



**TEC** | Tecnológico  
de Costa Rica

Instituto Tecnológico de Costa Rica

Campus Tecnológico Central Cartago

Proyecto de Ingeniería de Software IC-7841

IS 2025

## **Plan de Pruebas - ProfitQuest**

### **Estudiantes:**

Leonardo Céspedes Tenorio c.2022080602

Kevin Chang Chang c.2022039050

Greivin Mauricio Fernández Brizuela c.2022437510

Daniel Alonso Garbanzo Carvajal c.2022117129

### **Profesora:**

Prof. Alicia Marcela Salazar Hernández

9 de junio, 2025

## Índice

Detalles del Proyecto	3
Atributos de calidad seleccionados	6
Plan de pruebas para ProfitQuest	8
Pruebas estáticas	11
Prueba de Completitud Funcional (Revisión de Requisitos y Diseño)	11
Prueba de Correctitud Funcional (Revisión de Código y Lógica):	11
Prueba de Mantenibilidad:	12
Evaluación de Deuda Técnica (Análisis de Código Estático):	13
Pruebas dinámicas	13
Pruebas Unitarias	13
Pruebas de integración	16
Pruebas de Sistema	20
Pruebas de aceptación de usuario	23
Criterios de Aprobación/Rechazo	26
Entregables	27
Tareas de Prueba Restantes	27
Necesidades Ambientales	28
Calendario	28
Criterios de Suspensión y Requisitos de Reanudación	29
Planificación de Riesgos y Contingencias	29
Aprobaciones	30
Glosario	30
Herramientas Utilizadas para las Pruebas	31
Resultados de Pruebas para “ProfitQuest”	34
Resultados de Pruebas Estáticas	34
Prueba de Completitud Funcional (Revisión de Requisitos y Diseño)	34
Prueba de Correctitud Funcional (Revisión de Código y Lógica)	35
Prueba de Mantenibilidad	35
Evaluación de Deuda Técnica (Análisis de Código Estático)	36
Resultados de Pruebas Dinámicas	36
Pruebas Unitarias	36
Pruebas de Integración	39
Pruebas de Sistema	44
Pruebas de aceptación de usuario	46
Recomendaciones por atributo de calidad	48
Resumen por tipo de prueba	49

# Detalles del Proyecto

**Objetivo:** Proporcionar una experiencia gamificada para el aprendizaje de conceptos de producción y gestión de costos a estudiantes sin formación previa. El sistema busca simular procesos de producción, desarrollar niveles con mecánicas progresivas, implementar un sistema de evaluación basado en la optimización de recursos, y ofrecer una experiencia de usuario atractiva y accesible

## Naturaleza de la aplicación

ProfitQuest es un juego educativo independiente desarrollado en 2D con estilo pixel art utilizando Godot Engine. Está diseñado para ejecutarse en computadoras (Windows, macOS, Linux) sin requerir conexión a internet para su funcionamiento habitual, salvo para descarga inicial y actualizaciones. Sus funciones principales incluyen la simulación de producción (compra de materiales, fabricación de productos, optimización de beneficios), progresión basada en niveles con dificultad creciente, un sistema de evaluación de desempeño con retroalimentación, y un sistema de penalización y recompensas. La interfaz gráfica es intuitiva, facilitando la navegación y la interacción con los elementos del juego.

## Población meta

La población meta de ProfitQuest incluye estudiantes de administración y economía (público principal), docentes que deseen complementar sus clases y evaluar el desempeño de los estudiantes, y jugadores casuales interesados en juegos de simulación con mecánicas de optimización de recursos. Los usuarios deben poseer habilidades básicas en el uso de computadoras y navegación por interfaces gráficas.

## Fuente

Proyecto académico del curso de Proyecto de Ingeniería de Software, I Semestre, 2025, en el Instituto Tecnológico de Costa Rica.

## Tecnologías

- Motor de Juego: Godot Engine.
- Lenguaje de Programación Principal: GDScript.
- Estilo Visual: Pixel Art.

- **Gestión de Datos:** Almacenamiento en archivos de datos locales para partidas guardadas y configuraciones.

### **Listado de requerimientos o servicios funcionales:**

Los requerimientos funcionales de ProfitQuest se han agrupado en los siguientes módulos clave para facilitar las pruebas:

### **Módulo de Simulación Económica y Producción:**

- **PQ-01: Diseño de Niveles Preconfigurados:** Acceso y juego de niveles estáticos con estaciones de trabajo, recursos, materiales, objetivos y restricciones fijas.
- **PQ-06: Gestión de Presupuesto:** Manejo de un presupuesto inicial limitado en cada nivel, con seguimiento de gastos e ingresos.
- **PQ-07: Gestión del Inventario:** Manejo de inventario de materia prima por nivel, con incremento/disminución según uso.
- **PQ-11: Gestión de Productos:** Permite seleccionar, fabricar y gestionar productos según recursos y costos, con atributos específicos.
- **PQ-12: Gestión de Tipos de Producto:** Permite gestionar diferentes tipos de productos con características únicas (producción, costos, demanda), predefinidos.
- **PQ-13: Entrega de Productos:** Permite entregar productos terminados en una zona designada, otorgando recompensas o puntuaciones.
- **PQ-14: Compra de Materiales:** Permite la compra de materiales necesarios para la producción, utilizando recursos del juego y almacenándolos en el inventario.
- **PQ-20: Manejo de Desperdicio:** Permite reciclar materiales no utilizados o en exceso para recuperar parte del costo de producción (30-50%).

### **Módulo de Jugabilidad y Progresión:**

**PQ-02: Movimiento de Personaje:** Capacidad de moverse por el entorno 2D usando teclas asignadas, limitado a áreas designadas.

**PQ-03: Sistema de Penalización:** Aplicación automática de sanciones por productos incorrectos o materiales erróneos (restando recursos/tiempo).

**PQ-04: Manejo de Errores en Producción:** Uso de una estación para deshacer errores (hasta 3 por nivel), recuperando 50% de materiales gastados.

**PQ-05: Dificultad Progresiva:** Niveles con dificultad creciente, accesibles tras completar el tutorial.

**PQ-08: Validación de Desempeño por Partida:** Recopilación de métricas al finalizar un nivel para calcular el desempeño y determinar el avance.

**PQ-09: Tabla de Clasificación de Jugadores:** Clasificación de jugadores por nivel basada en puntajes de desempeño, mostrando el mayor al menor.

**PQ-10: Objetivos Opcionales por Nivel:** Objetivos adicionales que incrementan el puntaje de desempeño sin afectar la capacidad de pasar el nivel.

**PQ-15: Tutorial Inicial:** Proporciona un tutorial interactivo obligatorio en la primera sesión, con opción de omitir en posteriores.

**PQ-16: Tiempo Límite:** Establece un tiempo límite por nivel para gestionar recursos y decisiones estratégicas.

#### **Módulo de Gestión de Usuarios y Personalización:**

**PQ-17: Guardar Partida:** Permite guardar el progreso en cualquier momento (recursos, inventario, nivel, tiempo restante) y retomar la partida.

**PQ-18: Configuración del Avatar:** Permite personalizar el avatar (apariencia básica, colores, atuendos, accesorios), guardando los cambios automáticamente.

**PQ-19: Gestión de Usuarios:** Permite gestionar perfiles de usuario únicos (datos de progreso, configuraciones, estadísticas), con soporte para múltiples perfiles en un dispositivo.

