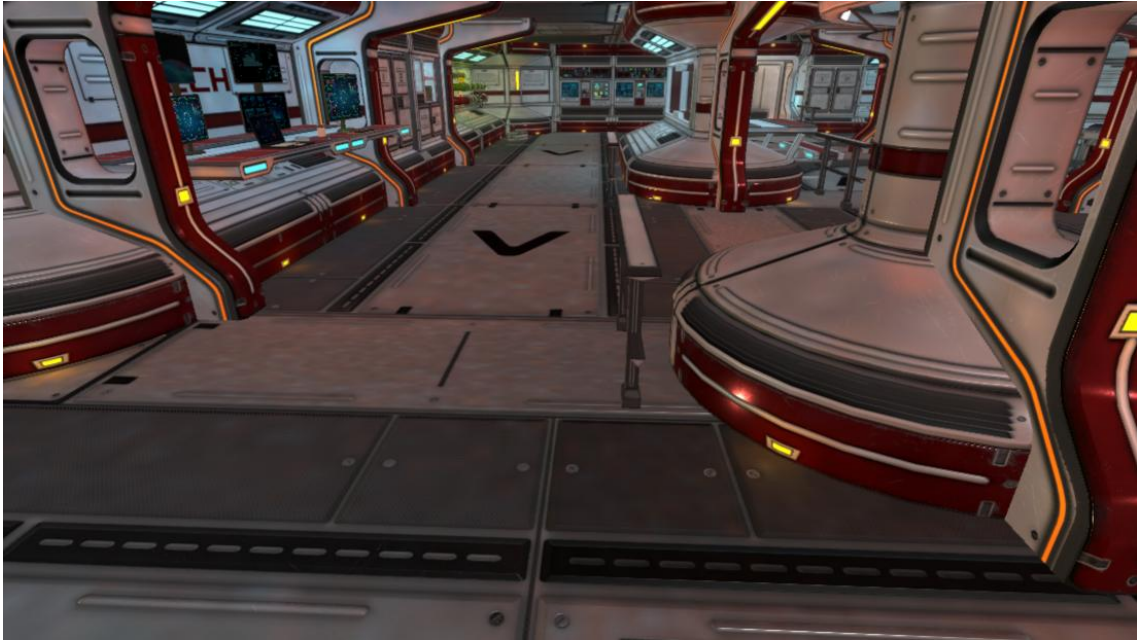


Figura 28. Visión del personaje principal.

Control.

El personaje principal es controlado, en su mayoría, por el usuario, lo que le otorga una sensación constante de presencia y responsabilidad. Puede desplazarse libremente por los entornos de la nave, interactuar con objetos, y gestionar recursos clave. Las acciones que puede realizar el personaje son las que se describen a continuación:

- **Giro de cámara.** Con el movimiento del ratón, el usuario podrá girar la cámara en los ejes x e y, simulando el movimiento natural de la cabeza.
- **Caminar.** Haciendo uso de las cuatro flechas del teclado, el jugador podrá moverse libremente por la escena.
 - ↑: Permite al jugador avanzar hacia delante.
 - ↓: Permite al jugador moverse hacia atrás.
 - →: Permite al jugador desplazarse hacia la derecha.
 - ←: Permite al jugador desplazarse hacia la izquierda.
- **Saltar.** Gracias a la tecla de espacio, el jugador podrá efectuar un salto en aquellos momentos en los que así lo considere.

Sin embargo, existen ciertos momentos en los que el control del usuario se suspende temporalmente, principalmente durante secuencias cinemáticas o transiciones narrativas importantes. En estas escenas, el personaje ejecuta acciones predefinidas, lo que permite mantener el ritmo de la narrativa sin interrupciones.

Progresión.

La progresión en *Prometheus-9* se construye en torno a una estructura no lineal guiada por misiones. Cada misión desbloquea nuevas áreas, objetos, grabaciones o fragmentos de la historia que el jugador puede interpretar para reconstruir lo sucedido en la nave.

El jugador avanza no solo superando retos técnicos, sino también interpretando y analizando pistas del entorno y conectando sucesos aparentemente inconexos. Las decisiones que tome influirán en el acceso a ciertas zonas y en el desenlace final de la historia, lo que añade un componente de rejugarabilidad y narrativa emergente.

Retos y dificultades.

Durante la experiencia de juego, el jugador se enfrenta a una variedad de retos que se clasifican en tres niveles de dificultad. Cada nivel está asociado a un tipo concreto de acciones dentro del juego, con el objetivo de ofrecer una experiencia equilibrada, que combine exploración relajada con desafíos que exijan atención y habilidad.

- **Dificultad baja.** Corresponde a acciones básicas como la recolección de objetos, donde basta con acercarse a un ítem del entorno y pulsar la tecla R para añadirlo al inventario. Estas acciones no requieren habilidades especiales ni tiempo límite, permitiendo al jugador familiarizarse con el entorno.
- **Dificultad media.** Representa la mayoría de las misiones interactivas, como activar paneles, resolver mecanismos o imitar secuencias. Suelen implicar cierta planificación, observación del entorno y reacción a tiempo, pero permiten varios intentos sin penalizaciones severas.
- **Dificultad alta.** Se reserva para misiones específicas con mecánicas complejas o con presión temporal, como tareas contrarreloj, misiones de sigilo o aquellas con penalizaciones por error acumulado. Estas situaciones elevan la tensión y requieren atención constante, reflejos rápidos y una buena comprensión del objetivo narrativo.

Esta clasificación por niveles de dificultad busca fomentar el ritmo de juego, mantener al jugador motivado y ofrecer una curva de aprendizaje progresiva. Al vincular el grado de dificultad con el tipo de acción, se logra una experiencia accesible pero desafiante, que refuerza la inmersión y la coherencia narrativa.

5.1.3. Historia

En este apartado se detallará la historia que envuelve este juego, además de cada uno de los personajes implicados en la misma.

Personajes.

El objetivo de este apartado es presentar a los personajes que conforman la historia.

- **Madison Vega – Comandante de la Misión.** Es una líder nata, condecorada por su experiencia en operaciones complejas y situaciones de alto riesgo. Su combinación de disciplina y empatía la convierten en el pilar emocional y estratégico del equipo. Es la responsable de guiar a la tripulación ante lo desconocido, y su figura aparece en varias grabaciones claves que revelan los conflictos internos surgidos durante la misión.
- **Liam Carter – Ingeniero de Sistemas.** Es el encargado de mantener el funcionamiento de la nave. Es una persona de carácter brillante y pragmático, con

una capacidad sobresaliente para solucionar fallos bajo presión. Su ausencia al inicio del juego genera tensión, ya que varios subsistemas de la nave están dañados y su experiencia se considera crucial.

- **Nicole Bennett – Bióloga Espacial.** Es la responsable de estudiar posibles formas de vida extraterrestre. Su pasión por la ciencia contrasta con la creciente amenaza biológica que se revela en el juego. Tiene un rol esencial en la narrativa relacionada con las muestras biológicas descubiertas.
- **Austin Prescott – Astrónomo y Explorador.** Es una persona soñadora y reflexiva, en cargada de liderar la interpretación de señales cósmicas y la cartografía del sistema Helion-3. Su presencia resulta clave para comprender la magnitud del descubrimiento alienígena.
- **Cooper Hartman – Especialista en IA y Comunicaciones.** Es el puente entre la tripulación y el resto del universo. Su experiencia en inteligencia artificial y comunicaciones interestelares será vital, no solo para mantener el contacto con la Tierra, sino también para interpretar cualquier intento de comunicación extraterrestre.

Explicación de la historia.

La explicación de la historia se fundamenta en la Figura 29, con el objetivo de proporcionar una visión global y una mejor comprensión de la misma.

