

Juego Educativo Enfocado en Gestión de Costos de Producción y Manejo de Recursos - ProfitQuest

Documento de Arquitectura de Software
Versión 1.0

Leonardo Céspedes Tenorio - 2022080602

Kevin Chang Chang - 2022039050

Mauricio Fernández Brizuela- 2022437510

Daniel Garbanzo Carvajal - 2022117129

24 de Marzo del 2025

Índice

1. Introducción.....	3
1.1. Propósito.....	3
1.2. Alcance.....	3
1.3. Definiciones, Acrónimos y Abreviaciones.....	4
1.4. Referencias.....	4
1.5. Vista General.....	5
2. Representación Arquitectural.....	6
3. Objetivos y Limitaciones Arquitecturales.....	6
3.1. Requerimientos Funcionales.....	7
3.2. Requerimientos No Funcionales.....	9
3.3. Diseño y Estrategia de Implementación.....	11
3.4. Herramientas.....	12
4. Vista de Escenarios.....	14
4.1. Módulo de Simulación de Producción.....	14
4.2. Módulo de Jugabilidad e Interacción.....	27
4.3. Módulo de Evaluación y Feedback.....	34
4.4. Módulo de Experiencia de Usuario.....	38
4.5. Módulo de Gestión de Usuarios.....	43
5. Vista Lógica.....	45
5.1. Vista General.....	45
5.2. Componentes de Diseño Específicos Relevantes.....	46
5.2.1. Nivel.....	46
5.2.2. Jugador.....	46
5.2.3. Sistema de Economía.....	46
5.2.4. UI.....	46
5.2.5. Evaluador.....	46
5.3. Casos de Uso.....	48
5.3.1. Simulación de Producción.....	48
5.3.1.1. Diseño de Niveles Preconfigurados.....	48
5.3.1.2. Gestión de Presupuesto.....	49
5.3.1.3. Gestión de Inventario.....	50
5.3.1.4. Gestión de Productos.....	51
5.3.1.5. Gestión de Tipos de Productos.....	52
5.3.1.6. Entrega de Productos.....	53
5.3.1.7. Compra de Materiales.....	54
5.3.1.8. Manejo de Desperdicio.....	55
5.3.2. Jugabilidad e Interacción.....	56
5.3.2.1. Movimiento de Personaje.....	56
5.3.2.2. Sistema de Penalización.....	57
5.3.2.3. Manejo de Errores en Producción.....	58
5.3.2.4. Dificultad Progresiva.....	59

5.3.2.5. Objetivos Opcionales por Nivel.....	60
5.3.2.6. Tiempo Límite.....	61
5.3.3. Evaluación y Feedback.....	62
5.3.3.1. Validación de Desempeño por Partida.....	62
5.3.3.2. Tabla de Clasificación de Jugadores.....	63
5.3.4. Experiencia de Usuario.....	64
5.3.4.1. Tutorial Inicial.....	64
5.3.4.2. Guardar Partida.....	65
5.3.4.3. Configuración del Avatar.....	66
5.3.5. Gestión de Usuarios y Datos.....	67
5.3.5.1. Gestión de Usuarios.....	67
6. Vista de Procesos.....	68
6.1. Proceso de Configuración Inicial.....	68
6.2. Proceso de Selección de Nivel.....	69
6.3. Proceso de Producción.....	70
6.4. Proceso de Entrega y Finalización.....	71
6.5. Proceso de Evaluación y Recompensa.....	72
7. Vista Física.....	73
8. Vista de Implementación.....	73
9. Vista de Datos.....	74
10. Tamaño y Desempeño.....	74
11. Calidad.....	76
11.1. Requerimientos de Fiabilidad.....	76
11.2. Consideraciones de Seguridad.....	76
11.3. Estándares de Programación.....	77
11.4. Consideraciones de Escalabilidad.....	78
12. Anexos.....	79

1. Introducción

Este documento corresponde al SAD (Software Architecture Document) del sistema ProfitQuest, el cual es un juego educativo diseñado para enseñar gestión de costos de producción y manejo de recursos en costos de producción a estudiantes sin formación previa en esta área. Su objetivo es ofrecer una visión panorámica de la arquitectura de software diseñada para facilitar la gestión de las funcionalidades esenciales. En la siguiente sección se mostrarán aspectos generales del contenido del documento. Se detalla el propósito del documento, el alcance que se espera con el proyecto, definiciones, acrónimos y abreviaciones utilizados en el documento, referencias bibliográficas utilizadas y por último la vista general en la que se abarca el resto del documento.

1.1. Propósito

Este documento tiene como propósito definir de forma clara y estructurada la arquitectura de software de “ProfitQuest”, un sistema educativo en forma de videojuego diseñado para simular escenarios de producción y fomentar el aprendizaje práctico de la gestión de recursos.

Servirá como guía de referencia para todos los interesados en el desarrollo del sistema, incluyendo al equipo de desarrollo, la profesora supervisora y otros posibles colaboradores. También facilitará futuras labores de mantenimiento y evolución del sistema, asegurando coherencia y continuidad en el diseño arquitectónico.

1.2. Alcance

Este documento aborda la arquitectura de software del sistema “ProfitQuest”, el cual será implementado como un videojuego educativo en 2D con estilo visual tipo pixel art. Está dirigido a estudiantes sin conocimientos previos en administración, y será ejecutado en computadoras de escritorio sin requerir conexión a internet continua.

- El sistema cuenta con los siguientes módulos clave:
- Simulación de procesos de producción.
- Gestión de presupuesto e inventario.
- Evaluación de desempeño del usuario.

- Progresión mediante niveles preconfigurados.
- Interfaz gráfica intuitiva con retroalimentación visual.

El SAD define cómo estos módulos interactúan desde una perspectiva arquitectónica, empleando un enfoque basado en vistas múltiples para facilitar la comprensión integral del sistema.

1.3. Definiciones, Acrónimos y Abreviaciones

Término	Descripción
ERS	Especificación de Requisitos de Software.
GUI	Interfaz Gráfica de Usuario.
SAD	Documento de Arquitectura de Software (Software Architecture Document).
Pixel Art	Estilo gráfico basado en píxeles, utilizado como estética visual del juego.
Godot	Motor de desarrollo de videojuegos empleado en la implementación de “ProfitQuest”.
MVC	Model-View-Controller. Patrón de diseño que separa la lógica, la interfaz de usuario y la interacción de usuario en diferentes componentes
ECS	Entity Control System, Patrón de diseño usado en juegos para representar objetos en el mundo del juego.

1.4. Referencias

Título	Autor/Organización	Relación	Año	Enlace
IEEE 830-1998	IEEE Computer Society	Plantilla de ERS utilizada como base de especificación	2008	https://standards.ieee.org/ieee/830/1222/
Godot Docs	Linietsky, J. & Manzur, A.	Documentación oficial del	2014	https://docs.godotengine.org/en/

		motor de desarrollo.		stable/
--	--	----------------------	--	-------------------------

1.5. Vista General

El presente documento está diseñado para ofrecer una descripción integral de la arquitectura del sistema “ProfitQuest”, cubriendo todos los aspectos necesarios para su implementación y evolución. A lo largo del texto, se presentan de manera estructurada los distintos componentes que conforman el sistema, así como su interacción, organización lógica, distribución física y comportamiento en tiempo de ejecución.

En primera instancia, se brinda una introducción al proyecto, donde se especifican el propósito, alcance, terminología y fuentes de referencia que respaldan la arquitectura propuesta. Posteriormente, se expone una representación arquitectural basada en el modelo de vistas múltiples, el cual facilita la visualización del sistema desde distintas perspectivas complementarias. Este enfoque permite descomponer la complejidad del software y organizar su diseño de manera coherente.

Más adelante, se detallan los objetivos arquitecturales y las restricciones relevantes, incluyendo tanto los requerimientos funcionales como no funcionales que el sistema debe satisfacer. Estas condiciones guían las decisiones de diseño y delimitan el alcance de la solución propuesta.

El documento también incorpora una serie de vistas arquitectónicas que permiten analizar el sistema desde diferentes ángulos. Se incluyen vistas de casos de uso, lógica, procesos, despliegue e implementación. Estas vistas permiten comprender el comportamiento del sistema ante diferentes escenarios de interacción, su estructura interna, la comunicación entre procesos, la forma en que se distribuyen los componentes en la infraestructura y cómo se organiza el código fuente y los módulos del proyecto.

Finalmente, se presentan aspectos relacionados con el rendimiento, la escalabilidad, la seguridad, los estándares de calidad y el mantenimiento, asegurando que el sistema sea robusto, eficiente y adaptable a futuras necesidades. En conjunto,

este documento constituye una guía esencial para el desarrollo ordenado de “ProfitQuest”, sirviendo como base para decisiones técnicas, validaciones posteriores y mejoras continuas del sistema.

2. Representación Arquitectural

Este juego se va a implementar como una aplicación para el ordenador, se seguirá una arquitectura apegada al manejo de lógica funcional, interfaz gráfica e interactividad entre el usuario y el juego. La idea es implementar una interfaz clara e intuitiva para la navegación de la aplicación y para el juego ambiente fácil de navegar, objetivos fáciles de entender y completar y capacidad del sistema de reaccionar y funcionar de la forma correcta para que el juego sea coherente.

Para diseñar la arquitectura del sistema, se adopta el modelo de vistas 4+1, que proporciona una comprensión holística desde diversas perspectivas. A continuación, se explica cómo se aplican estas vistas en el documento:

- Vista de Casos de Uso: Describe los escenarios o casos de uso del sistema, ilustrando cómo interactúan los componentes para cumplir con los requisitos utilizando diagramas de casos de uso.
- Vista Lógica: Representa la estructura lógica del sistema, mostrando los componentes de software y sus relaciones a través de diagramas de clases.
- Vista de Procesos: Muestra la organización y comunicación de los procesos del sistema, delineando cómo los componentes ejecutan y se comunican entre sí mediante diagramas de actividad.
- Vista Física: Esta vista muestra la distribución física de los componentes del sistema en el entorno físico, como hardware, dispositivos de red, servidores, etc. Se centra en cómo se organiza y despliega el hardware para admitir la arquitectura del software.

3. Objetivos y Limitaciones Arquitecturales

Esta sección tiene como objetivo dirigir el diseño y la implementación de la arquitectura, garantizando que el software final cumpla con los requisitos y expectativas establecidos. Aquí se abordarán los requisitos funcionales y no funcionales del sistema, así como el diseño y la estrategia para su implementación.

3.1. Requerimientos Funcionales

ID	Descripción
PQ-01	Los jugadores deben poder acceder y jugar niveles preconfigurados que fueron creados durante la fase de desarrollo.
PQ-02	El jugador debe ser capaz de moverse por el entorno 2D del respectivo nivel mediante el uso de teclas asignadas.
PQ-03	El sistema aplica sanciones de manera automática cuando el jugador fabrica productos incorrectos o mezcla materiales erróneos.
PQ-04	El jugador podrá utilizar una estación especial para deshacer hasta tres errores por nivel y recuperar el 50% de los materiales gastados.
PQ-05	El jugador será capaz de jugar niveles cuya dificultad aumenta progresivamente.
PQ-06	Los jugadores deben contar con un presupuesto inicial en cada nivel, el cual funcionará como recurso limitado para realizar acciones clave del juego y cumplir con los objetivos establecidos.
PQ-07	Los jugadores deben gestionar un inventario de materias primas que se actualiza dinámicamente según las acciones del jugador, como fabricar productos o comprar materiales.
PQ-08	El sistema debe evaluar automáticamente el desempeño del jugador al final de cada nivel, utilizando métricas como manejo del presupuesto, inventario y cumplimiento de objetivos.

PQ-09	El sistema debe mostrar una tabla de clasificación por nivel, ordenando a los jugadores con base en su puntaje de desempeño, permitiendo comparar resultados y acceder a detalles por jugador.
PQ-10	Los niveles deben incluir objetivos opcionales que, aunque no son necesarios para completar el nivel, permiten mejorar el puntaje de desempeño del jugador si se cumplen.
PQ-11	El sistema debe permitir la creación y gestión de productos utilizando recursos del inventario, considerando características como costo de producción y demanda del mercado.
PQ-12	Los productos dentro del juego deben clasificarse en diferentes tipos con atributos únicos, afectando la estrategia de producción del jugador según su progreso en el juego.
PQ-13	Los jugadores deben entregar los productos terminados en una zona específica del nivel para completar el ciclo de producción y recibir recompensas por su eficiencia y calidad.
PQ-14	Los jugadores podrán comprar materiales dentro del juego utilizando sus recursos disponibles, visualizando los costos antes de confirmar la compra y actualizando el inventario tras la transacción.
PQ-15	El sistema debe ofrecer un tutorial inicial obligatorio para nuevos jugadores, el cual introduce de forma progresiva las mecánicas básicas como presupuesto, inventario y producción.

PQ-16	Cada nivel debe tener un tiempo límite que obligue al jugador a tomar decisiones estratégicas bajo presión; al finalizar el tiempo, el nivel concluye automáticamente y se evalúa el desempeño.
PQ-17	Los jugadores deben poder guardar su progreso en cualquier momento, incluyendo estado del inventario, presupuesto, nivel en curso y tiempo restante, permitiendo continuar la partida posteriormente.
PQ-18	El jugador podrá personalizar su avatar antes de iniciar una partida, eligiendo entre opciones de color, atuendo y accesorios predefinidos, sin que esto afecte el desempeño del juego.
PQ-19	El sistema debe gestionar perfiles de usuario para que cada jugador tenga un historial de progreso, configuraciones personalizadas y estadísticas independientes dentro del mismo dispositivo.
PQ-20	Los jugadores deben tener la opción de reciclar materiales en exceso del inventario antes de ser utilizados, recuperando parcialmente su valor y evitando acumulación innecesaria de recursos.

3.2. Requerimientos No Funcionales

ID	Descripción
PQNF-01	El juego debe mantener una tasa mínima de 60 cuadros por segundo (FPS) en equipos que cumplan con los requisitos recomendados, garantizando fluidez en la experiencia de juego.

PQNF-02	La interfaz gráfica debe ser intuitiva y fácil de navegar, permitiendo a jugadores sin experiencia previa interactuar con el sistema sin complicaciones ni confusión.
PQNF-03	El sistema debe garantizar estabilidad durante la ejecución, evitando bloqueos o fallos críticos, y guardando automáticamente el progreso reciente ante cualquier cierre inesperado.
PQNF-04	El juego debe ser compatible con sistemas operativos Windows, macOS y Linux, adaptándose correctamente a distintas resoluciones de pantalla a partir de 1280x720 píxeles.
PQNF-05	Los datos del jugador, incluyendo progreso y configuraciones, deben almacenarse de manera segura, evitando accesos no autorizados y protegiendo la integridad de las partidas guardadas.
PQNF-06	El sistema debe estar diseñado para permitir la incorporación de nuevos niveles, mecánicas o funcionalidades en el futuro sin afectar el correcto funcionamiento del juego base.
PQNF-07	El código del juego debe seguir una estructura modular y buenas prácticas de desarrollo que faciliten su mantenimiento, corrección de errores y actualización en futuras versiones.

3.3. Diseño y Estrategia de Implementación

El diseño del sistema “ProfitQuest” se ha orientado a la modularidad, escalabilidad y claridad estructural, con el fin de facilitar tanto su implementación inicial como su mantenimiento y evolución futura. La arquitectura adoptada permite separar las distintas responsabilidades del sistema en componentes bien definidos, promoviendo la reutilización de código y el desarrollo paralelo por parte del equipo.

La estrategia de implementación parte de una organización lógica en capas, donde se diferencian claramente los módulos encargados de la lógica del juego, la interfaz gráfica del usuario (GUI), la gestión de datos, y el control de flujo de las partidas. Esta separación de responsabilidades se refleja directamente en la estructura de carpetas y scripts del motor Godot, permitiendo una navegación intuitiva del proyecto.

La lógica de negocio se encapsula en scripts dedicados que controlan mecánicas como la simulación de producción, el cálculo de costos, la administración de inventario y presupuesto, así como la validación de objetivos. Estos scripts son independientes entre sí en la medida de lo posible, comunicándose a través de señales y patrones de diseño como observadores o gestores centralizados (singleton) para reducir el acoplamiento.

Por su parte, la interfaz gráfica ha sido diseñada utilizando las herramientas nativas del motor Godot, con un enfoque en la usabilidad, consistencia visual y retroalimentación inmediata al usuario. Los elementos de la GUI están organizados en escenas reutilizables y controladas mediante nodos jerárquicos, lo que permite su integración sencilla en distintos contextos del juego (como menús, pantallas de resultados o ventanas emergentes).

En cuanto al control de flujo, se ha establecido un sistema de navegación basado en un gestor de escenas principal que orquesta las transiciones entre menús, niveles y pantallas auxiliares. Este enfoque centralizado permite controlar el estado del juego en todo momento, facilitar la implementación de guardado y cargado de partidas, y aplicar lógica de validación global sin duplicación innecesaria de código.

Además, se han aplicado principios de diseño orientado a objetos y buenas prácticas como SOLID, especialmente en la gestión de entidades como niveles, productos y jugadores. Estas entidades han sido modeladas como objetos que

contienen tanto atributos como comportamientos relevantes para su función dentro del juego.

Durante la implementación, se priorizan tareas de alto impacto funcional, como la construcción del motor de simulación y los módulos de evaluación del jugador, seguido por el diseño visual, el sistema de progreso por niveles, y la integración de funcionalidades complementarias como el tutorial o la tabla de clasificación. Esta estrategia iterativa permitirá validar la funcionalidad del núcleo del sistema de forma temprana y sobre ella construir componentes adicionales con mayor confianza.

El diseño e implementación de “ProfitQuest” se sustenta en una arquitectura clara, basada en módulos desacoplados, estructuras reutilizables y principios de desarrollo escalable. Esto garantiza un sistema sólido, eficiente y preparado para incorporar mejoras sin comprometer la calidad ni la estabilidad de la aplicación.

3.4. Herramientas

Para el desarrollo del sistema “ProfitQuest”, se ha seleccionado un conjunto de herramientas tecnológicas que permiten implementar una solución robusta, eficiente y sostenible. Se han priorizado programas de código abierto y de uso libre que proporcionen la flexibilidad necesaria para el desarrollo de un videojuego educativo en 2D, así como la creación de recursos visuales y auditivos personalizados.

Las principales herramientas que se utilizarán en el proyecto son las siguientes:

Componente	Herramienta seleccionada	Descripción
Motor de desarrollo	Godot Engine	Motor de videojuegos open source, ligero y altamente eficiente. Permite crear videojuegos 2D y 3D utilizando GDScript, un lenguaje inspirado en Python. Su sistema de escenas y nodos facilita la modularidad y escalabilidad del juego.
Lenguaje de programación	GDScript	Lenguaje de scripting propio de Godot, diseñado

		específicamente para el desarrollo de videojuegos, de sintaxis clara y fácil de aprender.
Diseño gráfico (Pixel Art)	Aseprite	Aseprite es una herramienta especializada para la creación de pixel art y animaciones 2D.
Edición de audio	LMMS	Estación de trabajo de audio digital de código abierto, ideal para la creación de música, efectos de sonido y mezcla de pistas. Permite la producción de audio compleja sin necesidad de licencias propietarias.
Control de versiones	Git + GitHub	Git se emplea para el control de versiones del código fuente, mientras que GitHub se utilizará como plataforma de colaboración, documentación y seguimiento de avances.
Edición de mapas / niveles	Tiled	Editor de mapas 2D basado en tiles, de código abierto, que permite diseñar niveles de forma visual y exportarlos en formatos compatibles con Godot. Ideal para organizar las escenas pre-configuradas del juego.
Bases de Datos	SQLite	Base de datos relacional simple. Librería utilizada por Godot para usar servicios de bases de datos
Documentación y diseño	Draw.io / Penpot / Figma	Herramientas libres para la creación de diagramas UML, flujos de interacción y diseño de interfaces. Penpot es una opción emergente open source enfocada en el diseño UI/UX. Figma es similar a

		Penpot pero no es open source.
--	--	--------------------------------

Estas herramientas permiten cubrir todas las áreas necesarias del proyecto sin recurrir a licencias comerciales, promoviendo la accesibilidad, colaboración y continuidad del desarrollo en entornos educativos o de recursos limitados.

