

**ERS**  
**(Especificación de Requisitos de Software)**

Para  
Juego Educativo: ProfitQuest

Preparado por:  
Leonardo Céspedes Tenorio (2022080602)  
Kevin Chang Chang (2022039050)  
Mauricio Fernández Brizuela (2022437510)  
Daniel Garbanzo Carvajal (2022117129)

*Instituto Tecnológico de Costa Rica, Sede Central*  
*Ingeniería en Computación*  
*I Semestre, 2025*

# Índice

<b>1. Introducción</b>	<b>4</b>
1.1. Propósito	4
1.2. Alcance	4
1.3. Definiciones, acrónimos y abreviaciones	5
1.4. Referencias	6
1.5. Visión General	6
<b>2. Descripción General</b>	<b>6</b>
2.1. Perspectiva del Producto	7
2.2. Funciones del Producto	7
2.3. Características de Usuario	8
2.4. Restricciones	8
2.5. Suposiciones y dependencias	9
2.6. Requisitos Futuros	9
<b>3. Requisitos Específicos</b>	<b>9</b>
3.1. Interfaces Externas	9
3.1.1. Interfaces de Usuario	9
3.1.2. Interfaces de Software	10
3.1.3. Interfaces de Hardware	11
3.1.4. Interfaces de Comunicación	11
3.2. Requisitos Funcionales	12
3.2.1. PQ-01: Diseño de Niveles Preconfigurados	12
3.2.2. PQ-02: Movimiento de Personaje	13
3.2.3. PQ-03: Sistema de Penalización	14
3.2.4. PQ-04: Manejo de Errores en Producción	15
3.2.5. PQ-05: Dificultad Progresiva	16
3.2.6. PQ-06: Gestión de Presupuesto	17
3.2.7. PQ-07: Gestión del Inventario	19
3.2.8. PQ-08: Validación de desempeño por partida	20
3.2.9. PQ-09: Tabla de clasificación de jugadores	22
3.2.10. PQ-10: Objetivos opcionales por nivel	23
3.2.11. PQ-11: Gestión de productos	25
3.2.12. PQ-12: Gestión de tipos de producto	27
3.2.13. PQ-13: Entrega de productos	28
3.2.14. PQ-14: Compra de materiales	29
3.2.15. PQ-15: Tutorial Inicial	32
3.2.16. PQ-16: Tiempo límite	34
3.2.17. PQ-17: Guardar partida	36
3.2.18. PQ-18: Configuración del avatar	38
3.2.19. PQ-19: Gestión de usuarios	41
3.2.20. PQ-20: Manejo de desperdicio	42
3.3. Requisitos no funcionales	44

3.3.1. Rendimiento	44
3.3.2. Experiencia de Usuario	44
3.3.3. Estabilidad	44
3.3.4. Portabilidad	45
3.3.5. Seguridad	45
3.3.6. Escalabilidad	45
3.3.7. Mantenibilidad	45
3.4. Restricciones de Diseño	46
3.4.1. Motor de Desarrollo	46
3.4.2. Estilo Visual y Gráfico	46
3.4.3. Interfaz y Experiencia de Usuario	46
3.4.4. Resolución y Compatibilidad	47
3.4.5. Mecánicas y Restricciones de Gameplay	47
3.4.6. Gestión de Datos y Guardado	47
3.4.7. Rendimiento y Optimización	47
3.5. Atributos del Sistema	48
3.6. Otros Requisitos	49
3.6.1. Requisitos de Seguridad	49
3.6.2. Requisitos de Seguridad	49
3.6.3. Requisitos de Tiempo de Inactividad	49
<b>4. Apéndices</b>	<b>50</b>

# 1. Introducción

El presente documento es la Especificación de Requerimientos de Software (ERS) 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. El sistema permite a los usuarios interactuar con escenarios de producción simulados, tomar decisiones estratégicas y recibir retroalimentación inmediata de su desempeño. Las funcionalidades específicas del sistema serán explicadas en próximas secciones.

## 1.1. Propósito

Este documento tiene como fin explicar los requerimientos funcionales y no funcionales que se necesitarán para cumplir con la implementación del sistema ProfitQuest. A lo largo del documento se definen todas las funcionalidades necesarias para cumplir con los objetivos del usuario y así funcionar de guía para el equipo de desarrollo.

El contenido del proyecto va dirigido hacia la profesora Biljhana Farah Guzmán, la cual debe confirmar si esta información cumple con las funcionalidades requeridas para el sistema mediante los requerimientos planteados.

Este texto busca asegurar que el producto final cumpla con los requisitos esperados, garantizando que el sistema sea útil, accesible y adecuado para su propósito educativo.

## 1.2. Alcance

El sistema **ProfitQuest** ofrecerá una experiencia gamificada para el aprendizaje de conceptos de producción a través de un **juego en 2D con estilo pixel art**, con los siguientes módulos clave:

- **Módulo de simulación de producción:** permitirá a los jugadores tomar decisiones sobre compra de materiales, fabricación y evaluación de resultados.
- **Módulo de progresión:** el juego incluirá niveles temáticos con distintos desafíos de producción y gestión de costos.
- **Módulo de evaluación y feedback:** proporcionará retroalimentación sobre la eficiencia en la toma de decisiones dentro del juego.

- **Módulo de interfaz gráfica intuitiva:** facilitará la interacción con el sistema y la comprensión de los conceptos presentados.

**Fuera del alcance del sistema:**

- No se implementará un modo multijugador.
- No habrá integración con plataformas educativas externas.
- No se desarrollará una versión para dispositivos móviles.
- No se incluirá contenido generado por usuarios.

**Beneficios esperados:**

- Facilitar el aprendizaje de conceptos de costos de producción de manera interactiva.
- Mejorar la retención de conocimientos al combinar elementos educativos y lúdicos.
- Proveer una herramienta complementaria para entornos educativos.

**Objetivos del sistema:**

- Simular de manera precisa procesos de producción dentro del juego.
- Desarrollar niveles con mecánicas progresivas y realistas.
- Implementar un sistema de evaluación basado en la optimización de recursos.
- Crear una experiencia de usuario atractiva y accesible.

### 1.3. Definiciones, acrónimos y abreviaciones

Término	Descripción
API	Interfaz de programación de aplicaciones
ERS	Especificación de Requerimientos de Software
GUI	Interfaz Gráfica de Usuario
Godot	Motor de desarrollo de videojuegos usado en este proyecto
Pixel Art	Estilo visual utilizado en el juego
SCRUM	Metodología ágil utilizada en el desarrollo del sistema

## 1.4. Referencias

Título	Autor/Organización	Relación	Año	Enlace
IEEE 830-1998	IEEE Computer Society	Plantilla	2008	<a href="https://standards.ieee.org/ieee/830/1222/">https://standards.ieee.org/ieee/830/1222/</a>

[1] Linietsky, J & Manzur, A. (2014). Godot Docs – 4.4 branch. Godot.  
<https://docs.godotengine.org/en/stable/>

## 1.5. Visión General

El documento actual se encuentra estructurado en cuatro secciones distintas. La primera sección corresponde a la introducción, donde se proporciona una explicación sobre el proyecto que será implementado, además de presentar cómo está organizado el documento Especificación de Requisitos del Sistema.

La segunda sección es la descripción general, la cual explica el entorno en el que los requerimientos operarán para llevar a cabo su función. El propósito es detallar todos los elementos que podrían influir en el sistema o en los propios requerimientos.

La tercera parte abarca los Requisitos Específicos. En esta sección, se realiza una descripción minuciosa de todos los requerimientos que serán incorporados en el sistema. Estos requisitos desempeñarán un papel fundamental en la realización de las diversas funciones requeridas por el usuario.

Finalmente, la cuarta sección lleva por título "Apéndices". Aquí se proporciona información adicional que tiene la finalidad de complementar aspectos específicos vinculados al desarrollo del sistema. Esto puede incluir detalles relacionados con el lenguaje de programación utilizado o análisis de costos relevantes.

## 2. Descripción General

Esta sección tiene como finalidad describir los factores que podrían afectar a las funcionalidades que deben implementarse en ProfitQuest.

## 2.1. Perspectiva del Producto

El producto que se desarrollará consiste en un juego educativo en 2D estilo pixel art que permitirá enseñar gestión de costos de producción y manejo de recursos en costos de producción a estudiantes que no tengan conocimiento en el área. Este juego funcionará como una herramienta de apoyo en el aprendizaje de conceptos económicos y administrativos mediante simulaciones interactivas.

ProfitQuest es un sistema independiente, el cual está diseñado para ejecutarse como una aplicación, sin necesidad de vinculación con plataformas externas. Utiliza una arquitectura basada en Godot Engine, lo que permite ofrecer una experiencia visual atractiva y fluida. ProfitQuest está diseñada para ejecutarse en computadoras, sin requerir una conexión a internet para su funcionamiento habitual. A nivel funcional, el sistema presenta una interfaz gráfica intuitiva que facilita la navegación del usuario y la administración de recursos dentro del juego. Se estructura en niveles de dificultad progresiva, donde cada nivel plantea nuevos desafíos relacionados con la optimización de costos y la toma de decisiones empresariales. El sistema también evalúa el desempeño del jugador mediante métricas detalladas, proporcionando retroalimentación basada en la eficiencia de la gestión realizada.