# Atributos de calidad seleccionados

Para ProfitQuest, se han seleccionado los siguientes atributos de calidad, fundamentales para su éxito como herramienta educativa interactiva:

- Rendimiento:
  - Métrica: Tiempo de respuesta promedio para interacciones clave y tasa de cuadros por segundo (FPS) promedio/mínima.
  - Criterios de Aceptación:
    - **Cumple totalmente:**
      - Tiempo de respuesta para interacciones clave (ej. fabricar producto, comprar material) menor a 300 ms
      - Tasa de cuadros por segundo (FPS) mínima de 60 FPS en hardware recomendado durante la jugabilidad.
      - Tiempos de carga entre pantallas y niveles no superan los 5 segundos.
      - Latencia en la respuesta de entrada del jugador no excede los 100 milisegundos.
    - **Cumple parcialmente:**
      - Tiempo de respuesta entre 300 ms y 600 ms.
      - FPS entre 30 y 59 FPS.
      - Tiempos de carga entre 5 y 10 segundos.
      - Latencia de entrada entre 100 ms y 200 ms
    - **No cumple:**
      - Tiempo de respuesta superior a 600 ms
      - FPS inferior a 30 FPS.
      - Tiempos de carga superiores a 10 segundos.
      - Latencia de entrada superior a 200 ms.
  - Se selecciona esta métrica, ya que, un rendimiento fluido es esencial para una experiencia de juego inmersiva y para la usabilidad, ya que las interacciones del jugador deben sentirse inmediatas y sin interrupciones.
- Seguridad:
  - Métrica: Número de vulnerabilidades detectadas en pruebas de integridad de datos guardados y configuraciones.
  - Criterios de Aceptación:

- **Cumple totalmente:** Sin vulnerabilidades críticas o mayores detectadas (ej. acceso no autorizado a archivos de guardado, corrupción de datos). Los archivos de guardado y configuraciones del usuario están protegidos contra modificaciones externas. No se expone información personal sensible.
  - **Cumple parcialmente:** Algunas vulnerabilidades menores detectadas, sin afectación de datos sensibles críticos. Puede haber vulnerabilidades menores en la protección de archivos de guardado que no comprometen la funcionalidad principal.
  - **No cumple:** Al menos una vulnerabilidad crítica o mayor detectada que compromete la integridad de datos o la experiencia de juego.
- Se selecciona esta métrica, ya que, es vital proteger el progreso y las configuraciones del jugador para evitar pérdida o corrupción de datos, garantizando una experiencia justa y confiable.
- Usabilidad:
  - Métrica: Tasa de éxito en la realización de tareas clave sin asistencia (porcentaje de usuarios que completan tareas clave sin ayuda) y claridad de la interfaz/feedback.
  - Criterios de Aceptación:
    - **Cumple totalmente:** Tasa de éxito del 90% o superior en tareas clave (ej. completar el tutorial, fabricar un producto, comprar materiales). La interfaz es intuitiva y fácil de navegar, con feedback visual/sonoro claro. El estilo visual (pixel art) es consistente y atractivo. El tutorial inicial es claro y progresivo.
    - **Cumple parcialmente:** Tasa de éxito entre el 70% y el 89%. Puede haber pequeños desafíos de navegación o claridad en algunos mensajes/elementos visuales.
    - **No cumple:** Tasa de éxito inferior al 70%. La interfaz es confusa, los controles son poco intuitivos o el feedback es ambiguo, lo que dificulta la interacción del jugador.
  - Justificación: La facilidad de uso es esencial para un juego educativo, asegurando que los jugadores puedan concentrarse en el aprendizaje de conceptos económicos en lugar de luchar con la interfaz del juego.

# Plan de pruebas para ProfitQuest

## Identificador del Plan de Pruebas:

Plan de Pruebas: ProfitQuest-2025

Fecha: 5 de junio de 2025

Versión: 1.0

## Referencias:

- Especificación de Requisitos de Software (ERS) para Juego Educativo: ProfitQuest (I Semestre, 2025).
- Estándar IEEE 829 para la documentación de pruebas de software.

## Introducción

Este plan de pruebas tiene como objetivo verificar y validar que el juego educativo "ProfitQuest" cumple con los requisitos funcionales y no funcionales establecidos en su Especificación de Requisitos de Software (ERS). Las pruebas abarcarán validaciones tanto estáticas (revisión de código y documentación) como dinámicas (ejecución del juego), e incluirán aspectos críticos como la funcionalidad de las mecánicas de juego, el rendimiento, la usabilidad y la seguridad de los datos del jugador. Se busca garantizar que todas las funcionalidades operen según lo esperado y que la experiencia educativa sea óptima y libre de errores.

## Elementos de Prueba:

Para "ProfitQuest", los elementos clave sujetos a prueba incluyen:

- **Motor de Juego y Código (Godot Engine, GDScript):** Verificación de la lógica de programación y la correcta implementación de todas las mecánicas.
- **Módulos de Jugabilidad:** Integridad y correcto funcionamiento de los sistemas de producción, inventario, presupuesto, penalización, recompensa y progresión de niveles.
- **Interfaz de Usuario (GUI):** Funcionalidad, estética y usabilidad de todos los menús, HUD y elementos interactivos del juego.
- **Activos del Juego:** Correcta carga y visualización de sprites, animaciones, fondos y efectos visuales (pixel art).
- **Sistemas de Entrada/Salida:** Respuesta de controles (teclado, ratón, gamepad) y funcionamiento del guardado/carga de partidas.
- **Audio:** Correcta reproducción y sincronización de efectos de sonido y música.



### **Problemas de Riesgo de Software:**

Se han identificado los siguientes problemas de riesgo que podrían afectar la calidad de "ProfitQuest":

- Bugs de colisión o movimiento errático del personaje en el entorno 2D.
- Errores lógicos en los cálculos de presupuesto, inventario o penalizaciones.
- Desbalance en la dificultad de los niveles o en las mecánicas de recompensa/penalización.
- Degradación del rendimiento (bajos FPS, tiempos de carga excesivos) en hardware de gama media o en niveles complejos.
- Corrupción o manipulación no autorizada de los archivos de partida guardada.
- Problemas de usabilidad derivados de una interfaz poco intuitiva o mensajes de error ambiguos.
- Inconsistencias visuales o de audio que rompan la inmersión del juego.

### **Características a Testear**

- **Funcionalidad:**
  - Mecánicas de Juego: Validación de la lógica de compra de materiales, fabricación de productos, gestión de inventario y presupuesto.
  - Progresión: Verificación del flujo entre niveles, el sistema de dificultad progresiva y la implementación de objetivos principales y opcionales.
  - Interacciones: Confirmación de que el personaje puede moverse libremente y que las interacciones con estaciones de trabajo, menús y objetos son correctas.
  - Guardado/Carga: Prueba de la persistencia del estado del juego (recursos, inventario, nivel, tiempo) mediante el guardado y la carga de partidas.
  - Feedback al Jugador: Verificación de las notificaciones de penalización, recompensa, mensajes de error y retroalimentación visual/sonora general.
  - Tutorial: Aceptación y completitud del flujo del tutorial inicial, incluyendo la opción de omitir.
  - Gestión de Datos: Correcto funcionamiento de la tabla de clasificación de jugadores y la personalización del avatar.

- **Seguridad:**

- Integridad de Archivos Guardados: Protección contra la modificación externa o corrupción de los archivos de progreso y configuraciones del jugador.
- Exposición de Datos: Asegurarse de que no se exponen datos sensibles del usuario.

- **Rendimiento:**

- Tasa de Cuadros por Segundo (FPS): Medición de la estabilidad de los FPS en diferentes escenarios y niveles del juego.
- Tiempos de Carga: Evaluación del tiempo que tarda el juego en cargar entre pantallas y niveles.
- Latencia de Entrada: Medición de la respuesta del juego a los comandos del jugador (teclado, ratón, gamepad).
- Optimización de Recursos: Observación del consumo de memoria y CPU/GPU durante la ejecución del juego.

- **Usabilidad:**

- Intuitividad de la Interfaz: Evaluar la facilidad de navegación a través de los menús y la comprensión de los elementos del HUD.
- Controles: Verificar la naturalidad y respuesta de los controles de movimiento y acción.
- Claridad del Feedback: Comprobar que los mensajes de error, advertencias y retroalimentación general son claros y útiles para el jugador.
- Consistencia Visual: Asegurar que el estilo pixel art y la paleta de colores son uniformes en todo el juego.

### **Características que no se van a Testear:**

- Modo multijugador.
- Integración con plataformas educativas externas.
- Versión para dispositivos móviles.
- Capacidad de modificación de niveles por parte de los jugadores.
- Recuperación de contraseñas de usuario (ProfitQuest no gestiona contraseñas en línea).