4. Vista de Escenarios

4.1. Módulo de Simulación de Producción

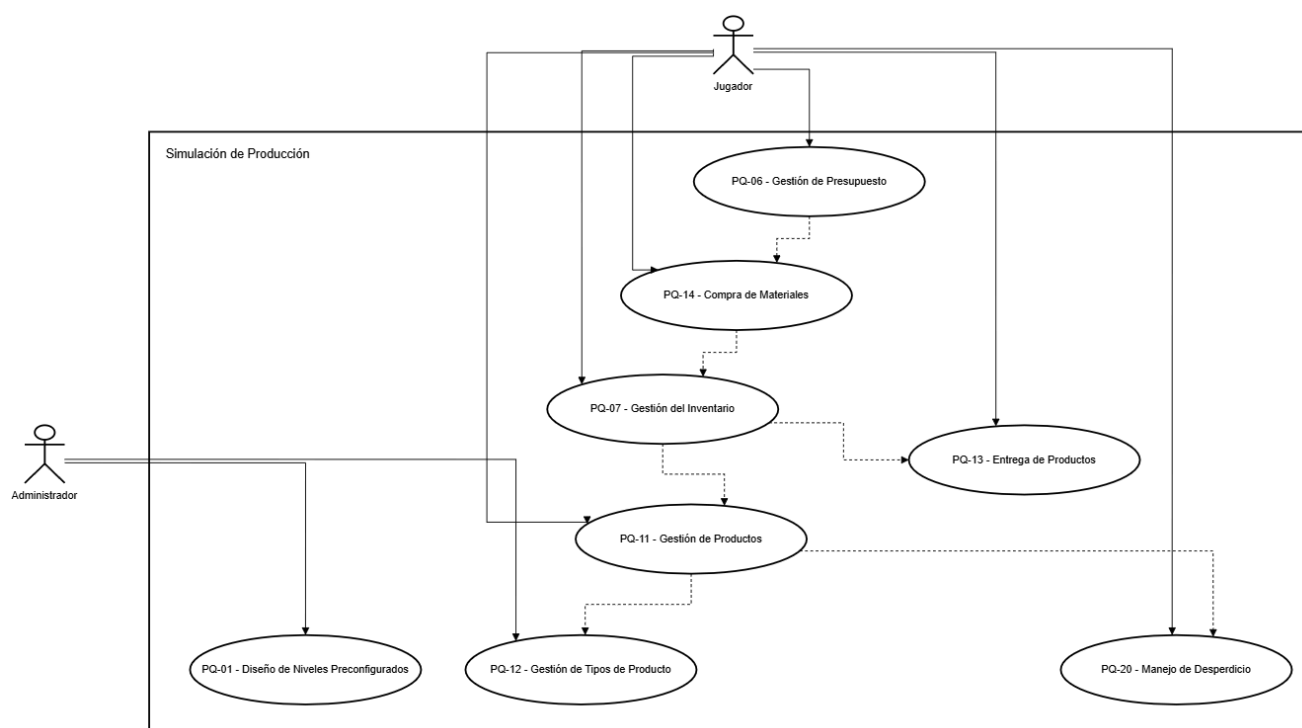


Figura 01: Diagrama de Casos de Uso para el módulo de simulación de producción

Ficha de Descripción de Caso de Uso	
ID	PQ-01
Nombre	Diseño de Niveles Preconfigurados
Descripción	Los jugadores deben poder acceder y jugar niveles estáticos preconfigurados que fueron creados durante la fase de desarrollo. Cada nivel incluye estaciones de trabajo, recursos iniciales, materiales disponibles,

	objetivos y restricciones fijas. Los objetivos incluyen mantener los gastos dentro del presupuesto, calcular los costos de la operación y manejo general y eficiente de recursos y costos en el proceso de producción.
Actores	Jugador
Precondiciones	El jugador ha desbloqueado el nivel según su progreso en el juego. El sistema ha cargado correctamente los niveles preconfigurados.
Postcondiciones	El jugador ha completado o abandonado el nivel y su progreso se ha registrado. Se mantiene la configuración del nivel para futuras partidas.
Evento activador	El jugador selecciona un nivel para empezar la partida.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador selecciona un nivel preconfigurado desde el menú. 2. El sistema verifica que el nivel está desbloqueado y carga su configuración. 3. El jugador inicia la partida en el nivel elegido. 4. Se renderiza la escena con estaciones de trabajo, recursos iniciales y objetivos establecidos.
Flujos alternativos	<p>Nivel bloqueado: El sistema indica que el nivel no está disponible hasta cumplir los requisitos previos.</p> <p>Error en la carga: Si los datos del nivel no se cargan correctamente, el sistema devuelve un mensaje de error y sugiere reiniciar la aplicación.</p>
Restricciones	<p>Cada nivel debe incluir al menos tres estaciones de trabajo, objetivos claros y recursos iniciales predefinidos.</p> <p>Los jugadores no pueden modificar la estructura del nivel.</p>

	Los niveles deben cargarse sin errores y mantener su configuración predefinida en cada partida.
Casos de uso relacionados	PQ-05, PQ-06, PQ-07, PQ-08, PQ-10, PQ-15

Ficha de Descripción de Caso de Uso	
ID	PQ-06
Nombre	Gestión de Presupuesto
Descripción	Permite al jugador gestionar un presupuesto limitado asignado al inicio de cada nivel. El presupuesto funciona como un recurso clave para realizar acciones como la compra de materiales, producción de bienes y gestión general de recursos. El jugador debe completar los objetivos sin exceder el presupuesto establecido.
Actores	Jugador / Sistema
Precondiciones	El jugador ha iniciado un nivel preconfigurado. El sistema ha cargado correctamente el presupuesto inicial del nivel.
Postcondiciones	El jugador ha completado el nivel utilizando el presupuesto asignado. El sistema ha registrado las modificaciones al presupuesto realizadas durante la partida.
Evento activador	El jugador inicia un nivel del juego.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador selecciona e inicia un nivel. 2. El sistema presenta el presupuesto inicial en pantalla. 3. El jugador realiza acciones (compra, producción, etc.). 4. El sistema descuenta automáticamente el monto

	<p>correspondiente del presupuesto.</p> <p>5. El jugador completa los objetivos del nivel.</p> <p>6. El sistema evalúa el uso del presupuesto y finaliza el nivel.</p>
Flujos alternativos	<p>Si el jugador intenta realizar una acción cuyo costo supera el presupuesto restante, el sistema cancela la acción e informa que no hay fondos suficientes.</p> <p>Si el jugador abandona el nivel antes de completarlo, el presupuesto se reinicia en la próxima partida.</p> <p>Si el jugador falla el nivel, el presupuesto y las condiciones del nivel se restablecen.</p>
Restricciones	<p>El presupuesto es fijo por nivel y no puede ser modificado por el jugador.</p> <p>No se pueden realizar acciones si el presupuesto restante no lo permite.</p>
Casos de uso relacionados	PQ-01, PQ-07, PQ-08, PQ-14, PQ-16

Ficha de Descripción de Caso de Uso	
ID	PQ-07
Nombre	Gestión del Inventario
Descripción	<p>Permite al jugador gestionar un inventario de materias primas en cada nivel. El inventario refleja en tiempo real los materiales disponibles, los utilizados en la producción y los adquiridos mediante compras. Su correcta administración es esencial para cumplir los objetivos del nivel sin agotar recursos clave.</p>
Actores	Jugador / Sistema
Precondiciones	El jugador ha iniciado un nivel preconfigurado.

	El sistema ha generado el inventario inicial correspondiente al nivel.
Postcondiciones	<p>El inventario ha sido actualizado de acuerdo con las acciones del jugador.</p> <p>Solo se han utilizado los materiales disponibles.</p> <p>El estado del inventario refleja con precisión el consumo de recursos.</p>
Evento activador	El jugador accede a un nivel de juego.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador entra a un nivel preconfigurado. 2. El sistema genera y presenta el inventario inicial con materias primas predeterminadas. 3. El jugador utiliza materiales del inventario para fabricar productos. 4. El sistema descuenta automáticamente los materiales utilizados. 5. El jugador completa el nivel utilizando únicamente los recursos disponibles.
Flujos alternativos	<p>Si el jugador reinicia el nivel manualmente, el sistema restaura el inventario a su estado original.</p> <p>Si el jugador falla el nivel, el sistema reinicia el nivel y restablece el inventario.</p> <p>Si el jugador intenta utilizar materiales inexistentes o agotados, el sistema bloquea la acción y muestra una advertencia.</p>
Restricciones	<p>Cada nivel tiene un inventario predefinido e independiente.</p> <p>No se pueden utilizar materiales no presentes o agotados en el inventario.</p> <p>Cada material tiene un costo asociado que impacta el presupuesto total.</p>

Casos de uso relacionados	PQ-01, PQ-06, PQ-11, PQ-14, PQ-20
----------------------------------	-----------------------------------

Ficha de Descripción de Caso de Uso	
ID	PQ-11
Nombre	Gestión de productos
Descripción	Permite al jugador seleccionar y fabricar productos utilizando los materiales disponibles en su inventario. Cada producto tiene requisitos específicos como costo, tiempo de fabricación y demanda. La correcta elección y gestión de productos es esencial para avanzar en el juego y optimizar los recursos.
Actores	Jugador / Sistema
Precondiciones	El jugador ha desbloqueado la opción de fabricación de productos. El sistema ha cargado la lista de productos disponibles. El jugador dispone de los materiales requeridos para fabricar al menos un producto.
Postcondiciones	El producto seleccionado ha sido fabricado exitosamente y agregado al inventario. Los materiales necesarios han sido descontados del inventario del jugador. El sistema ha actualizado el estado del inventario y el presupuesto.
Evento activador	El jugador accede al menú de productos o interactúa con una estación de producción.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador abre el menú de productos disponibles. 2. El sistema muestra la lista con costos, tiempos y requisitos de cada producto. 3. El jugador selecciona un producto a fabricar.

	<p>4. El sistema verifica la disponibilidad de materiales.</p> <p>5. Si se cumplen los requisitos, el sistema inicia la producción y descuenta los recursos.</p> <p>6. Una vez completado el proceso, el producto se agrega al inventario del jugador.</p>
Flujos alternativos	<p>Si el jugador no tiene suficientes materiales, el sistema muestra un mensaje y sugiere comprar más.</p> <p>Si el jugador cancela la fabricación, los materiales parcialmente usados se devuelven.</p> <p>Si se produce un error de sistema durante la producción, se cancela el proceso y se devuelven los recursos.</p>
Restricciones	<p>Los productos deben estar predefinidos y no pueden ser modificados por el jugador.</p> <p>No se permite fabricar productos si no se cuentan con todos los materiales requeridos.</p> <p>Cada producto tiene atributos específicos e inalterables.</p>
Casos de uso relacionados	PQ-03, PQ-07, PQ-12, PQ-13, PQ-14

Ficha de Descripción de Caso de Uso	
ID	PQ-12
Nombre	Gestión de tipos de producto
Descripción	Permite al jugador acceder a distintos tipos de productos, clasificados por categorías con características únicas como costo, tiempo de fabricación y nivel de demanda. El sistema determina qué tipos de productos están disponibles en función del progreso del jugador, incentivando la planificación estratégica y la progresión en el juego.
Actores	Jugador / Sistema

Precondiciones	<p>El jugador ha avanzado lo suficiente como para desbloquear tipos de productos adicionales.</p> <p>El sistema ha cargado correctamente la información y requisitos de los productos disponibles.</p> <p>El jugador tiene acceso a la estación de producción.</p>
Postcondiciones	<p>El jugador ha fabricado un producto del tipo seleccionado y este se ha agregado a su inventario.</p> <p>El sistema ha actualizado los recursos utilizados y el estado del jugador.</p>
Evento activador	El jugador selecciona un tipo de producto desde el menú de fabricación o estación de producción.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador accede al menú de fabricación. 2. El sistema muestra los tipos de productos disponibles según el progreso del jugador. 3. El jugador selecciona un tipo de producto. 4. El sistema muestra los materiales y costos requeridos. 5. El jugador inicia la producción. 6. El sistema descuenta los materiales y agrega el producto terminado al inventario.
Flujos alternativos	<p>Si el tipo de producto aún no ha sido desbloqueado, el sistema muestra los requisitos necesarios.</p> <p>Si no hay suficientes materiales, el sistema impide iniciar la producción y sugiere adquirir más.</p> <p>Si el jugador intenta modificar atributos del producto, el sistema lo impide y muestra una notificación.</p>
Restricciones	<p>Solo se permiten tipos de productos desbloqueados según el avance del jugador.</p> <p>Los atributos de los productos no pueden ser modificados por el jugador.</p>

	La producción sólo puede iniciarse si se cumplen todos los requisitos del producto seleccionado.
Casos de uso relacionados	PQ-07, PQ-08, PQ-11, PQ-13, PQ-14

Ficha de Descripción de Caso de Uso	
ID	PQ-13
Nombre	Entrega de productos
Descripción	Permite a los jugadores entregar productos ya fabricados en una zona de salida designada dentro del nivel. La entrega marca el cierre del ciclo de producción y permite al jugador recibir recompensas, puntaje o avanzar en el nivel según la calidad y eficiencia de la entrega.
Actores	Jugador / Sistema
Precondiciones	El jugador ha completado la fabricación de al menos un producto. El producto se encuentra en el inventario y está marcado como terminado. El sistema ha activado la zona de entrega dentro del nivel.
Postcondiciones	El producto entregado ha sido eliminado del inventario. El jugador ha recibido una recompensa o puntaje en función de la calidad del producto entregado. El sistema ha registrado la entrega como parte del progreso del nivel.
Evento activador	El jugador transporta un producto terminado a la zona de salida e interactúa con ella.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador finaliza la producción de un producto. 2. El jugador transporta el producto hasta la zona de entrega dentro del nivel.

	<ol style="list-style-type: none"> 3. El sistema verifica que el producto está completo y apto para entrega. 4. El jugador confirma la acción de entrega. 5. El sistema procesa la entrega, elimina el producto del inventario y otorga una recompensa o puntaje. 6. El sistema actualiza el estado del jugador en ese nivel.
Flujos alternativos	<p>Si el producto no está terminado, el sistema bloquea la entrega y muestra un mensaje de advertencia.</p> <p>Si el producto no cumple con requisitos de calidad mínimos, el sistema aplica una penalización.</p> <p>Si el jugador intenta entregar un objeto inválido, el sistema lo rechaza y muestra una notificación.</p>
Restricciones	<p>Solo pueden entregarse productos completamente terminados.</p> <p>La entrega debe realizarse en la zona específica definida por el sistema.</p> <p>No se puede revertir una entrega una vez confirmada.</p>
Casos de uso relacionados	PQ-03, PQ-07, PQ-08, PQ-11, PQ-12

Ficha de Descripción de Caso de Uso	
ID	PQ-14
Nombre	Compra de materiales
Descripción	Permite al jugador adquirir materiales necesarios para la producción mediante el uso de recursos obtenidos dentro del juego. Los materiales comprados se almacenan en el inventario del jugador y están sujetos a disponibilidad, presupuesto y espacio. Esta funcionalidad es clave para mantener la cadena de producción activa.

Actores	Jugador / Sistema
Precondiciones	<p>El jugador se encuentra en una partida activa.</p> <p>El jugador tiene suficientes recursos para realizar la compra.</p> <p>Hay materiales disponibles en la tienda del juego.</p> <p>El inventario del jugador tiene espacio disponible.</p>
Postcondiciones	<p>Los materiales seleccionados han sido añadidos al inventario del jugador.</p> <p>El sistema ha descontado la cantidad correspondiente de recursos del jugador.</p> <p>Se ha actualizado el estado del inventario y presupuesto.</p>
Evento activador	El jugador accede a la tienda de materiales dentro del juego.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador accede al menú de la tienda de materiales. 2. El sistema muestra los materiales disponibles con sus respectivos precios. 3. El jugador selecciona los materiales deseados. 4. El sistema verifica los recursos y espacio disponible. 5. El jugador confirma la compra. 6. El sistema descuenta los recursos, añade los materiales al inventario y muestra una confirmación.
Flujos alternativos	<p>Si el jugador no tiene suficientes recursos, el sistema muestra un mensaje de advertencia y cancela la transacción.</p> <p>Si el inventario está lleno, el sistema impide la compra e informa al jugador.</p> <p>Si el jugador cancela la compra antes de confirmar, el</p>

	<p>sistema no realiza ninguna modificación.</p> <p>Si el jugador abandona la tienda sin comprar, no se registra ninguna transacción.</p>
Restricciones	<p>Solo se pueden comprar materiales si el jugador dispone de recursos suficientes.</p> <p>No se puede realizar la compra si el inventario está lleno.</p> <p>La compra está disponible únicamente en los niveles donde esta mecánica ha sido habilitada.</p>
Casos de uso relacionados	PQ-06, PQ-07, PQ-08, PQ-11, PQ-20

Ficha de Descripción de Caso de Uso	
ID	PQ-20
Nombre	Manejo de desperdicio
Descripción	<p>Permite al jugador reciclar materiales no utilizados o en exceso en su inventario antes de que sean procesados. A través del reciclaje, el jugador puede recuperar parcialmente el valor de los materiales, incentivando una gestión eficiente de recursos y evitando acumulación innecesaria en el inventario.</p>
Actores	Jugador / Sistema
Precondiciones	<p>El jugador posee materiales en su inventario que no han sido utilizados en producción.</p> <p>El sistema ha identificado que los materiales seleccionados son reciclables.</p>
Postcondiciones	<p>El material reciclado ha sido eliminado del inventario.</p> <p>El jugador ha recibido una cantidad parcial de recursos (dinero o puntos) como reembolso.</p> <p>Se ha actualizado el inventario y el sistema económico del</p>

	juego.
Evento activador	El jugador accede al inventario y selecciona la opción de reciclar materiales.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador entra al inventario y selecciona uno o varios materiales en exceso. 2. El sistema verifica la reciclabilidad de los materiales y muestra el porcentaje de recuperación. 3. El jugador confirma la acción de reciclaje. 4. El sistema elimina los materiales seleccionados del inventario. 5. El sistema otorga al jugador una cantidad parcial de recursos, dependiendo del material.
Flujos alternativos	<p>Si el jugador intenta reciclar materiales que ya fueron procesados, el sistema muestra una notificación de error.</p> <p>Si el jugador no posee materiales reciclables, la opción de reciclaje se desactiva.</p> <p>Si el jugador cancela la acción, no se realizan cambios en el inventario.</p>
Restricciones	<p>Solo se pueden reciclar materiales no utilizados.</p> <p>No se puede reciclar productos terminados o materiales ya procesados.</p> <p>El reembolso otorgado es parcial (entre el 30% y el 50% del costo original).</p>
Casos de uso relacionados	PQ-06, PQ-07, PQ-08, PQ-11, PQ-14

4.2. Módulo de Jugabilidad e Interacción

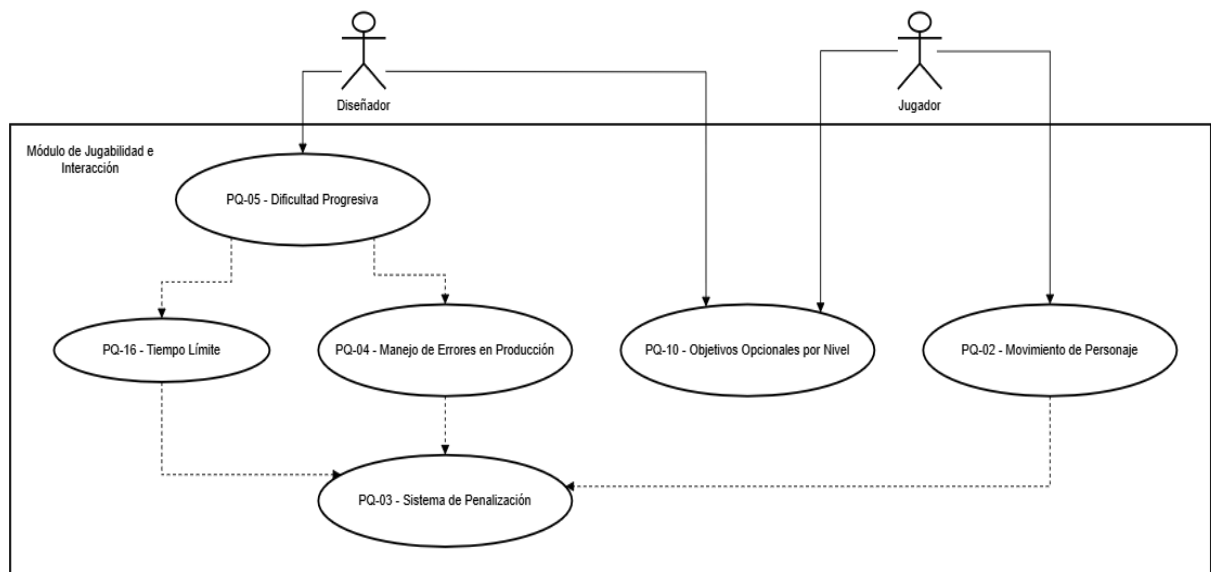


Figura 02: Diagrama de Casos de Uso para el módulo de jugabilidad e interacción

Ficha de Descripción de Caso de Uso	
ID	PQ-02
Nombre	Movimiento de Personaje
Descripción	El jugador debe ser capaz de moverse por el entorno 2D del respectivo nivel mediante el uso de teclas asignadas.
Actores	Jugador
Precondiciones	El jugador ha iniciado una partida.
Postcondiciones	La posición del jugador debe ser actualizada correctamente, ubicándose en la posición deseada al finalizar el movimiento.
Evento activador	El jugador presiona las teclas asignadas para el movimiento.
Flujo normal	<ol style="list-style-type: none"> 1. Se presionan las teclas de movimiento definidas. 2. El sistema actualiza la posición del personaje y muestra interacciones disponibles.

Flujos alternativos	Colisión con obstáculos: El personaje se detiene y reproduce una animación de impacto.
Restricciones	El movimiento debe estar limitado a áreas designadas por el juego.
Casos de uso relacionados	N/A

Ficha de Descripción de Caso de Uso	
ID	PQ-03
Nombre	Sistema de Penalización
Descripción	El sistema aplica sanciones de manera automática cuando el jugador fabrica productos incorrectos o mezcla materiales erróneos.
Actores	Sistema
Precondiciones	El jugador comete un error en producción
Postcondiciones	Se aplican las sanciones necesarias y el juego continúa.
Evento activador	El jugador comete un error de producción.
Flujo normal	<ol style="list-style-type: none"> 1. Se detecta un error por parte del jugador. 2. Se restan los recursos configurados, ya sea materiales o tiempo restante. 3. Se muestra una notificación explicando el error y la sanción.
Flujos alternativos	N/A
Restricciones	Las penalizaciones se calculan en tiempo real.
Casos de uso relacionados	N/A

Ficha de Descripción de Caso de Uso
--

ID	PQ-04
Nombre	Manejo de Errores en Producción
Descripción	El jugador podrá utilizar una estación especial para deshacer hasta tres errores por nivel y recuperar el 50% de los materiales gastados.
Actores	Jugador
Precondiciones	El jugador ha cometido un error y tiene reversiones disponibles.
Postcondiciones	El jugador tiene los materiales recuperados en el inventario. El jugador no posee el producto destruido.
Evento activador	El jugador utiliza la estación de destruir.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador crea un producto no deseado 2. El jugador se mueve hacia la estación correspondiente y selecciona el producto por destruir 3. El sistema devuelve el 50% del costo en materiales del producto correspondiente
Flujos alternativos	Sin usos disponibles: El sistema muestra un mensaje: "Límite de reversiones alcanzado".
Restricciones	No se recupera el tiempo perdido en el proceso La estación tiene un uso máximo de tres veces por nivel
Casos de uso relacionados	N/A

Ficha de Descripción de Caso de Uso	
ID	PQ-05

Nombre	Dificultad Progresiva
Descripción	El jugador será capaz de jugar niveles cuya dificultad aumenta progresivamente.
Actores	Jugador
Precondiciones	El jugador ha completado el tutorial inicial y se encuentra jugando los niveles preconfigurados.
Postcondiciones	Los niveles posteriores incluyen mecánicas avanzadas y restricciones.
Evento activador	El jugador termina un nivel y selecciona la opción de continuar al siguiente.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador termina un nivel 2. Se coloca al jugador en el nivel siguiente 3. Se le indica al jugador el presupuesto, materiales y productos por crear 4. El jugador inicia la partida
Flujos alternativos	Puntuación insuficiente: El jugador no logra terminar el nivel, no se le teletransporta al siguiente.
Restricciones	La dificultad aumenta en cada nivel.
Casos de uso relacionados	PQ-01

Ficha de Descripción de Caso de Uso	
ID	PQ-10
Nombre	Objetivos opcionales por nivel
Descripción	Permite al jugador completar objetivos adicionales no obligatorios dentro de cada nivel. Estos objetivos opcionales brindan recompensas extra y aumentan el

	puntaje final, incentivando una experiencia más desafiante y estratégica sin interferir con el cumplimiento básico del nivel.
Actores	Jugador / Sistema
Precondiciones	<p>El jugador ha iniciado un nivel preconfigurado.</p> <p>El sistema ha cargado los objetivos principales y opcionales del nivel.</p> <p>Los objetivos opcionales están claramente definidos y visibles para el jugador.</p>
Postcondiciones	<p>El jugador ha cumplido o ignorado uno o más objetivos opcionales.</p> <p>El sistema ha evaluado el cumplimiento de estos objetivos y ajustado el puntaje final.</p> <p>El nivel ha sido completado con o sin dichos objetivos.</p>
Evento activador	El jugador comienza un nivel y visualiza los objetivos opcionales presentados por el sistema.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador inicia un nivel y el sistema muestra los objetivos principales y opcionales. 2. El jugador avanza en el nivel cumpliendo los objetivos principales y decide intentar los opcionales. 3. Al cumplir un objetivo opcional, el sistema lo registra y otorga una bonificación interna. 4. Al finalizar el nivel, el sistema considera el cumplimiento de estos objetivos al calcular el puntaje final.
Flujos alternativos	<p>Si el jugador ignora los objetivos opcionales, el sistema no aplica penalización y evalúa solo los objetivos principales.</p> <p>Si el jugador cumple parcialmente los objetivos</p>