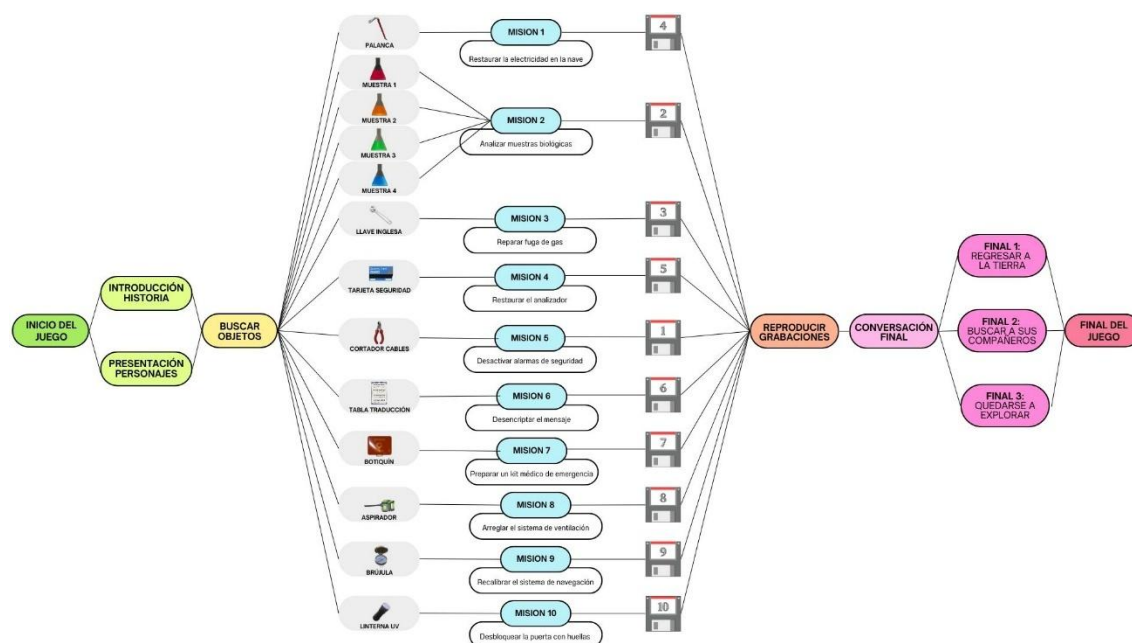


Figura 29. Diagrama explicativo de la historia del juego.

La estructura narrativa inicial del juego se desarrolla a través de una serie de secuencias que utilizan un sistema de cámaras virtuales para representar una transmisión oficial de la Agencia Espacial Global, en la que se presenta la misión al sistema estelar Helion-3 y se introducen los cinco miembros de la tripulación. Tras esto, se encadena una transición cinematográfica que muestra el despegue de la nave y su progresivo alejamiento de la Tierra. Posteriormente, se activa una escena desde el interior de la nave donde comienzan

a producirse pequeñas vibraciones que culminan en una cinemática en la que el personaje principal pierde el conocimiento a causa de las mismas.

Una vez finalizada dicha introducción, el control se transfiere al jugador en una vista en primera persona, estableciendo el inicio de la experiencia interactiva. El entorno se encuentra a oscuras, con sistemas críticos inactivos, lo que obliga al jugador a explorar el escenario e interactuar con objetos claves mediante mecánicas básicas como el uso de la tecla R para añadir ítems al inventario.

Los objetos coleccionables y sus funciones mecánicas son los siguientes:

1. **Palanca**

Permite completar la Misión 1 (restaurar la electricidad en la nave). Se utiliza para forzar una puerta metálica que contiene un cuadro eléctrico en su interior, con una palanca principal y cinco interruptores. El jugador debe utilizar la tecla X para accionar la palanca y las teclas 1 a 5 para operar los interruptores, con el objetivo de apagar todo el sistema y volver a encenderlo siguiendo una secuencia lógica. El éxito en esta tarea reactiva el sistema eléctrico de la nave y habilita nuevas zonas de exploración. También se desbloquea el código de acceso a la Grabación 4, que puede observarse en la Figura 30.



Figura 30. Código de acceso de la misión 1.

2. **Muestras biológicas (1 a 4)**

Están directamente relacionadas con la resolución de la Misión 2 (análisis de muestras biológicas). El jugador debe colocar cada muestra en el Sistema de Identificación Genética pulsando la tecla X, lo que activa una barra de progreso automática para cada uno de los análisis. Esta tarea debe completarse para las cuatro muestras antes de que expire un temporizador de 5 minutos. Al finalizar, se muestra una secuencia numérica en pantalla, de la cual cuatro dígitos se encuentran resaltados en color naranja, y son los que corresponden al código de desbloqueo de la Grabación 2, que se muestra en la Figura 31.



Figura 31. Código de acceso de la misión 2.

3. Llave inglesa

Utilizada en la Misión 3 (reparación de fuga de gas). Al interactuar con la tubería pulsando la tecla X, el jugador accederá a un minijuego de precisión visual basado en una barra de colores (rojo – amarillo – verde – amarillo – rojo). Una flecha se desplaza constantemente de un lado a otro de la barra y el jugador debe presionar Espacio para detenerla. El grado de precisión determina cuántas veces se debe repetir la acción para ajustar la válvula de salida del gas. Si el jugador consigue colocar la flecha en el segmento verde, se aplicará una fuerza de ajuste sobre la válvula de 1, si la flecha queda en el segmento amarillo, la fuerza será de 0.5 y si la flecha queda en el rojo, no se aplicará fuerza de ajuste alguna. Una vez selladas las tres válvulas de fuga, se desbloquea el código de la Grabación 3, Figura 32.



Figura 32. Código de acceso de la misión 3.

4. Tarjeta de seguridad

Permite resolver la Misión 4 (restauración del estado del analizador). Tras escanear la tarjeta en el dispositivo correspondiente, se activa un proceso de reinicio asociado a una mecánica basada en la memoria y la repetición de secuencias alfabéticas. De esta manera, el sistema muestra en pantalla una cadena de letras durante un breve periodo de tiempo, que el jugador debe recordar e introducir correctamente usando el teclado. Al superar esta prueba, se muestra un mensaje que contiene implícito el código de acceso a la Grabación 5, tal y como se puede observar en la Figura 33.

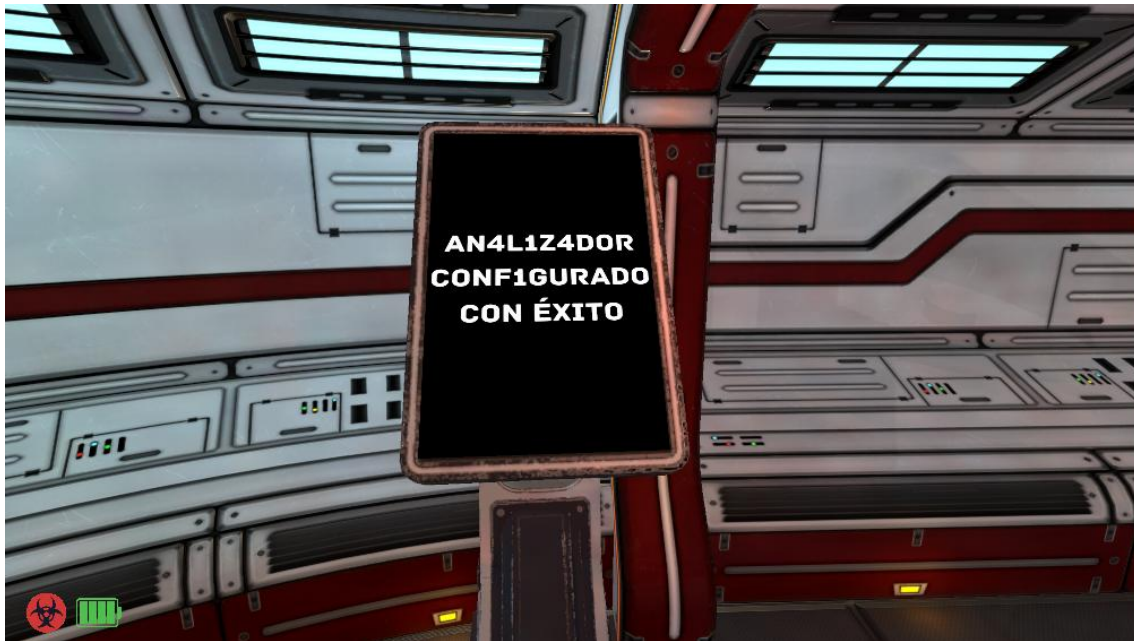


Figura 33. Código de acceso de la misión 4.