## Enfoque

El enfoque de las pruebas en "ProfitQuest" será integral, combinando metodologías estáticas y dinámicas para garantizar la calidad del producto.

## Pruebas estáticas

Estas pruebas se realizarán sin ejecutar el juego, enfocándose en la verificación del código, la documentación y los activos.

### Prueba de Completitud Funcional (Revisión de Requisitos y Diseño)

- **Objetivo:** Verificar que todos los requisitos funcionales y sus respectivas especificaciones están completamente documentados en el ERS y reflejados en los documentos de diseño de niveles, interfaz y lógica de negocio.
- **Método:**
  - Revisar el ERS para identificar todos los módulos y requerimientos (PQ-XX).
  - Comparar los detalles de cada requerimiento con los diseños específicos (ej. planos del ambiente del juego, diseño de la interfaz de usuario, lógica de presupuesto y penalización) para asegurar que todos los elementos necesarios están contemplados.
  - Verificar que las reglas de negocio (ej. cálculos de costos, fórmulas de penalización) están definidas claramente.
- **Criterios de Aceptación:**
  - Todos los requisitos funcionales y no funcionales del ERS están representados en la documentación de diseño del juego.
  - No hay ambigüedades o inconsistencias entre el ERS y los documentos de diseño.
  - Todas las restricciones y dependencias de los requerimientos están consideradas en el diseño.

### Prueba de Correctitud Funcional (Revisión de Código y Lógica):

- **Objetivo:** Validar que la lógica implementada en los scripts GDScript para las mecánicas principales es coherente con las especificaciones del sistema.

- **Método:**
  - Realizar revisiones de código (Code Reviews) por pares para los scripts GDScript clave.
  - Analizar las implementaciones de las reglas de negocio para confirmar su adherencia al ERS.
  - Utilizar herramientas de análisis estático (linters de GDScript si están disponibles) para identificar posibles errores de lógica o violaciones de estilo.
- **Criterios de Aceptación:**
  - La lógica de negocio en GDScript es consistente con los requisitos del ERS.
  - Los errores lógicos son manejados adecuadamente (ej. no se permite una acción si el presupuesto es insuficiente).
  - El código es legible y sigue convenciones de programación.

## Prueba de Mantenibilidad:

- **Objetivo:** Evaluar si la estructura del código del juego en Godot es modular, fácil de entender y puede mantenerse y extenderse con facilidad.
- **Método:**
  - Revisar la organización de los nodos, escenas y scripts en Godot para confirmar una separación clara entre la lógica del juego, la interfaz de usuario y los activos.
  - Verificar la existencia y calidad de los comentarios en el código y la documentación técnica de módulos críticos.
  - Identificar el nivel de acoplamiento entre los diferentes scripts y sistemas.
- **Criterios de Aceptación:**
  - El código está modularizado y bien documentado, facilitando futuras actualizaciones.
  - La lógica de juego está separada de la representación visual/UI.
  - La complejidad de las funciones críticas es manejable.

## Evaluación de Deuda Técnica (Análisis de Código Estático):

- **Objetivo:** Identificar problemas en el código fuente que puedan aumentar la deuda técnica del sistema a largo plazo, como duplicación, lógica acoplada o falta de optimización.
- **Método:**
  - Inspeccionar el código GDScript en busca de patrones repetitivos o funciones excesivamente complejas.
  - Revisar la gestión de recursos (texturas, sprites, memoria) para identificar oportunidades de optimización que podrían afectar el rendimiento.
  - Asegurarse de que los activos siguen las pautas de optimización.
- **Criterios de Aceptación:**
  - Se minimiza la duplicación de código.
  - Las funciones críticas están bien documentadas.
  - No se identifican problemas críticos de diseño o estructura que puedan impactar la estabilidad o el rendimiento a largo plazo.

## Pruebas dinámicas

Las pruebas dinámicas en "ProfitQuest" se centran en la ejecución del juego y la interacción con sus componentes para validar su comportamiento en tiempo real. Dada la naturaleza visual e interactiva del videojuego, las grabaciones de video, capturas de pantalla y logs serán las principales formas de evidencia para documentar los resultados de estas pruebas.

## Pruebas Unitarias

Las pruebas unitarias se enfocan en verificar el correcto funcionamiento de los componentes de software más pequeños y aislados del juego, como funciones específicas de scripts GDScript o nodos atómicos, que gestionan lógicas y cálculos fundamentales.

1. PU-001: Prueba de Cálculo de Reducción de Presupuesto
  - a. **Objetivo:** Verificar que el módulo de gestión de presupuesto (PQ-06) deduce la cantidad correcta de dinero al realizar una compra o acción que implique gasto.

b. **Método:**

- i. Acceder a un entorno de prueba donde se pueda inicializar un presupuesto específico para el jugador.
- ii. Simular una acción de compra o fabricación con un costo predefinido.
- iii. Verificar el valor final del presupuesto tras la deducción.
- iv. Repetir con diferentes valores de presupuesto inicial y costos de acción.

c. **Criterios de Aceptación:**

- i. El presupuesto final debe ser exactamente el esperado Presupuesto Inicial - Costo de Acción.
- ii. No deben presentarse valores negativos inesperados si la lógica lo prohíbe.

d. **Evidencia:** Grabación de pantalla breve si existe una interfaz de depuración para este valor.

2. PU-002: Prueba de Denegación por Presupuesto Insuficiente

a. **Objetivo:** Asegurar que el sistema de gestión de presupuesto (PQ-06) deniega acciones cuando el jugador no tiene recursos suficientes.

b. **Método:**

- i. Inicializar un presupuesto bajo que sea insuficiente para una acción de costo conocido.
- ii. Intentar realizar la acción que excede el presupuesto.
- iii. Verificar que la acción es denegada y no se realiza el gasto.
- iv. Confirmar que se muestra un mensaje de error o notificación adecuada.

c. **Criterios de Aceptación:**

- i. La acción debe ser bloqueada y el presupuesto no debe modificarse.
- ii. El sistema debe proporcionar una retroalimentación clara de la denegación.

d. **Evidencia:** Grabación de video mostrando el intento de acción y la denegación con su respectivo mensaje.

3. PU-003: Prueba de Actualización de Inventario por Consumo

a. **Objetivo:** Verificar que la cantidad de materiales en el inventario (PQ-07) se reduce correctamente cuando se utilizan para fabricar un producto.

b. **Método:**

- i. Inicializar el inventario del jugador con cantidades conocidas de materiales.
- ii. Simular la fabricación de un producto que requiere cantidades específicas de esos materiales.
- iii. Verificar que las cantidades de los materiales consumidos se reducen exactamente como se espera en el inventario.

c. **Criterios de Aceptación:**

- i. Las cantidades de los materiales en el inventario deben coincidir con Cantidad Inicial - Cantidad Consumida
- ii. No deben presentarse valores negativos o inconsistencias.

d. **Evidencia:** Grabación de pantalla si hay una interfaz de depuración de inventario.

4. PU-004: Prueba de Cálculo de Puntaje de Desempeño

a. **Objetivo:** Validar que la lógica de cálculo del puntaje de desempeño (PQ-08) procesa correctamente las variables (costos, presupuesto, objetivos) para determinar el resultado de la partida.

b. **Método:**

- i. Simular diferentes escenarios de fin de nivel con datos conocidos
- ii. Disparar la función de cálculo de desempeño.
- iii. Verificar que el puntaje y la decisión de pasar/fallar el nivel corresponden a los valores esperados según la lógica.

c. **Criterios de Aceptación:**

- i. El puntaje calculado debe ser preciso para cada combinación de variables de entrada.
- ii. La decisión de pasar/fallar el nivel debe ser consistente con el puntaje y los umbrales definidos.

d. **Evidencia:** Grabación de vídeo de la pantalla de desempeño.