ProfitQuest se diferencia de otras herramientas educativas al combinar la enseñanza de conceptos económicos con mecánicas de juego, incentivando el aprendizaje activo y el desarrollo de habilidades analíticas. Su diseño modular permite futuras expansiones, incluyendo un posible modo multijugador o el diseño de nuevos niveles.

## 2.2. Funciones del Producto

El sistema ProfitQuest proporciona una serie de funcionalidades diseñadas para facilitar el aprendizaje sobre la gestión de costos de producción mediante mecánicas de juego interactivas. Entre sus funciones principales, se encuentra la simulación de producción, la cual permite a los jugadores gestionar la compra de materiales, fabricar productos y optimizar su rendimiento con el objetivo de maximizar beneficios. Además, el sistema cuenta con una progresión basada en niveles, donde se ofrecen distintos escenarios con retos de dificultad creciente.

Otra característica relevante es la evaluación de desempeño, que analiza la eficiencia del jugador en cada nivel y genera una retroalimentación sobre su rendimiento. Para motivar a los jugadores, el sistema incorpora un sistema de penalización y recompensas, otorgando

bonificaciones por una gestión eficiente y aplicando penalizaciones por errores cometidos en la producción. Finalmente, ProfitQuest ofrece una interfaz intuitiva que permite a los jugadores interactuar de manera sencilla y efectiva con los elementos del juego.

### 2.3. Características de Usuario

Los usuarios objetivo de ProfitQuest incluyen estudiantes de administración y economía, docentes y jugadores casuales. Los estudiantes de administración y economía representan el grupo principal, ya que el sistema está diseñado para personas sin experiencia previa en gestión de costos de producción y busca introducir estos conceptos mediante una metodología interactiva. Por otro lado, los docentes pueden utilizar la herramienta para complementar sus clases y evaluar el desempeño de sus estudiantes a través de métricas detalladas. Finalmente, los jugadores casuales interesados en juegos de simulación con mecánicas de optimización de recursos pueden encontrar en ProfitQuest una experiencia entretenida y educativa. Para garantizar una experiencia fluida, los usuarios deben poseer habilidades básicas en el uso de computadoras y navegación por interfaces gráficas, ya que el sistema se apoya en una interfaz visualmente intuitiva.

### 2.4. Restricciones

El sistema está sujeto a diversas restricciones que limitan su funcionalidad en la versión actual. En primer lugar, ProfitQuest no soportará modo multijugador, ya que está diseñado para experiencias individuales. Además, el sistema no contará con integración con plataformas educativas externas. Otra restricción importante es que no habrá una versión móvil en esta fase del desarrollo, lo que significa que el juego solo estará disponible en los sistemas operativos Windows, macOS y Linux. Finalmente, no se permitirá la modificación de niveles por parte de los jugadores, asegurando que la experiencia de aprendizaje siga una estructura predefinida.



## 2.5. Suposiciones y dependencias

El desarrollo de ProfitQuest se basa en una serie de suposiciones y dependencias clave. Se asume que los usuarios dispondrán de un dispositivo compatible con los requisitos mínimos del juego, lo que garantiza un rendimiento óptimo. Asimismo, el sistema se desarrollará utilizando Godot Engine, un motor de videojuegos que proporciona las herramientas necesarias para su implementación. Otra suposición relevante es que el juego no requiere conexión a internet para su funcionamiento, salvo para la descarga inicial y posibles actualizaciones. Esto permite a los jugadores disfrutar de la experiencia sin depender de una conexión constante.

## 2.6. Requisitos Futuros

En futuras versiones de ProfitQuest, se podrían implementar diversas mejoras y expansiones. Entre las características más relevantes se encuentra la inclusión de un modo multijugador, que permitiría que varios jugadores compitan o colaboren en la gestión de recursos. También se contempla la expansión de niveles, agregando nuevos escenarios con mecánicas avanzadas para aumentar la variedad de desafíos.

# 3. Requisitos Específicos

## 3.1. Interfaces Externas

En esta sección se describen las distintas interfaces que serán usadas por el juego. Esta aplicación cuenta con diversas interfaces, las cuales varían mucho en cuanto uso y diseño. En este apartado, se explican detalladamente como funcionan cada una de las interfaces.

### 3.1.1. Interfaces de Usuario

ProfitQuest contará con diversas interfaces de usuario que facilitarán la interacción con el sistema. El menú principal proporcionará acceso a opciones de inicio de partida,

configuración y tabla de clasificaciones. La pantalla de juego mostrará el entorno de producción, el inventario y los recursos disponibles, permitiendo a los jugadores gestionar su estrategia en tiempo real.

El sistema también incluirá un panel de gestión, donde los jugadores podrán administrar su presupuesto, materiales y productos. Adicionalmente, se implementará un menú de pausa que permitirá guardar la partida, modificar configuraciones o salir del juego sin perder el progreso.

La interfaz del juego tendrá un diseño enfocado en estética, calidad, intuición y calidad general al usuario. Algunos de los aspectos más importantes de la interfaz de usuario son:

- Paleta de colores: Se van a utilizar colores que sean fáciles y atractivos de ver. Los colores de los objetos en el juego serán relacionados con la realidad y los menús usarán colores que no distraigan mucho.
- Navegación intuitiva: Navegar los diferentes menús tiene que ser sencillo e intuitivo. El flujo de navegación entre pantallas e interfaces debe ser fluido.
- Jugabilidad: Las interfaces de juego deben de ser fluidas, intuitivas y debe de ser fácil interactuar con el ambiente del juego. Los movimientos de la cámara del juego tienen que tener sentido físico y permitir al jugador hacer lo que desea.

### 3.1.2. Interfaces de Software

Las interfaces de software se refieren a las interacciones y funcionalidades que se dan entre el sistema brindado al usuario y otros sistemas. En cuanto al juego ProfitQuest, se presentan las siguientes funcionalidades:

- Menú en el juego: Se permite al usuario entrar a un menú donde puede reiniciar la partida (reiniciando el estado del juego a su estado inicial) o salirse de la partida.
- Botón de guardado: Se permite al usuario guardar el estado actual del juego para poder continuarlo en otro momento, el estado se puede establecer cuando el usuario jugador desee.

- Pantalla de clasificación de jugadores: El sistema busca todos los puntajes de todos los usuarios que hayan completado un nivel y presenta en forma descendente a los usuarios para mostrar cuáles jugadores tuvieron el mejor desempeño en un dado nivel.
- Menú principal: Permite al usuario interactuar con todas las opciones y funcionalidades del juego.
- Ambiente de juego: Es el ambiente principal donde el usuario juega el juego. Se pueden lograr los diferentes objetivos y acciones permitidas por el juego y el usuario puede interactuar con el ambiente de forma libre en los límites del juego.

### 3.1.3. Interfaces de Hardware

Las interfaces de hardware se refieren a las interacciones que tendrá el equipo máquina con el software. Estas tienen como propósito recibir entradas del usuario para poder generar las funciones deseadas. Dentro de la interfaz de hardware se tiene:

- Acceso a sistemas de Input: Para la jugabilidad del juego, se deberá contar con acceso a un teclado y un mouse para poder interactuar con ella y todos los atributos de la aplicación.
- Requisitos de hardware: Para tener una buena experiencia de usuario se recomienda contar con una computadora con al menos 4 GB de RAM y que tenga una resolución de al menos 1024x768 píxeles.

### 3.1.4. Interfaces de Comunicación

La comunicación entre el sistema y el juego se debe llevar a cabo de forma eficiente y confiable. El sistema debe de completar las funciones necesarias para que el juego funcione correctamente y se presente la experiencia que se desea. Se presentan las siguientes características.

- Actualización en tiempo real: El juego debe de guardar partidas, calcular costos, limitar presupuestos y actualizar el estado del juego a tiempo real mientras el usuario elabora diferentes acciones dentro del juego. El sistema debe de reaccionar correctamente y de forma casi inmediata y fluida.

## 3.2. Requisitos Funcionales

Esta sección se enfoca en presentar los diferentes requisitos, su rol en la aplicación y su deseado funcionamiento en el sistema. Para cada una de las funciones se detalla su nombre, prioridad, detalles, autores, iniciadores, dependencias, restricciones, precondiciones o entradas, descripción de flujo, descripción de flujos alternos, postcondiciones o salidas y criterios de aceptación. Se presentan los requisitos funcionales:

### 3.2.1. PQ-01: Diseño de Niveles Preconfigurados

**Prioridad:** Alta

**Detalle:** 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.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:** No aplica

**Restricciones:**

- Cada nivel debe incluir al menos tres estaciones de trabajo, objetivos claros y recursos iniciales predefinidos.
- Los jugadores no pueden modificar la estructura del nivel.
- Los niveles deben cargarse sin errores y mantener su configuración predefinida en cada partida.

**Precondiciones:**

- El jugador ha desbloqueado el nivel según su progreso en el juego.
- El sistema ha cargado correctamente los niveles preconfigurados.

**Descripción del Flujo:**

Actor	Sistema
El jugador selecciona un nivel preconfigurado desde el menú.	

	Verifica que el nivel está desbloqueado y carga su configuración.
Inicia la partida en el nivel elegido.	
	Renderiza la escena con estaciones de trabajo, recursos iniciales y objetivos establecidos.