8. Cortador de cables

Este objeto será de gran utilidad para resolver la Misión 5 (desactivación de las alarmas de seguridad). Para ello, el jugador debe atravesar una sala equipada con un conjunto de sensores de movimiento, que la recorren continuamente de un lado a otro, evitando el contacto directo con ellos. La colisión con algunos de estos sensores reinicia el progreso y penaliza el estado de salud del jugador. Una vez recolectado el cortador de cables, debe desplazarse hasta el panel donde se ubican los cables de alimentación de las alarmas e interactuar con el mismo pulsando la tecla X para ello. A partir de este momento, la misión del jugador será cortar los 4 cables que mantienen activada la alarma, para lo que debe pulsar repetidamente la tecla Espacio hasta completar una barra de progreso asociada a cada cable. Una vez completado el proceso, las alarmas se desactivan y se revela en un teclado numérico el código de desbloqueo de la Grabación 1, conforme a la información visualizada en la Figura 34.

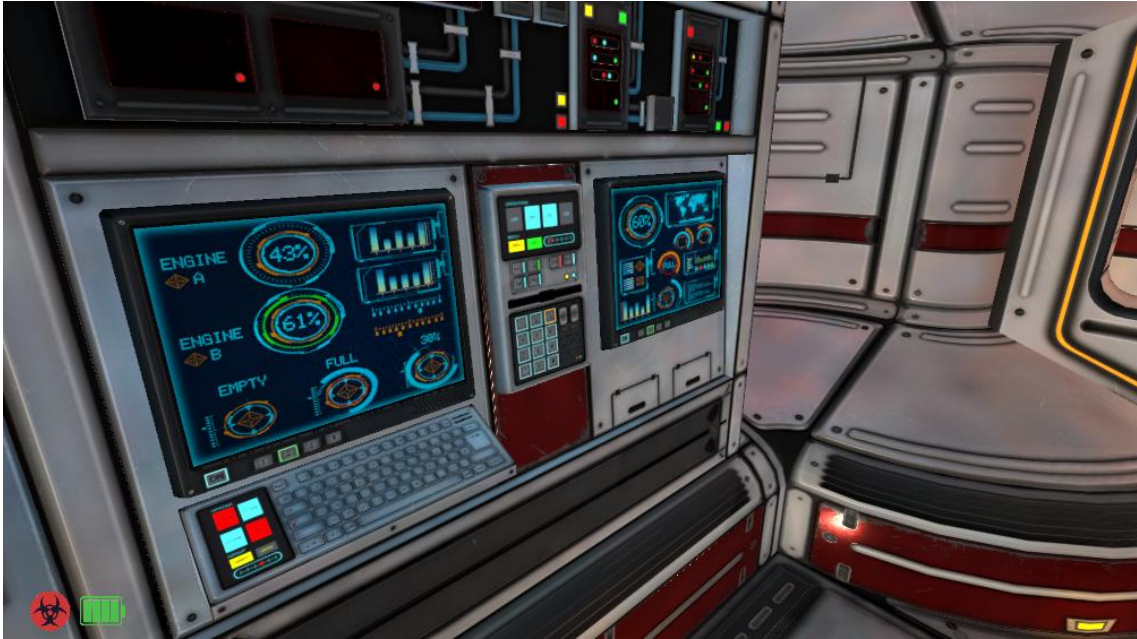


Figura 34. Código de acceso de la misión 5.

9. Tabla de traducción

La utilización de este objeto, permitirá al jugador la resolución de la Misión 6 (descifrado del mensaje alienígena). El jugador tiene acceso a un libro que contiene símbolos alienígenas escritos en una página, por lo que deberá utilizar la tabla de traducción, para asociar cada símbolo con letras o números. La tarea consiste en transcribir línea a línea la traducción correcta en un campo de texto habilitado para ello. Una vez completado, el jugador pulsa Enter para confirmar la solución y, si la decodificación es correcta, habrá traducido el código para desbloquear la Grabación 6, como se ilustra en la Figura 35.



Figura 35. Código de acceso de la misión 6.

10. Botiquín

Está directamente relacionado con la resolución de la Misión 7 (preparación de un kit médico de emergencia). Para ello, el jugador deberá colocar el botiquín sobre una caja mediante la pulsación de la tecla X. En ese mismo momento, queda activado un temporizador de 3 minutos. Durante ese tiempo, el jugador debe explorar el almacén y localizar 8 ítems médicos repartidos por el entorno. Cada vez que detecta un objeto, lo recoge utilizando la tecla R y este se coloca automáticamente en el botiquín. Al completar el conjunto, el jugador puede visualizar una el código de desbloqueo de la Grabación 7 en una etiqueta médica, de acuerdo a lo representado en la Figura 36.



Figura 36. Código de acceso de la misión 7.

11. Aspirador

Permite completar la Misión 8 (limpieza del sistema de ventilación) mediante su recolección en un armario de la zona de limpieza de la nave. Una vez iniciado el proceso de limpieza (pulsando la tecla X sobre la rejilla de ventilación), el jugador tendrá control total sobre el aspirador con las teclas de dirección. La misión consiste en eliminar todos los elementos de suciedad, que desaparecerán tras dos segundos de entrar en contacto con el aspirador. Esta tarea debe completarse antes de que se agote un temporizador de 4 minutos. Una vez realizada, se desbloquea un panel de estado con el código de acceso a la Grabación 8, como se aprecia en la Figura 37.



Figura 37. Código de acceso de la misión 8.

12. Brújula

El jugador utilizará este objeto para completar la Misión 9 (recalibración del sistema de navegación). Al interactuar con el panel de calibración principal de la nave, accederá a un 8-puzzle digital, el cual debe resolver en un número limitado de movimientos utilizando las teclas de dirección para ello. Las piezas deben reorganizarse hasta formar la imagen original del panel de navegación. El número de movimientos óptimos es calculado internamente, y superar ese límite penaliza al jugador. Al resolver correctamente el puzzle, el panel muestra un código que permite desbloquear la Grabación 9, según se muestra en la Figura 38.



Figura 38. Código de acceso de la misión 9.

13. Linterna UV

La linterna de luz ultravioleta va a ser de gran utilidad para la resolución de la Misión 10 (desbloqueo de una puerta mediante la identificación y reconstrucción de huellas dactilares). Con la linterna activada, el jugador debe inspeccionar una puerta de seguridad en busca de tres huellas ocultas, que aparecerán al contacto con dicha luz. La interacción del jugador con dichas huellas activará un minijuego de reconstrucción, donde la huella aparecerá dividida en 4 fragmentos desordenados, y el jugador debe reorganizarlos utilizando las teclas \leftarrow \rightarrow , y confirmando su elección con Enter. Al completar correctamente el patrón de las tres huellas, se muestra en la pantalla de la puerta el código para desbloquear la Grabación 10, cerrando así la narrativa principal del juego.