5. PU-005: Prueba de Decremento del Temporizador

a. **Objetivo:** Verificar que el temporizador por nivel (PQ-16) se decrementa de forma precisa y continua.

b. **Método:**

- i. Inicializar el temporizador de un nivel con un valor conocido.
  - ii. Monitorear el valor del temporizador durante un período de tiempo.
  - iii. Verificar que el valor disminuye de forma constante y a la velocidad esperada.
- c. **Criterios de Aceptación:**
  - i. El temporizador debe decrementar con precisión.
  - ii. No debe haber saltos o inconsistencias en la cuenta regresiva.
- d. **Evidencia:** Grabación de video si el temporizador se muestra en una interfaz de depuración.

## Pruebas de integración

Las pruebas de integración se centran en verificar la correcta interacción y comunicación entre diferentes módulos o componentes del juego. Se busca asegurar que, al trabajar en conjunto, los sistemas se comportan como se espera y no introducen errores.

1. PI-001: Integración Compra de Materiales y Gestión de Presupuesto/Inventario
  - a. **Objetivo:** Validar que la compra de materiales (PQ-14) afecta correctamente el presupuesto (PQ-06) y actualiza el inventario (PQ-07).
  - b. **Método:**
    - i. Iniciar una partida con un presupuesto y un inventario iniciales conocidos.
    - ii. Acceder a la tienda de materiales y seleccionar un material para comprar.
    - iii. Confirmar la compra.
    - iv. Verificar que el presupuesto se ha reducido en el costo exacto del material y que la cantidad del material en el inventario ha aumentado.
    - v. Realizar una compra que supere el presupuesto y verificar que es denegada con mensaje apropiado.
  - c. **Criterios de Aceptación:**
    - i. El presupuesto y el inventario se actualizan de manera atómica y precisa después de una compra exitosa.



- ii. Las compras con presupuesto insuficiente son correctamente denegadas y notificadas al jugador.
  - d. **Evidencia:** Grabación de video del flujo de compra (interfaz de la tienda, confirmación, actualización visible de HUD)
2. PI-002: Integración Fabricación de Producto y Gestión de Inventario/Presupuesto
- a. **Objetivo:** Confirmar que la fabricación de un producto (PQ-11) consume los materiales correctos del inventario (PQ-07) y que los costos asociados (PQ-06) son descontados, además de añadir el producto terminado.
  - b. **Método:**
    - i. Asegurar que el inventario contiene los materiales necesarios para un producto específico.
    - ii. Seleccionar el producto para fabricar en la estación correspondiente.
    - iii. Iniciar el proceso de fabricación.
    - iv. Verificar que las cantidades de los materiales requeridos se reducen en el inventario y que el presupuesto se ha ajustado por los costos de fabricación.
    - v. Comprobar que el producto terminado aparece en el inventario de productos fabricados.
    - vi. Intentar fabricar un producto sin los materiales necesarios y verificar que es denegado.
  - c. **Criterios de Aceptación:**
    - i. El inventario de materiales y el presupuesto se actualizan correctamente tras la fabricación.
    - ii. El producto terminado se añade al inventario.
    - iii. La fabricación es bloqueada si los materiales son insuficientes.
  - d. **Evidencia:** Grabación de video del proceso de fabricación, mostrando las interfaces de inventario y presupuesto antes y después.
3. PI-003: Integración Sistema de Penalización y Manejo de Errores en Producción
- a. **Objetivo:** Validar que el sistema de penalización (PQ-03) interactúa correctamente con la estación de manejo de errores (PQ-04), permitiendo deshacer errores y recuperar materiales.
  - b. **Método:**

- i. Configurar un escenario en el juego donde se pueda cometer un error de producción que dispare una penalización.
- ii. Cometer el error y observar la aplicación de la penalización (ej. reducción de recursos).
- iii. Moverse a la estación de manejo de errores y activar la opción de deshacer el último error.
- iv. Verificar que se recupera el 50% de los materiales gastados en el error y que el contador de usos de la estación se decrementa.
- v. Repetir hasta alcanzar el límite de 3 usos y verificar que la estación deniega más usos.

c. **Criterios de Aceptación:**

- i. La recuperación de materiales es precisa (50% del costo original).
- ii. El contador de usos de la estación de errores se actualiza correctamente.
- iii. La estación de errores bloquea su uso una vez alcanzado el límite.

- d. **Evidencia:** Grabación de video del ciclo completo (cometer error -> penalización -> usar estación de deshacer -> verificación de inventario/contador).

4. PI-004: Integración Guardado de Partida y Carga de Nivel/Estado del Jugador

- a. **Objetivo:** Asegurar que el sistema de guardado de partida (PQ-17) almacena y recupera correctamente el estado del juego, incluyendo el nivel actual (PQ-01), el presupuesto (PQ-06), el inventario (PQ-07) y el tiempo restante (PQ-16).

b. **Método:**

- i. Iniciar un nivel, realizar algunas acciones (gastos, fabricaciones), y dejar el temporizador correr un poco para establecer un estado único.
- ii. Acceder al menú de pausa y seleccionar "Guardar partida".
- iii. Salir del juego completamente.
- iv. Reiniciar el juego y seleccionar "Cargar partida" desde el menú principal.
- v. Verificar que el juego se carga en el nivel correcto, con el presupuesto, inventario y tiempo exactamente como estaban al momento de guardar.

- vi. Realizar una prueba de guardado automático (ej. cerrando el juego abruptamente) y verificar la recuperación.

c. **Criterios de Aceptación:**

- i. Todos los elementos del estado del juego (nivel, presupuesto, inventario, tiempo) se cargan con valores idénticos a los guardados.
  - ii. El proceso de guardado y carga es fluido y sin errores.
  - iii. El guardado automático funciona como respaldo.
- d. **Evidencia:** Grabación de video del proceso de guardado, salida, y carga, mostrando la verificación de los valores en el HUD. Logs que confirmen los valores almacenados y recuperados.

5. PI-005: Integración Personalización de Avatar y Persistencia de Perfil de Usuario

- a. **Objetivo:** Validar que los cambios realizados en la configuración del avatar (PQ-18) se guardan automáticamente y persisten en el perfil de usuario (PQ-19) en futuras sesiones.

b. **Método:**

- i. Iniciar el juego y seleccionar un perfil de usuario existente.
- ii. Acceder al menú de configuración del avatar.
- iii. Realizar cambios en la apariencia del avatar (ej. color, atuendo).
- iv. Confirmar los cambios y salir al menú principal o iniciar una partida.
- v. Salir del juego y reiniciarlo.
- vi. Seleccionar el mismo perfil de usuario y verificar que el avatar mantiene la personalización aplicada.

c. **Criterios de Aceptación:**

- i. Los cambios en el avatar se guardan y se reflejan correctamente al reiniciar el juego con el mismo perfil.
  - ii. La personalización del avatar no afecta el rendimiento del juego.
- d. **Evidencia:** Grabación de video del proceso de personalización, salida y reinicio del juego, mostrando el avatar persistente. Capturas de pantalla del antes y después de la personalización.

6. PI-006: Integración Diseño de Niveles y Movimiento de Personaje

- a. **Objetivo:** Asegurar que el personaje (PQ-02) puede moverse libremente y colisiona correctamente con los elementos del nivel preconfigurado (PQ-01), respetando las áreas designadas y evitando glitches de colisión.
- b. **Método:**
  - i. Cargar un nivel preconfigurado que contenga diferentes tipos de terreno, obstáculos (ej. paredes, cajas, estaciones de trabajo) y límites definidos.
  - ii. Mover el personaje por todo el nivel utilizando las teclas de movimiento asignadas (arriba, abajo, izquierda, derecha, salto).
  - iii. Intentar mover el personaje a través de los límites del nivel y objetos estáticos (paredes, estaciones) para verificar la correcta colisión.
  - iv. Mover el personaje cerca de bordes, esquinas y objetos interactivos para detectar cualquier "clipping" (atravesar objetos) o comportamiento inesperado.
  - v. Realizar saltos sobre y alrededor de obstáculos para confirmar la física de colisión.
- c. **Criterios de Aceptación:**
  - i. El movimiento del personaje debe ser fluido y sin interrupciones visuales o físicas.
  - ii. El personaje debe colisionar de forma consistente con todos los objetos y límites del nivel.
  - iii. No deben presentarse glitches de colisión, como atravesamiento de objetos o "teletransportaciones" inesperadas.
  - iv. Las áreas de movimiento designadas deben ser respetadas por el personaje.
- d. **Evidencia:** Grabación de video de la jugabilidad en el nivel, mostrando los movimientos, interacciones con el entorno y cualquier anomalía de colisión o movimiento.