#### **Flujos Alternativos:**

- **Nivel bloqueado:** El sistema indica que el nivel no está disponible hasta cumplir los requisitos previos.
- **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.

#### **Postcondiciones:**

- El jugador ha completado o abandonado el nivel y su progreso se ha registrado.
- Se mantiene la configuración del nivel para futuras partidas.

#### **Criterios de Aceptación:**

- Los niveles preconfigurados se cargan sin errores y son jugables.
- Las estaciones y recursos definidos son consistentes en todas las partidas.
- La dificultad aumenta progresivamente según el orden de los niveles.

### **3.2.2. PQ-02: Movimiento de Personaje**

**Prioridad:** Alta

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

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:** No aplica

#### **Restricciones:**

- El movimiento debe estar limitado a áreas designadas por el juego.

#### **Precondiciones:**

- El jugador ha iniciado una partida.

**Descripción del Flujo:**

Actor	Sistema
Se presionan las teclas de movimiento definidas.	
	Actualiza la posición del personaje y muestra interacciones disponibles.

**Flujos Alternativos:**

- **Colisión con obstáculos:** El personaje se detiene y reproduce una animación de impacto.

**Postcondiciones:**

- La posición del jugador debe ser actualizada correctamente, ubicándose en la posición deseada al finalizar el movimiento.

**Criterios de Aceptación:**

- Tiempo de respuesta del movimiento
- Manejo adecuado de colisiones

### 3.2.3. PQ-03: Sistema de Penalización

**Prioridad:** Media

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

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Sistema

**Dependencias:** PQ-11: Gestión de Productos

**Restricciones:**

- Las penalizaciones se calculan en tiempo real.

**Precondiciones:**

- El jugador comete un error en producción

**Descripción del Flujo:**

Actor	Sistema
	Se detecta un error por parte del jugador.
	Se restan los recursos configurados, ya sea materiales o tiempo restante.
	Se muestra una notificación explicando el error y la sanción.

**Flujos Alternativos:**

- No aplica

**Postcondiciones:**

- Se aplican las sanciones necesarias y el juego continúa.

**Criterios de Aceptación:**

- Las penalizaciones se aplican dentro de un rango menor a 2 segundos tras detectar el error.

### 3.2.4. PQ-04: Manejo de Errores en Producción

**Prioridad:** Baja

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

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:** No aplica

**Restricciones:**

- No se recupera el tiempo perdido en el proceso
- La estación tiene un uso máximo de tres veces por nivel

**Precondiciones:**

- El jugador ha cometido un error y tiene reversiones disponibles.

**Descripción del Flujo:**

Actor	Sistema
-------	---------

El jugador crea un producto no deseado	
El jugador se mueve hacia la estación correspondiente y selecciona el producto por destruir	
	Devuelve el 50% del costo en materiales del producto correspondiente

#### **Flujos Alternativos:**

- **Sin usos disponibles:** El sistema muestra un mensaje: "Límite de reversiones alcanzado".

#### **Postcondiciones:**

- El jugador tiene los materiales recuperados en el inventario.
- El jugador no posee el producto destruido.

#### **Criterios de Aceptación:**

- Se aplica el límite de tres usos por nivel.
- El inventario del jugador es actualizado correctamente.

### **3.2.5. PQ-05: Dificultad Progresiva**

**Prioridad:** Media

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

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:** PQ-01: Diseño de Niveles Preconfigurados

#### **Restricciones:**

- La dificultad aumenta en cada nivel.

#### **Precondiciones:**

- El jugador ha completado el tutorial inicial y se encuentra jugando los niveles preconfigurados.

#### **Descripción del Flujo:**

Actor	Sistema
-------	---------



El jugador termina un nivel	
	Se coloca al jugador en el nivel siguiente
	Se le indica al jugador el presupuesto, materiales y productos por crear
El jugador inicia la partida	

#### **Flujos Alternativos:**

- **Puntuación insuficiente:** El jugador no logra terminar el nivel, no se le teletransporta al siguiente.

#### **Postcondiciones:**

- Los niveles posteriores incluyen mecánicas avanzadas y restricciones.

#### **Criterios de Aceptación:**

- Conforme mayor es el nivel, mayor dificultad posee.

### **3.2.6. PQ-06: Gestión de Presupuesto**

**Prioridad:** Alta

**Detalle:** Como función principal del juego, el usuario debe tener un presupuesto inicial en cada nivel, el cual se va a utilizar como recurso limitado para poder completar las diferentes acciones y objetivos del juego. Se debe de gestionar la cantidad, los efectos y las interacciones que hay utilizando el presupuesto en el juego

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Usuario Jugador

**Dependencias:** PQ-01: Diseño de Niveles Preconfigurados

#### **Restricciones:**

- Cada nivel debe de tener un presupuesto dado
- El presupuesto debe de estar calculado para tomar efecto en un nivel dado
- El presupuesto es independiente de cada nivel a otro
- El presupuesto no puede ser alterado o cambiado

#### **Precondiciones:**

- Los cálculos de gasto de presupuesto deben de existir.

- Se debe de haber iniciado el juego y entrado a un nivel.

#### **Descripción del Flujo:**

Actor	Sistema
El usuario entra al nivel preconfigurado	
	El sistema presenta al presupuesto del usuario en el nivel dado
El usuario usa el presupuesto como recurso limitado para poder llevar a cabo diferentes acciones en el juego.	
	El sistema reduce o incrementa el presupuesto, acorde a las acciones del usuario.
El usuario completa los objetivos dados en el nivel utilizando el presupuesto dado.	

#### **Flujos Alternativos:**

- El usuario se sale del nivel antes de ser completado, el presupuesto debe de reiniciarse.
- El usuario no completa el nivel satisfactoriamente por lo que el sistema resetea las condiciones y estados del nivel y del presupuesto.

#### **Postcondiciones:**

- Se finaliza el nivel sin excederse del presupuesto dado (Se completan los objetivos).
- El presupuesto se reduce y reacciona de forma acorde a las diferentes acciones que el usuario puede hacer cuando juega un nivel.

#### **Criterios de Aceptación:**

- Los niveles preconfigurados se cargan sin errores durante la ejecución del juego y presentan el presupuesto correspondiente.
- El presupuesto reacciona correspondientemente a las diferentes acciones del usuario, ya sea comprar materiales, pagar costos fijos, etc. (Incrementa o reduce).

- El sistema de gestión de presupuesto debe de denegar acciones que requieran cierta cantidad de dinero cuando el usuario tiene menos de esta cantidad.

### 3.2.7. PQ-07: Gestión del Inventario

**Prioridad:** Alta

**Detalle:** Cada nivel preconfigurado debe de venir con un inventario de materia prima que se usa para crear los diferentes bienes de cada nivel. El inventario debe de incrementar o disminuir dependiendo del uso de cada producto y las acciones del usuario. Es el inventario virtual en el juego y se debe poder gestionar como un inventario real.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Usuario Jugador

**Dependencias:**

- PQ-01: Diseño de Niveles Preconfigurados
- PQ-06: Gestión de Presupuesto

**Restricciones:**

- Cada producto en el inventario debe de corresponder a un costo específico para poder funcionar junto al presupuesto.
- Cada nivel tiene su inventario predeterminado e independiente a los otros niveles.

**Precondiciones:**

- El usuario debe de haber iniciado el nivel para generar el inventario.

**Descripción del Flujo:**

Actor	Sistema
El usuario entra al nivel preconfigurado	
	El sistema genera el inventario predeterminado con la materia prima necesaria para hacer el producto específico del nivel al que se entró
El usuario selecciona los objetos del inventario y los utiliza para crear los productos.	

	El sistema reduce el producto escogido por el usuario del inventario, se presenta visualmente el cambio (Reduce el número)
El usuario completa los objetivos del nivel haciendo uso del inventario.	

#### **Flujos Alternativos:**

- El usuario reinicia manualmente el nivel y el sistema debe de reiniciar el inventario a su estado original.
- El usuario no completa los objetivos y pierde el nivel, el sistema reinicia el nivel y el inventario vuelve a su estado original.
- El usuario intenta utilizar materiales que no existen en el inventario, el sistema no le permite esta acción.

#### **Postcondiciones:**

- Se finaliza el nivel utilizando únicamente los materiales disponibles en el inventario.

#### **Criterios de Aceptación:**

- El inventario debe de incrementar y reducir acorde a las acciones del usuario.
- El inventario debe de estar predefinido en cada nivel
- No se pueden utilizar materiales que se acabaron o ya no existen en el inventario.

### **3.2.8. PQ-08: Validación de desempeño por partida**

**Prioridad:** Alta

**Detalle:** Al finalizar un nivel, el sistema debe de recompilar todas las métricas necesarias para calcular el desempeño del jugador en un nivel dado. Se toma en cuenta los objetivos logrados ya sea el cálculo correcto de los costos, el uso correcto del presupuesto o el manejo óptimo del inventario. Se toman estos factores y el sistema decide si el jugador pasa o falla el nivel. Se redefinen los objetivos necesarios para pasar el nivel y se calcula si se lograron los objetivos en base a las métricas anteriores.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** El sistema/juego

**Dependencias:**

- PQ-01: Diseño de Niveles Preconfigurados

**Restricciones:**

- No se puede finalizar de calcular el desempeño si el nivel se reinicia, el cálculo también se reinicia.
- El cálculo del desempeño debe ser equitativo en cada nivel.