Figura 39. Código de acceso de la misión 10.

Como ya se ha comentado en múltiples ocasiones, a lo largo del juego, el jugador desbloquea progresivamente 10 grabaciones de seguridad que sirven como eje narrativo. Cada una aporta información esencial sobre el origen del conflicto, el sabotaje de la nave y el contacto con una inteligencia desconocida. Estas grabaciones, están ordenadas cronológicamente y cubren desde 24 horas antes del incidente hasta 3 horas después.

A continuación, se detallará el contenido de cada una de ellas con el objetivo de una mejor comprensión de la historia.

Grabación 1: Inicio del Plan (24 horas antes)

Esta grabación tiene lugar en la zona de observación de la nave, y en ella se puede observar cómo Cooper contempla el espacio mientras manipula un dispositivo de transmisión. Pronuncia una frase reveladora sobre el costo de su objetivo, de manera que esta escena introduce su motivación personal y deja entrever su intención de contactar con entidades externas a espaldas del resto de la tripulación.

Grabación 2: Robo Crítico (20 horas antes)

Esta grabación ocurre en el laboratorio, donde se observa a Nicole analizando una muestra biológica de origen desconocido. Cooper, con actitud inquisitiva, se interesa por su potencial de interacción electromagnética. En un descuido, roba una de las muestras. Esta grabación revela el primer acto de sabotaje y sugiere que Cooper está dispuesto a comprometer la seguridad de la misión.

Grabación 3: Manipulación del Sistema (18 horas antes)

Por su parte, esta grabación tiene lugar en la sala de control de comunicaciones, cuando Liam detecta una anomalía en las transmisiones. Cooper, encubriendo sus acciones, finge investigar mientras redirige la antena para establecer una transmisión clandestina. Reafirma su intención de actuar en solitario y engañar a la tripulación.

Grabación 4: Sabotaje de la IA (14 horas antes)

La cuarta grabación acontece en el departamento de tecnología de la nave. En ella, Cooper conecta un dispositivo externo a la IA de la nave, forzando su inestabilidad. A pesar de las advertencias del sistema, introduce comandos para desactivarlas. Esta grabación muestra su voluntad de inutilizar los sistemas de control internos con el fin de asegurar que su plan no sea detenido.

Grabación 5: Detectando el Problema (12 horas antes)

La zona de tecnología es nuevamente el escenario de esta grabación. Aquí puede observarse cómo Madison, Austin y Nicole analizan señales electromagnéticas anómalas. Las discusiones científicas entre ellos comienzan a evidenciar tensiones y sospechas. Cooper escucha sin intervenir, lo que incrementa la sensación de amenaza.

Grabación 6: El Contacto (Momento exacto del suceso)

Esta grabación se desarrolla en la zona de despresurización. En ella se ve claramente a Cooper colocando la muestra robada en un dispositivo de transmisión de información, mientras lo activa. Una onda energética emana en el exterior, revelando la aparición de una nave alienígena. Las alarmas se disparan. Esta grabación representa el punto de inflexión: se confirma el contacto alienígena y comienza la desestabilización general.

Grabación 7: Caos Inicial (10 minutos después)

La séptima grabación transcurre en la sala de control de comunicaciones. Madison y Liam trabajan desesperadamente para restaurar las comunicaciones. Los sistemas están gravemente comprometidos y Madison deduce que ha habido un sabotaje interno. Se transmite la desesperación creciente de la tripulación ante la pérdida de control.

Grabación 8: Confrontación Crítica (30 minutos después)

La grabación número 8 tiene lugar en una sala especial de la nave habilitada para la transmisión de señales. En ella puede observarse a Nicole intentando analizar el origen de las señales recibidas, cuando Cooper irrumpe. Nicole intenta detenerle advirtiéndole de que podría poner en peligro la misión, pero, ante su negativa a detenerse, él la incapacita liberando un gas. Esta escena confirma que Cooper está dispuesto a utilizar la fuerza contra sus propios compañeros para completar su plan.

Grabación 9: Llegada Alienígena (1 hora después)

En esta grabación, que tiene lugar en los pasillos principales de la nave, las cámaras captan cómo diversos seres alienígenas se desplazan rápidamente por los corredores, causando interferencias eléctricas. Se representa por primera vez la presencia física de los entes externos y su impacto directo en la nave.

Grabación 10: Aterrizaje Forzoso (3 horas después)

Esta grabación se produce nuevamente en la sala de observación. En ella puede verse cómo la nave, parcialmente inutilizada, entra en un protocolo de aterrizaje automático. Austin cae al suelo completamente desorientado. Esta grabación cierra la narrativa visualizando el colapso de la misión y la incertidumbre del destino de la tripulación.

Tras completar las diez misiones y recolectar todas las grabaciones, el jugador accede finalmente a la sala de comunicaciones, donde encontrará un dispositivo en el que reproducir dichas grabaciones. Además, es aquí donde se activa la conversación final, un intercambio de mensajes con la estación de control terrestre, que culmina con una decisión definitiva sobre el destino del protagonista y el desenlace de la historia.

La conversación se desarrolla mediante un sistema de diálogo interactivo ramificado, en el que el jugador puede elegir entre diferentes respuestas en cada turno. Estas opciones permiten al jugador expresar sus emociones, reconstruir lo vivido y dar forma a su versión de los hechos. A medida que responde, se activa una secuencia de preguntas por parte del control terrestre, con líneas que se adaptan según las respuestas elegidas. Las distintas líneas de diálogo revelan elementos clave como la traición de Cooper, la naturaleza del contacto alienígena o la desaparición de la tripulación.

La Figura 40 muestra en detalle dicha conversación.

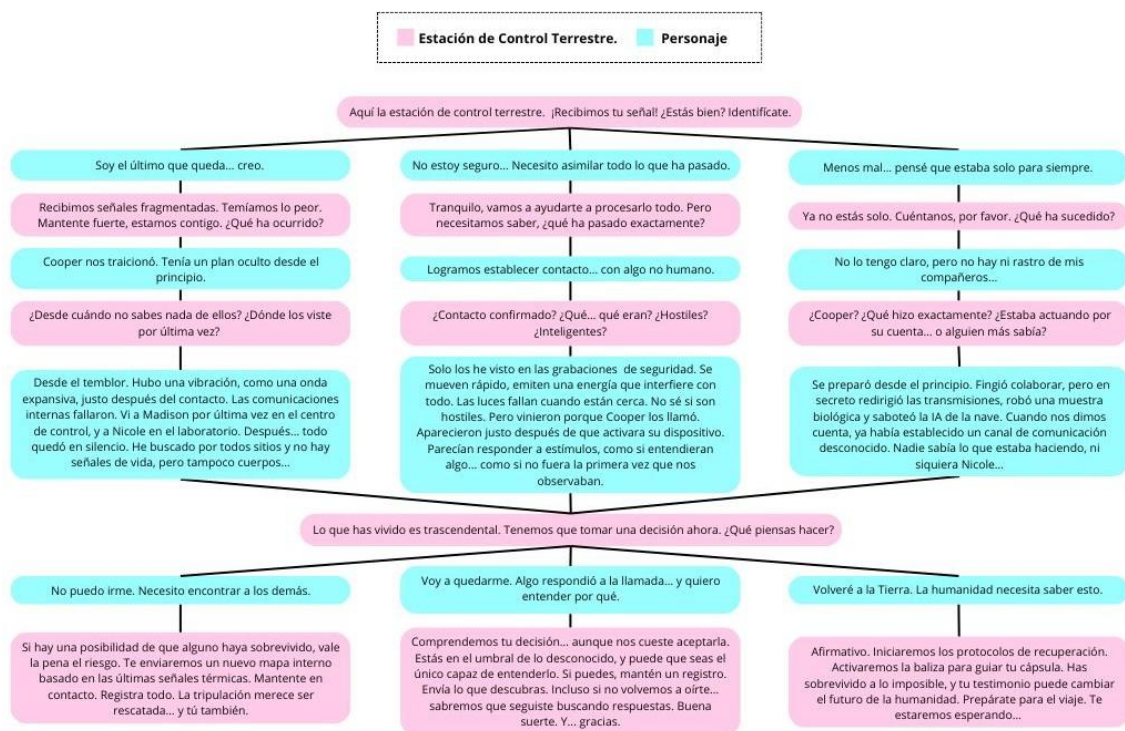


Figura 40. Ramificaciones de la conversación final.