## Pruebas de Sistema

Validar que el sistema "ProfitQuest" en su totalidad cumple con todos los requisitos funcionales y no funcionales en un entorno que simula lo más fielmente posible el entorno de

producción. Se busca la estabilidad general, el rendimiento, la experiencia del juego de principio a fin, y la correcta interacción de todos sus módulos en un ciclo completo de juego.

1. PS-001: Flujo Completo del Juego (De Principio a Fin)

a. **Objetivo:** Verificar que el juego puede ser jugado desde el inicio hasta la finalización de varios niveles, confirmando la correcta integración de todas las mecánicas y sistemas sin fallos.

b. **Método:**

- i. Iniciar el juego desde cero (primera sesión de usuario).
- ii. Completar el tutorial inicial (PQ-15).
- iii. Progresar a través de al menos un nivel preconfigurado con dificultad progresiva (PQ-01, PQ-05), completando sus objetivos principales y opcionales (PQ-10).
- iv. Durante el juego, realizar acciones de gestión de producción: compra de materiales (PQ-14), fabricación de productos de varios tipos (PQ-11, PQ-12), gestión de presupuesto (PQ-06) e inventario (PQ-07), entrega de productos (PQ-13) y manejo de desperdicio (PQ-20).
- v. Intencionalmente cometer errores de producción y utilizar el sistema de manejo de errores (PQ-03, PQ-04).
- vi. Realizar operaciones de guardado y carga de partida (PQ-17) en diferentes puntos del flujo de juego.
- vii. Verificar que el tiempo límite (PQ-16) funciona correctamente en el nivel, tanto si se completa antes como si se agota.
- viii. Al finalizar el nivel, verificar la validación de desempeño (PQ-08) y la actualización de la tabla de clasificación (PQ-09).
- ix. Confirmar que la personalización del avatar (PQ-18) y los perfiles de usuario (PQ-19) persisten a lo largo de las sesiones.

c. **Criterios de Aceptación:**

- i. El juego debe permitir un avance fluido y sin bloqueos a través de todos los niveles y funcionalidades.
- ii. Todas las interacciones entre los módulos deben ser consistentes y correctas.
- iii. Los sistemas de penalización, recompensa y manejo de errores deben funcionar según lo esperado.

- iv. Los datos de progreso del jugador deben guardarse y cargarse de forma fiable en todo momento.
- d. **Evidencia:** Grabación de video de una sesión completa o de partes representativas de este flujo de juego, destacando las transiciones y la funcionalidad de cada módulo.

## 2. PS-002: Rendimiento General y Fluidez de Juego

- a. **Objetivo:** Medir y validar que el juego mantiene los estándares de rendimiento especificados, incluyendo FPS, tiempos de carga y latencia de entrada, en diversas condiciones operativas.
- b. **Método:**
  - i. Ejecutar el juego en máquinas que cumplan con los requisitos de hardware mínimos y recomendados.
  - ii. Utilizar herramientas de monitoreo para registrar los FPS promedio y mínimos durante sesiones de juego extensas.
  - iii. Medir los tiempos de carga entre el menú principal y cada nivel, así como entre diferentes escenas dentro de los niveles.
  - iv. Realizar pruebas de respuesta de entrada (latencia) para acciones clave del personaje (movimiento, interacción con objetos) usando teclado y ratón, y un gamepad si es compatible.
  - v. Simular escenarios de "alta carga visual" o "alta interacción" dentro de un nivel y monitorear el rendimiento.
- c. **Criterios de Aceptación:**
  - i. Los FPS promedio deben mantenerse en 60 FPS o más en hardware recomendado, y no caer por debajo de 30 FPS en hardware mínimo.
  - ii. Los tiempos de carga entre pantallas y niveles no deben superar los 5 segundos.
  - iii. La latencia de respuesta de entrada no debe exceder los 100 milisegundos.
  - iv. El consumo de recursos (CPU, GPU, RAM) debe ser razonable y no excesivo para un juego 2D.
- d. **Evidencia:** Grabaciones de video de las sesiones de monitoreo

## 3. PS-003: Compatibilidad y Adaptación a Entornos Variados

- a. **Objetivo:** Asegurar que "ProfitQuest" se ejecuta y se visualiza correctamente en los diferentes sistemas operativos compatibles (Windows, macOS, Linux) y se adapta a diversas resoluciones de pantalla.
- b. **Método:**
  - i. Instalar y ejecutar el juego en al menos una versión reciente de Windows, macOS y una distribución común de Linux.
  - ii. En cada sistema operativo, probar el juego en la resolución mínima 1280x720 píxeles y en varias resoluciones mayores.
  - iii. Verificar que la interfaz de usuario, los sprites y los elementos gráficos escalan correctamente sin distorsiones o recortes.
  - iv. Probar el control del juego con teclado y ratón, y con un gamepad si es compatible con el sistema operativo.
- c. **Criterios de Aceptación:**
  - i. El juego debe iniciarse y ejecutarse sin errores en todos los sistemas operativos especificados.
  - ii. La interfaz y los gráficos deben adaptarse correctamente a todas las resoluciones de pantalla probadas, sin distorsiones ni problemas visuales.
  - iii. Los controles deben ser funcionales y responder de manera consistente en todas las plataformas y con los periféricos compatibles.
- d. **Evidencia:** Retroalimentación de las diferentes corridas en diversos entornos.

## Pruebas de aceptación de usuario

Obtener la confirmación de los usuarios finales (estudiantes sin formación previa en economía, docentes) de que el juego "ProfitQuest" cumple con sus expectativas, es fácil de usar, educativo y adecuado para su propósito. Estas pruebas son las más cercanas a un escenario de uso real y se centran en la percepción y experiencia del usuario.

### 1. PA-001: Usabilidad y Claridad General del Juego

- a. **Objetivo:** Evaluar la facilidad con la que los usuarios pueden navegar por la interfaz, comprender las mecánicas y completar tareas clave sin asistencia, y la claridad de los mensajes de error y del tutorial.

**b. Método:**

- i. Seleccionar un grupo representativo de usuarios finales (estudiantes sin conocimientos de economía, docentes).
- ii. Asignarles una serie de tareas clave en el juego, completar el tutorial (PQ-15), comprar todos los materiales para un producto, fabricar 3 productos, entregar un producto, usar la estación de reciclaje, personalizar su avatar (PQ-18), y consultar la tabla de clasificación (PQ-09)).
- iii. Observar a los usuarios mientras realizan estas tareas, registrando el tiempo que tardan y si necesitan ayuda.
- iv. Recopilar feedback directo sobre la intuitividad de la interfaz (menús, HUD), la comprensión de las mecánicas de juego, la claridad del tutorial y la comprensibilidad de cualquier mensaje de error o notificación.

**c. Criterios de Aceptación:**

- i. Tasa de éxito del 90% o superior en la realización de las tareas clave sin asistencia.
- ii. La interfaz de usuario debe ser percibida como intuitiva y fácil de navegar por la mayoría de los usuarios.
- iii. El tutorial inicial debe ser claro y efectivo para introducir las mecánicas básicas.
- iv. Los mensajes de error y las notificaciones deben ser comprendidos fácilmente por los usuarios.

- d. Evidencia:** Notas de observación de los testers y el equipo de diseño, registros de tiempo para completar tareas.

**2. PA-002: Relevancia Educativa y Efectividad del Aprendizaje**

- a. Objetivo:** Determinar si "ProfitQuest" enseña efectivamente los conceptos de gestión de costos y recursos, y si las mecánicas de dificultad y objetivos (PQ-05, PQ-10) contribuyen al aprendizaje.