**Precondiciones:**

- El usuario debe de haber iniciado el nivel para poder iniciar los cálculos para definir el desempeño de la partida.
- El desempeño de cada nivel es independiente a los otros.
- Se deben de pre-definir los objetivos de cada nivel.

**Descripción del Flujo:**

Actor	Sistema
El usuario inicia un nivel	
	El sistema empieza a calcular el puntaje y desempeño según las acciones del usuario
El usuario finaliza el nivel	
	El sistema en base a la cantidad de objetivos que logra y qué tan lejos de los objetivos se encuentra, produce un puntaje y le presenta al usuario su desempeño en el nivel
	El sistema presenta los cálculos que hace para representar el desempeño
	El sistema decide si el usuario jugador puede continuar al siguiente nivel o si debe reiniciar el nivel y volver a intentar

**Flujos Alternativos:**

- Si el usuario reinicia el nivel, el sistema debe reiniciar los cálculos de desempeño e iniciar nuevamente.

**Postcondiciones:**

- El sistema presenta el desempeño del usuario en una partida dada y lo califica en un rango de 0 a 10.

**Criterios de Aceptación:**

- El sistema debe usar el mismo sistema de cálculos en base a los objetivos logrados en cada nivel.
- Debe de calcular correctamente el desempeño y si el usuario puede pasar de nivel.
- El sistema debe de guardar el desempeño del usuario.

### 3.2.9. PQ-09: Tabla de clasificación de jugadores

**Prioridad:** Alta

**Detalle:** La idea es clasificar el desempeño de los jugadores en un dado nivel basándose en el cálculo de desempeño por partida. Se debe de guardar el puntaje de cada usuario jugador diferente y presentarlo en una lista mostrando el mayor puntaje al menor por cada nivel.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** El sistema/juego

**Dependencias:**

- PQ-08: Validación de desempeño por partida

**Restricciones:**

- Las tablas de clasificación de puntaje son únicamente por niveles individuales.

**Precondiciones:**

- Un usuario debe de haber completado un nivel para que la tabla se genere.

**Descripción del Flujo:**

Actor	Sistema
El usuario inicia la aplicación y en el menú principal entra a la opción de tabla de clasificación y escoge el nivel que desea ver.	

	El sistema recolecta los datos sobre los puntajes de cada usuario que ha completado el nivel y los presenta en orden descendente de arriba para abajo.
El usuario puede ver los rankings y seleccionar un usuario específico para ver datos específicos sobre la partida en el que consiguió dicho puntaje	
	El sistema presenta los datos específicos del usuario seleccionado.

#### **Flujos Alternativos:**

- El usuario selecciona otro nivel y se presentan los datos de dicho nivel
- El usuario selecciona la opción de retroceder y el sistema lo devuelve al menú principal.

#### **Postcondiciones:**

- Se muestra la pantalla de la tabla de rankings.
- El usuario puede navegar por la pantalla con la tabla de rankings.

#### **Criterios de Aceptación:**

- La tabla de rankings sirve correctamente y presenta las posiciones de cada usuario según su mejor puntaje en un dado nivel.
- Se permite filtrar la tabla por nombre de usuario, nivel y puntaje.
- La tabla solo se muestra cuando al menos 1 jugador haya completado el nivel al que se quiere visualizar el ranking
- El sistema ordena correctamente los puntajes de cada usuario.

### **3.2.10. PQ-10: Objetivos opcionales por nivel**

**Prioridad:** Alta

**Detalle:** Además de los objetivos principales definidos en la preconfiguración de cada nivel, se van a agregar objetivos específicos y opcionales a cada nivel. Estos objetivos no afectarán la capacidad de completar el nivel del jugador, pero sí afectarán el cálculo de desempeño únicamente incrementándolo.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** El sistema/juego

**Dependencias:**

- PQ-01: Diseño de Niveles Preconfigurados
- PQ-08: Validación de desempeño por partida

**Restricciones:**

- No afecta la habilidad del usuario de pasar un nivel.
- No disminuye el desempeño si no se hacen, pero sí lo puede llegar a incrementar.

**Precondiciones:**

- El usuario debe de estar dentro de un nivel.
- Cada nivel tiene una lista distinta de objetivos opcionales-

**Descripción del Flujo:**

Actor	Sistema
El usuario entra a un nivel	
	El sistema presenta objetivos necesarios para pasar el nivel y adicionalmente objetivos opcionales.
El usuario completa alguno de los objetivos opcionales.	
	El sistema detecta el cumplimiento del objetivo opcional.
	El sistema calcula el desempeño del usuario y le suma puntos por cada objetivo opcional completado.

**Flujos Alternativos:**



- El usuario no completa ningún objetivo adicional, el sistema sirve normalmente.

**Postcondiciones:**

- El usuario puede llegar a recibir un puntaje más alto de lo normal si completa los objetivos opcionales en adición a los objetivos normales.

**Criterios de Aceptación:**

- Los objetivos opcionales son únicos en cada nivel.
- Los objetivos opcionales son tareas que se enfocan en atributos específicos de cada nivel.
- Los objetivos opcionales no afectan los objetivos normales del juego, son un tipo de contenido adicional.

### **3.2.11. PQ-11: Gestión de productos**

**Prioridad:** Alta

**Detalle:** El sistema debe permitir la administración de productos dentro del juego, asegurando que los jugadores puedan seleccionar, fabricar y gestionar productos en función de los recursos disponibles y los costos de producción. Cada producto tendrá características específicas que impactarán la estrategia del jugador.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:**

- No aplica

**Restricciones:**

- Cada producto debe tener atributos específicos, como costo de producción, tiempo de fabricación y demanda.
- Los productos deben estar predefinidos y no podrán ser creados o modificados por el usuario en esta versión.
- No se podrá fabricar un producto si el jugador no dispone de los materiales necesarios..

**Precondiciones:**

- El jugador ha desbloqueado la opción de fabricación de productos según su progreso en el juego.
- El sistema ha cargado correctamente la lista de productos disponibles.

**Descripción del Flujo:**

Actor	Sistema
El jugador accede al menú de productos.	
	Muestra la lista de productos disponibles junto con sus costos y requisitos.
El jugador selecciona un producto para fabricar.	
	Verifica si el jugador cuenta con los materiales necesarios.
	Si tiene los materiales, inicia el proceso de producción y descuenta los recursos utilizados.
	Una vez completada la fabricación, el producto terminado se agrega al inventario del jugador.

#### **Flujos Alternativos:**

- El jugador no tiene suficientes materiales, el sistema muestra una advertencia y sugiere comprar más recursos.
- Si el jugador cancela la fabricación, los materiales usados parcialmente se devuelven al inventario.

#### **Postcondiciones:**

- El producto ha sido fabricado y añadido al inventario del jugador.
- Se ha actualizado el estado de los recursos utilizados en la fabricación.

#### **Criterios de Aceptación:**

- La lista de productos se carga correctamente en el sistema.
- La creación del producto se realiza solo si se cumplen todos los requisitos.
- El proceso de fabricación debe reflejar el impacto en los recursos del jugador.

### 3.2.12. PQ- 12: Gestión de tipos de producto

**Prioridad:** Alta

**Detalle:** El sistema debe permitir la gestión de diferentes tipos de productos dentro del juego, cada uno con características únicas de producción, costos y demanda. Los productos deben pertenecer a categorías predefinidas y afectar la estrategia del jugador.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:**

- No aplica

**Restricciones:**

- El jugador ha avanzado lo suficiente para desbloquear nuevos tipos de productos.
- No se permite la modificación de las propiedades de los productos por parte del jugador.

**Precondiciones:**

- El jugador ha desbloqueado la opción de fabricación de productos según su progreso en el juego.
- El sistema ha cargado correctamente los datos de los productos en el juego.

**Descripción del Flujo:**

Actor	Sistema
El jugador selecciona un producto para fabricar.	
	Verifica si el producto está disponible según el progreso del jugador.
	Muestra las características del producto, incluyendo costos y materiales requeridos.
El jugador inicia la fabricación.	
	Deduce los recursos del inventario y actualiza el progreso de producción.
Finaliza la producción.	

	Entrega el producto terminado y actualiza la puntuación del jugador.
--	--

#### **Flujos Alternativos:**

- El producto aún no ha sido desbloqueado, el sistema muestra un mensaje indicando los requisitos necesarios para acceder a él.
- El jugador no tiene suficientes materiales, el sistema advierte que no puede iniciar la producción y sugiere adquirir más recursos.

#### **Postcondiciones:**

- El producto ha sido fabricado con éxito y añadido al inventario del jugador.
- Se registra el uso de recursos y costos asociados.

#### **Criterios de Aceptación:**

- La lista de productos se carga correctamente en el sistema.
- La creación del producto se realiza solo si se cumplen todos los requisitos.
- El proceso de fabricación debe reflejar el impacto en los recursos del jugador.

### **3.2.13. PQ-13: Entrega de productos**

**Prioridad:** Alta

**Detalle:** El sistema debe permitir que los productos terminados sean entregados en una zona de salida designada. Esta acción marcará la finalización del ciclo de producción y otorgará recompensas o puntuaciones al jugador según su desempeño.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

#### **Dependencias:**

- PQ-11: Gestión de Productos

#### **Restricciones:**

- Los productos solo pueden ser entregados si están completamente fabricados.
- La entrega debe realizarse en una zona específica dentro del juego.