	<p>opcionales, el sistema aplica una bonificación proporcional.</p> <p>Si un objetivo opcional es fallado, el sistema lo marca como no completado pero permite continuar el nivel.</p>
Restricciones	<p>Los objetivos opcionales no pueden sustituir el cumplimiento de los objetivos principales.</p> <p>El sistema no debe impedir avanzar al siguiente nivel si los objetivos opcionales no se cumplen.</p> <p>Los objetivos deben estar claramente diferenciados entre principales y opcionales.</p>
Casos de uso relacionados	PQ-01, PQ-06, PQ-07, PQ-08, PQ-09

Ficha de Descripción de Caso de Uso	
ID	PQ-16
Nombre	Tiempo límite por nivel
Descripción	<p>Establece un tiempo máximo para completar cada nivel. El sistema inicia un temporizador al comenzar la partida, y el jugador debe completar todos los objetivos dentro del tiempo asignado. Una vez que el tiempo se agota, el nivel finaliza automáticamente y se procede a la evaluación del desempeño.</p>
Actores	Jugador / Sistema
Precondiciones	<p>El jugador ha iniciado un nivel preconfigurado.</p> <p>El sistema ha cargado correctamente el tiempo límite asignado para dicho nivel.</p>
Postcondiciones	<p>El nivel ha finalizado por cumplimiento de los objetivos o agotamiento del tiempo.</p> <p>El sistema ha registrado el estado final del nivel, incluyendo el tiempo usado y el puntaje alcanzado.</p>

	El jugador puede revisar su desempeño y decidir si repetir o continuar.
Evento activador	El jugador inicia un nuevo nivel de juego.
Flujo normal	<p>El jugador inicia el nivel.</p> <p>El sistema activa el temporizador con el tiempo límite definido.</p> <p>El jugador realiza sus acciones dentro del nivel.</p> <p>El sistema muestra el tiempo restante de forma visible.</p> <p>Si el jugador finaliza los objetivos antes de que el tiempo termine, el sistema detiene el cronómetro y evalúa el desempeño.</p> <p>Si el tiempo se agota, el sistema finaliza el nivel automáticamente y procede con la evaluación.</p>
Flujos alternativos	<p>Si el jugador pausa la partida, el temporizador se congela y se reanuda al continuar.</p> <p>Si el jugador guarda y sale, al reanudar se restablece el tiempo restante exacto.</p> <p>Si el jugador reinicia el nivel, el temporizador se reinicia con el tiempo completo.</p>
Restricciones	<p>El jugador no puede extender el tiempo asignado.</p> <p>El temporizador debe ser visible en pantalla durante toda la partida.</p> <p>Si el tiempo llega a cero, el sistema debe finalizar el nivel sin posibilidad de continuar.</p>
Casos de uso relacionados	PQ-01, PQ-06, PQ-08, PQ-10, PQ-17

4.3. Módulo de Evaluación y Feedback

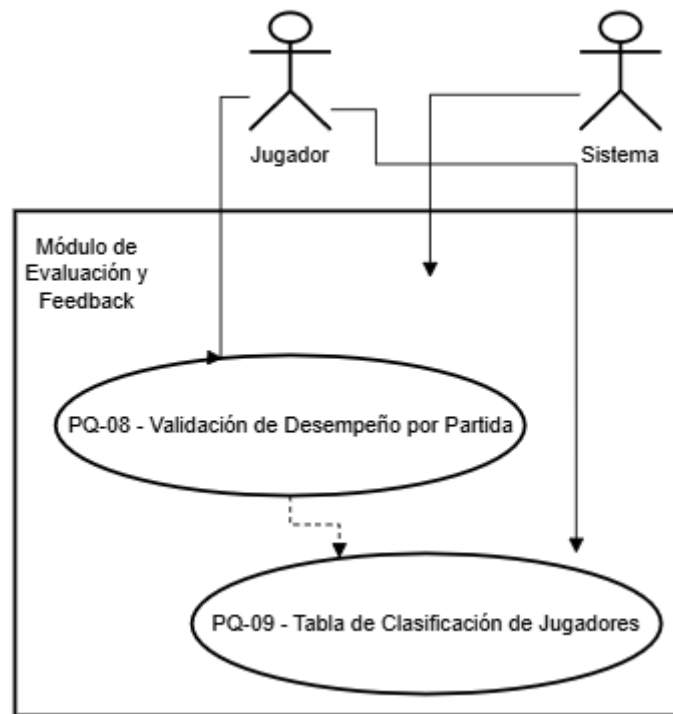


Figura 03: Diagrama de Casos de Uso para el módulo de evaluación y feedback

Ficha de Descripción de Caso de Uso	
ID	PQ-08
Nombre	Validación de desempeño por partida
Descripción	Permite al sistema evaluar automáticamente el desempeño del jugador al finalizar un nivel. La evaluación se basa en métricas como el uso eficiente del presupuesto, la correcta gestión del inventario y el cumplimiento de los objetivos del nivel. El resultado determina si el jugador puede avanzar o debe repetir el nivel.
Actores	Jugador / Sistema
Precondiciones	El jugador ha iniciado y finalizado un nivel preconfigurado. El sistema ha registrado las acciones del jugador durante la partida.

	Los objetivos del nivel están predefinidos y cargados correctamente.
Postcondiciones	<p>El jugador ha recibido un puntaje del 0 al 10 que refleja su desempeño.</p> <p>El sistema ha determinado si el jugador puede avanzar al siguiente nivel.</p> <p>El puntaje ha sido almacenado para uso en rankings y seguimiento de progreso.</p>
Evento activador	El jugador finaliza un nivel (ya sea cumpliendo los objetivos o agotando el tiempo).
Flujo normal	<ol style="list-style-type: none"> 1. El jugador inicia un nivel y comienza a realizar acciones. 2. El sistema monitorea y registra el uso del presupuesto, materiales y cumplimiento de objetivos. 3. Al finalizar el nivel, el sistema compara el rendimiento del jugador con los objetivos preestablecidos. 4. Se genera un puntaje de desempeño y se presenta al jugador. 5. El sistema determina si el jugador puede avanzar o debe repetir el nivel.
Flujos alternativos	<p>Si el jugador reinicia el nivel, el sistema reinicia también el cálculo de desempeño.</p> <p>Si el jugador abandona el nivel antes de terminarlo, no se genera evaluación.</p>
Restricciones	<p>No se puede calcular el desempeño si el nivel ha sido reiniciado o abandonado.</p> <p>El sistema debe aplicar el mismo criterio de evaluación en todos los niveles para garantizar equidad.</p>

	El puntaje solo puede calcularse si se han registrado todas las métricas necesarias.
Casos de uso relacionados	PQ-01, PQ-06, PQ-07, PQ-09, PQ-10

Ficha de Descripción de Caso de Uso	
ID	PQ-09
Nombre	Tabla de clasificación de jugadores
Descripción	Permite al sistema mostrar una tabla de clasificación que organiza a los jugadores según su puntaje de desempeño en niveles específicos. Los jugadores pueden consultar el ranking por nivel, visualizar los mejores puntajes y acceder a detalles adicionales de cada partida registrada.
Actores	Jugador / Sistema
Precondiciones	Al menos un jugador ha completado un nivel con éxito. El sistema ha registrado correctamente los puntajes de desempeño de las partidas. El jugador ha accedido al menú principal del juego.
Postcondiciones	El jugador visualiza una tabla ordenada por puntaje descendente para un nivel específico. El sistema permite consultar detalles de partidas individuales según el puntaje mostrado. El sistema mantiene los rankings actualizados tras cada nueva partida registrada.
Evento activador	El jugador selecciona la opción de “Tabla de clasificación” en el menú principal del juego.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador accede al menú principal y selecciona la opción de tabla de clasificación. 2. El sistema solicita al jugador elegir un nivel para

	<p>visualizar los puntajes.</p> <ol style="list-style-type: none"> 3. El sistema recopila los datos de desempeño de todos los jugadores que han completado ese nivel. 4. El sistema muestra una tabla con los puntajes ordenados de mayor a menor. 5. El jugador puede seleccionar un nombre de usuario para ver los detalles de su partida.
Flujos alternativos	<p>Si el jugador cambia de nivel, el sistema actualiza la tabla con los datos del nivel seleccionado.</p> <p>Si el jugador aún no ha completado ningún nivel, el sistema muestra un mensaje indicando que no hay datos disponibles.</p> <p>Si el jugador regresa al menú principal, el sistema cierra la vista del ranking.</p>
Restricciones	<p>La tabla solo muestra resultados de jugadores que hayan completado un nivel.</p> <p>El sistema organiza los datos únicamente por nivel individual, no hay un ranking global.</p> <p>No se pueden modificar manualmente los resultados mostrados en la tabla.</p>
Casos de uso relacionados	PQ-01, PQ-08, PQ-10, PQ-19

4.4. Módulo de Experiencia de Usuario

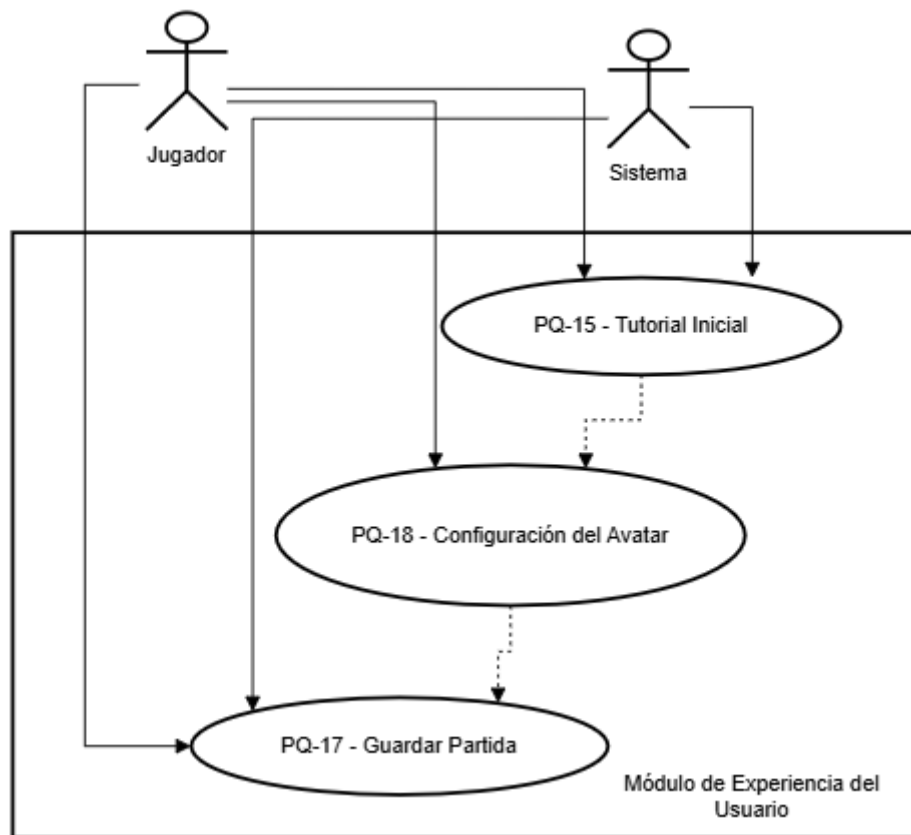


Figura 04: Diagrama de Casos de Uso para el módulo de experiencia de usuario

Ficha de Descripción de Caso de Uso	
ID	PQ-15
Nombre	Tutorial Inicial
Descripción	Permite al sistema presentar un tutorial interactivo la primera vez que un jugador accede al juego. El tutorial guía paso a paso al usuario a través de las mecánicas básicas como la compra de materiales, gestión del presupuesto, uso del inventario, producción y entrega de productos. Su objetivo es asegurar que el jugador comprenda los fundamentos antes de comenzar a jugar niveles completos.
Actores	Jugador / Sistema

Precondiciones	<p>El jugador ha iniciado el juego por primera vez.</p> <p>No existe un progreso previo guardado en la cuenta del jugador.</p> <p>El sistema ha cargado correctamente los elementos necesarios del tutorial.</p>
Postcondiciones	<p>El jugador ha completado o saltado el tutorial inicial.</p> <p>El sistema ha registrado que el tutorial fue completado u omitido, y no lo mostrará automáticamente en futuras sesiones.</p> <p>El jugador puede iniciar el juego conociendo las mecánicas básicas.</p>
Evento activador	<p>El jugador inicia el juego sin registros previos en el sistema.</p>
Flujo normal	<ol style="list-style-type: none"> 1. El jugador inicia sesión por primera vez. 2. El sistema detecta que es una nueva sesión y activa el tutorial automáticamente. 3. El sistema presenta las instrucciones básicas y guía al jugador paso a paso. 4. El jugador realiza las acciones solicitadas (comprar materiales, fabricar un producto, usar el presupuesto, etc.). 5. El sistema valida cada acción antes de permitir avanzar al siguiente paso. 6. Al finalizar el recorrido, se muestra un mensaje indicando que el tutorial ha terminado. 7. El sistema guarda el progreso y desactiva la repetición automática del tutorial.
Flujos alternativos	<p>Si el jugador elige omitir el tutorial, el sistema registra la omisión y lo lleva al menú principal.</p> <p>Si el jugador no completa una sección del tutorial, el</p>

	<p>sistema bloquea el avance hasta que cumpla la acción requerida.</p> <p>Si el jugador cierra el juego durante el tutorial, el sistema guarda el progreso y lo reanuda al reiniciar la sesión.</p>
Restricciones	<p>El tutorial solo se activa automáticamente en la primera sesión.</p> <p>El jugador no puede avanzar al siguiente paso sin realizar correctamente la acción solicitada.</p> <p>No se puede repetir el tutorial automáticamente después de completarlo (solo de forma manual desde el menú).</p>
Casos de uso relacionados	PQ-01, PQ-06, PQ-07, PQ-11, PQ-14, PQ-19

Ficha de Descripción de Caso de Uso	
ID	PQ-17
Nombre	Guardado de progreso
Descripción	<p>Permite al jugador guardar su progreso en cualquier momento de la partida. El sistema registra el estado del inventario, presupuesto, nivel actual, tiempo restante y demás variables relevantes, permitiendo retomar la sesión desde el mismo punto en el futuro.</p>
Actores	Jugador / Sistema
Precondiciones	<p>El jugador ha iniciado sesión en el juego.</p> <p>El jugador se encuentra en una partida activa.</p> <p>El sistema tiene acceso a los recursos de almacenamiento necesarios.</p>
Postcondiciones	<p>El progreso actual del jugador ha sido guardado exitosamente.</p> <p>El sistema ha almacenado todos los datos clave del estado</p>

	<p>de juego.</p> <p>El jugador puede reanudar la partida desde el punto exacto en que la guardó.</p>
Evento activador	El jugador selecciona manualmente la opción de “Guardar progreso” desde el menú del juego.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador accede al menú de pausa o configuración. 2. Selecciona la opción “Guardar progreso”. 3. El sistema recopila el estado actual del juego (nivel, inventario, presupuesto, tiempo, etc.). 4. El sistema guarda los datos en un archivo o base de datos asociado al perfil del jugador. 5. El sistema confirma el guardado exitoso y permite continuar o salir del juego.
Flujos alternativos	<p>Si ocurre un error durante el guardado, el sistema muestra una advertencia e intenta guardarlo nuevamente.</p> <p>Si el jugador apaga el dispositivo sin guardarlo manualmente, el sistema intenta realizar un guardado automático del último estado disponible.</p> <p>Si el jugador guarda y luego reinicia el juego, se ofrece la opción de continuar desde el último punto guardado.</p>
Restricciones	<p>El guardado no debe realizarse durante eventos no plausibles o fases críticas del nivel.</p> <p>Cada perfil de jugador debe tener un único estado de guardado activo por nivel.</p> <p>El guardado automático no debe sobrescribir manualmente el progreso guardado por el jugador sin confirmación.</p>
Casos de uso relacionados	PQ-06, PQ-07, PQ-15, PQ-16, PQ-19

Ficha de Descripción de Caso de Uso	
ID	PQ-18
Nombre	Personalización de avatar
Descripción	Permite al jugador personalizar su avatar antes de comenzar una partida. La personalización incluye opciones estéticas como colores, atuendos y accesorios prediseñados. Estas configuraciones no afectan el rendimiento del jugador, pero aportan una experiencia más inmersiva y personal.
Actores	Jugador / Sistema
Precondiciones	El jugador ha iniciado sesión o ha creado un perfil. El sistema ha cargado correctamente las opciones de personalización disponibles.
Postcondiciones	El jugador ha seleccionado una combinación visual para su avatar. El sistema ha almacenado la configuración para que se aplique en las partidas posteriores. La apariencia personalizada se visualiza correctamente en el juego.
Evento activador	El jugador accede al menú de personalización desde el menú principal o al iniciar una partida nueva.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador accede al menú de personalización de avatar. 2. El sistema presenta una serie de opciones visuales organizadas por categoría (colores, ropa, accesorios, etc.). 3. El jugador selecciona las opciones que desea aplicar. 4. El sistema actualiza en tiempo real la vista previa

	<p>del avatar.</p> <p>5. El jugador guarda los cambios y el sistema almacena la configuración en su perfil.</p> <p>6. El avatar personalizado se aplica en las partidas siguientes.</p>
Flujos alternativos	<p>Si el jugador cancela los cambios, el sistema restaura la última configuración guardada.</p> <p>Si el jugador no realiza cambios, el sistema mantiene la apariencia predeterminada.</p> <p>Si se produce un error al guardar la configuración, el sistema informa y sugiere reintentar.</p>
Restricciones	<p>La personalización es únicamente visual y no modifica atributos del juego.</p> <p>No se permite combinar elementos de distintas categorías si generan conflicto gráfico.</p> <p>Las opciones disponibles deben ser predefinidas por el sistema, sin carga de contenido externo.</p>
Casos de uso relacionados	PQ-01, PQ-17, PQ-19

4.5. Módulo de Gestión de Usuarios

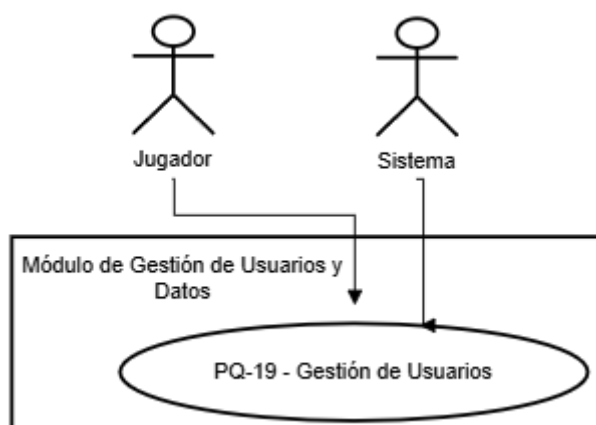


Figura 05: Diagrama de Casos de Uso para el módulo de gestión de usuarios

Ficha de Descripción de Caso de Uso

ID	PQ-19
Nombre	Gestión de usuarios
Descripción	Permite al sistema gestionar múltiples perfiles de usuario, asegurando que cada jugador tenga un historial de progreso, configuraciones y estadísticas personalizadas. Esto permite a varios jugadores utilizar el mismo dispositivo sin que se altere su experiencia individual ni sus avances.
Actores	Jugador / Sistema
Precondiciones	El sistema se encuentra activo y operativo. Existen perfiles creados o el jugador tiene la opción de crear uno nuevo.
Postcondiciones	El jugador ha accedido a su perfil personal. El sistema ha cargado el historial de progreso, configuraciones y datos asociados. Los cambios realizados durante la sesión se guardan únicamente en el perfil seleccionado.
Evento activador	El jugador inicia el juego o accede a la opción de cambiar o crear perfil.
Flujo normal	<ol style="list-style-type: none"> 1. El jugador inicia el juego. 2. El sistema muestra los perfiles existentes y la opción de crear uno nuevo. 3. El jugador selecciona o crea su perfil. 4. El sistema carga el progreso, configuración, avatar y estadísticas asociadas a ese perfil. 5. El jugador comienza a jugar bajo su identidad única. 6. Al cerrar sesión o salir del juego, el sistema guarda los cambios en el perfil correspondiente.

Flujos alternativos	<p>Si el jugador intenta crear un perfil con un nombre ya existente, el sistema solicita uno diferente.</p> <p>Si se corrompe un perfil, el sistema ofrece restaurar desde una copia de seguridad si está disponible.</p> <p>Si el jugador desea eliminar un perfil, el sistema solicita confirmación antes de proceder.</p>
Restricciones	<p>Cada perfil debe tener un identificador único (por ejemplo, un nombre no repetido).</p> <p>No se puede acceder a los datos de otros perfiles sin autorización.</p> <p>El sistema debe mantener la integridad y separación de datos entre perfiles.</p>
Casos de uso relacionados	PQ-08, PQ-09, PQ-15, PQ-17, PQ-18

5. Vista Lógica

Esta sección describe la arquitectura lógica del sistema y ProfitQuest, un juego educativo de simulación de producción. Se detallan los componentes principales, su organización interna y el comportamiento ante funciones clave del sistema.

5.1. Vista General

La arquitectura lógica se basa en un enfoque modular utilizando el motor de videojuegos Godot. El sistema se divide en los siguientes módulos principales:

- **Nivel:** Contiene las escenas, estaciones de trabajo, objetivos, inventario y lógica de evaluación.
- **Jugador:** Representa al avatar del usuario, incluyendo su movimiento, acciones y personalización.
- **Sistema de Economía:** Gestiona el presupuesto, inventario, productos y penalizaciones.

- **Interfaz de Usuario (UI):** Maneja menús, pantalla de juego, paneles informativos, tablas de clasificación y configuraciones.
- **Motor de Evaluación:** Evalúa el desempeño con base en métricas definidas para cada nivel.

5.2. Componentes de Diseño Específicos Relevantes

5.2.1. Nivel

- Escenario: Carga y organiza los elementos visuales del nivel.
- EstacionesTrabajo: Controlan el proceso de fabricación.
- Objetivos: Define los objetivos principales y opcionales.

5.2.2. Jugador

- Avatar: Controla el movimiento y las colisiones.
- ControladorJugador: Procesa entradas del teclado y acciones.

5.2.3. Sistema de Economía

- PresupuestoManager: Controla ingresos y egresos.
- InventarioManager: Administra materiales y productos.
- PenalizacionManager: Calcula penalizaciones en tiempo real.