**b. Método:**



- i. Después de completar un segmento del juego, aplicar un cuestionario o realizar entrevistas con los usuarios participantes para evaluar su comprensión de los conceptos económicos introducidos.
- ii. Preguntar a los usuarios sobre la relevancia y claridad de los objetivos de cada nivel (PQ-10).
- iii. Recopilar feedback sobre si la progresión de dificultad (PQ-05) se siente adecuada y desafiante, pero no frustrante, para el aprendizaje.
- iv. Analizar la calidad de la retroalimentación del sistema de penalización (PQ-03) y evaluación de desempeño (PQ-08) en términos de si ayuda a los usuarios a corregir sus errores y mejorar sus decisiones.

**c. Criterios de Aceptación:**

- i. Los usuarios deben demostrar una comprensión básica o mejorada de los conceptos económicos relevantes tras jugar.
- ii. Los objetivos y la dificultad de los niveles deben ser percibidos como claros y apropiados para el aprendizaje.
- iii. El feedback del juego (penalizaciones, evaluaciones) debe ser útil para el proceso educativo.

**d. Evidencia:** Resultados de cuestionarios de comprensión

**3. PA-003: Experiencia General del Usuario y Satisfacción**

- a. Objetivo:** Recopilar feedback cualitativo sobre la experiencia general del juego, incluyendo la fluidez del movimiento del personaje, el atractivo visual, la satisfacción con la personalización del avatar, y la percepción de las recompensas y penalizaciones.

**b. Método:**

- i. Al finalizar las sesiones de juego, realizar una entrevista abierta o una encuesta de satisfacción general a los usuarios.
- ii. Solicitar comentarios sobre aspectos específicos como la fluidez del movimiento del personaje (PQ-02), la claridad y estética del pixel art, la satisfacción con las opciones de personalización del avatar (PQ-18).
- iii. Preguntar sobre la sensación de "justicia" del sistema de penalización (PQ-03) y la motivación generada por la tabla de clasificación (PQ-09) y las recompensas.

- iv. Identificar cualquier punto de fricción o elemento que disminuya la satisfacción general.

**c. Criterios de Aceptación:**

- i. Alta satisfacción general del usuario con la experiencia de juego.
- ii. El estilo visual pixel art debe ser percibido como atractivo y consistente.
- iii. Los usuarios deben sentirse motivados por el sistema de progresión y recompensas.
- iv. La fluidez del movimiento del personaje debe ser satisfactoria.

**d. Evidencia:** Análisis cualitativo de las respuestas de las encuestas

## Criterios de Aprobación/Rechazo

Para que las pruebas de "ProfitQuest" se consideren aprobadas y el juego esté listo para su lanzamiento o la siguiente fase de desarrollo, se deben cumplir los siguientes criterios:

1. **Funcionalidad Completa:** Todos los requisitos funcionales (PQ-XX) definidos en el ERS deben estar implementados y operar correctamente según sus especificaciones, con un porcentaje de aprobación del 95% o superior en los casos de prueba funcionales.
2. **Calidad No Funcional:**
3. **Rendimiento:** El juego debe cumplir con los criterios de "Cumple totalmente" para el Rendimiento, es decir, FPS estables mayor a 60, tiempos de carga rápidos menor a 5 segundos y baja latencia de entrada menor a 100 milisegundos.
4. **Seguridad:** El sistema debe cumplir con los criterios de "Cumple totalmente" para la Seguridad, sin vulnerabilidades críticas o mayores detectadas en la protección de datos de guardado y perfiles.
5. **Usabilidad:** El juego debe alcanzar una tasa de éxito del 90% o superior en las tareas clave durante las pruebas de Aceptación de Usuario y la interfaz debe ser percibida como intuitiva por la mayoría de los usuarios.
6. **Estabilidad:** No se deben detectar fallos críticos (crashes o bloqueos) durante las sesiones de pruebas de sistema prolongadas o bajo estrés. El sistema de guardado automático debe funcionar correctamente como respaldo.

7. **Integración:** Todas las interacciones entre los módulos del juego deben ser fluidas y sin errores, asegurando que los subsistemas trabajan armoniosamente.
8. **Defectos Resueltos:** Todos los defectos de prioridad "Bloqueante" y "Crítico" identificados durante las pruebas deben haber sido corregidos y verificados por QA. Menos del 5% de los defectos "Mayores" pueden quedar pendientes, siempre que no impacten el núcleo de la jugabilidad o el propósito educativo.

Las pruebas se considerarán rechazadas si no se cumplen uno o más de los criterios anteriores, lo que implica la necesidad de más desarrollo y un nuevo ciclo de pruebas.

## Entregables

Los siguientes documentos y artefactos serán generados como parte del proceso de pruebas:

1. Plan de Pruebas
2. Registro de Resultados de Pruebas
3. Informe Final de Pruebas:

## Tareas de Prueba Restantes

Una vez completada esta fase del plan de pruebas, las siguientes tareas serán cruciales para la mejora continua y la preparación final del juego:

1. Coordinar pruebas de usabilidad con usuarios finales adicionales para obtener una muestra más amplia y validar la experiencia educativa.
2. Realizar pruebas de carga y estrés intensivas en entornos simulados para simular escenarios de uso extremo (si el juego contempla un alto volumen de operaciones internas).
3. Implementar y verificar las correcciones de todos los bugs reportados como "Bloqueante" y "Crítico".
4. Ejecutar un ciclo de pruebas de regresión completo después de la integración de las correcciones de bugs y nuevas características.
5. Realizar una última revisión de compatibilidad en todas las plataformas objetivo antes del despliegue final.

## Necesidades Ambientales

Para ejecutar el plan de pruebas de "ProfitQuest" de manera efectiva, se requieren las siguientes condiciones y recursos:

1. Servidor de Desarrollo/Compilación: Un servidor estable con la última versión del software del juego implementado y compilado, asegurando que los testers siempre trabajen con la build más reciente y funcional.
2. Entornos de Prueba de Hardware:
  - a. Múltiples equipos (PC) con diferentes configuraciones de hardware que representen los requisitos mínimos y recomendados del juego (CPU, GPU, RAM, espacio en disco).
  - b. Equipos con los sistemas operativos objetivo: Windows (versiones recientes), macOS (versiones recientes), y al menos una distribución popular de Linux.
  - c. Periféricos de Entrada: Conjunto de teclados, ratones y al menos un gamepad compatible para probar la respuesta de los controles.
3. Herramientas de Pruebas:
  - a. Captura de Video: Software de grabación de pantalla en cada máquina de prueba para la documentación de evidencia.
  - b. Comunicación: Una plataforma de comunicación en equipo para colaboración rápida y resolución de dudas entre testers y desarrolladores.

## Calendario

El siguiente calendario establece las fases y duraciones estimadas para las actividades de pruebas de "ProfitQuest".

Actividad	Fecha Inicio	Fecha Fin
Desarrollo del plan de pruebas	15 de mayo de 2025	16 de mayo de 2025
Configuración de entornos y herramientas	16 de mayo de 2025	17 de mayo de 2025
Pruebas estáticas	18 de mayo de 2025	20 de mayo de 2025

Pruebas unitarias	21 de mayo de 2025	24 de mayo de 2025
Pruebas de integración	25 de mayo de 2025	28 de mayo de 2025
Pruebas de sistema	29 de mayo de 2025	3 de junio de 2025
Pruebas de aceptación de usuario	4 de junio de 2025	6 de junio de 2025
Informe Final	7 de junio de 2025	8 de junio de 2025

## Criterios de Suspensión y Requisitos de Reanudación

**Criterios de Suspensión:** Las pruebas se suspenderán si se detectan fallos críticos que impidan la ejecución de un segmento significativo de los casos de prueba o que corrompan el entorno de pruebas de forma irreversible.

**Requisitos de Reanudación:** Las pruebas se reanudarán una vez que los fallos críticos que causaron la suspensión hayan sido corregidos y verificados por el equipo de desarrollo. Se deberá proporcionar una nueva build estable, y se realizará un pequeño ciclo de regresión inicial para confirmar que el problema original no ha reaparecido y que no se han introducido nuevos bloqueadores.