#### **Precondiciones:**

- El jugador ha finalizado la fabricación del producto.
- El sistema ha verificado que el producto está listo para su entrega.

#### **Descripción del Flujo:**

Actor	Sistema
El jugador transporta el producto terminado a la zona de salida.	
	Verifica que el producto esté completo y listo para su entrega.
El jugador confirma la entrega.	
	Procesa la entrega y otorga la recompensa o puntuación correspondiente.
	Actualiza el inventario del jugador eliminando el producto entregado.

#### **Flujos Alternativos:**

- El producto no está terminado, el sistema muestra un mensaje de error y no permite la entrega.
- Si el producto no cumple con los requisitos de calidad, el sistema notifica al jugador y aplica una penalización.

#### **Postcondiciones:**

- El producto ha sido entregado y retirado del inventario del jugador.
- Se ha otorgado una puntuación o recompensa al jugador según su desempeño.

#### **Criterios de Aceptación:**

- Los productos solo pueden ser entregados si cumplen con los requisitos establecidos.
- La entrega debe otorgar recompensas basadas en la eficiencia y calidad del producto.

### **3.2.14. PQ-14: Compra de materiales**

**Prioridad:** Alta

**Detalle:** El sistema permitirá a los jugadores realizar la compra de materiales necesarios para la producción dentro del juego. La compra se realizará utilizando los recursos obtenidos a través de las mecánicas del juego, y los materiales adquiridos serán almacenados en el inventario del jugador. Se debe garantizar que el usuario pueda visualizar el costo de cada

material antes de confirmar la compra y que el sistema actualice correctamente el inventario y los recursos disponibles después de cada transacción.

**Iniciador:** Jugador

**Dependencias:**

- **PQ-08:** Validación de desempeño por partida
- **PQ-07:** Gestión del inventario

**Restricciones:**

- Solo se podrán comprar materiales si el jugador dispone de recursos suficientes.
- La compra de materiales estará limitada a los niveles donde esta mecánica sea aplicable.
- No se permitirá la compra de materiales si el inventario del jugador está lleno.

**Precondiciones:**

- El usuario debe estar en una partida activa.
- El jugador debe haber acumulado recursos suficientes para realizar la compra.
- Debe haber materiales disponibles en la tienda del juego.

**Descripción del Flujo:**

Actor	Sistema
El usuario accede a la tienda de materiales dentro del juego.	
	El sistema muestra la lista de materiales disponibles junto con sus costos.
El usuario selecciona los materiales que desea comprar.	
	El sistema verifica que el jugador tenga suficientes recursos y que haya espacio en su inventario.
Si la compra es válida, el usuario confirma la transacción.	
	El sistema descuenta los recursos correspondientes y añade los materiales al

	inventario del jugador.
Se muestra un mensaje confirmando la compra exitosa.	
	El sistema actualiza la información del inventario y los recursos disponibles del jugador.

### **Flujos Alternativos:**

- **El usuario no tiene suficientes recursos:**
  - El sistema muestra un mensaje indicando que los recursos son insuficientes y la compra no se procesa.
- **El inventario del usuario está lleno:**
  - El sistema muestra una advertencia y no permite la compra.
- **El usuario cancela la compra antes de confirmar:**
  - El sistema no realiza ninguna modificación en los recursos ni el inventario.
- **El usuario regresa al menú principal sin realizar compras:**
  - El sistema retorna al menú sin modificar los recursos o el inventario.

### **Postcondiciones:**

- El sistema ha actualizado correctamente el inventario del jugador y los recursos disponibles tras la compra.
- Se ha registrado la transacción en el historial del usuario.
- El usuario puede continuar con el juego y utilizar los materiales adquiridos.

### **Criterios de Aceptación:**

- El sistema solo permite la compra si el usuario tiene suficientes recursos.
- El inventario del usuario se actualiza correctamente tras cada compra.
- Se muestra un mensaje de confirmación tras cada compra exitosa.
- Se prohíbe la compra si el inventario del usuario está lleno.
- Se permite visualizar información detallada sobre cada material antes de la compra.

### 3.2.15. PQ-15: Tutorial Inicial

**Prioridad:** Alta

**Detalle:** El sistema debe proporcionar un tutorial interactivo que guíe al jugador a través de las mecánicas básicas del juego en su primera sesión. El tutorial debe presentar de manera progresiva los conceptos clave, incluyendo la compra de materiales, la producción, la gestión de costos y la toma de decisiones estratégicas. Este tutorial debe ser obligatorio para nuevos jugadores, con la opción de omitir en partidas posteriores.

**Iniciador:** El sistema/juego

**Dependencias:**

- PQ-01: Diseño de Niveles preconfigurados
- PQ-06: Gestión de Presupuesto
- PQ-07: Gestión del Inventario
- PQ-11: Gestión de Productos
- PQ-14: Compra de Materiales

**Restricciones:**

- El tutorial solo se activa en la primera sesión del jugador.
- Se debe permitir omitir el tutorial después de la primera ejecución.
- El tutorial debe adaptarse al progreso del jugador, asegurando que cada mecánica se practique antes de avanzar a la siguiente.

**Precondiciones:**

- El jugador ha iniciado sesión en el juego por primera vez.
- No existe un progreso previo registrado en la cuenta del usuario.
- El sistema ha cargado correctamente los elementos necesarios para la simulación del tutorial.

**Descripción del Flujo:**

Actor	Sistema
El usuario inicia el juego por primera vez.	
	El sistema detecta que es una nueva sesión y



	activa el tutorial.
	El sistema guía al usuario a través de las mecánicas básicas.
	Se presentan instrucciones paso a paso y acciones requeridas.
El usuario completa cada sección del tutorial.	
	El sistema valida que las acciones del usuario sean correctas antes de avanzar.
Una vez finalizado el tutorial, se notifica al usuario que puede comenzar a jugar.	
	El sistema guarda el progreso y desactiva la repetición automática del tutorial.

### Flujos Alternativos:

- **El usuario omite el tutorial:**
  - El sistema registra la omisión y redirige al jugador al menú principal.
- **El usuario no completa una sección del tutorial:**
  - El sistema evita avanzar hasta que el usuario realice la acción requerida.
- **El usuario cierra el juego antes de terminar el tutorial:**
  - El sistema guarda el progreso del tutorial y lo reanuda en la siguiente sesión.

### Postcondiciones:

- El usuario ha sido introducido correctamente a las mecánicas básicas del juego.
- Se ha registrado que el tutorial fue completado o omitido en la cuenta del usuario.
- El sistema ya no mostrará el tutorial en futuras sesiones, salvo que el usuario lo reinicie manualmente.

### Criterios de Aceptación:

- El tutorial se activa solo en la primera sesión del usuario.
- Se presenta de manera secuencial y solo avanza cuando el usuario completa cada paso.
- Se permite omitir el tutorial tras la primera sesión.
- Se guardan correctamente los datos de progreso del usuario.
- El tutorial no se repite automáticamente después de completarlo.

### 3.2.16. PQ-16: Tiempo límite

**Prioridad:** Alta

**Detalle:** El sistema debe establecer un tiempo límite para cada nivel dentro del juego, asegurando que los jugadores gestionen eficientemente los recursos y tomen decisiones estratégicas en un tiempo determinado. Una vez agotado el tiempo, el nivel finalizará automáticamente y se evaluará el desempeño del jugador en función de las decisiones tomadas y los resultados obtenidos.

**Autores:** Equipo de desarrollo de ProfitQuest

**Iniciador:** El sistema/juego

**Dependencias:**

- PQ-06: Gestión de Presupuesto
- PQ-07: Gestión del Inventario
- PQ-08: Validación de desempeño por partida
- PQ-17: Guardar partida

**Restricciones:**

- Cada nivel tendrá un tiempo predefinido, que variará en función de su complejidad.
- El tiempo límite no podrá ser extendido ni pausado por el usuario.
- Una vez alcanzado el tiempo límite, se deshabilitarán las interacciones del jugador y se generará la evaluación del desempeño.

**Precondiciones:**

- El usuario ha iniciado un nivel dentro del juego.
- El temporizador del nivel se encuentra en estado activo.
- El sistema ha cargado correctamente los parámetros de tiempo del nivel seleccionado.

**Descripción del Flujo:**

Actor	Sistema
-------	---------

El usuario inicia un nivel en el juego.	
	El sistema activa el temporizador y lo muestra en la pantalla.
El usuario gestiona sus recursos y toma decisiones durante el nivel.	
	El sistema reduce el tiempo progresivamente hasta llegar a cero.
Si el jugador completa el nivel antes de agotar el tiempo, finaliza la partida.	
	El sistema detiene el temporizador y evalúa el desempeño del jugador.
Si el tiempo llega a cero antes de completar el nivel, la partida finaliza automáticamente.	
	El sistema muestra un mensaje de finalización y genera la evaluación del jugador.