Finalmente, la estación de control solicita al jugador que tome una decisión. En este punto, se presentan tres finales distintos, cada uno asociado a una filosofía de acción. Independientemente de la decisión tomada por el jugador, no existe un final definido como “correcto”, todos se presentan como legítimos según la experiencia, interpretación y valores del jugador. Así, el cierre del juego se convierte en un espejo de las decisiones éticas y emocionales acumuladas durante la travesía.

Además, el sistema de decisiones permite rejugabilidad: los jugadores pueden iniciar una nueva partida, tomar otras decisiones y reconstruir el relato desde nuevas perspectivas, reinterpretando las grabaciones o descubriendo matices previamente ignorados.

5.1.4. Interfaz

En *Prometheus-9*, la interfaz de usuario ha sido diseñada bajo un enfoque minimalista e inmersivo, donde todos los elementos visibles tienen una razón narrativa y funcional clara. La intención ha sido evitar distracciones innecesarias, integrando la mayor parte de la información de estado en el entorno y las sensaciones del propio jugador, en lugar de sobrecargar la pantalla con indicadores tradicionales.

Los únicos elementos permanentes visibles en la interfaz son dos iconos discretos ubicados en la esquina inferior izquierda de la pantalla: el medidor de oxígeno y el de energía. Ambos utilizan un sistema de codificación por colores (verde, amarillo, naranja y rojo) para reflejar el estado actual del jugador, permitiendo una rápida lectura sin necesidad de texto o números. Este sistema no solo facilita la comprensión, sino que refuerza la tensión y la urgencia conforme los niveles descienden hacia zonas críticas.

Más allá de estos iconos, el resto de los sistemas de estado del jugador están integrados en la experiencia de forma sensorial:

- **Vida.** Su deterioro se refleja mediante un oscurecimiento progresivo de la pantalla y una reducción en la claridad visual, lo que simula el debilitamiento del personaje. No hay barra visible, lo que obliga al jugador a estar atento a señales visuales sutiles.
- **Energía.** Se manifiesta principalmente a través de una reducción de la velocidad de movimiento, especialmente al correr, y un cambio en el sonido de la respiración del personaje, que se vuelve más agitado a medida que se fatiga.
- **Oxígeno.** Su disminución afecta directamente a la percepción visual. La pantalla adopta un efecto de visión borrosa y se escucha una alerta sonora automatizada (una voz robótica que repite “*low oxygen*”), reforzando la sensación de emergencia y desorientación. Esta alerta simula un sistema interno de aviso de la nave, integrando la narrativa con la interfaz de forma coherente e inmersiva.

Este enfoque de diseño consigue que la interfaz no actúe como una barrera entre jugador y juego, sino como una extensión natural de la narrativa. La información está presente, pero es orgánica. Con esta estrategia, *Prometheus-9* fomenta una conexión más profunda entre el jugador y el protagonista, al trasladar el estado físico del personaje al propio campo sensorial del usuario, lo que refuerza el tono introspectivo del juego, alineándose con su propuesta narrativa centrada en la supervivencia, la soledad y la tensión creciente.

5.2. ARQUITECTURA FINAL DEL JUEGO

En este apartado se detallará la arquitectura del juego, comenzando por lo más general y avanzando gradualmente hasta las partes más específicas. Es importante resaltar que todo el código del juego está escrito en C# ya que es el lenguaje de programación en el que se escriben los scripts de Unity.

La arquitectura general del juego está conformada por un total de 28 scripts relacionados entre sí, que se encuentran categorizados de manera cromática, tal como puede apreciarse en la Figura 41.

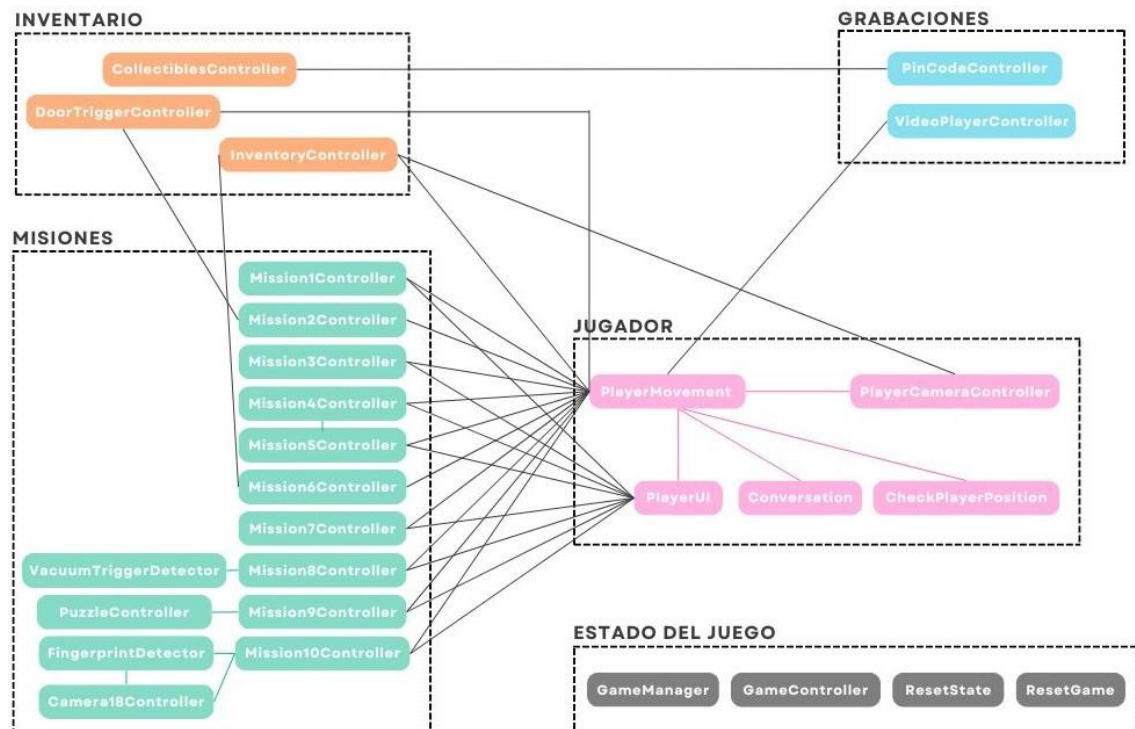


Figura 41. Arquitectura general del juego por scripts.

5.2.1. Inventario

- ***CollectiblesController***. Encargado de la recolección de objetos interactivos (herramientas y grabaciones), este script detecta cuando el jugador pulsa R cerca de un elemento y lo añade al inventario si está desbloqueado. En el caso de grabaciones protegidas, activa una interfaz de código vinculada a *PinCodeController*. Cada objeto o grabación recolectada actualiza el *GameManager*, garantizando persistencia. Colabora con *InventoryController* para reflejar visualmente los elementos adquiridos.
- ***DoorTriggerController***. Este script gestiona la interacción del jugador con puertas animadas, asociadas a eventos narrativos o misiones clave. Permite abrir y cerrar puertas mediante la tecla E y permite recoger objetos usando R si se cumplen unas condiciones concretas (que la puerta donde se encuentra el objeto esté abierta). Además, guarda y carga el estado de puertas y objetos mediante el

GameManager, asegurando persistencia entre sesiones. Está vinculado a animaciones, control de cámara y restricciones de movimiento temporal.