5.2.4. UI

- MenuPrincipal, PantallaJuego, PantallaEvaluacion, TablaClasificacion, ConfiguracionAvatar.

5.2.5. Evaluador

- DesempenoCalculator: Calcula el puntaje del jugador.
- GestorRecompensas: Asigna bonificaciones o restricciones.

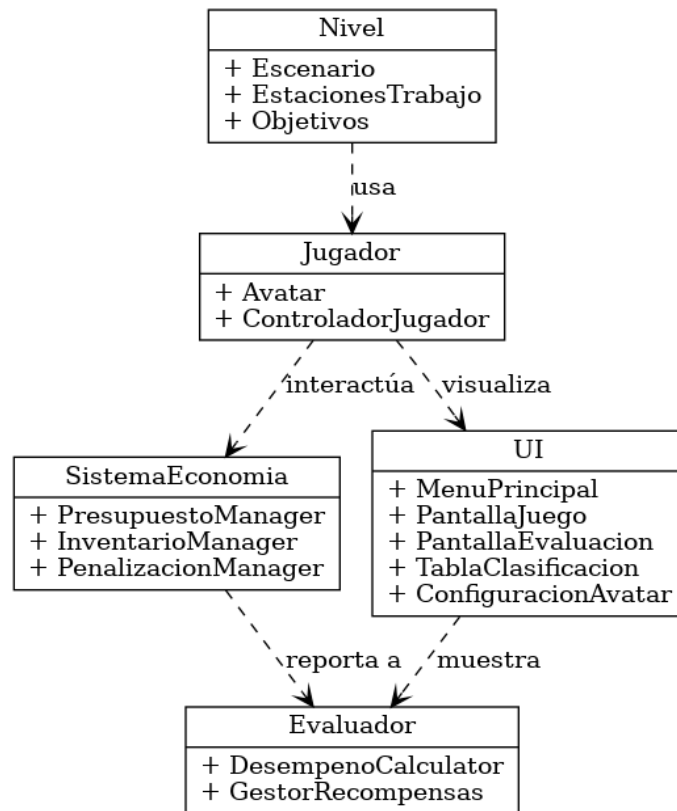


Figura 06: Arquitectura General de la Vista Lógica

5.3. Casos de Uso

5.3.1. Simulación de Producción

5.3.1.1. Diseño de Niveles Preconfigurados

Los jugadores deben poder acceder y jugar niveles estáticos preconfigurados que fueron creados durante la fase de desarrollo. Cada nivel incluye estaciones de trabajo, recursos iniciales, materiales disponibles, objetivos y restricciones fijas. Los objetivos incluyen mantener los gastos dentro del presupuesto, calcular los costos de la operación y manejo general y eficiente de recursos y costos en el proceso de producción.

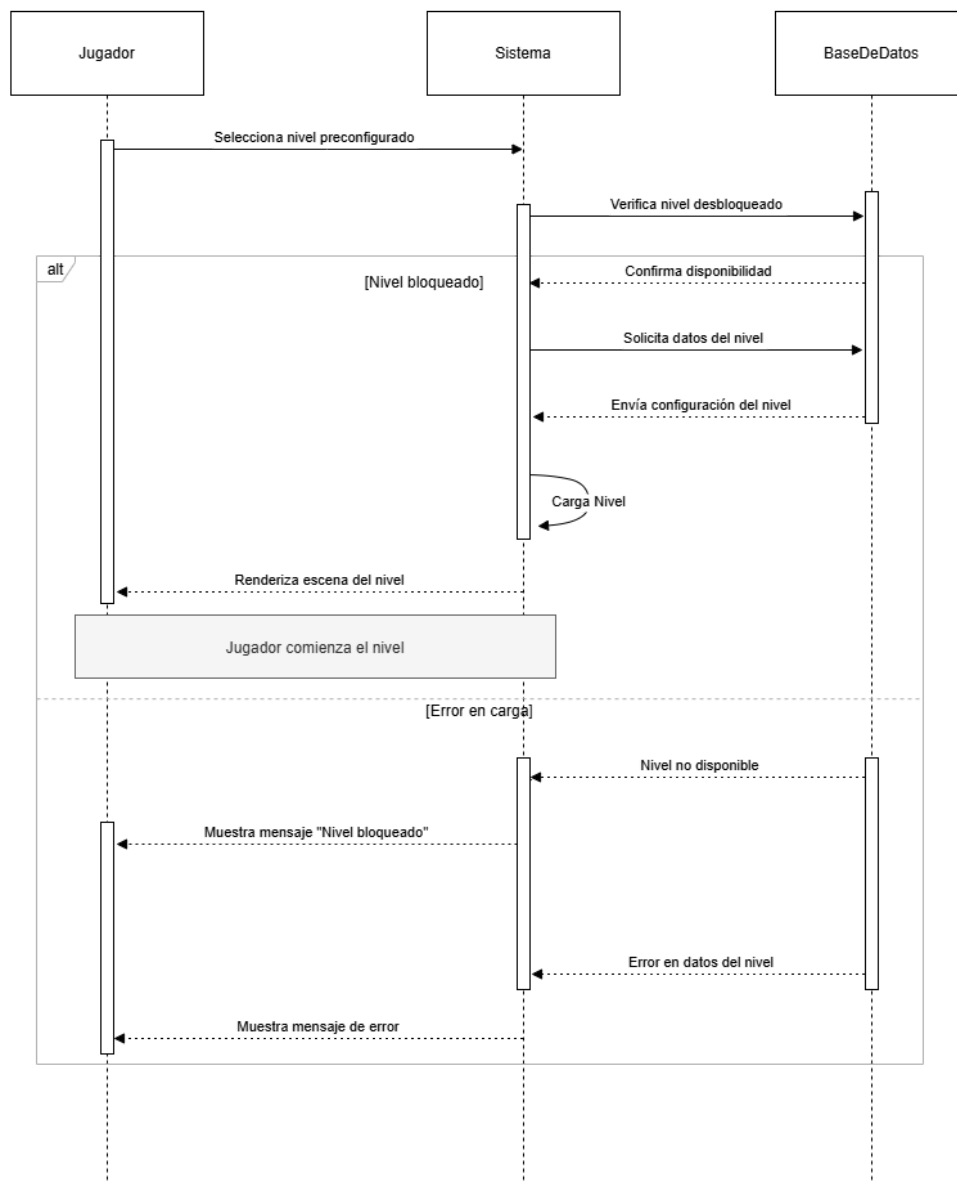


Figura 07: Diagrama de Secuencia de Caso de Uso PQ-01

5.3.1.2. Gestión de Presupuesto

El jugador recibe un monto fijo de recursos al inicio de cada nivel, este se usa como recurso para completar los diferentes objetivos del nivel.

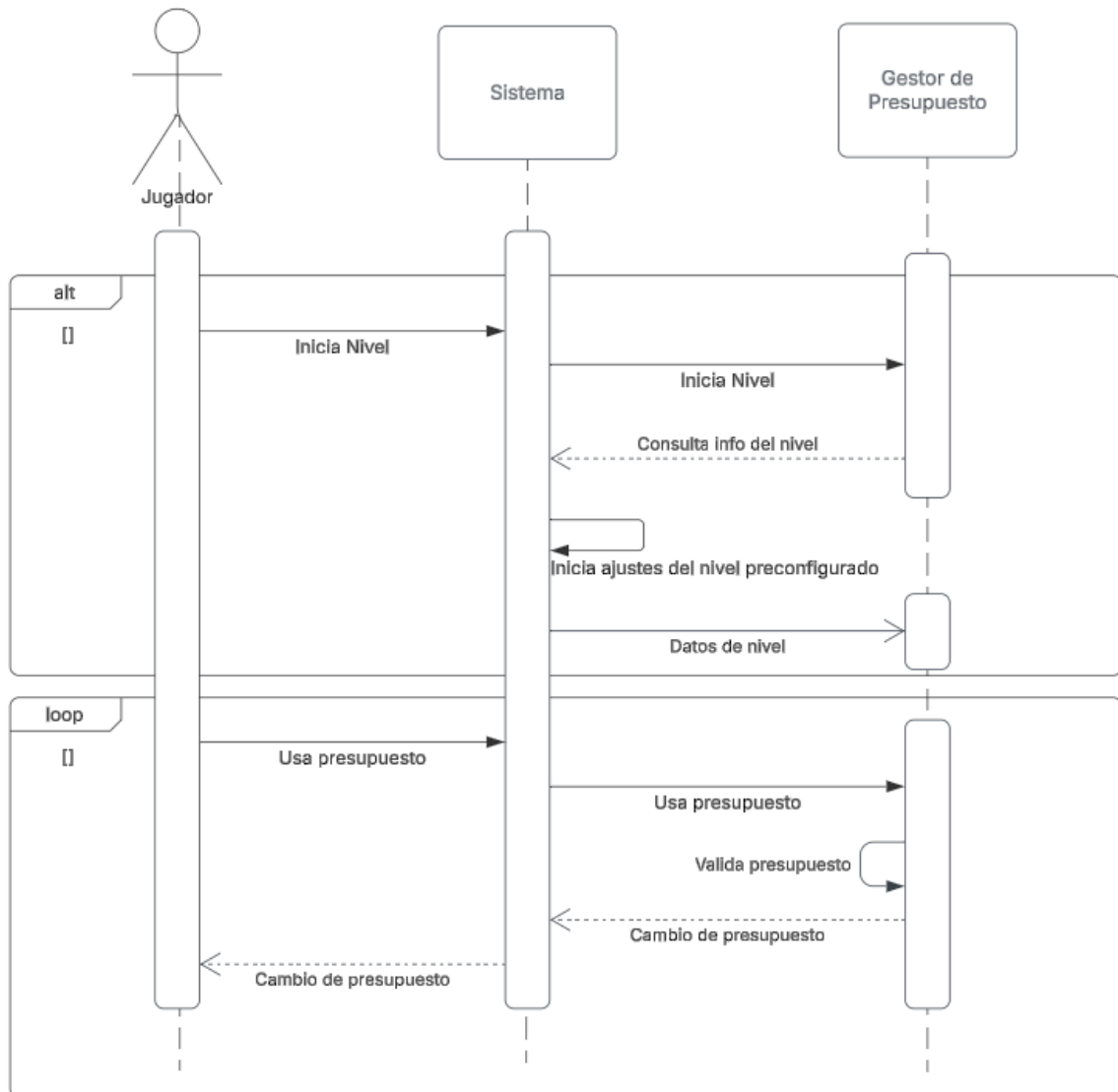


Figura 08: Diagrama de Secuencia de Caso de Uso PQ-06

5.3.1.3. Gestión de Inventario

El sistema tiene un inventario predefinido de recursos y productos que se utilizan para completar los diferentes objetivos del juego, el inventario debe de ser responsivo y claro.

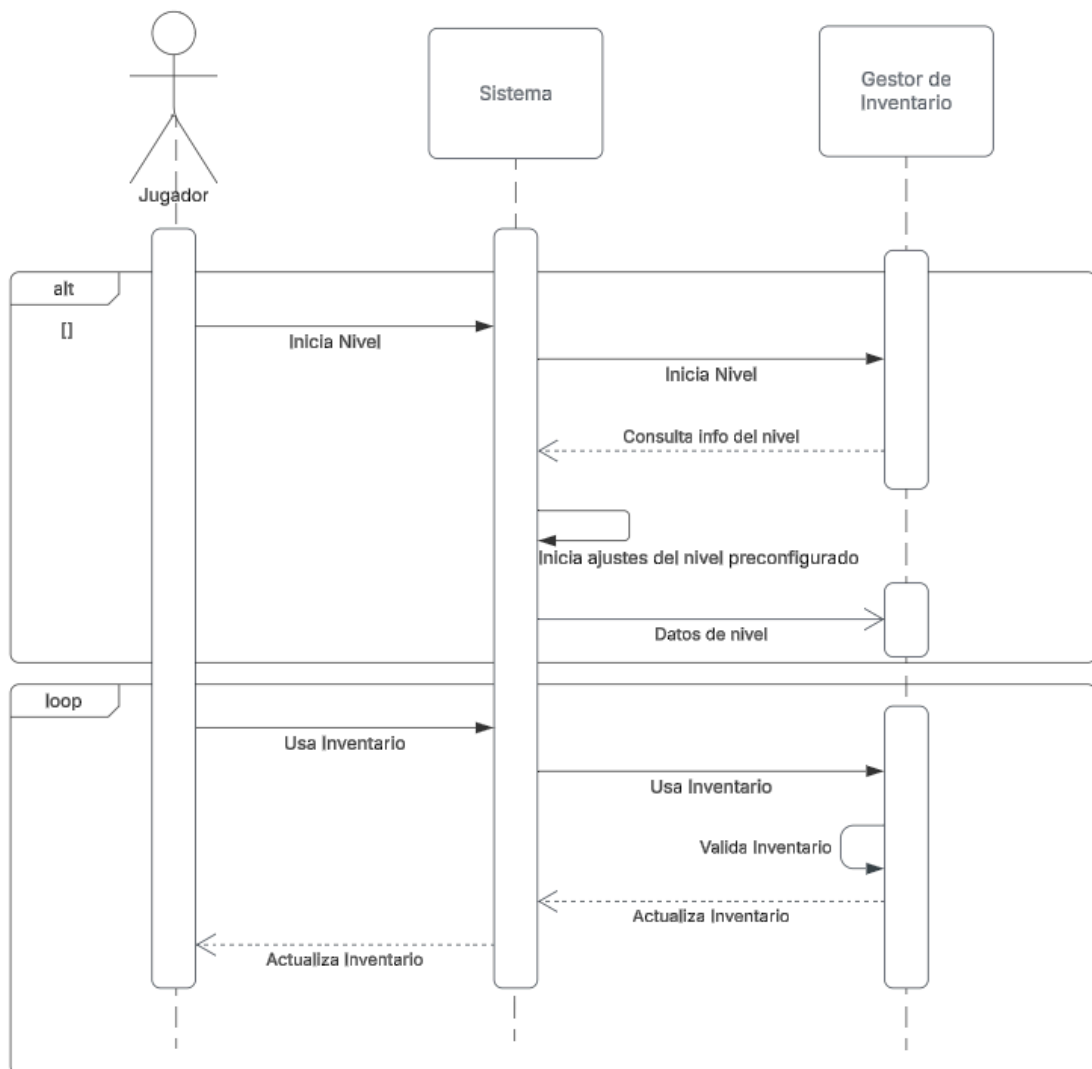


Figura 09: Diagrama de Secuencia de Caso de Uso PQ-07

5.3.1.4. Gestión de Productos

El sistema cuenta con un catálogo predefinido de productos que los jugadores pueden fabricar y gestionar a lo largo del juego. Cada producto posee atributos específicos, como costo de producción, tiempo de fabricación y demanda, estos influyen directamente en la estrategia del jugador.

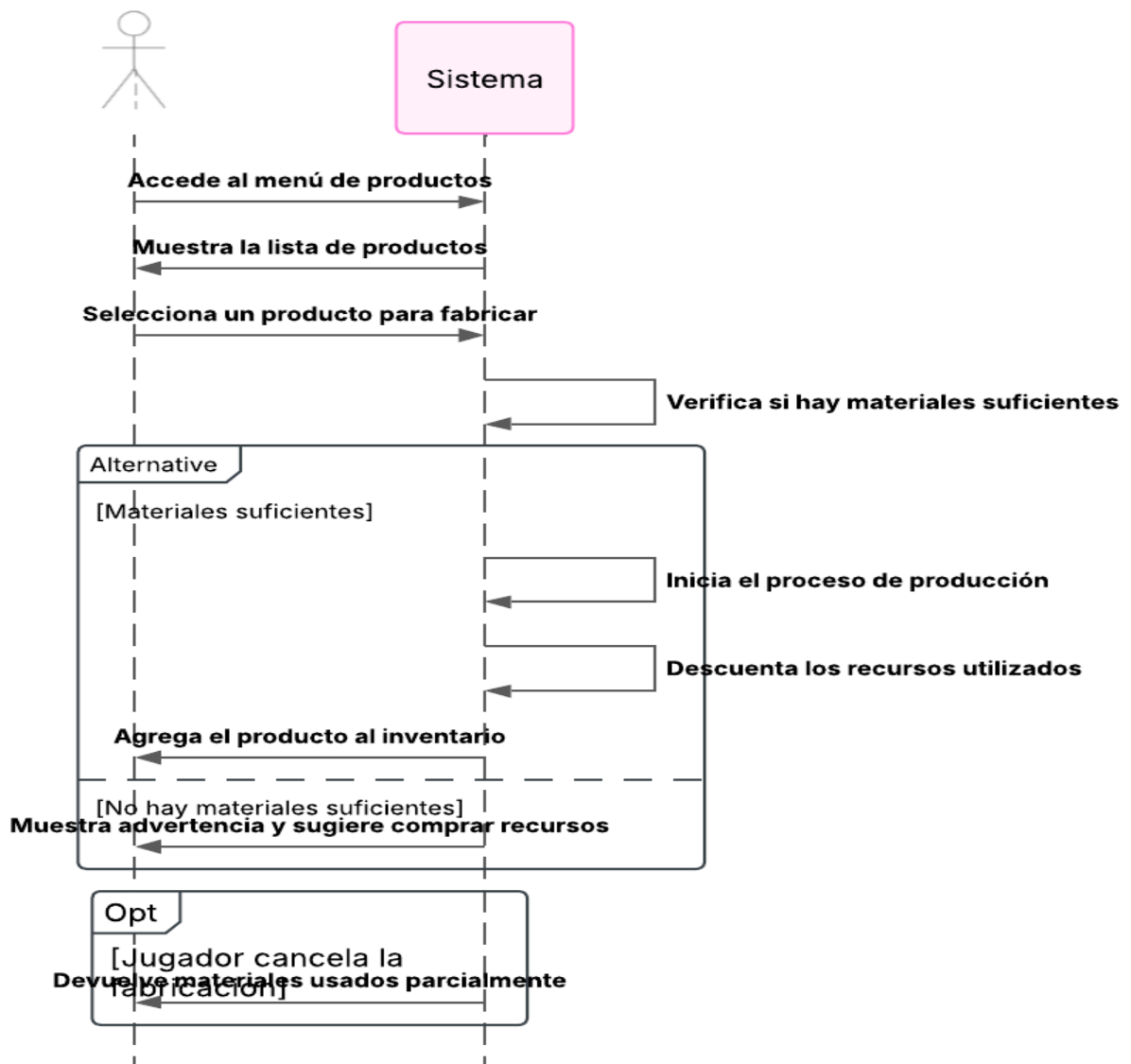


Figura 10: Diagrama de Secuencia de Caso de Uso PQ-11

5.3.1.5. Gestión de Tipos de Productos

La gestión se realiza de manera ordenada y clara: el sistema verifica el desbloqueo y la disponibilidad de recursos antes de permitir la fabricación, asegurando que el proceso se refleje correctamente en el inventario y la puntuación del jugador.

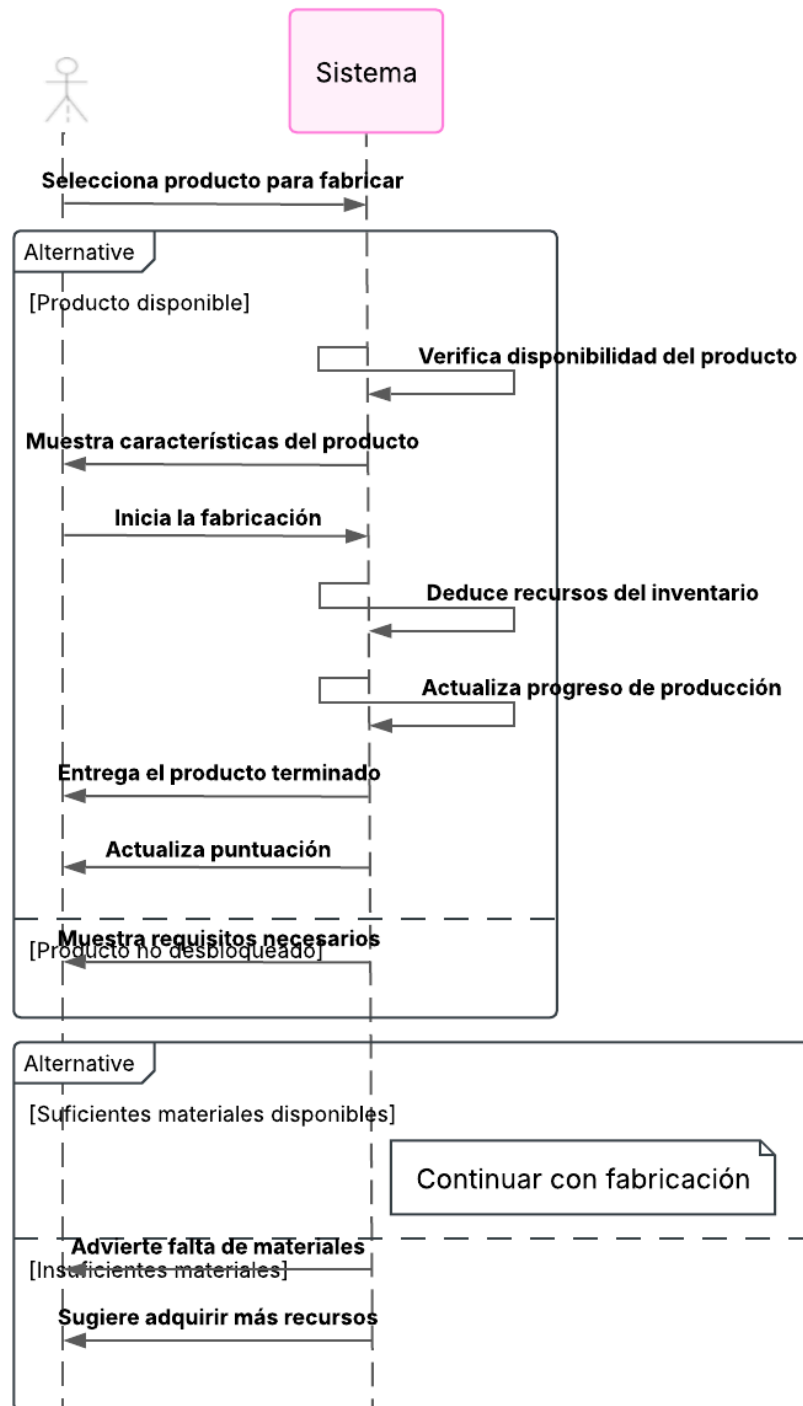


Figura 11: Diagrama de Secuencia de Caso de Uso PQ-12

5.3.1.6. Entrega de Productos

El sistema permite que los productos terminados sean entregados en una zona de salida designada, marcando la finalización del ciclo de producción. Una vez que el jugador transporta el producto y confirma la entrega, el sistema verifica que el producto esté completamente fabricado y cumpla con los requisitos de calidad. Si se cumplen las condiciones, el sistema procesa la entrega, actualiza el inventario eliminando el producto entregado y otorga la recompensa o puntuación correspondiente al desempeño del jugador.