## Planificación de Riesgos y Contingencias

La gestión de riesgos es fundamental para el éxito del proceso de pruebas:

1. **Riesgo:** Retrasos por problemas en la infraestructura o entornos de prueba.
  - a. **Mitigación:** Garantizar la disponibilidad y estabilidad de todos los entornos de prueba (hardware, SO) y herramientas antes de iniciar el ciclo de pruebas. Mantener imágenes de respaldo de los entornos. Establecer comunicación constante con el equipo de infraestructura.
2. **Riesgo:** Identificación tardía de bugs críticos.

- a. **Mitigación:** Priorizar las pruebas de los flujos de juego principales y los requisitos más críticos (PQ-01, PQ-05, PQ-06, PQ-07, PQ-11, PQ-14, PQ-17) al inicio del ciclo de pruebas. Implementar sesiones diarias de "bug triage" con desarrollo.
3. **Riesgo:** Falta de recursos humanos (testers) o técnicos.
  - a. **Mitigación:** Asegurar que el equipo de QA está suficientemente capacitado y tiene tiempo asignado. Escalar las necesidades de hardware con anticipación.
4. **Riesgo:** Discrepancia entre expectativas del usuario y el producto.
  - a. **Mitigación:** Involucrar a los usuarios finales (docentes, estudiantes) en las fases tempranas de diseño y en las pruebas de Aceptación de Usuario, recopilando feedback constante. Asegurar que el ERS es claro y consensuado.

## Aprobaciones

Este Plan de Pruebas ha sido revisado y aprobado por los siguientes responsables:

Nombre	Rol	Firma	Fecha
Daniel Garbanzo	Líder de pruebas	Daniel	7 de junio de 2025
Mauricio Fernández	Responsable de calidad de software	Mauricio	7 de junio de 2025
Leonardo Céspedes	Líder de desarrollo	Leonardo	7 de junio de 2025
Kevin Chang	Desarrollador principal	Kevin	7 de junio de 2025

## Glosario

**Atributo de Calidad:** Característica no funcional de un sistema que describe cómo se comporta o qué tan bien funciona.

**ERS (Especificación de Requisitos de Software):** Documento que describe los requisitos funcionales y no funcionales de un sistema.

**FPS (Frames Per Second):** Medida de la fluidez del rendimiento visual del juego.

**GDScript:** Lenguaje de programación nativo y ligero de Godot Engine.

**Godot Engine:** Motor de desarrollo de videojuegos 2D y 3D, de código abierto.

**Lint:** Herramienta de análisis estático de código que marca errores de programación, bugs, errores estilísticos y construcciones sospechosas.

**Pixel Art:** Estilo visual donde las imágenes se construyen a nivel de píxel individual.

**Pruebas de Aceptación de Usuario (UAT):** Pruebas realizadas por usuarios finales o representantes del cliente para verificar que el sistema cumple con las expectativas y es adecuado para su uso en producción.

**Pruebas de Integración:** Pruebas que verifican la interacción y correcta comunicación entre dos o más módulos o componentes del sistema.

**Pruebas de Sistema:** Pruebas que validan que el sistema completo cumple con los requisitos funcionales y no funcionales en un entorno que simula la producción.

**Pruebas Dinámicas:** Tipo de pruebas de software que implican la ejecución del código del programa para verificar su comportamiento.

**Pruebas Estáticas:** Tipo de pruebas de software que implican el análisis del código sin ejecutarlo, a menudo a través de revisiones o herramientas de análisis estático.

**Pruebas Unitarias:** Tipo de pruebas que validan el correcto funcionamiento de la unidad o componente de código más pequeño y aislado del sistema.

## Herramientas Utilizadas para las Pruebas

Para la ejecución de este plan de pruebas en "ProfitQuest", se emplearán las siguientes herramientas:

### 1. Herramientas de Captura de Video y Pantalla:

- a. Propósito: Grabar la ejecución del juego para documentar visualmente el comportamiento esperado, los bugs, los problemas de rendimiento y la experiencia del usuario.
- b. Funcionalidad Clave:
- c. OBS Studio: Para grabaciones de alta calidad y personalizables de la pantalla de juego en PC (Windows, macOS, Linux).

- d. Beneficios: Proporciona evidencia irrefutable de los bugs y el comportamiento del juego, facilitando la reproducción y el diagnóstico por parte de los desarrolladores. Crucial para comunicar la experiencia de usuario.

## **2. Herramientas de Monitoreo de Rendimiento:**

- a. Propósito: Medir métricas de rendimiento clave del juego como FPS, tiempos de carga, latencia de entrada y consumo de recursos.
- b. Funcionalidad Clave:
  - i. Godot Profiler (Integrado en Godot Engine): Permite analizar el rendimiento a nivel de motor (tiempo de procesamiento de scripts, renderizado, físicas, etc.).
  - ii. MSI Afterburner (para PC): Herramienta para monitorear el uso de CPU, GPU, RAM, temperaturas y FPS en tiempo real, con opciones de registro.
  - iii. HWMonitor: Para monitoreo general de hardware.
- c. Beneficios: Identifica cuellos de botella en el rendimiento, asegura el cumplimiento de los requisitos no funcionales y optimiza la experiencia del jugador.

## **3. Herramientas de Análisis Estático de Código:**

- a. Propósito: Realizar pruebas estáticas en el código GDScript sin ejecutar el juego, identificando problemas de calidad, estilo y posibles errores lógicos.
- b. Funcionalidad Clave:
  - i. Revisiones Manuales de Código (Code Review): Proceso de revisión por pares que, aunque no es una "herramienta" en sí, se apoya en el editor de Godot y sistemas de control de versiones.
- c. Beneficios: Mejora la mantenibilidad del código, reduce la deuda técnica y permite la detección temprana de bugs.

## **4. Herramientas de Comunicación y Colaboración:**

- a. Propósito: Facilitar la comunicación fluida y la colaboración entre el equipo de pruebas y el equipo de desarrollo.
- b. Funcionalidad Clave: Slack o Microsoft Teams para discusiones en tiempo real, compartir información y resolver dudas rápidamente.



- c. Beneficios: Agiliza el proceso de reporte y resolución de bugs, y fomenta un ambiente de trabajo colaborativo.

# Resultados de Pruebas para “ProfitQuest”

**Sistema:** ProfitQuest

**Fecha de ejecución:** 7 de junio de 2025

**Responsables:** Leonardo Céspedes, Kevin Chang, Mauricio Fernández, Daniel Garbanzo

## Introducción

Este documento presenta los resultados obtenidos de las pruebas realizadas al juego educativo "ProfitQuest". Las pruebas incluyeron verificaciones estáticas del código y la documentación, así como validaciones dinámicas del juego en ejecución. Los objetivos principales fueron garantizar la completitud funcional de las mecánicas de producción y progresión, la correcta integración entre los módulos, el rendimiento y la usabilidad del sistema, así como la seguridad de los datos del jugador y la satisfacción general de los usuarios finales.

## Resultados de Pruebas Estáticas

Las pruebas estáticas consistieron en la revisión del ERS, el código GDScript, la documentación técnica de Godot, los diseños de niveles y activos, y el análisis de la deuda técnica utilizando herramientas de análisis de código estático (linters).

### Prueba de Completitud Funcional (Revisión de Requisitos y Diseño)

**Objetivo:** Verificar que todos los requisitos funcionales y sus respectivas especificaciones (ej. límites, objetivos, restricciones) están completamente documentados en el ERS y reflejados en los documentos de diseño de niveles, interfaz y lógica de negocio.

**Resultado:** Todos los requisitos funcionales y no funcionales definidos en el ERS están bien representados en la documentación de diseño del juego. Los módulos de Simulación Económica y Producción, Jugabilidad y Progresión, y Gestión de Usuarios y Personalización tienen sus funcionalidades bien delineadas. Sin embargo, se identificó que, si bien el ERS especifica un tutorial interactivo completo para introducir las mecánicas básicas, la documentación de diseño actual y los prototipos indican un tutorial breve, que puede no ser

tan exhaustivo como se esperaba inicialmente. No se encontraron ambigüedades o inconsistencias significativas entre el ERS y los documentos de diseño en otros aspectos.