- *InventoryController*. Controlador principal del sistema de inventario, que permite al jugador visualizar y gestionar objetos y grabaciones mediante la tecla I (abrir/cerrar). Integra navegación en cuadrícula por secciones (objetos y grabaciones) usando las teclas de dirección. Los elementos desbloqueados se muestran con su Sprite y nombre, y al seleccionarlos, se reproduce una animación del jugador (sujetarlos con la mano derecha). Dispone de métodos para cargar progreso, agregar ítems y sincronizar el estado. Todo el inventario se mantiene sincronizado con el GameManager.

5.2.2. Grabaciones

- *PinCodeController*. Este script permite la introducción de códigos numéricos para desbloquear cajas de seguridad que contienen las grabaciones. Captura la entrada por teclado de dígitos, Backspace y Enter, mostrando el código en tiempo real. Al confirmar con Enter, valida el código contra un array predefinido asociado a cada caja. Si es correcto, activa la animación de apertura, registra la grabación como desbloqueada y actualiza el contador de códigos correctos. Además, guarda el estado del juego a través de GameManager y permite reiniciar la entrada en cualquier momento.
- *VideoPlayerController*. Este script controla la reproducción de grabaciones de seguridad cuando el jugador interactúa con objetos del entorno. Gestiona la transición entre cámaras virtuales para cambiar entre vista libre, animaciones y reproducción del vídeo. Al pulsar X sobre una grabación, se bloquea el movimiento del jugador y se inicia el vídeo correspondiente. Al finalizar el vídeo, se reactiva el control del jugador y se registra el evento en recordsPlayed, actualizando el progreso en GameManager.

5.2.3. Misiones

- *Mission1Controller*. Gestiona la lógica de la primera misión del juego, centrada en la activación manual del sistema eléctrico. Requiere que el jugador apague y encienda cinco interruptores en orden, usando el teclado, junto a una palanca principal. Controla el movimiento del jugador, alterna cámaras con Cinemachine, y proporciona retroalimentación visual y auditiva al finalizar. Al completarse, se desbloquean nuevas zonas y se guarda el progreso mediante el GameManager. También muestra un código para acceder a la primera grabación.
- *Mission2Controller*. Este script gestiona la segunda misión, en la que el jugador debe analizar cuatro muestras en un sistema genético antes de que acabe un temporizador. Cada análisis activa una animación y barra de progreso. Se usan cámaras virtuales para alternar vistas durante el proceso. Al finalizar el análisis de las muestras, se desbloquea un código, se actualiza el progreso en

GameManager, y se penaliza al jugador si el tiempo se agota. La interfaz muestra estado y resultados del análisis en tiempo real.

- *Mission3Controller*. Controla un minijuego de precisión donde el jugador sella tres puntos de fuga usando una llave inglesa. Se utiliza una barra de gradación de colores con una flecha oscilante para simular presión de válvulas. Al fallar, se penaliza al jugador con pérdida de oxígeno. Cada punto de fuga tiene su propia cámara y animación sincronizada. Al completar las tres fases, se guarda el progreso y se muestra el código para la siguiente grabación.
- *Mission4Controller*. Implementa una mecánica de repetición de secuencias alfabéticas. El jugador debe memorizar y reproducir correctamente tres secuencias, usando una tarjeta de seguridad como activador. Controla el movimiento, cámaras y animaciones. Con cada fallo, la misión se reinicia con penalizaciones. Al completarse correctamente, se reactiva el analizador, se desbloquea la siguiente grabación y se guarda el progreso en GameManager.
- *Mission5Controller*. Gestiona una misión donde el jugador debe cortar cuatro cables sin ser detectado por sensores. Incluye secuencias de cámara, detección de colisiones con alarmas, penalizaciones por fallo, y barra de progreso para cortar cada cable. Si se activa una alarma, se reduce energía y se oscurece el entorno. Al completar la misión, se muestra un código en una secuencia animada y se registra el avance. Controla con precisión cámaras, animaciones y estados del jugador.
- *Mission6Controller*. Gestiona una misión de lógica textual basada en el descifrado de símbolos mediante una tabla de traducción. Al interactuar con el libro, el jugador activa un modo de entrada de texto controlado por teclado, donde debe escribir tres palabras clave. Se valida cada entrada progresivamente, restringiendo inventario y movimiento. Al completar correctamente el desafío, se muestra un código, se guarda el progreso y se restablece el control del jugador. La misión incorpora cambio de cámaras y animaciones para aumentar la inmersión.
- *Mission7Controller*. Controla una misión contrarreloj donde el jugador debe recolectar 8 ítems médicos esparcidos por el almacén y colocarlos en un botiquín. Se activa mediante interacción con el kit, inicia un temporizador, y permite recoger elementos con la tecla R. Si se completa a tiempo, se guarda el progreso y se revela un nuevo código. Si no, se penaliza al jugador con pérdida de vida y reinicio con menor tiempo. Se usan cámaras virtuales, animaciones y efectos visuales para reforzar el desafío.
- *Mission8Controller*. Esta misión asigna al jugador el control de un aspirador móvil para limpiar partículas tóxicas en rejillas de ventilación. El movimiento se gestiona con flechas direccionales, dentro de un espacio tridimensional. Si el temporizador expira, se activan penalizaciones como pérdida de oxígeno y efectos visuales y sonoros. Al limpiar todas las zonas, se muestra el código de desbloqueo, se guarda el progreso y se restituye la jugabilidad. Utiliza múltiples cámaras, partículas y sonidos para reforzar la ambientación.
- *Mission9Controller*. Permite al jugador resolver un puzzle tipo 8-puzzle como parte del proceso de sincronización del sistema de navegación. Tras usar una brújula,

se inicia una animación y se presenta el puzzle bajo una nueva cámara. El jugador reorganiza piezas hasta completar la imagen objetivo. Al completarse, se desbloquea un nuevo código, se guarda el progreso y se cambia la cámara a la normal. Se sincronizan animaciones, interfaces y cámaras mediante corutinas y CanvasGroup para mejorar la narrativa visual.

- *Mission10Controller*. Gestiona la última misión del juego, donde el jugador usa una linterna UV para detectar huellas ocultas en una puerta. Cada huella está fragmentada en cuatro partes que deben reordenarse correctamente. Si se excede el número de intentos o se fallan combinaciones, se activan penalizaciones y las alarmas de la nave. Al completar las tres huellas, se abre la puerta final, se muestra el código de la última caja y se habilita la transición a la escena final. Integra control de cámaras, puzzles visuales y lógica de validación.
- *VacuumTriggerController*. Este script detecta la interacción entre el aspirador y partículas de suciedad mediante colisiones tipo trigger. Al entrar en contacto, se inicia una corutina que reduce progresivamente la opacidad del objeto hasta desactivarlo, impidiendo repeticiones con una estructura de control (HashSet). El sistema actualiza el contador de suciedad activa y notifica al mission8Controller cuando la limpieza está completa. Refuerza la dinámica de tiempo limitado con feedback visual de eliminación progresiva.
- *PuzzleController*. Gestiona el funcionamiento del puzzle 3x3 que representa la recalibración del sistema de navegación. Permite mover piezas adyacentes a una casilla vacía mediante las flechas del teclado, controlando la lógica de movimiento y resolución. Superar cierto número de movimientos genera penalizaciones al jugador. La posición correcta de las piezas se valida con comprobaciones de coordenadas. Cuando el puzzle se resuelve dentro de los límites establecidos, se notifica al controlador de misión para marcar la tarea como completada.
- *FingerprintController*. Gestiona la detección y activación visual de las huellas ocultas en la puerta final mediante la linterna UV. Detecta colisiones con las tres huellas dactilares y activa su visualización gradual mediante una corutina. Al completarse, muestra un icono para indicar que la huella puede analizarse. Si el jugador pulsa R, se activa el análisis de la huella a través de mission10Controller. El script también gestiona el control de la cámara durante la secuencia.
- *Camera18Controller*. Permite controlar la rotación de la cámara en primera persona mediante el ratón durante la inspección de huellas dactilares. Usa Input.GetAxis para capturar el movimiento en los ejes X e Y, aplicando un clamp vertical para limitar la rotación. Se activa mediante un interruptor booleano que permite o bloquea la entrada según el momento de la misión. Su propósito es ofrecer libertad visual controlada en una secuencia clave del juego, sin romper la inmersión narrativa.