#### Flujos Alternativos:

- **El usuario completa el nivel antes de que el tiempo se agote:**
  - El sistema detiene el temporizador y genera la evaluación final.
- **El tiempo llega a cero antes de que el usuario complete el nivel:**
  - El sistema finaliza el nivel automáticamente y evalúa el desempeño con base en el progreso alcanzado.
- **El usuario cierra el juego antes de que termine el nivel:**
  - El sistema guarda el progreso y reanuda el nivel en la siguiente sesión con el tiempo restante.

#### Postcondiciones:

- El nivel finaliza al alcanzar el tiempo límite o cuando el jugador completa los objetivos.
- Se genera un reporte de desempeño basado en la eficiencia de las decisiones tomadas.
- El sistema registra los resultados y permite la visualización del puntaje obtenido.

**Criterios de Aceptación:**

- El temporizador inicia automáticamente al comenzar el nivel y se muestra en pantalla.
- El sistema detiene todas las interacciones del jugador cuando el tiempo se agota.
- El tiempo límite es fijo y no puede ser modificado por el usuario.
- La evaluación del desempeño se genera correctamente tras finalizar el nivel.
- Si el jugador guarda el juego, el sistema guarda el tiempo restante y lo reanuda en la siguiente sesión.

### **3.2.17. PQ-17: Guardar partida**

**Prioridad:** Alta

**Detalle:** El sistema debe permitir a los jugadores guardar su progreso en cualquier momento dentro del juego. Esto asegurará que puedan retomar la partida desde el último punto guardado sin perder avances. La funcionalidad de guardado incluirá el estado actual de los recursos, inventario, nivel en curso y tiempo restante. El usuario podrá acceder a partidas guardadas desde el menú principal y continuar el juego sin interrupciones.

**Iniciador:** Jugador

**Dependencias:**

- PQ-06: Gestión de Presupuesto
- PQ-07: Gestión del Inventario
- PQ-16: Tiempo límite

**Restricciones:**

- Solo se permitirá guardar la partida fuera de eventos críticos (como la evaluación de desempeño o la finalización de un nivel).
- El usuario podrá tener un número limitado de archivos de guardado según lo definido por la configuración del sistema.

- El guardado debe ser automático al cierre del juego o en momentos clave predefinidos.

**Precondiciones:**

- El jugador debe estar dentro de una partida activa.
- No debe encontrarse en un evento crítico (evaluación de desempeño o fin de nivel).
- El sistema debe tener espacio de almacenamiento disponible para registrar el progreso.

**Descripción del Flujo:**

Actor	Sistema
El usuario accede al menú de pausa y selecciona la opción de guardar partida.	
	El sistema recopila el estado actual de los recursos, inventario, nivel en curso y tiempo restante.
El usuario confirma el guardado.	
	El sistema almacena los datos de la partida en un archivo de guardado.
Se muestra una notificación confirmando que la partida ha sido guardada exitosamente.	
	El sistema permite continuar el juego o regresar al menú principal.

**Flujos Alternativos:**

- **El usuario intenta guardar durante un evento crítico:**
  - El sistema muestra un mensaje indicando que no es posible guardar en ese momento.

- **El usuario supera el límite de archivos de guardado disponibles:**
  - El sistema solicita eliminar un archivo existente antes de guardar uno nuevo.
- **El usuario cierra el juego abruptamente (sin guardar manualmente):**
  - El sistema activa el guardado automático y almacena el progreso más reciente.
- **El usuario selecciona una partida guardada desde el menú principal:**
  - El sistema carga el estado almacenado y permite continuar el juego desde el último punto guardado.

**Postcondiciones:**

- El sistema ha almacenado correctamente el estado de la partida en el archivo de guardado.
- El usuario puede continuar la partida desde el último punto guardado sin pérdida de progreso.
- Se ha actualizado la lista de partidas guardadas disponibles.

**Criterios de Aceptación:**

- El usuario puede guardar la partida manualmente desde el menú de pausa.
- Se genera un archivo de guardado con el estado exacto del juego en ese momento.
- El guardado automático se activa correctamente en eventos clave o al cierre del juego.
- El sistema carga correctamente una partida guardada y permite continuar el juego sin problemas.
- Se muestra un mensaje de confirmación cuando el guardado es exitoso.

### **3.2.18. PQ-18: Configuración del avatar**

**Prioridad:** Media

**Detalle:** El sistema debe permitir a los jugadores personalizar su avatar dentro del juego. La personalización incluirá la selección de apariencia básica, como colores, atuendos y

accesorios. Los cambios realizados en la configuración del avatar deben guardarse automáticamente para su uso en futuras partidas.

**Iniciador:** Jugador

**Dependencias:**

- PQ-17: Guardar partida

**Restricciones:**

- La configuración del avatar no afecta el rendimiento del juego ni otorga ventajas competitivas.
- Los jugadores solo podrán personalizar su avatar dentro de las opciones predefinidas por el sistema.
- La personalización se debe realizar antes de iniciar una partida o desde el menú de configuración.

**Precondiciones:**

- El usuario ha iniciado sesión en el juego.
- El sistema ha cargado correctamente las opciones de personalización disponibles.
- El jugador accede al menú de configuración del avatar.

**Descripción del Flujo:**

Actor	Sistema
El usuario accede al menú de configuración del avatar.	
	El sistema carga las opciones disponibles de personalización.
El usuario selecciona características como color, atuendo y accesorios.	
	El sistema actualiza la vista previa del avatar con los cambios seleccionados.
El usuario confirma la personalización.	
	El sistema guarda automáticamente la

	configuración del avatar en el perfil del jugador.
El usuario regresa al menú principal o inicia una partida.	
	El sistema mantiene los cambios aplicados en futuras sesiones de juego.

### Flujos Alternativos:

- **El usuario cancela la personalización sin guardar:**
  - El sistema descarta los cambios y mantiene la configuración anterior del avatar.
- **El usuario intenta aplicar una configuración no permitida:**
  - El sistema muestra un mensaje de error y no aplica los cambios.
- **El usuario selecciona una opción bloqueada o no desbloqueada:**
  - El sistema notifica que la opción aún no está disponible.
- **El usuario cambia de avatar y luego inicia una partida:**
  - El sistema guarda automáticamente los cambios y usa la nueva configuración en la partida.

### Postcondiciones:

- El usuario puede acceder al menú de personalización en cualquier momento antes de iniciar una partida.
- Se permite seleccionar entre las opciones predefinidas sin afectar el rendimiento del juego.
- Los cambios realizados en la configuración del avatar se guardan automáticamente.
- La personalización del avatar se refleja correctamente en el juego.
- Se impide la selección de opciones bloqueadas o no disponibles.

### Criterios de Aceptación:



- La tabla de rankings sirve correctamente y presenta las posiciones de cada usuario según su mejor puntaje en un dado nivel.
- Se permite filtrar la tabla por nombre de usuario, nivel y puntaje.
- La tabla solo se muestra cuando al menos 1 jugador haya completado el nivel al que se quiere visualizar el ranking
- El sistema ordena correctamente los puntajes de cada usuario.

### 3.2.19. PQ-19: Gestión de usuarios

**Prioridad:** Alta

**Detalle:** El sistema debe permitir la gestión de usuarios, asegurando que cada jugador tenga un perfil con datos de progreso, configuraciones personalizadas y estadísticas de desempeño. Esto permitirá guardar y recuperar la información de cada jugador en sesiones distintas.

Autores: Equipo de desarrollo de ProfitQuest

**Iniciador:** Jugador

**Dependencias:**

- No aplica.

**Restricciones:**

- Cada usuario debe contar con un perfil único dentro del sistema.
- El sistema debe permitir múltiples perfiles en un mismo dispositivo, pero cada sesión debe gestionarse de forma independiente.
- No se requiere integración con cuentas en línea o redes externas en esta versión.

**Precondiciones:**

- El usuario ha creado o seleccionado un perfil al iniciar el juego.
- El sistema ha cargado correctamente la información del perfil del usuario.

**Descripción del Flujo:**

Actor	Sistema
El jugador inicia el juego.	
	Muestra la pantalla de selección de usuario.
El jugador crea un nuevo perfil o selecciona uno existente.	

El jugador accede al menú principal.	Permite la personalización de configuraciones y estadísticas de usuario.
--------------------------------------	--

#### **Flujos Alternativos:**

- Si el perfil del usuario no existe o está dañado, el sistema solicitará crear un nuevo perfil.
- Si hay varios usuarios en el mismo dispositivo, se mostrará una lista para seleccionar el perfil deseado.

#### **Postcondiciones:**

- El usuario ha iniciado sesión en su perfil y su progreso ha sido cargado correctamente.
- Cualquier configuración o progreso guardado se mantiene para la siguiente sesión.

#### **Criterios de Aceptación:**

- El sistema permite crear, seleccionar y administrar perfiles de usuario correctamente.
- La información del usuario se guarda y se recupera sin errores.
- La gestión de múltiples usuarios en un mismo dispositivo funciona sin interferencias entre perfiles.

### **3.2.20. PQ-20: Manejo de desperdicio**

**Prioridad:** Alta

**Detalle:** El sistema debe permitir que los jugadores reciclen materiales no utilizados o en exceso para recuperar parte del costo de producción. Esto incentivará la optimización de recursos y reducirá la acumulación innecesaria de materiales en el inventario del jugador.

**Iniciador:** Jugador

#### **Dependencias:**

- No aplica.