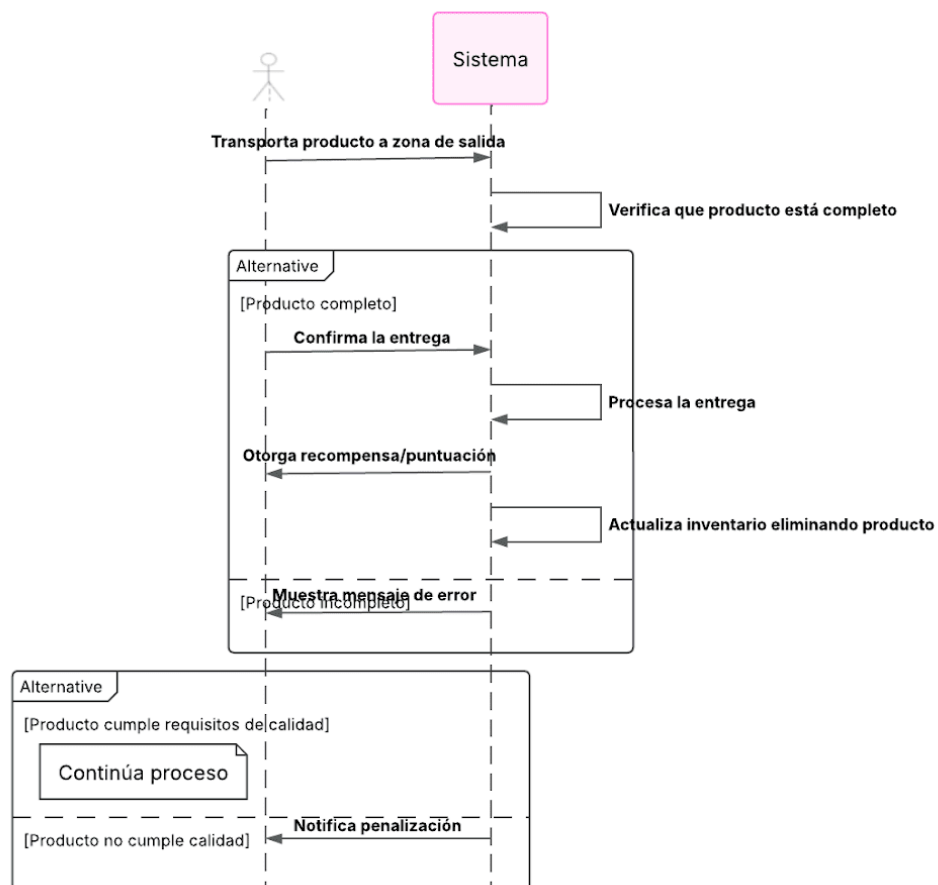


Figura 12: Diagrama de Secuencia de Caso de Uso PQ-13

5.3.1.7. Compra de Materiales

El sistema permite a los jugadores adquirir materiales dentro del juego utilizando los recursos disponibles. Esta funcionalidad se integra con el inventario y el presupuesto, validando en tiempo real la disponibilidad de fondos y espacio. La compra eficiente de materiales es clave para mantener la producción activa y alcanzar los objetivos del nivel.

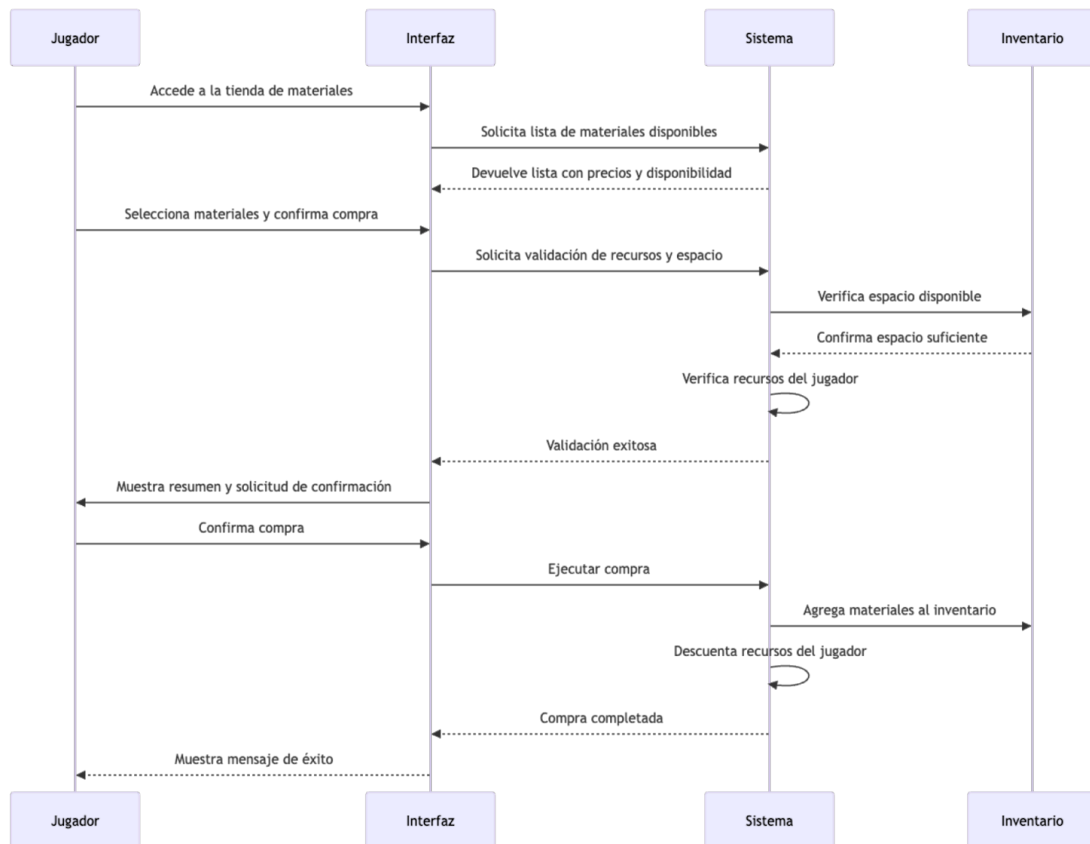


Figura 13: Diagrama de Secuencia de Caso de Uso PQ-14

5.3.1.8. Manejo de Desperdicio

El sistema facilita el manejo del desperdicio al permitir que los jugadores reciclen materiales en exceso o no utilizados, recuperando así parte del costo de producción. Este proceso incentiva la optimización de recursos y previene la acumulación innecesaria en el inventario.

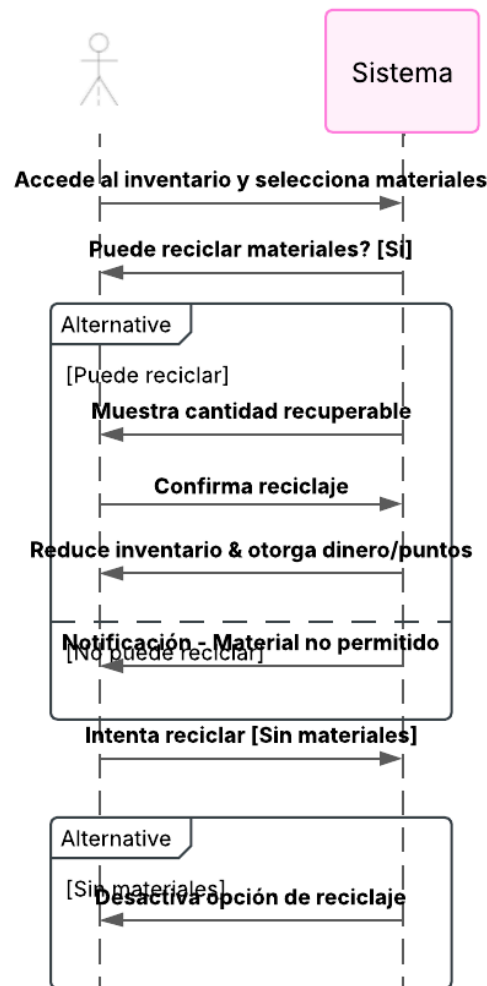


Figura 14: Diagrama de Secuencia de Caso de Uso PQ-20

5.3.2. Jugabilidad e Interacción

5.3.2.1. Movimiento de Personaje

El jugador debe ser capaz de moverse por el entorno 2D del respectivo nivel mediante el uso de teclas asignadas.

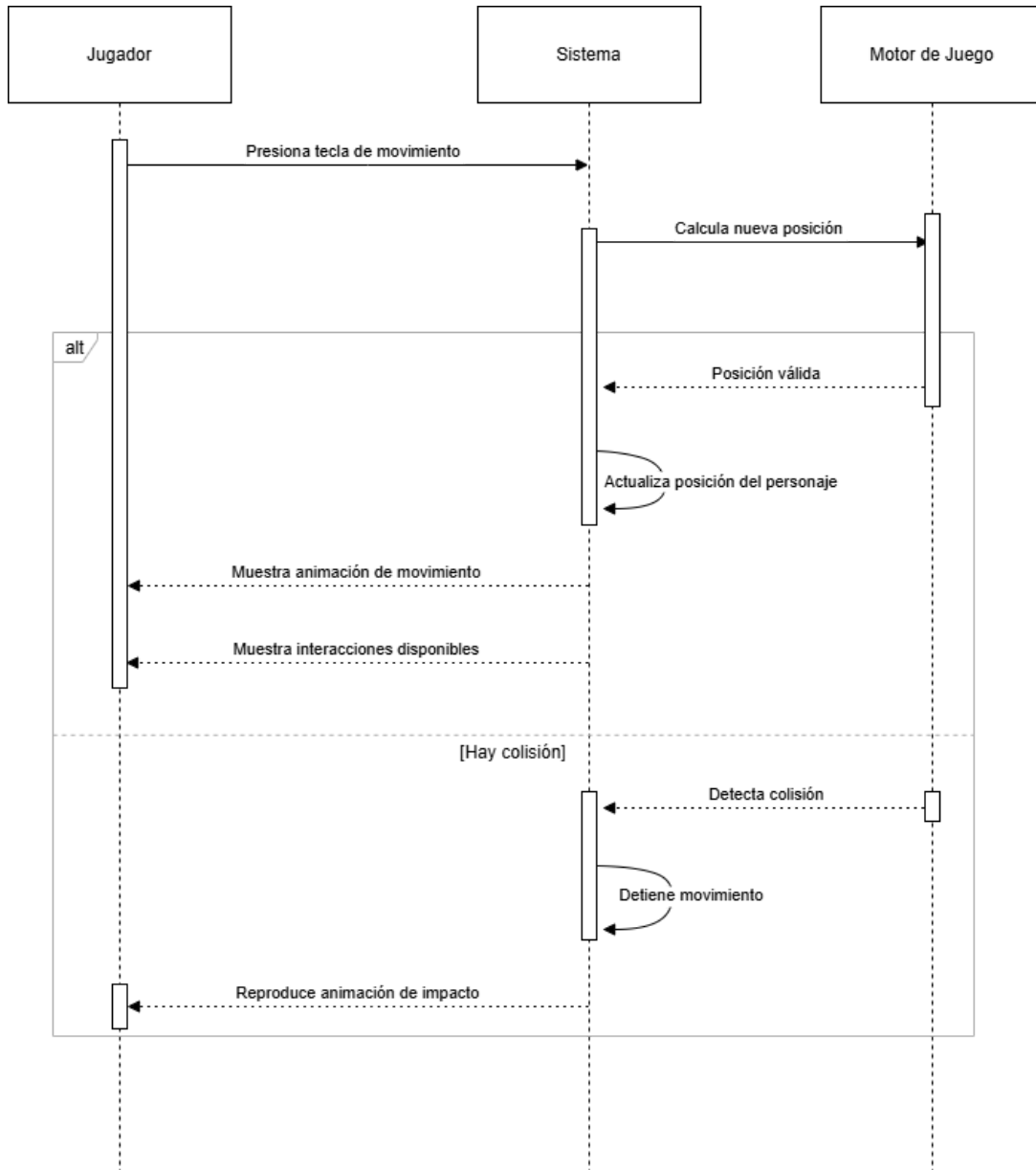


Figura 15: Diagrama de Secuencia de Caso de Uso PQ-02

5.3.2.2. Sistema de Penalización

El sistema aplica sanciones de manera automática cuando el jugador fabrica productos incorrectos o mezcla materiales erróneos.

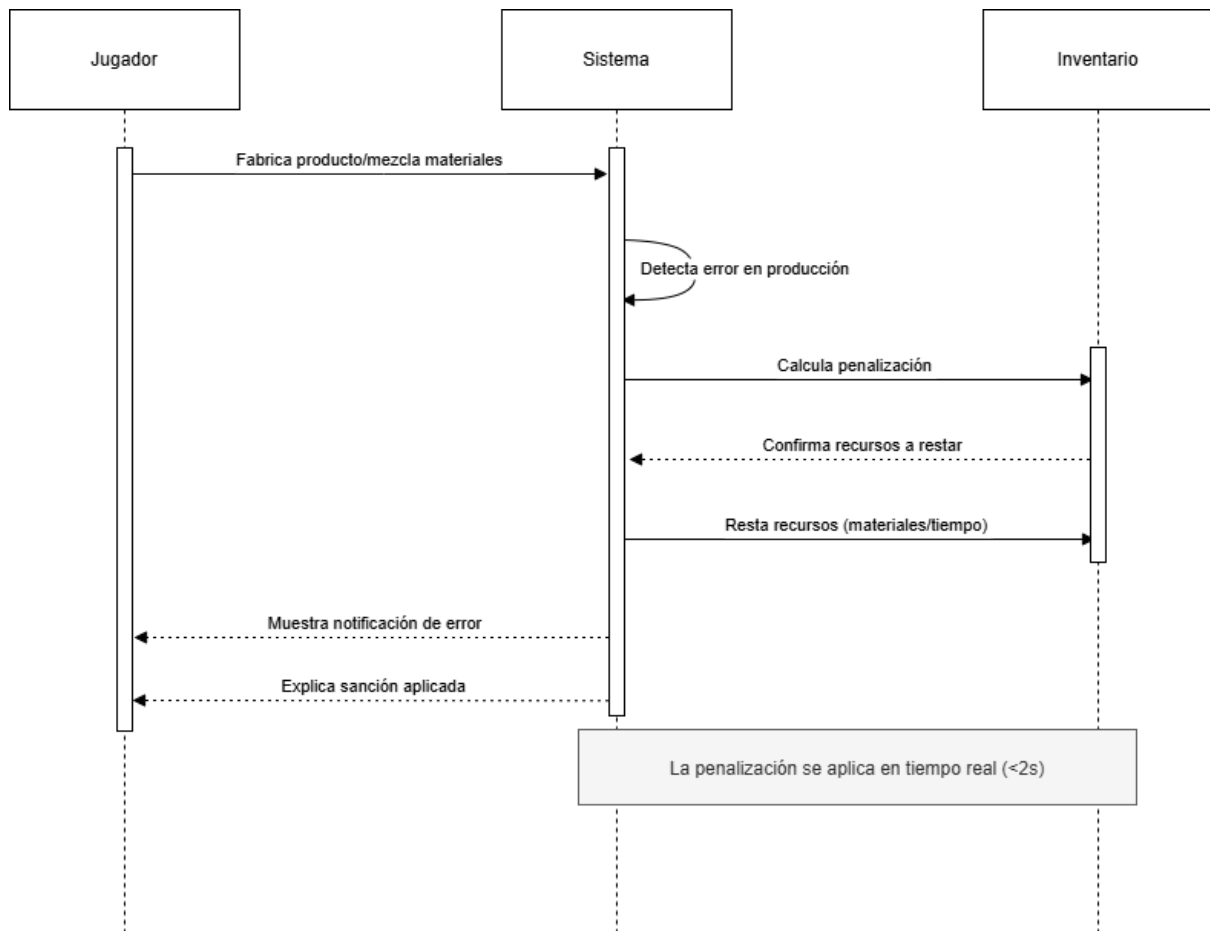


Figura 16: Diagrama de Secuencia de Caso de Uso PQ-03

5.3.2.3. Manejo de Errores en Producción

El jugador podrá utilizar una estación especial para deshacer hasta tres errores por nivel y recuperar el 50% de los materiales gastados.

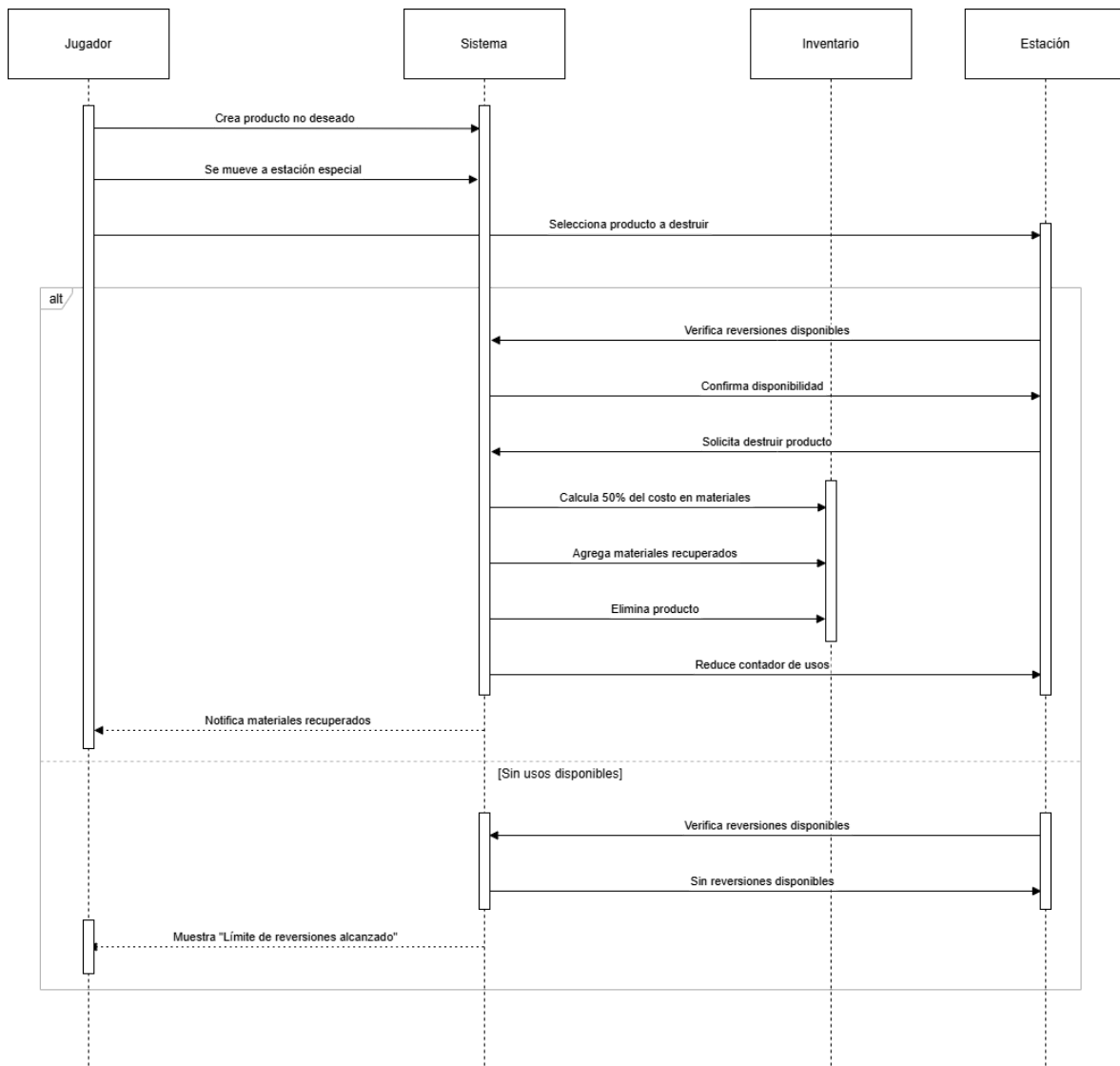


Figura 17: Diagrama de Secuencia de Caso de Uso PQ-04

5.3.2.4. Dificultad Progresiva

El jugador será capaz de jugar niveles cuya dificultad aumenta progresivamente.