5.2.4. Jugador

- *PlayerMovement*. Gestiona el desplazamiento del jugador en primera persona, incluyendo movimiento direccional, salto y gravedad. Utiliza CharacterController y Animator para controlar el movimiento fluido, con restricciones condicionadas por el sistema de inventario o cinemáticas. Permite saltos si el jugador está en el suelo y calcula la altura con física clásica. El script impide el desplazamiento si canMove o playerMov está desactivado, garantizando coherencia con el resto del sistema de juego.
- *PlayerCameraController*. Permite al jugador controlar la rotación de la cámara usando el ratón para lograr una experiencia inmersiva. Rota el cuerpo del jugador en el eje Y y ajusta el ángulo vertical de la cámara con límites entre -80° y 80° para evitar distorsión visual. Solo se activa si el jugador puede moverse y el inventario está cerrado. El cursor se bloquea y oculta al iniciar la escena, reforzando la sensación de control total en primera persona.
- *PlayerUI*. Controla de forma inmersiva los estados de vida, energía y oxígeno del jugador, utilizando cambios visuales, sonoros y en el rendimiento. La energía y el oxígeno influyen directamente en velocidad, visión y sonidos, y se representan mediante iconos dinámicos. Al alcanzar valores críticos, se activan efectos como desenfoque o sonidos de advertencia. Incorpora regeneración progresiva y funciones de daño, consumo energético y oxígeno. Se optimiza la actualización solo cuando hay cambios.
- *Conversation*. Controla el sistema de conversación final entre jugador y estación terrestre, en tres fases con múltiples opciones que determinan el final del juego. Integra reproducción de audios, textos en pantalla, cámaras y vídeos previos. Las elecciones del jugador influyen directamente en los desenlaces, sin establecer un final correcto. Se integra con GameManager para verificar el progreso y desactiva el control del jugador durante la conversación. Al concluir, se hace un fade out y se carga la escena final correspondiente.
- *CheckPlayerPosition*. Este script posiciona al jugador en el punto adecuado según su progreso guardado, y activa o desactiva scripts de misión de forma dinámica. Usa triggers físicos para registrar el avance y habilita scripts específicos con manageScripts. También reposiciona de forma segura al jugador desactivando temporalmente el CharacterController. La lógica asegura que solo se activen componentes necesarios en cada etapa, mejorando el rendimiento y la coherencia narrativa del juego.

5.2.5. Estado del juego

- *GameManager*. Este script centraliza la gestión del estado del juego usando el patrón Singleton y garantiza persistencia entre escenas con DontDestroyOnLoad. Almacena el progreso mediante arrays de desbloques, objetos, grabaciones y misiones completadas. Los métodos SaveProgress() y LoadProgress() serializan datos usando PlayerPrefs, asegurando restauración completa en cada sesión. Proporciona acceso dinámico a arrays mediante nombres, lo que facilita la interacción desde otros scripts. Es el núcleo del sistema de guardado del juego.

- *GameController*. Controla la visualización del mensaje o pantalla de introducción al comenzar el juego. Emplea un CanvasGroup y corutinas para mostrar el contenido inicial tras un retardo, seguido de una transición tipo fade-out. Una vez completado, libera el control al jugador. Este sistema mejora la inmersión narrativa mediante una entrada suave y estilizada antes de la exploración activa.
- *ResetState*. Restaura dinámicamente todos los elementos visuales y funcionales del juego según el estado guardado. Agrupa referencias a objetos del inventario, grabaciones, puertas, cajas, puzzles, sistemas ambientales, etc. Ejecuta la lógica de restauración en initializeStates(), ocultando objetos recogidos y activando elementos según misiones completadas. A través de resetMissionsStates() y checkRecordsIndex(), aplica animaciones, textos, y efectos del entorno, reflejando con precisión el avance del jugador.
- *ResetGame*. Permite iniciar el juego en un estado avanzado predefinido, útil para testing. Desbloquea objetos, grabaciones y misiones específicas, configura el progreso del jugador, y guarda el estado con GameManager.SaveProgress(). Se ejecuta automáticamente al iniciar la escena, luego desactiva su ejecución para evitar duplicaciones. Facilita la verificación de etapas avanzadas del juego sin pasar por el progreso completo.

CAPÍTULO 6. CONCLUSIONES Y PROPUESTAS DE TRABAJO FUTURO

El desarrollo de *Prometheus-9* ha supuesto un proceso de aprendizaje exhaustivo sobre lo que implica diseñar, programar y pulir un videojuego narrativo con múltiples capas. En este capítulo se recogen las conclusiones derivadas de la experiencia, así como los aprendizajes y competencias adquiridas, las limitaciones identificadas y las rutas posibles para una evolución futura del proyecto.

6.1. CONCLUSIONES GENERALES

6.1.1. Unir narrativa y jugabilidad

Uno de los principales logros de *Prometheus-9* ha sido demostrar cómo la narrativa puede ser el núcleo estructural del juego. En muchos títulos comerciales, la historia se presenta a través de cinemáticas separadas del gameplay, lo que a veces rompe la inmersión. En este proyecto, se procuró que cada mecánica, misión y objeto estuviera integrado en la historia. Esta unión diseño-narrativa hace que el jugador se sienta motivado a resolver misiones, debido a la curiosidad de conocer la historia completa.

6.1.2. Diseño modular y escalable

Otro acierto significativo fue la implementación de una arquitectura de software modular, inspirada en principios sólidos de ingeniería. Gracias a ello:

- Cada misión es un bloque relativamente independiente que puede modificarse o mejorarse sin afectar a las demás.
- Sistemas como la interfaz, los datos globales, el inventario o las cámaras están desacoplados entre sí mediante el GameManager y eventos, lo que facilitó depurar y ajustar componentes individuales.
- El sistema resultante puede escalar fácilmente: añadir misiones nuevas, más objetos o incluso escenas adicionales no requeriría reescribir la base, sólo seguir extendiendo con nuevos módulos.

La escalabilidad lograda da margen para que el proyecto continúe creciendo si se desea.

6.1.3. La narrativa como motor emocional

La historia del juego fue construida para provocar emociones en el jugador: miedo ante lo desconocido, sentido de responsabilidad, dudas morales, sensación de aislamiento, etc. Se evitó deliberadamente el uso de combates para generar emoción; y, en su lugar, se buscó que el jugador sintiera tensión al:

- Quedarse sin energía y ver parpadear las luces.
- Fracasas una misión crítica y oír alarmas.
- Escuchar una grabación con implicaciones morales fuertes.
- Enfrentar una decisión con consecuencias irreversibles en el final.

Este enfoque se alinea con una corriente de diseño narrativo más moderna que apuesta por lo emocional y reflexivo, más allá del reto mecánico tradicional.