#### **Restricciones:**

- Solo se podrán reciclar materiales antes de ser utilizados en la producción.
- La recuperación del costo debe ser parcial (por ejemplo, entre el 30% y el 50% del costo original).
- No se podrá reciclar productos terminados ni materiales ya procesados.

#### **Precondiciones:**

- El jugador tiene materiales en exceso en su inventario.
- El sistema ha verificado que los materiales son reciclables.

#### **Descripción del Flujo:**

Actor	Sistema
El jugador accede al inventario y selecciona materiales en exceso.	
	Verifica si los materiales pueden ser reciclados y muestra la cantidad recuperable.
El jugador confirma el reciclaje.	
	Reduce los materiales del inventario y otorga la cantidad de dinero o puntos correspondiente.

#### **Flujos Alternativos:**

- Si el jugador intenta reciclar un material no permitido, el sistema muestra una notificación indicando que no es posible.
- Si el jugador no tiene materiales para reciclar, el sistema desactiva la opción de reciclaje..

#### **Postcondiciones:**

- El material reciclado ha sido eliminado del inventario.
- Se ha otorgado al jugador la cantidad correspondiente de dinero o puntos por el reciclaje.

#### **Criterios de Aceptación:**

- La cantidad de recursos recuperados se refleja correctamente en la economía del juego.
- No se permite el reciclaje de materiales ya utilizados o procesados.

### 3.3. Requisitos no funcionales

Esta sección describe los aspectos no funcionales del sistema *ProfitQuest*, los cuales definen atributos esenciales como rendimiento, experiencia de usuario, seguridad y escalabilidad, garantizando un funcionamiento óptimo del juego.

#### 3.3.1. Rendimiento

- El juego debe mantener una tasa de cuadros por segundo (FPS) mínima de 60 en hardware que cumpla con los requisitos recomendados.
- Los tiempos de carga entre pantallas y niveles no deben superar los 5 segundos en condiciones normales de hardware.
- La latencia en la respuesta de entrada del jugador no debe exceder los 100 milisegundos.
- La transición entre menús y niveles debe realizarse de manera fluida, sin retrasos perceptibles para el usuario.

#### 3.3.2. Experiencia de Usuario

- La interfaz gráfica debe ser intuitiva y fácil de navegar para jugadores sin experiencia previa.
- Los elementos interactivos deben proporcionar retroalimentación visual y/o sonora al ser seleccionados.
- El diseño visual debe ser consistente con el estilo *pixel art*, asegurando una estética atractiva y uniforme en todo el juego.
- El tutorial inicial debe ser claro y progresivo, evitando sobrecargar al jugador con demasiada información en poco tiempo.

#### 3.3.3. Estabilidad

- El juego no debe presentar bloqueos o fallos críticos que afecten la experiencia del usuario.
- Si ocurre un fallo inesperado, el sistema debe guardar automáticamente el progreso reciente del usuario para evitar pérdida de datos.

- El sistema debe manejar correctamente errores de ejecución, evitando interrupciones abruptas del juego.

#### **3.3.4. Portabilidad**

- *ProfitQuest* debe ser compatible con Windows, macOS y Linux en sus versiones más utilizadas.
- El juego debe adaptarse correctamente a distintas resoluciones de pantalla, con un mínimo de 1280x720 píxeles.
- La configuración de controles debe permitir el uso de teclado y mouse, con opción a compatibilidad con controladores externos.

#### **3.3.5. Seguridad**

- Los datos de los jugadores, incluyendo progreso y configuraciones, deben almacenarse de manera segura para evitar pérdida o corrupción.
- No se deben exponer datos sensibles del usuario, asegurando que la información guardada en el sistema sea únicamente la necesaria para la experiencia de juego.
- Las partidas guardadas deben estar protegidas contra modificaciones externas que puedan comprometer la integridad del juego.

#### **3.3.6. Escalabilidad**

- El sistema debe permitir la adición de nuevos niveles, mecánicas y elementos sin afectar el funcionamiento del juego base.

#### **3.3.7. Mantenibilidad**

- La estructura del código debe facilitar actualizaciones y correcciones sin comprometer la estabilidad del sistema.
- Se deben aplicar buenas prácticas en la optimización del código y los recursos gráficos para garantizar un rendimiento estable en futuras versiones.

### 3.4. Restricciones de Diseño

Esta sección define las limitaciones y estándares técnicos que deben respetarse durante el desarrollo de *ProfitQuest*. Estas restricciones garantizan la coherencia en la implementación, optimización y compatibilidad del sistema.

#### 3.4.1. Motor de Desarrollo

- *ProfitQuest* debe desarrollarse utilizando Godot Engine, asegurando compatibilidad con sus características y herramientas nativas.
- Se utilizará GDScript como lenguaje principal de programación, evitando el uso de otros lenguajes dentro del motor para mantener coherencia en el desarrollo.

#### 3.4.2. Estilo Visual y Gráfico

- El juego debe seguir un estilo pixel art, con una resolución base de 16x16 o 32x32 píxeles para los sprites.
- Se deben evitar elementos gráficos que rompan la estética retro del juego.
- La paleta de colores utilizada debe ser consistente a lo largo de los niveles y menús, siguiendo un esquema definido previamente.

#### 3.4.3. Interfaz y Experiencia de Usuario

- La interfaz gráfica de usuario (GUI) debe ser clara y minimalista, evitando elementos innecesarios que puedan distraer al jugador.
- Todos los menús deben ser navegables mediante teclado y mouse, asegurando accesibilidad en todos los dispositivos compatibles.
- Se debe garantizar que los elementos interactivos tengan retroalimentación visual o sonora para mejorar la experiencia del usuario.

#### **3.4.4. Resolución y Compatibilidad**

- ProfitQuest debe ser compatible con resoluciones de 1280x720 píxeles en adelante, adaptándose automáticamente a resoluciones mayores sin distorsionar los elementos gráficos.
- No se permitirá una resolución inferior a 1280x720 píxeles para evitar problemas de interfaz.
- El juego debe ejecutarse en Windows, macOS y Linux, sin dependencia de plataformas propietarias específicas.

#### **3.4.5. Mecánicas y Restricciones de Gameplay**

- No se permitirá una mecánica de juego que involucre elementos aleatorios que afecten el aprendizaje del usuario.
- La progresión de niveles debe seguir un orden estructurado, sin la opción de omitir niveles clave sin cumplir requisitos previos.
- La dificultad del juego debe estar diseñada para fomentar el aprendizaje, evitando picos de dificultad desbalanceados.

#### **3.4.6. Gestión de Datos y Guardado**

- Las partidas guardadas deben almacenarse en archivos de datos locales y protegidos para evitar modificaciones externas no autorizadas.
- El sistema de guardado debe garantizar la integridad de los datos, evitando la corrupción del archivo en caso de cierres inesperados.
- No se permitirá acceso del usuario a archivos internos del juego que puedan comprometer la estabilidad del sistema.

#### **3.4.7. Rendimiento y Optimización**

- El uso de texturas y sprites debe estar optimizado para evitar sobrecarga de memoria, utilizando técnicas como atlas de sprites y compresión de texturas.
- Las animaciones y efectos deben mantener la fluidez del juego, asegurando un rendimiento de 60 FPS en hardware recomendado.

- Se debe minimizar el uso de procesos en segundo plano para evitar consumo innecesario de recursos.

### 3.5. Atributos del Sistema

El sistema *ProfitQuest* está diseñado para proporcionar una experiencia de juego estable, eficiente y accesible, garantizando su correcto funcionamiento bajo diferentes condiciones.

En términos de fiabilidad, el juego debe asegurar que todas las mecánicas y sistemas internos operen de manera consistente, evitando fallos críticos que afecten la jugabilidad. Las partidas guardadas serán almacenadas de forma segura y el sistema deberá prevenir la corrupción de datos mediante procesos de validación y respaldos automáticos. En caso de un cierre inesperado, el juego restaurará el estado más reciente del usuario, minimizando la pérdida de progreso.

La portabilidad del sistema permitirá su ejecución en entornos de escritorio que utilicen Windows, macOS y Linux, asegurando compatibilidad con hardware de gama media. No será necesario software adicional para su instalación y ejecución, más allá de los requisitos estándar del sistema operativo. Además, la interfaz gráfica será adaptable a distintas resoluciones de pantalla, asegurando una correcta visualización en diferentes dispositivos con un mínimo de 1280x720 píxeles.

En cuanto a rendimiento, *ProfitQuest* debe mantener una tasa de 60 cuadros por segundo (FPS) en hardware recomendado, garantizando una experiencia fluida y sin retrasos. Los tiempos de carga entre niveles y secciones del juego no deben superar los 5 segundos en condiciones normales de uso. Las interacciones del usuario dentro del juego deben reflejarse de inmediato, con una latencia máxima de 100 milisegundos en la respuesta a comandos y acciones. Para optimizar el uso de recursos, se minimizará el consumo de memoria y procesamiento mediante técnicas de compresión de texturas y optimización del código.

En relación con la seguridad, *ProfitQuest* protegerá los datos del jugador evitando accesos no autorizados a los archivos del sistema. La información relacionada con el progreso del usuario, como niveles completados y configuraciones



personalizadas, será almacenada de forma encriptada o protegida para evitar modificaciones externas que puedan comprometer la integridad del juego. No se permitirá la manipulación directa de archivos de guardado sin pasar por los mecanismos internos del sistema.