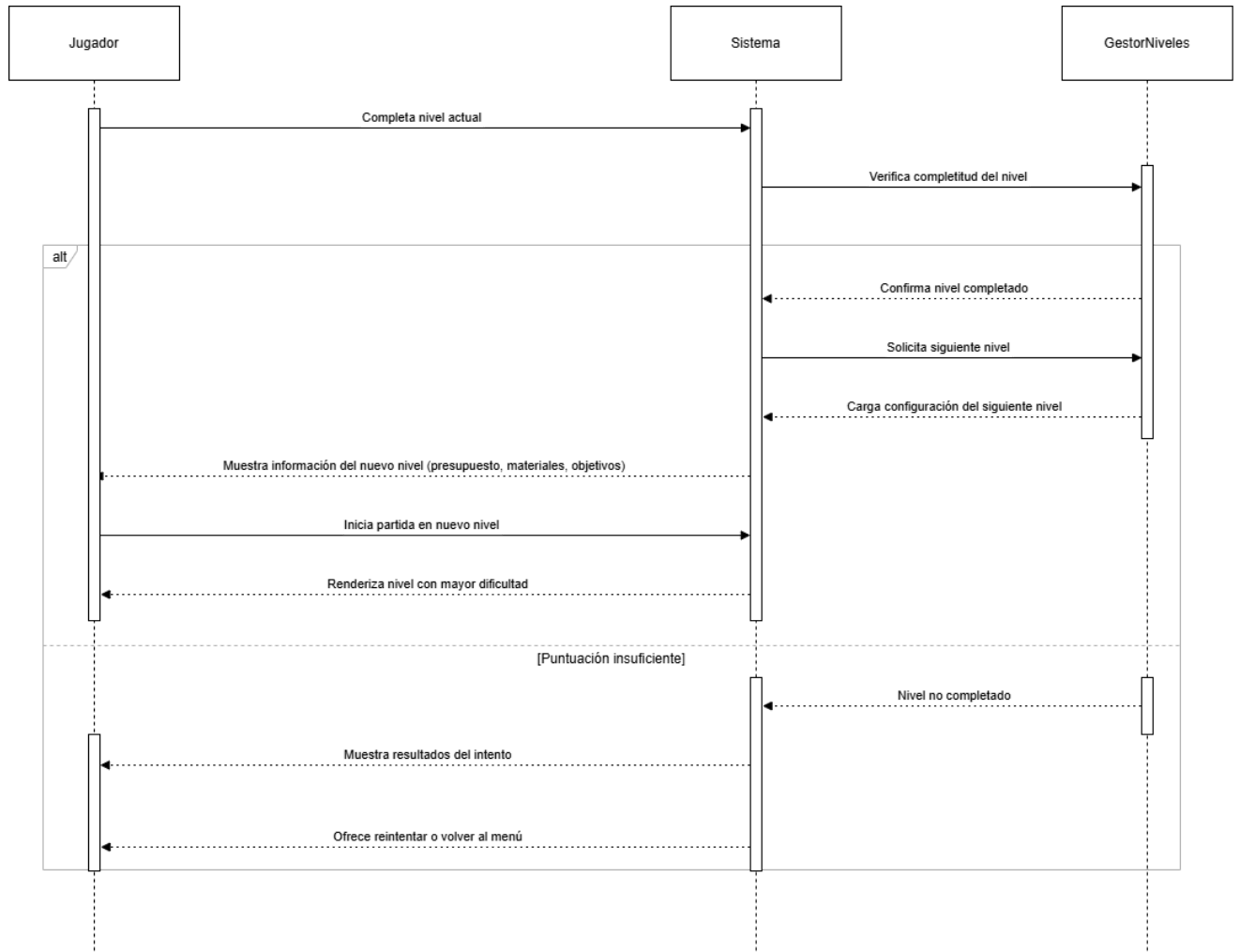


Figura 18: Diagrama de Secuencia de Caso de Uso PQ-05

5.3.2.5. Objetivos Opcionales por Nivel

Cada nivel tiene objetivos adicionales a los objetivos principales. Estos son diferentes y únicos por nivel y pueden afectar positivamente el puntaje y desempeño del jugador

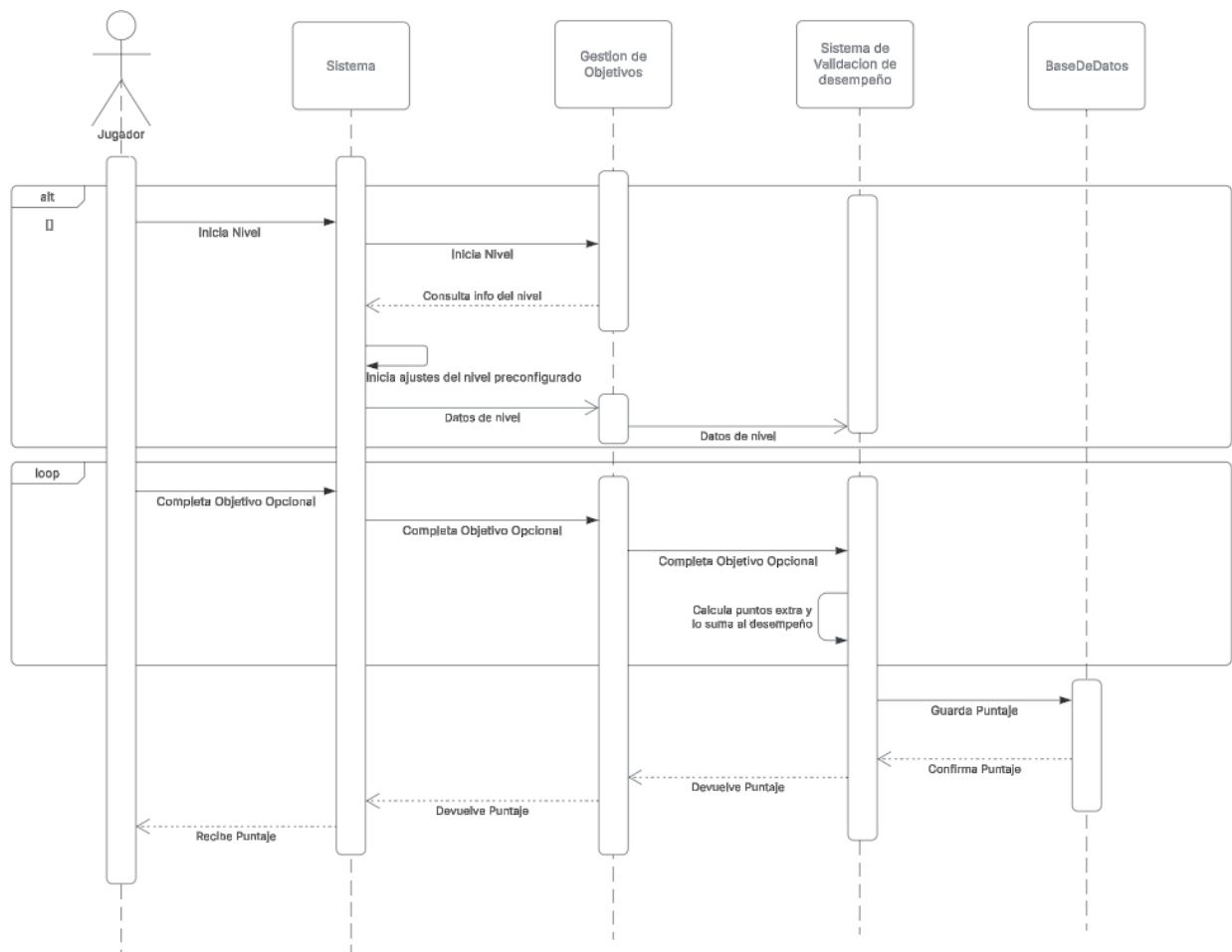


Figura 19: Diagrama de Secuencia de Caso de Uso PQ-010

5.3.2.6. Tiempo Límite

El sistema impone un límite de tiempo para cada nivel, obligando al jugador a completar los objetivos antes de que expire el cronómetro. Esta mecánica introduce un componente de presión y estrategia, ya que el tiempo restante influye directamente en la evaluación del desempeño. Al finalizar el tiempo, el sistema cierra automáticamente la partida y procede con la validación.

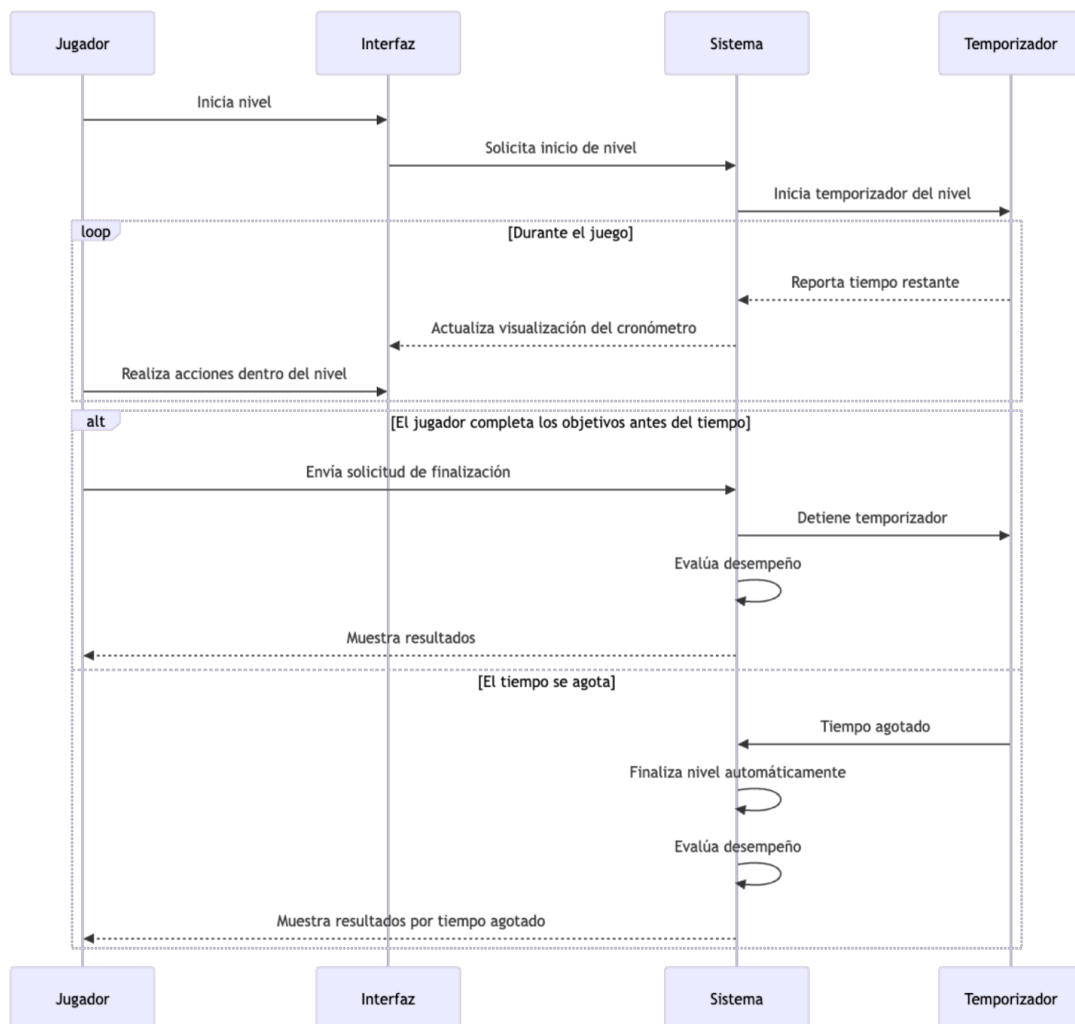


Figura 20: Diagrama de Secuencia de Caso de Uso PQ-16

5.3.3. Evaluación y Feedback

5.3.3.1. Validación de Desempeño por Partida

El sistema calcula según los objetivos completados el desempeño objetivo del jugador en un dado nivel y partida.

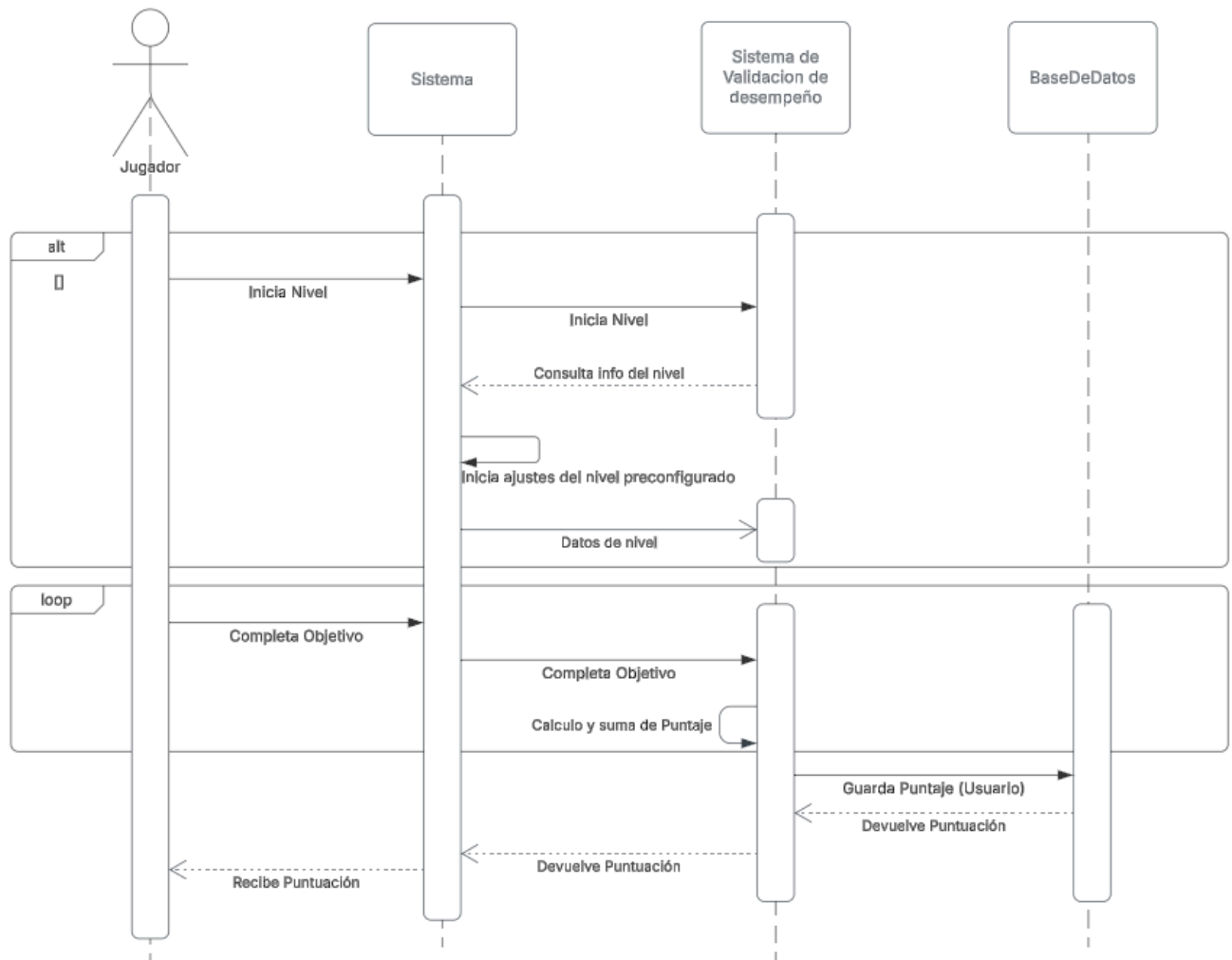


Figura 21: Diagrama de Secuencia de Caso de Uso PQ-08

5.3.3.2. Tabla de Clasificación de Jugadores

El sistema, basándose en una lista de puntajes de desempeño de cada usuario, forma una tabla de clasificación descendente de los jugadores. Se divide por niveles.

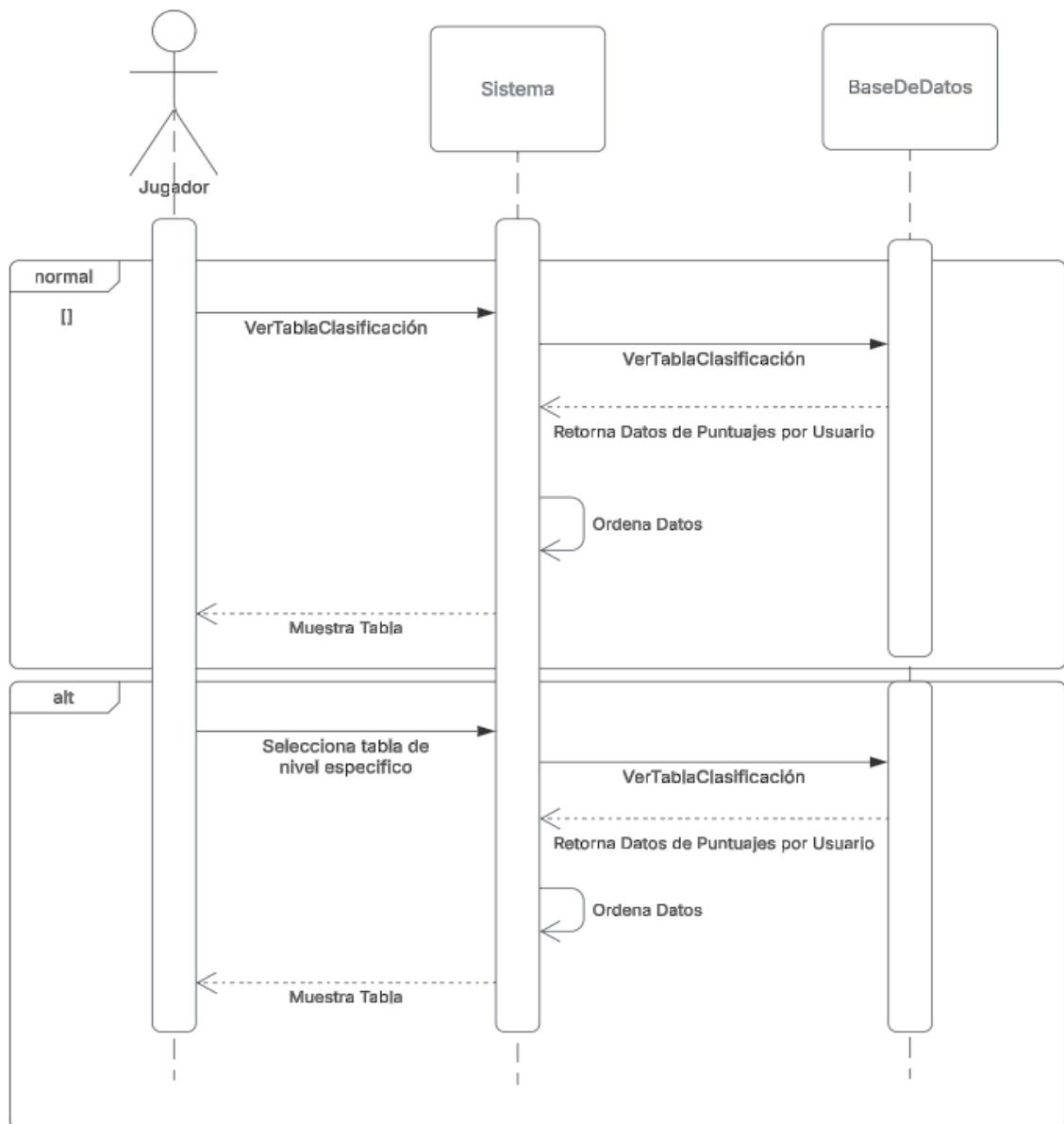


Figura 22: Diagrama de Secuencia de Caso de Uso PQ-09

5.3.4. Experiencia de Usuario

5.3.4.1. Tutorial Inicial

El sistema presenta automáticamente un tutorial interactivo la primera vez que el jugador accede al juego. Este tutorial guía al usuario paso a paso por las mecánicas esenciales, como la compra de materiales, el uso del inventario y la fabricación de productos. Su objetivo es garantizar que el jugador comprenda el funcionamiento básico del sistema antes de comenzar a jugar niveles completos.

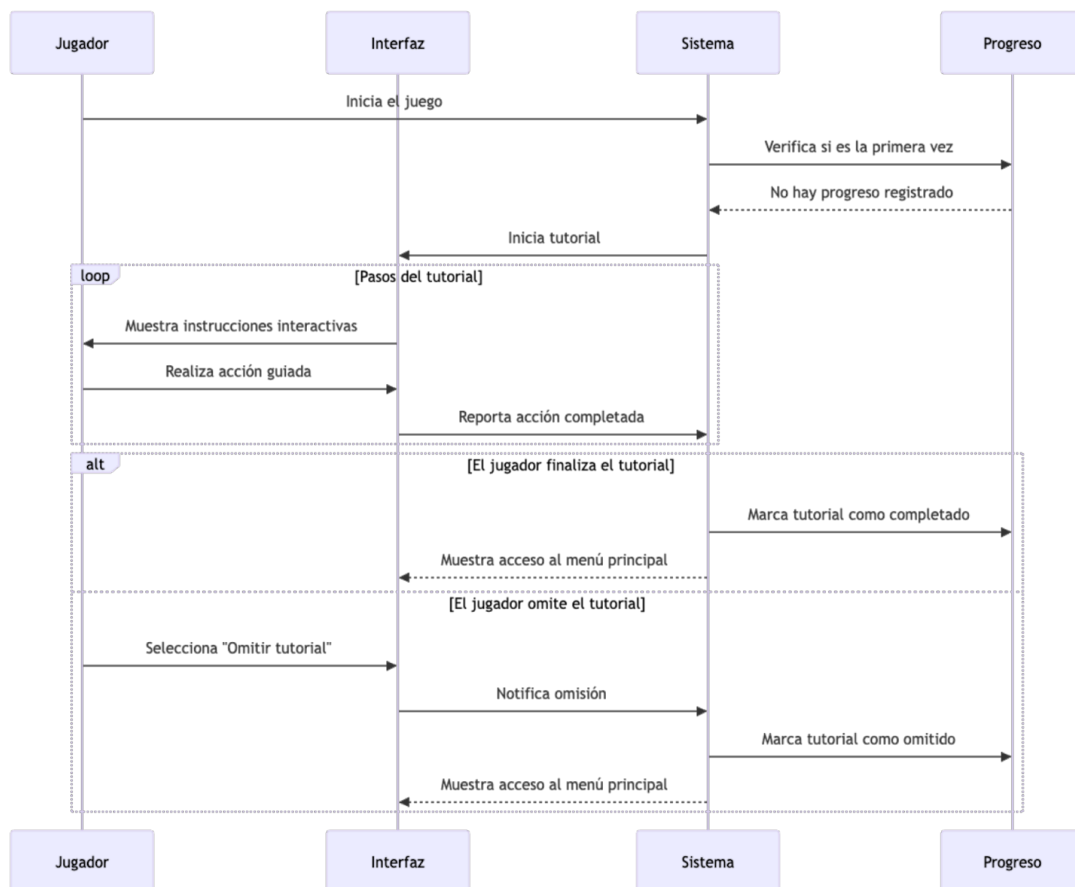


Figura 23: Diagrama de Secuencia de Caso de Uso PQ-15

5.3.4.2. Guardar Partida

El sistema permite al jugador guardar su progreso en cualquier momento durante una partida. Esta función almacena el estado actual del juego, incluyendo nivel, inventario, presupuesto y tiempo restante. Gracias a esto, el jugador puede pausar su experiencia y retomarla posteriormente desde el mismo punto, sin pérdida de datos relevantes.

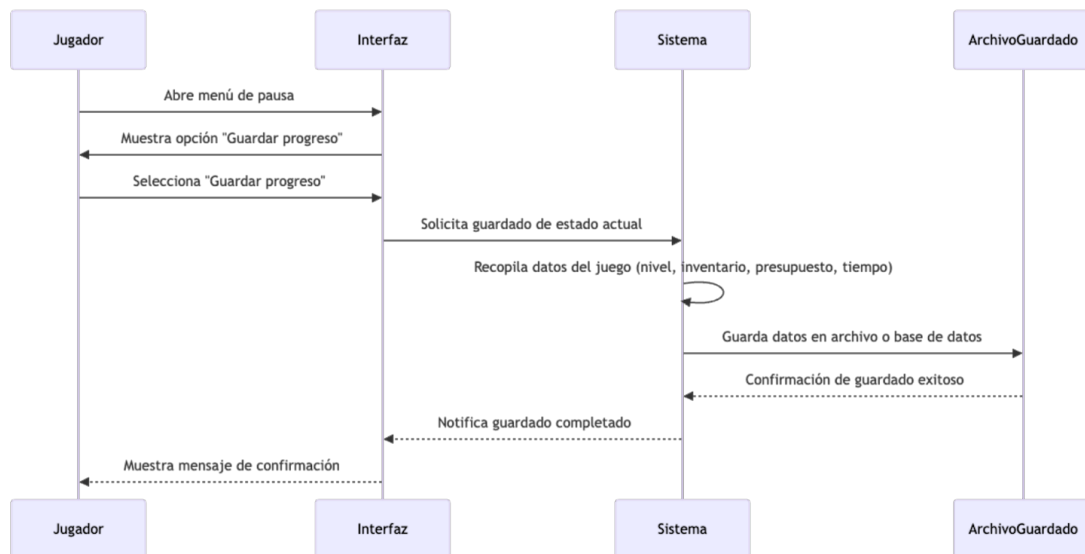


Figura 24: Diagrama de Secuencia de Caso de Uso PQ-17

5.3.4.3. Configuración del Avatar

El sistema permite al jugador personalizar su avatar antes de iniciar una partida. Esta personalización incluye la selección de colores, ropa y accesorios prediseñados. Las opciones disponibles no afectan el rendimiento del jugador, pero mejoran la inmersión al permitirle expresarse visualmente dentro del entorno de juego.