6.2. OBJETIVOS ALCANZADOS

6.2.1. Revisión exhaustiva de los objetivos propuestos

A continuación, se enumeran los objetivos iniciales planteados al proyecto y cómo fueron materializados en el producto final.

1. Crear un videojuego narrativo jugable de principio a fin: se desarrolló una experiencia completa con introducción, 10 misiones de desarrollo, un clímax y un desenlace ramificado en tres finales.
2. Integrar historia y mecánicas de forma coherente: cada mecánica está justificada dentro de la narrativa y aporta a la comprensión de la historia.
3. Desarrollar múltiples tipos de interacción: el juego incluye secciones de sigilo, calibración de sistemas, introducción de códigos, puzzles de lógica, minijuegos de habilidad y limpieza, cubriendo un amplio espectro de interacciones.
4. Aplicar elementos de gamificación para motivar al jugador: se incorporaron barras de salud y energía, penalizaciones por error, recompensas por éxito, un sistema de progresión por misiones y desbloqueo gradual de contenido, manteniendo al jugador comprometido.
5. Desarrollar una arquitectura escalable y mantenible: todos los módulos del juego son reutilizables, están bien documentados en el código y desacoplados entre sí; se pueden extender fácilmente con contenido nuevo sin cambiar la base.
6. Utilizar metodologías ágiles adaptadas al trabajo individual: se utilizó un backlog con sprints de implementación cortos, retrospectivas y ajustes iterativos constantes que guiaron el proyecto de forma flexible pero controlada.
7. Documentar el proceso de forma rigurosa: se generó una memoria técnica completa, que incluye la base teórica, los objetivos, el desarrollo, los resultados y reflexiones críticas sobre el proyecto.

De esta manera, todos los objetivos planteados inicialmente fueron cumplidos. Esto no significa que no haya aspectos mejorables, pero sí que el proyecto logró hacer realidad sus aspiraciones fundamentales.

6.3. COMPETENCIAS ADQUIRIDAS

El proceso de creación de *Prometheus-9* ha supuesto un entrenamiento en múltiples disciplinas relacionadas con el desarrollo de videojuegos. Entre las competencias más valiosas adquiridas por el desarrollador a lo largo del proyecto, se destacan las siguientes.

6.3.1. Competencias técnicas avanzadas

- Programación orientada a objetos en C#, incluyendo gestión de memoria, optimización y depuración en tiempo real.
- Desarrollo con Unity a nivel intermedio-avanzado: uso de prefabs, animaciones, físicas, *Canvas UI*, sistema de eventos y corutinas, integración de plugins y manejo de escenas múltiples.
- Patrones de diseño aplicados a videojuegos: implementación de Singletons, gestores, controladores de estado, máquina de estados sencilla para misiones, etc., entendiendo sus beneficios y riesgos en un entorno de juego.
- Sistemas de interacción compleja: uso de *raycasting*, *triggers*, detección de mirada y encadenamiento de condiciones lógicas entre objetos del mundo.
- Gestión de datos globales persistentes entre escenas con DontDestroyOnLoad y serialización simple de estado.
- Optimización de rendimiento en Unity: técnicas de occlusion culling, baking de iluminación, reducción de draw calls y control de complejidad de escena.

6.3.2. Competencias creativas y de diseño narrativo

- Escritura de narrativa interactiva: elaborar una historia fragmentada en registros, diseñar personajes, crear finales múltiples coherentes, etc.
- Diseño emocional del jugador: aplicación de principios de diseño emocional, generando tensión y alivio alternados, manejando la incertidumbre, etc.

6.3.3. Competencias organizativas y de gestión de proyecto

- Planificación ágil: organización con Scrum, estableciendo *sprints* semanales, priorizando tareas, y adaptándose a cambios sin perder de vista el objetivo global.
- Documentación técnica clara y estructurada: la elaboración de este documento ha reforzado la capacidad de explicar decisiones de diseño y arquitectura, y reflexionar sobre el propio trabajo de manera objetiva.

Estas habilidades no solo son relevantes para el desarrollo de videojuegos, sino que muchas de ellas son transferibles a otros entornos de la ingeniería de software.

6.4. LIMITACIONES TÉCNICAS Y NARRATIVAS

Aunque el proyecto ha cumplido sus objetivos principales, es importante reconocer sus límites actuales, tanto en el ámbito técnico como en el creativo, los cuales ofrecen espacio para futuras mejoras.

6.4.1. Alcance narrativo limitado

- La historia, si bien inmersiva, está centrada exclusivamente en el punto de vista del protagonista y los registros de la tripulación. No hay escenas donde interactúen personajes en tiempo real.
- No hay eventos que cambien en función de decisiones intermedias del jugador; la ramificación narrativa solo ocurre al final. Esto significa que durante el desarrollo del juego la historia es lineal.
- No existen diálogos activos. La interacción narrativa es unidireccional.

6.4.2. Recursos visuales genéricos

- Todos los modelos 3D de la nave y objetos son assets gratuitos, lo que, si bien ahorra tiempo, conlleva que la dirección artística no sea única.
- Falta una coherencia estética completa: los distintos módulos de la nave podrían tener identidades visuales más diferenciadas. Actualmente comparten elementos genéricos, lo que empobrece un poco la sensación de lugar real.

6.4.3. Inteligencia Artificial limitada

- No hay otros NPCs ni enemigos con comportamientos complejos o dinámicos. La IA es inexistente porque el entorno está vacío de seres vivos.

6.4.4. Duración corta

- La experiencia completa se puede terminar en aproximadamente 2-3 horas, siendo de una duración modesta. Esto es apropiado para el alcance, pero en comparación con juegos comerciales es corto.
- No hay rejugabilidad mecánica más allá de ver distintos finales. Es decir, el núcleo jugable será igual en cada partida; el incentivo para rejugar es solo narrativo.

6.5. TRABAJO FUTURO

Pensando en convertir *Prometheus-9* en un juego más completo, inmersivo y pulido, a continuación, se proponen diversas líneas de expansión y mejora que podrían abordarse en trabajos futuros.

6.5.1. Ampliación narrativa

- Incorporar diálogos interactivos en tiempo real: por ejemplo, mediante un sistema de comunicación por radio con algún superviviente o IA de la nave. Esto añadiría dimensiones a la narrativa y rompería la soledad absoluta.
- Incluir decisiones morales a lo largo del juego, no solo al final. Por ejemplo, decidir entre dos cursos de acción en una misión afectando a eventos posteriores.
- Crear más finales, algunos secretos o desbloqueables tras ciertas condiciones.

6.5.2. Mejora de la estructura jugable

- Introducir un sistema de exploración libre una vez se completan ciertas misiones.
- Crear contenido opcional: salas ocultas, recursos o easter eggs que no sean necesarios para terminar el juego, pero enriquezcan el lore o supongan un reto.

6.5.3. Evolución audiovisual

- Sustituir las cinemáticas de grabaciones en vídeo por eventos interactivos in-game. En vez de mostrar un vídeo al encontrar una cinta, se podría recrear la escena como un holograma 3D que el jugador presencia dentro de la nave, haciendo las transiciones menos bruscas.

6.6. CONCLUSIÓN FINAL

Prometheus-9 no es solo un juego para efectos académicos, sino que se concibió como una experiencia emocional y una propuesta narrativa que demuestra cómo los videojuegos pueden contar historias con profundidad, tensión y dilemas. A lo largo del desarrollo se integraron disciplinas diversas para servir a una visión unificada, y el resultado es un mundo interactivo pequeño pero completo.

Este prototipo es solo un punto de partida. La historia de *Prometheus-9*, sus tripulantes y el misterioso contacto alienígena todavía tiene mucho que revelar. El proyecto deja abierta la posibilidad de expandirse, enriquecer su contenido y seguir explorando las posibilidades que ofrece la narrativa interactiva dentro del medio videojuego.