Finalmente, el juego ha sido diseñado para ser mantenible y escalable. Su código seguirá una estructura modular que facilitará la incorporación de nuevos niveles, mecánicas y ajustes sin afectar la estabilidad del sistema. Se utilizarán estándares de programación en GDScript dentro del motor Godot Engine, garantizando un desarrollo ordenado y eficiente. Cualquier actualización futura podrá implementarse sin necesidad de rediseñar la arquitectura base del juego, permitiendo mejoras continuas en la experiencia del usuario.

## **3.6. Otros Requisitos**

### **3.6.1. Requisitos de Seguridad**

Los archivos de guardado y configuraciones del usuario deben estar protegidos contra modificaciones externas o corrupción. No se almacenará información personal sensible, y se evitará la ejecución de código no autorizado dentro del sistema.

### **3.6.2. Requisitos de Seguridad**

Se realizarán pruebas unitarias, de integración y de estrés para verificar el correcto funcionamiento del juego. Además, se evaluará la usabilidad con jugadores objetivo para asegurar una experiencia intuitiva y sin errores críticos.

### **3.6.3. Requisitos de Tiempo de Inactividad**

Las interrupciones por mantenimiento o actualización deben ser mínimas. En caso de cierre inesperado, el sistema debe restaurar la partida desde el último guardado sin pérdida significativa de progreso.

## 4. Apéndices

Repositorio de GitHub: Pendiente...

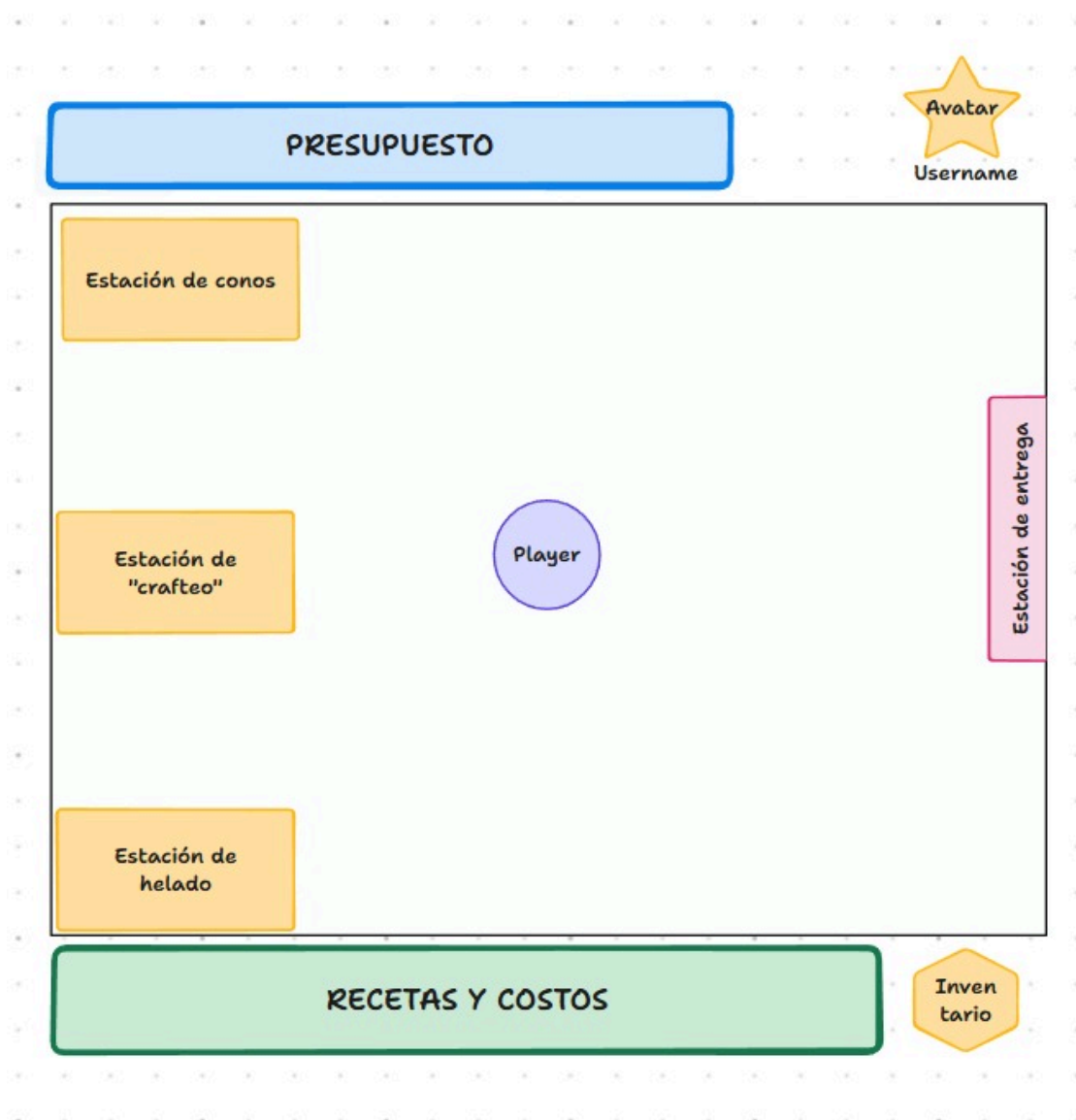


Figura 1: Planos del Ambiente del Juego

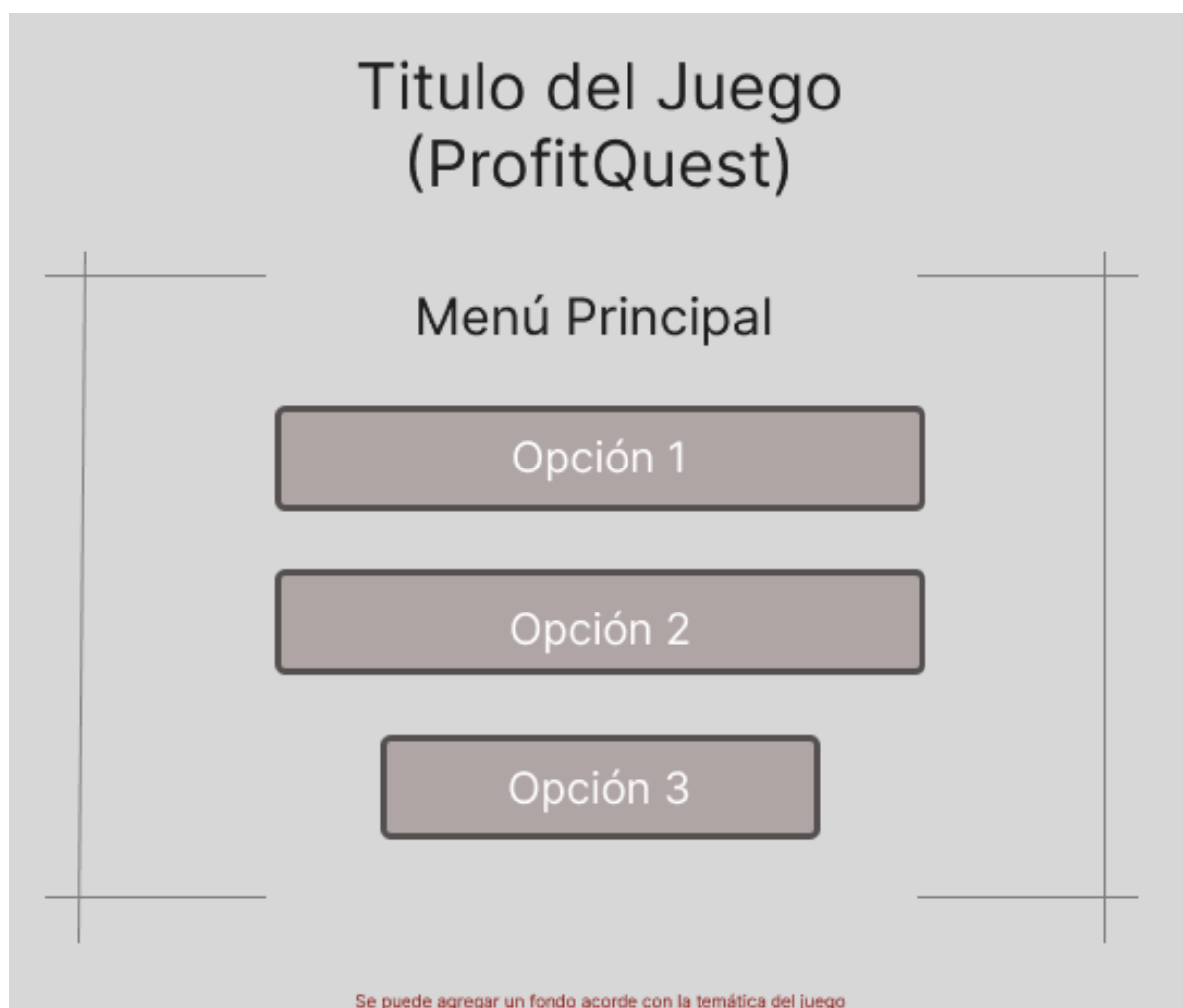


Figura 2: Planos del Menú principal