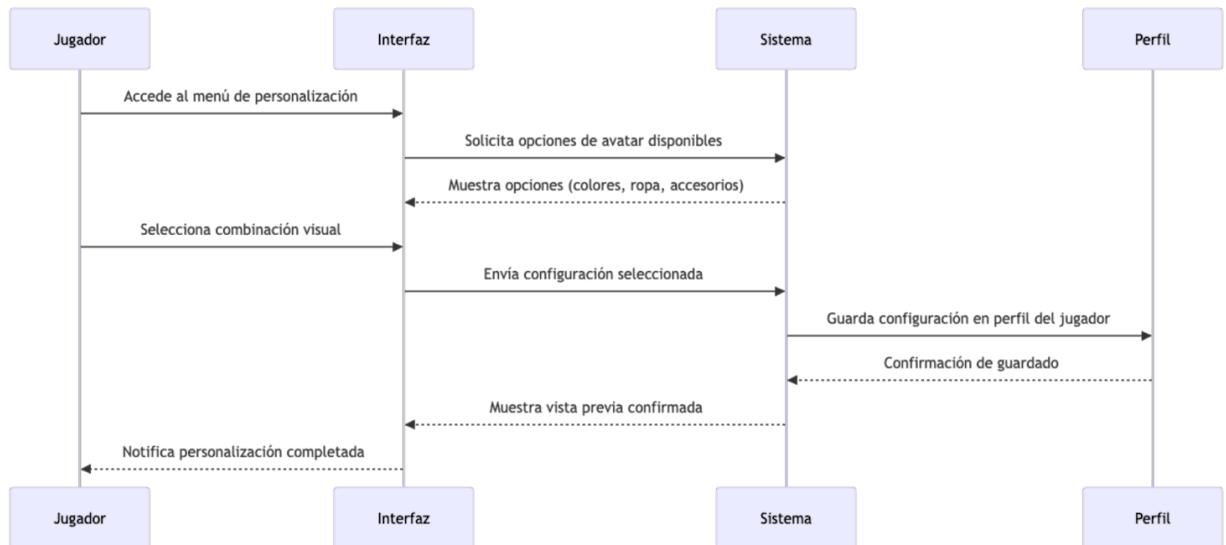


Figura 25: Diagrama de Secuencia de Caso de Uso PQ-18

5.3.5. Gestión de Usuarios y Datos

5.3.5.1. Gestión de Usuarios

El sistema permite la gestión integral de usuarios, garantizando que cada jugador disponga de un perfil único que almacena su progreso, configuraciones personalizadas y estadísticas de desempeño. Al iniciar el juego, el usuario crea o selecciona un perfil existente, permitiendo que la información se guarde y recupere de manera independiente en cada sesión.

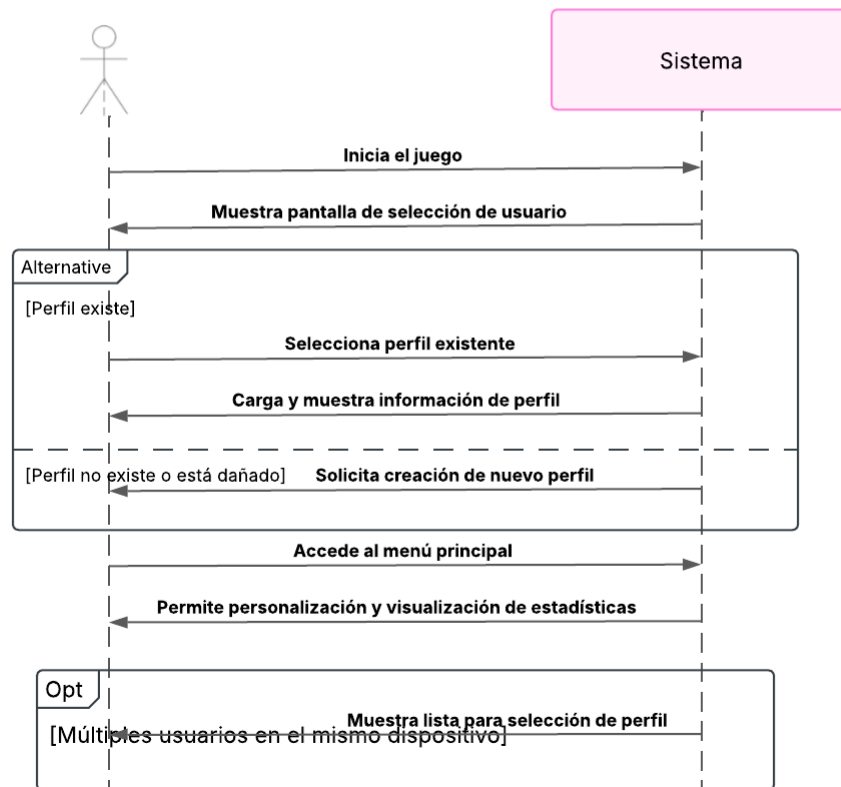


Figura 26: Diagrama de Secuencia de Caso de Uso PQ-19

6. Vista de Procesos

6.1. Proceso de Configuración Inicial

Se requiere que el usuario jugador configure su perfil de usuario

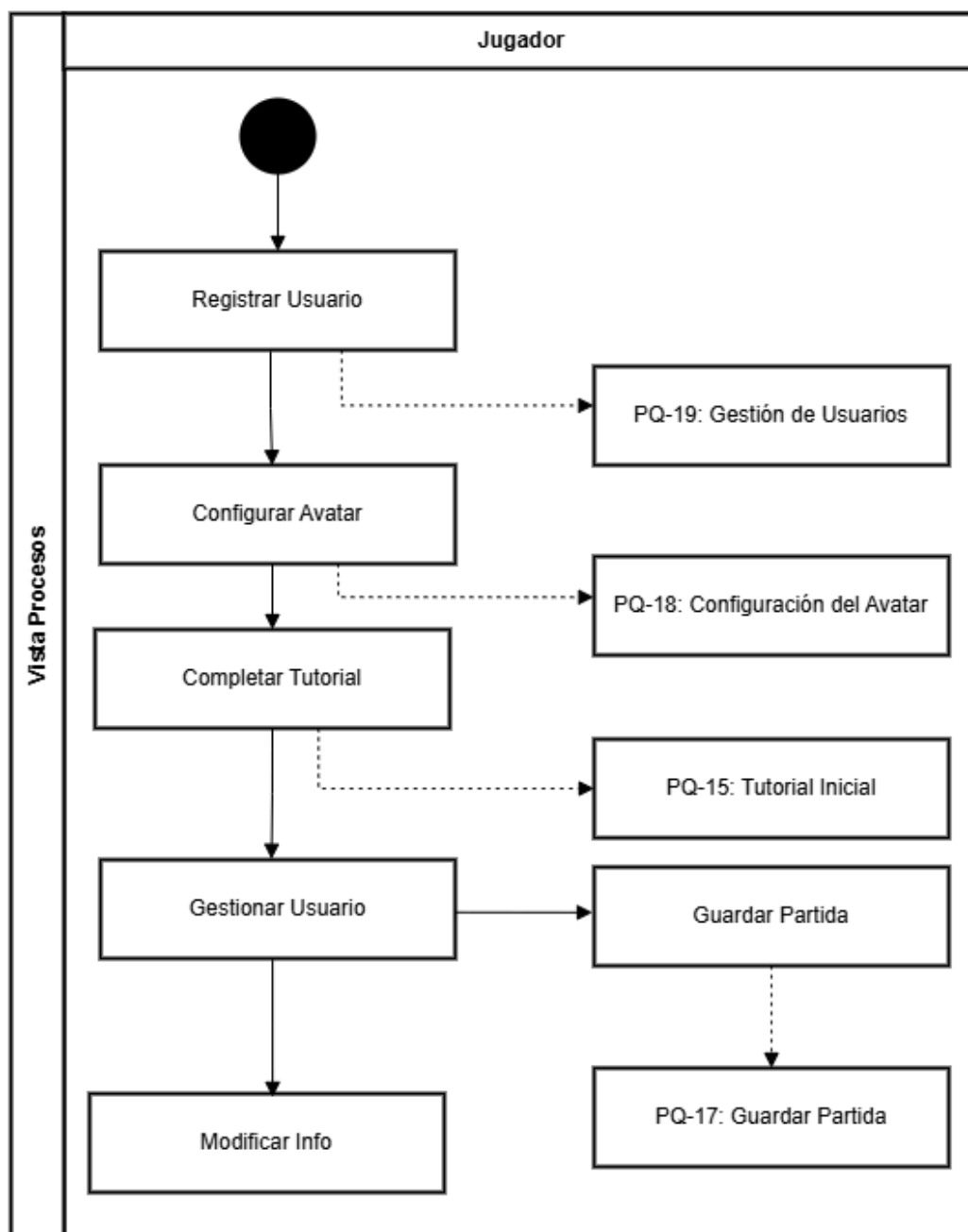


Figura 27: Proceso de Configuración Inicial

6.2. Proceso de Selección de Nivel

Se requiere que el usuario jugador esté iniciado en su cuenta personal

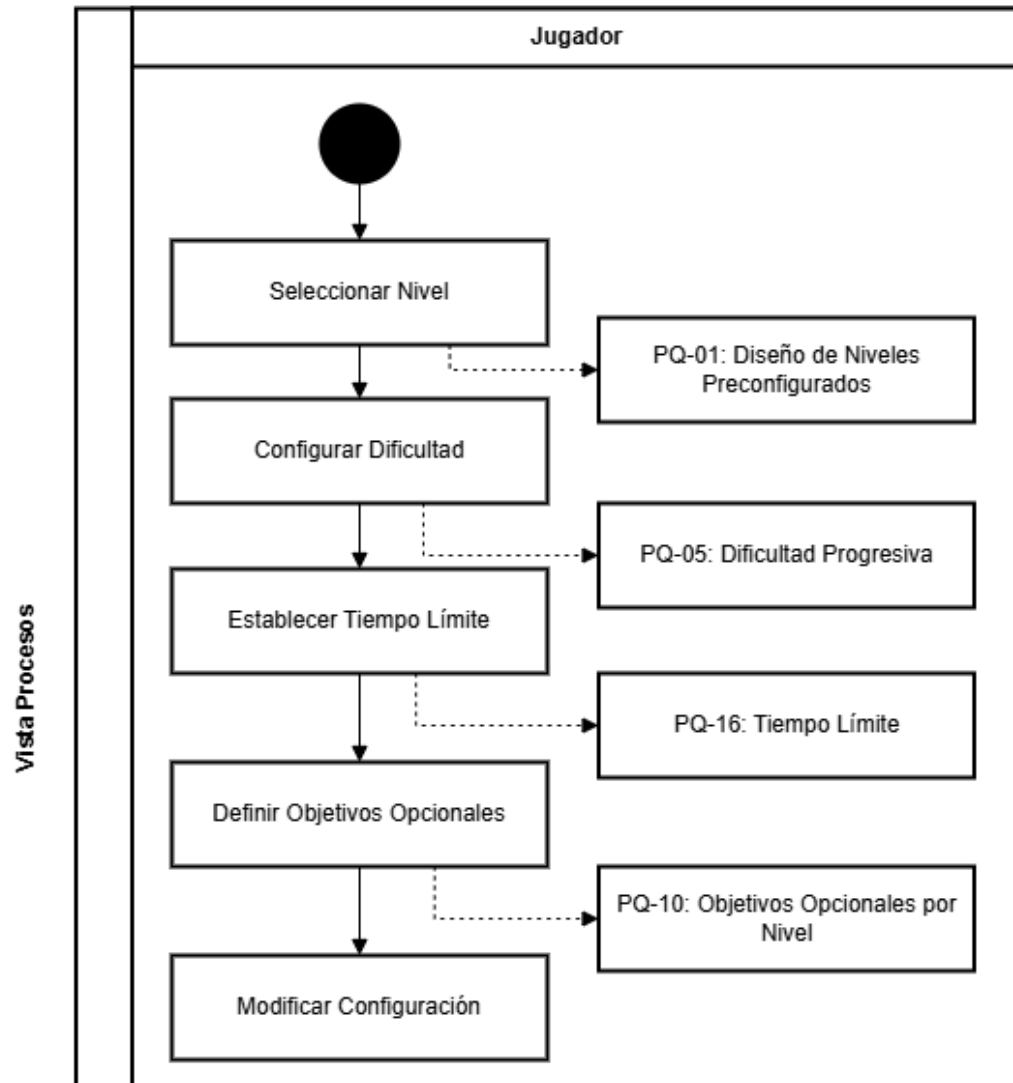


Figura 28: Proceso de Selección de Nivel

6.3. Proceso de Producción

Se requiere que el usuario jugador se encuentre iniciado en su cuenta y entre a alguno de los niveles.

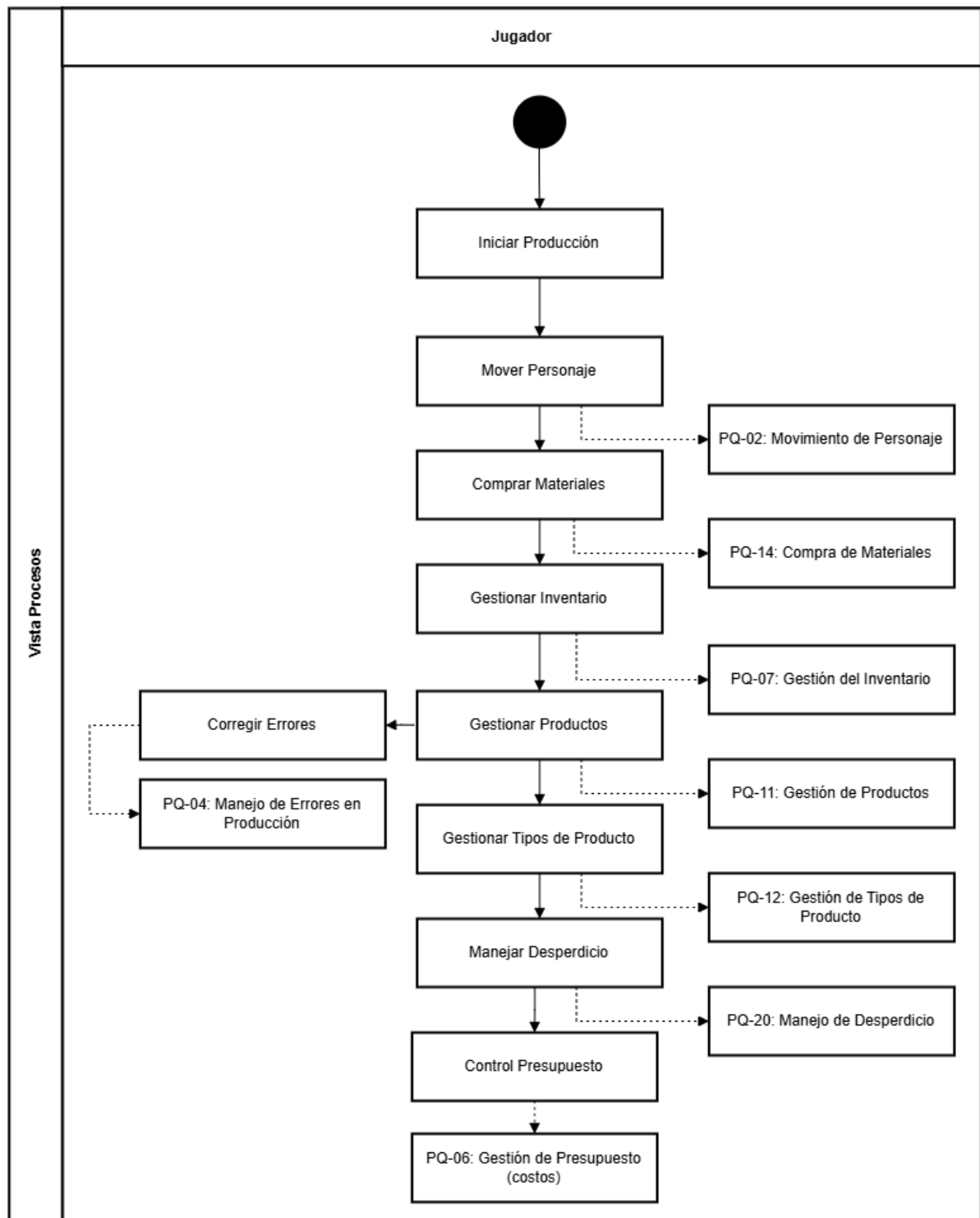


Figura 29: Proceso de Producción

6.4. Proceso de Entrega y Finalización

El usuario jugador completa objetivos dentro de un nivel.

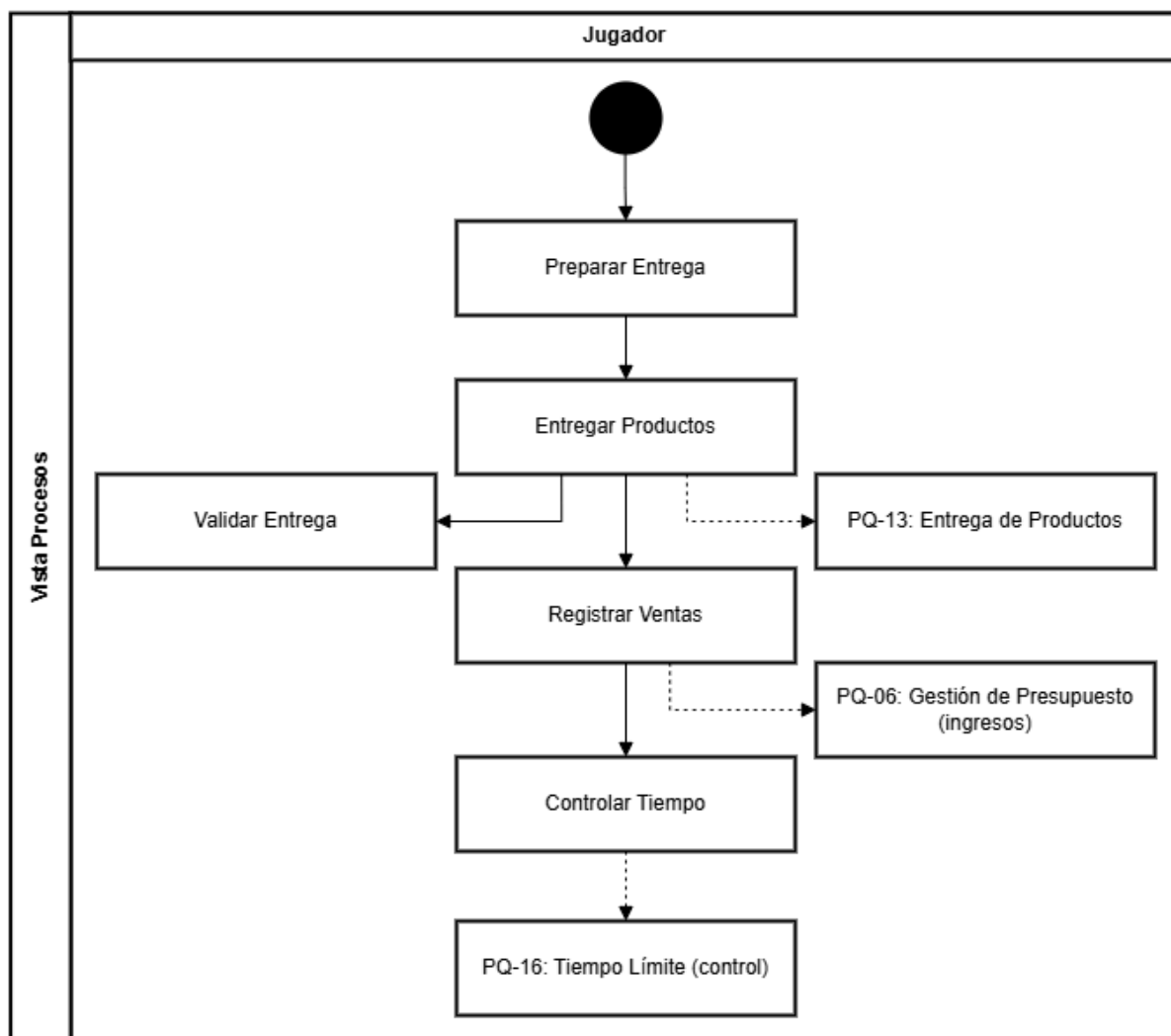


Figura 30: Proceso de Entrega y Finalización

6.5. Proceso de Evaluación y Recompensa

Se requiere que el usuario jugador se encuentre iniciado en su cuenta personal.

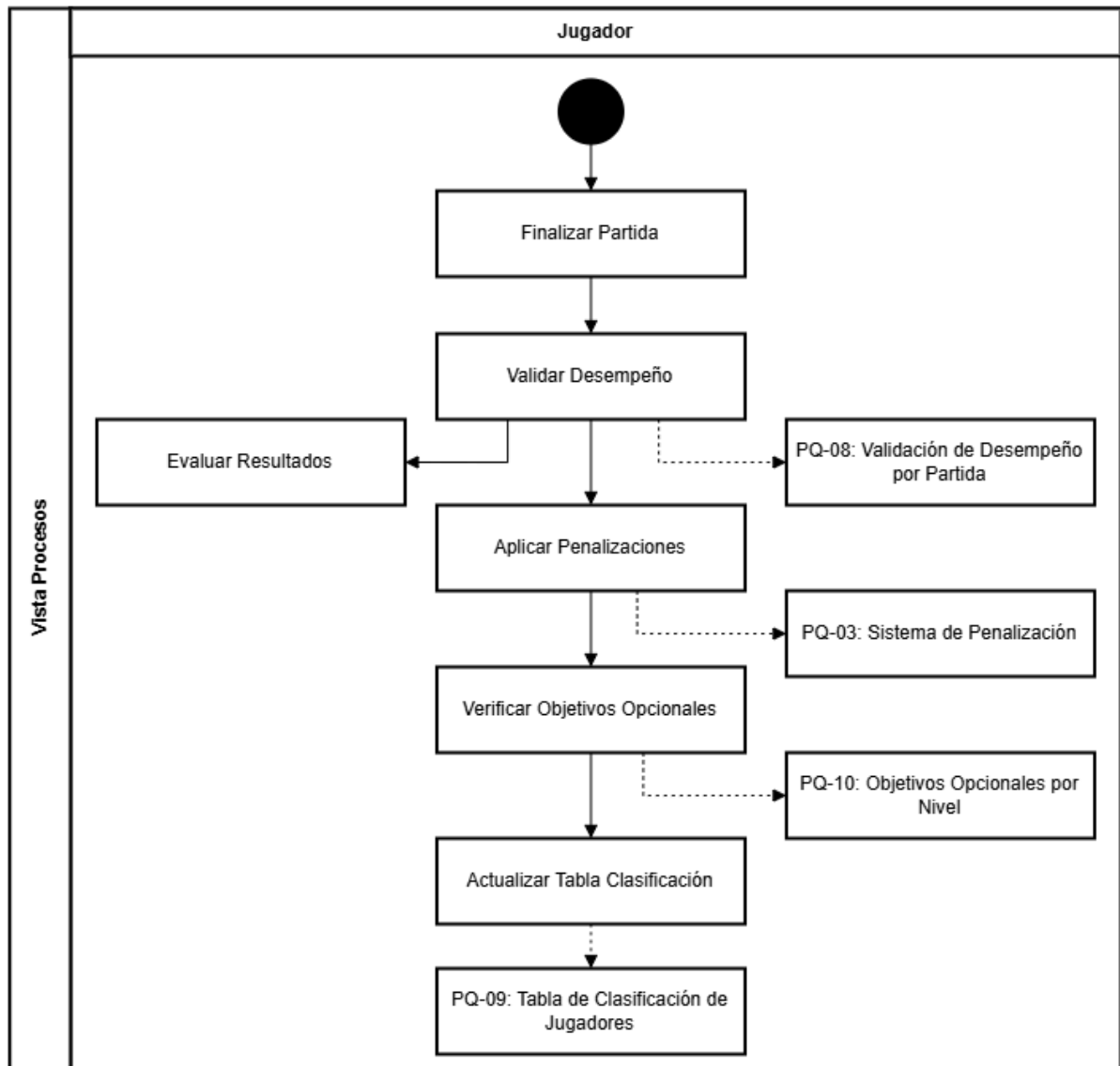


Figura 31: Proceso de Evaluación y Recompensa

7. Vista Física

La vista física del sistema proporciona una representación detallada de cómo se distribuyen físicamente los diferentes componentes del software en el entorno real. En esta sección, se explorará la disposición concreta de los elementos del sistema, como servidores, dispositivos de red y otros recursos de hardware, con el fin de comprender cómo interactúan en el entorno físico.

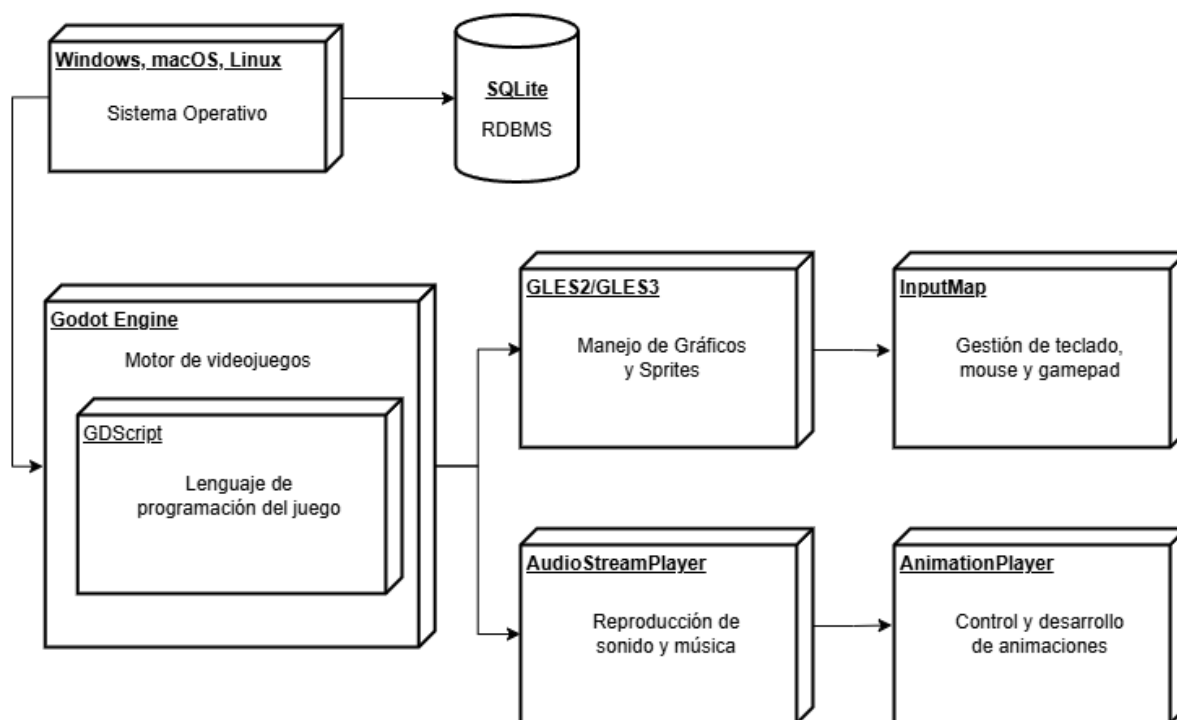


Figura 32: Vista Física

8. Vista de Implementación

La vista de implementación es una representación detallada de cómo se estructuran y relacionan los módulos dentro de una aplicación de software. Para ello, se va a usar un diagrama de componentes donde se despliega cómo es que el sistema funciona y realiza sus tareas. En este diagrama existen 3 módulos principales los cuales son: usuario jugador, sistema y base de datos.

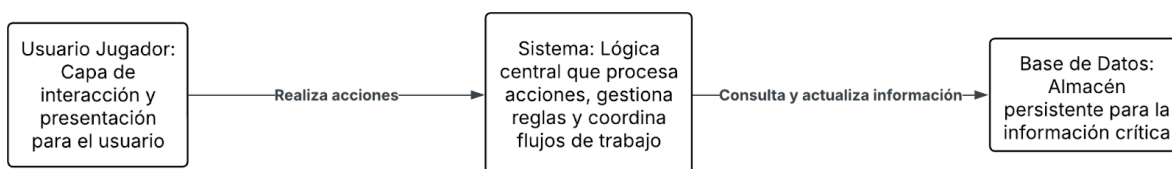


Figura 33: Vista de Implementación

9. Vista de Datos

En esta sección se describe la vista de datos del software, incluyendo cómo se diseñará y organizará la base de datos SQLite para satisfacer los requisitos de almacenamiento y acceso a la información sobre usuarios y el desempeño de los usuarios entre otros datos que se necesitan que sean permanentes.

SQLite es un sistema de gestión de base de datos relacional (RDBMS) de código abierto (Librería), ligero y autónomo, que no requiere un servidor separado y se integra directamente en las aplicaciones, ideal para dispositivos con recursos limitados. Godot utiliza SQLite para su gestión de bases de datos.

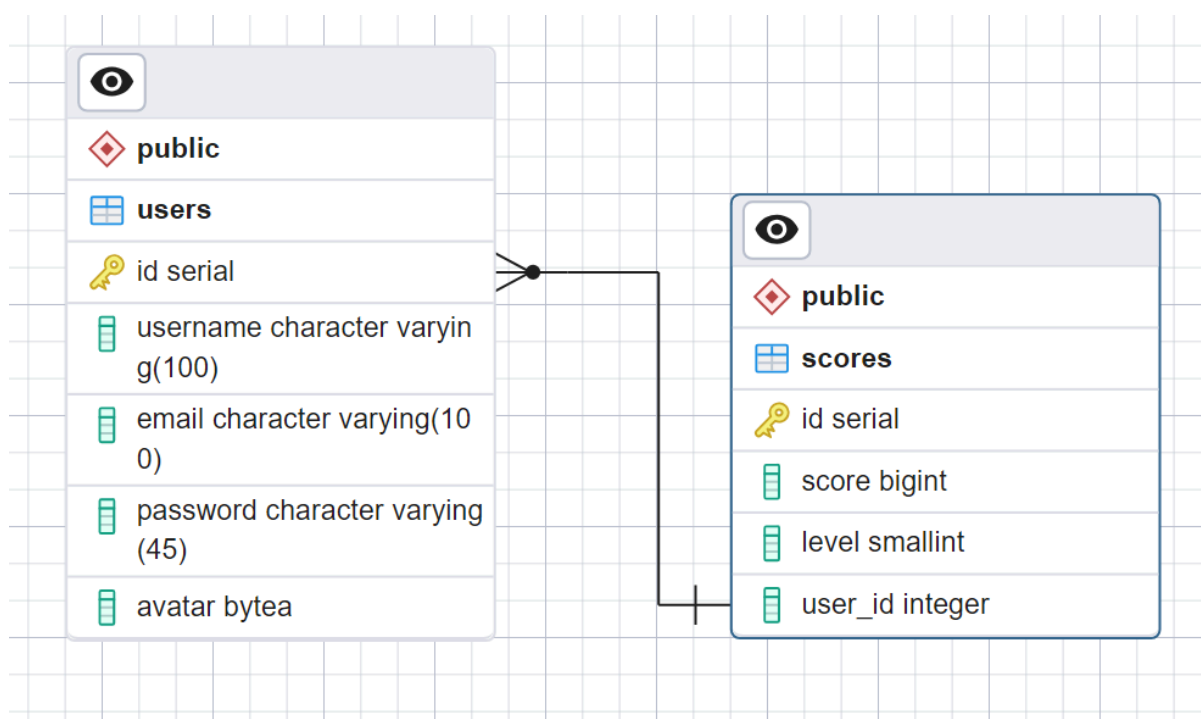


Figura 34: Vista de Datos

10. Tamaño y Desempeño

El sistema “ProfitQuest” ha sido diseñado con el objetivo de ofrecer un rendimiento fluido en una amplia gama de dispositivos, garantizando al mismo tiempo una experiencia de usuario satisfactoria durante la ejecución del juego. Dado que el sistema será implementado como una aplicación de escritorio en 2D utilizando el motor de videojuegos Godot, se espera que el consumo de recursos sea moderado, permitiendo su operación incluso en equipos con especificaciones limitadas.

En cuanto al tamaño, se estima que el paquete completo del sistema, incluyendo los recursos gráficos, archivos de configuración y ejecutables, tendrá un tamaño inferior a 500 MB. Este valor puede variar dependiendo de la inclusión de nuevos niveles, elementos visuales o contenido adicional en futuras versiones, sin afectar significativamente la capacidad de distribución ni el tiempo de descarga.

Respecto al desempeño, el sistema debe garantizar tiempos de respuesta inmediatos en la mayoría de las interacciones del usuario. Las transiciones entre menús, la ejecución de animaciones y el procesamiento de decisiones dentro del juego deben realizarse con una latencia mínima para mantener la inmersión del jugador. La arquitectura del sistema asegura que las operaciones críticas, como la gestión de inventario, cálculo de presupuesto, validación de desempeño y renderizado de escenas, se ejecuten en tiempo real sin interrupciones perceptibles.

El motor Godot permite optimizaciones específicas a nivel gráfico y de lógica de juego, lo cual contribuye a un rendimiento eficiente. Además, se utilizarán técnicas de precarga de recursos y almacenamiento en caché para minimizar los tiempos de carga durante la navegación por los distintos módulos del sistema.

Se espera que el sistema pueda ejecutarse sin inconvenientes en equipos que cumplan con los siguientes requisitos mínimos:

- Procesador de doble núcleo.
- 4 GB de memoria RAM.
- Resolución mínima de pantalla de 1024x768 píxeles.
- Sistema operativo Windows, macOS o Linux.

Finalmente, el sistema ha sido diseñado para escalar de forma adecuada, permitiendo en futuras versiones la incorporación de funcionalidades adicionales sin comprometer el rendimiento base. Esto se logra mediante una arquitectura modular, donde cada componente opera de forma desacoplada, lo cual facilita la optimización y mantenimiento del sistema a lo largo del tiempo.

11. Calidad

La arquitectura de “ProfitQuest” ha sido concebida para asegurar un producto final de alta calidad, confiable, seguro y fácil de mantener. Para lograrlo, se han considerado aspectos fundamentales que garantizan el cumplimiento de los requisitos funcionales y no funcionales definidos en la especificación original. Esta sección describe los principales lineamientos que sustentan la calidad del sistema desde diversas perspectivas, incluyendo fiabilidad, seguridad, estandarización en el desarrollo y capacidad de escalabilidad.

11.1. Requerimientos de Fiabilidad

El sistema debe garantizar un comportamiento estable y predecible en todos los escenarios de uso previstos. Para ello, se han implementado mecanismos de control de errores que aseguren la integridad de los datos del jugador, el correcto funcionamiento de las mecánicas del juego y la persistencia del progreso en caso de fallos inesperados.

Entre los requerimientos específicos de fiabilidad se incluyen:

- Prevención de pérdidas de datos mediante un sistema de guardado robusto.
- Manejo de errores en la carga de niveles o recursos visuales, con mensajes claros para el usuario.
- Capacidad de recuperación ante fallos, como la interrupción inesperada del sistema.
- Consistencia del comportamiento del juego a través de múltiples sesiones.

Los niveles preconfigurados, validaciones de desempeño y las mecánicas de simulación han sido diseñadas para mantener resultados coherentes bajo condiciones normales de uso, reforzando la confianza del usuario en el sistema.

11.2. Consideraciones de Seguridad

Dado que “ProfitQuest” está diseñado como una aplicación de escritorio sin conexión continua a internet, los vectores de ataque externos son limitados. No

obstante, se han contemplado medidas de seguridad para proteger la integridad del sistema y la privacidad de los datos locales del usuario.

Las consideraciones de seguridad incluyen:

- Almacenamiento seguro de archivos de progreso y configuraciones del usuario.
- Restricciones de modificación de niveles o manipulación de recursos mediante verificación de integridad.
- Prevención de acceso no autorizado a archivos sensibles mediante cifrado básico o estructuración en directorios de acceso restringido.

En versiones futuras donde se contemple funcionalidad en línea, se evaluarán mecanismos de autenticación, comunicación cifrada y control de acceso adecuado para salvaguardar la información del usuario.

11.3. Estándares de Programación

El desarrollo de “ProfitQuest” se rige por estándares que promueven la mantenibilidad, claridad y eficiencia del código fuente. Se aplican principios SOLID en el diseño orientado a objetos, así como convenciones de nomenclatura y estructuración de archivos alineadas con las buenas prácticas del motor Godot y del lenguaje GDScript.

Los estándares contemplados incluyen:

- Separación clara entre lógica de juego, interfaz gráfica y gestión de datos.
- Comentarios explicativos en fragmentos críticos del código para facilitar el mantenimiento.
- Reutilización de componentes y scripts mediante modularidad.
- Validación continua mediante pruebas internas durante la integración de nuevas funcionalidades.

Estas prácticas permiten un desarrollo colaborativo organizado y reducen la complejidad durante futuras expansiones o correcciones del sistema.

11.4. Consideraciones de Escalabilidad

“ProfitQuest” ha sido diseñado con una arquitectura modular que facilita su evolución a lo largo del tiempo. Esta característica permite incorporar nuevas funcionalidades, niveles, modos de juego o adaptaciones gráficas sin necesidad de reestructurar por completo el sistema existente.

Algunas consideraciones clave en cuanto a escalabilidad incluyen:

- Separación de niveles y mecánicas en archivos independientes, permitiendo su extensión o modificación de forma aislada.
- Estructura de menús y lógica de juego desacopladas para facilitar nuevas ramas de navegación.
- Uso eficiente de memoria y procesamiento, asegurando que nuevas funcionalidades no comprometan el rendimiento en sistemas con recursos limitados.
- Capacidad de integrar nuevos módulos, como un modo multijugador o editor de niveles, sin alterar la funcionalidad base.

Gracias a esta visión escalable, el sistema puede adaptarse a distintos contextos educativos, evolucionar con nuevos contenidos y ajustarse a necesidades cambiantes de los usuarios.

12. Anexos

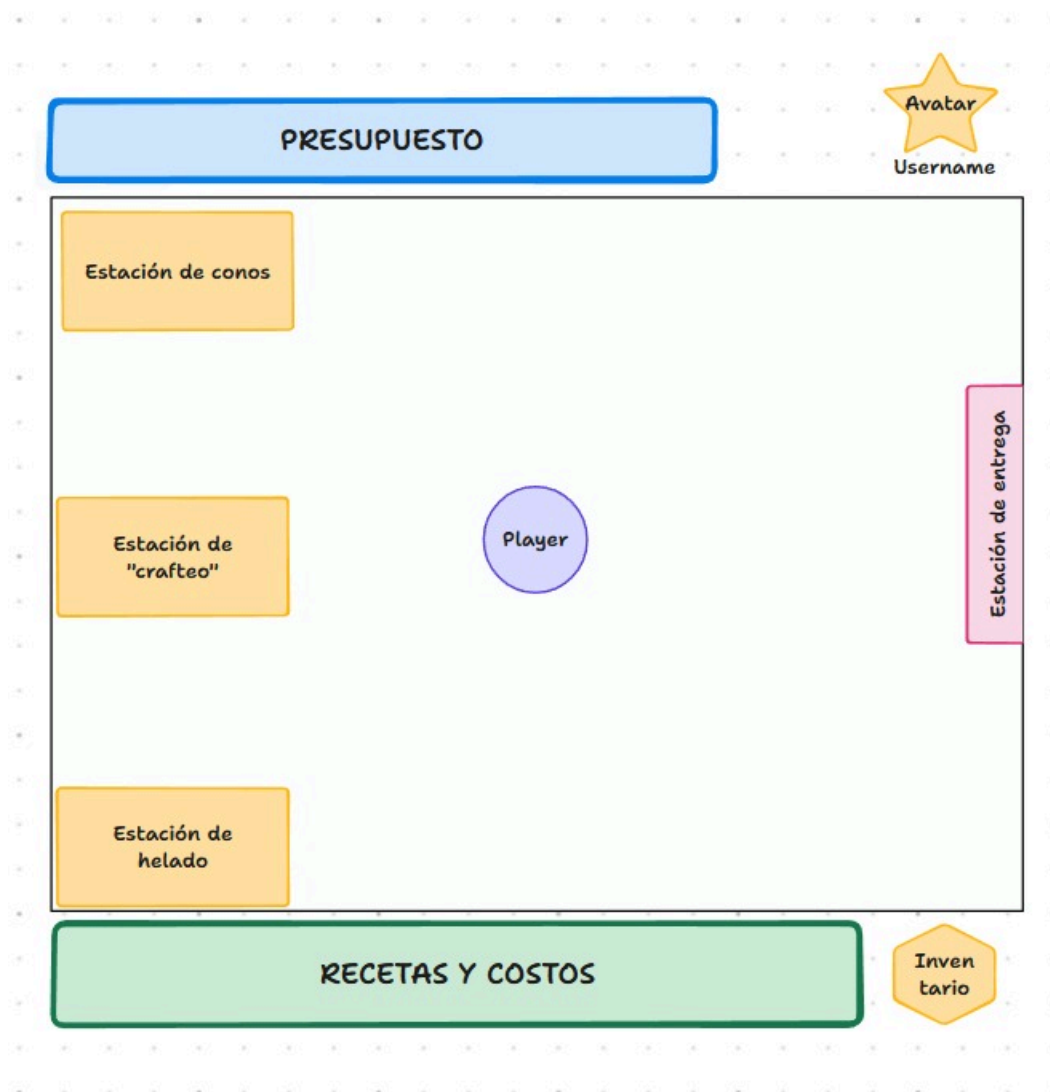


Figura 35: Planos del Ambiente del Juego

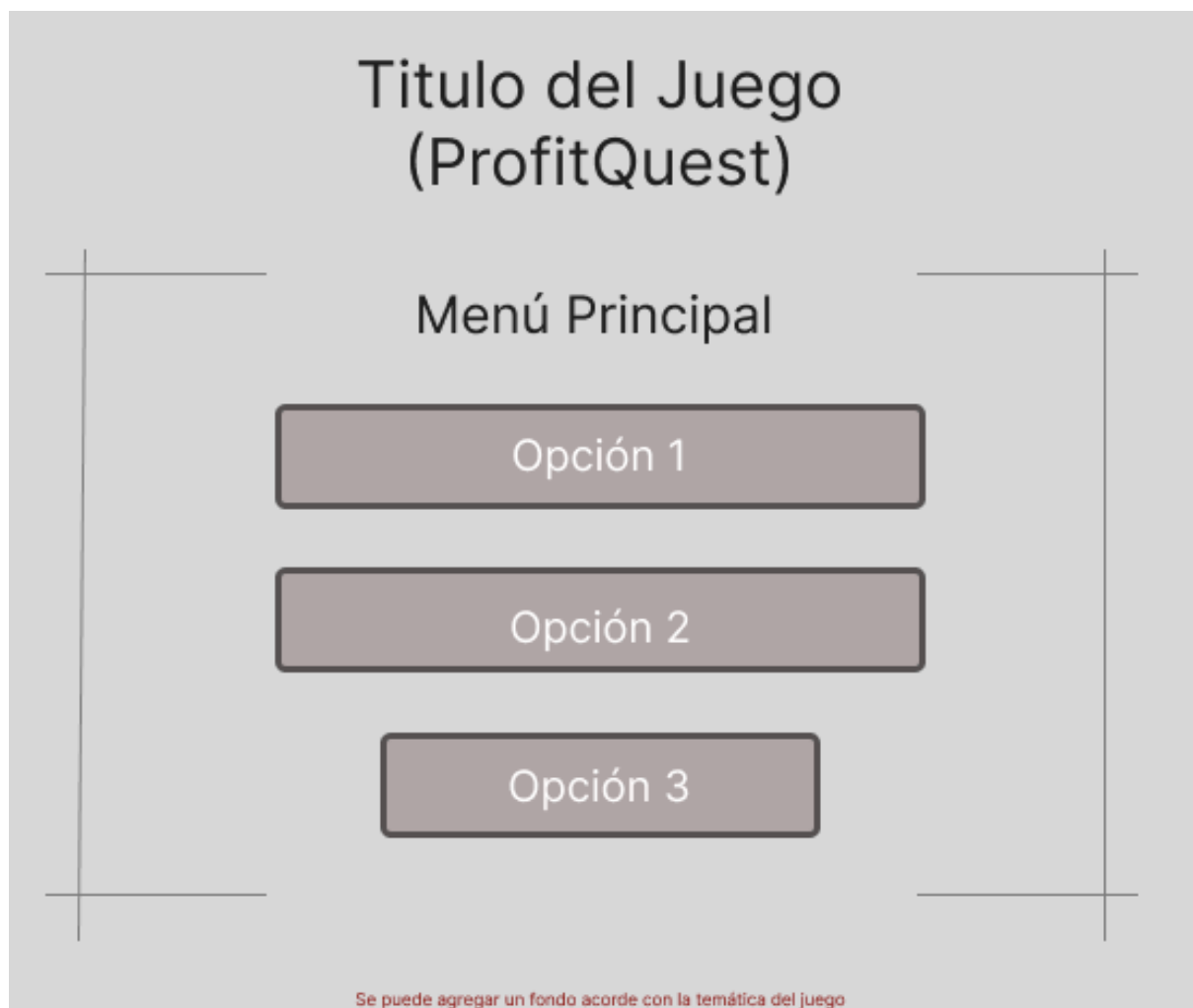


Figura 36: Planos del Menú principal