**Recomendaciones:** Se recomienda una revisión del alcance y detalle del tutorial inicial (PQ-15) en la documentación de diseño para asegurar que cumple con el nivel de completitud requerido por el ERS, o ajustar el ERS si la implementación de un tutorial más extenso se considera fuera de alcance para la versión actual.

**Estado:** Aprobado con observaciones.

## **Prueba de Correctitud Funcional (Revisión de Código y Lógica)**

**Objetivo:** Validar que la lógica implementada en los scripts GDScript para las mecánicas principales es coherente con las especificaciones del sistema.

**Resultado:** La lógica de negocio en los scripts GDScript para la gestión de presupuesto (PQ-06), inventario (PQ-07), penalización (PQ-03) y cálculo de desempeño (PQ-08) es consistente con los requisitos del ERS. Los errores lógicos detectados en revisiones previas están siendo manejados adecuadamente. El código GDScript es legible y sigue la mayoría de las convenciones de programación establecidas. No se identificaron fallas críticas en la lógica de negocio durante esta fase.

**Estado:** Aprobado.

## **Prueba de Mantenibilidad**

**Objetivo:** Evaluar si la estructura del código del juego en Godot es modular, fácil de entender y puede mantenerse y extenderse con facilidad.

**Resultado:** La estructura del código en Godot es modular, con una clara separación entre los nodos, escenas y scripts que gestionan la lógica del juego, la interfaz de usuario y los activos visuales. Se ha implementado un buen nivel de comentarios en los scripts GDScript clave, facilitando la comprensión del código. Aunque se identificaron algunas áreas con un ligero acoplamiento entre sistemas, la mayoría de los módulos críticos son independientes y reusables. La documentación técnica cubre los módulos principales de forma adecuada.

**Estado:** Aprobado con observaciones.

## **Evaluación de Deuda Técnica (Análisis de Código Estático)**

**Objetivo:** Identificar problemas en el código fuente que puedan aumentar la deuda técnica del sistema a largo plazo, como duplicación, lógica acoplada o falta de optimización.

**Resultado:** No se detectaron problemas críticos de diseño o estructura que pongan en riesgo la estabilidad del sistema a largo plazo. Se observó una minimización de la duplicación de código en la mayoría de las funciones. La gestión de recursos (texturas, sprites, memoria) sigue pautas de optimización básicas. Las funciones críticas están bien documentadas. Sin embargo, se identificaron algunas funciones con una complejidad ligeramente superior a la recomendada en ciertas validaciones, y oportunidades menores para refactorizar y consolidar la lógica de interacción de ciertos nodos.

**Recomendaciones:** Se sugiere una revisión de las funciones de mayor complejidad para buscar simplificaciones y la posibilidad de consolidar fragmentos de código repetitivos.

**Estado:** Aprobado con observaciones.

## **Resultados de Pruebas Dinámicas**

Las pruebas dinámicas involucran la ejecución de "ProfitQuest" en un entorno de desarrollo para validar el comportamiento en tiempo real de sus componentes, módulos y sistemas completos. Se utilizaron grabaciones de video, capturas de pantalla y logs de la consola como evidencia para documentar los hallazgos.

### **Pruebas Unitarias**

Las pruebas unitarias se realizaron para verificar el correcto funcionamiento de los componentes más pequeños y aislados del juego, asegurando que las lógicas internas y los cálculos fundamentales operan según lo especificado.

PU-001: Prueba de Cálculo de Reducción de Presupuesto

**Objetivo:** Verificar que el módulo de gestión de presupuesto (PQ-06) deduce la cantidad correcta de dinero al realizar una compra o acción que implique gasto.

**Método:** Se inicializan varios escenarios con presupuestos específicos y se simularon acciones de compra y fabricación con costos predefinidos. Se verificó el valor final del presupuesto en el depurador y mediante logs.

**Resultado:** El presupuesto se redujo en la cantidad exacta esperada en todos los escenarios probados, incluyendo casos con números enteros y decimales. No se observaron valores negativos inesperados. La lógica de cálculo funciona perfectamente.

**Estado:** Aprobado.

**Atributos Evaluados:** Rendimiento, Funcionalidad.

**Métricas:**

- Tiempo de respuesta: 50 milisegundos.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Grabación de pantalla.

**Vídeo de YouTube:** [https://youtu.be/NcuaHxyUI9M?si=Q\\_XJ9NPKoYWVrITc](https://youtu.be/NcuaHxyUI9M?si=Q_XJ9NPKoYWVrITc)

PU-002: Prueba de Denegación por Presupuesto Insuficiente

**Objetivo:** Asegurar que el sistema de gestión de presupuesto (PQ-06) deniega acciones cuando el jugador no tiene recursos suficientes.

**Método:** Se configuró un presupuesto mínimo y se intentaron realizar compras y fabricaciones que excedían este monto. Se observó el comportamiento del sistema y los mensajes mostrados.

**Resultado:** La acción fue correctamente denegada y el presupuesto no se modificó en todos los casos. Sin embargo, no hubo mensaje de error.

**Estado:** No aprobado con comentarios.

**Atributos Evaluados:** Usabilidad, Funcionalidad.

**Métricas:**

- Tiempo de respuesta: 70 milisegundos.
  - Estado (Métrica): Cumple totalmente.

**Observaciones:** Debería de haber un mensaje de error indicando que no hay presupuesto suficiente.

**Recomendaciones:** Implementar mensaje de denegación para indicar "Presupuesto insuficiente" o similar.

**Evidencia:** Grabación de video mostrando la denegación y el mensaje.

**Vídeo de YouTube:** <https://youtu.be/pvxIKGVhOk?si=crZi-TJ0mtbYUiT7>

#### PU-003: Prueba de Actualización de Inventario por Consumo

**Objetivo:** Verificar que la cantidad de materiales en el inventario (PQ-07) se reduce correctamente cuando se utilizan para fabricar un producto.

**Método:** Se preparó un inventario con cantidades conocidas de materiales. Se simuló la fabricación de diversos productos que consumen distintas cantidades de estos materiales, y se verificaron las existencias resultantes.

**Resultado:** Las cantidades de los materiales en el inventario se redujeron de forma precisa y consistente según el consumo de cada producto fabricado. No se detectaron inconsistencias ni valores negativos.

**Estado:** Aprobado.

**Atributos Evaluados:** Funcionalidad.

**Métricas:**

- Tiempo de respuesta: 60 milisegundos.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Grabación de pantalla mostrando el consumo en el inventario.

**Vídeo de YouTube:** <https://youtu.be/QSRvs2qlw2w?si=aeGoqaW3REXwPJn3>

#### PU-004: Prueba de Cálculo de Puntaje de Desempeño

**Objetivo:** Validar que la lógica de cálculo del puntaje de desempeño (PQ-08) procesa correctamente las variables (costos, presupuesto, objetivos) para determinar el resultado de la partida.

**Método:** Se simularon varios escenarios de fin de nivel con diferentes combinaciones de objetivos logrados, presupuesto restante y costos incurridos. Se verificó el puntaje resultante y la decisión de pasar/fallar el nivel.

**Resultado:** La función de cálculo de desempeño generó puntajes precisos y consistentes para todos los escenarios de entrada, y la decisión de avance de nivel fue correcta según los umbrales definidos. La lógica se alinea con el ERS.

**Estado:** Aprobado.

**Atributos Evaluados:** Funcionalidad.

**Métricas:**

- Tiempo de respuesta: 100 milisegundos.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Grabación de pantalla mostrando el puntaje.

**Vídeo de YouTube:** <https://youtu.be/FEsqBEBhKrI?si=y1AfgQT-NCu5Ewaq>

PU-005: Prueba de Decremento del Temporizador

**Objetivo:** Verificar que el temporizador por nivel (PQ-16) se decrementa de forma precisa y continua.

**Método:** Se inició el temporizador de un nivel con un valor de 60 segundos y se monitorea su decremento a través de una interfaz de depuración visible.

**Resultado:** El temporizador se redujo correctamente, mostrando el tiempo restante de forma precisa. No se observaron saltos abruptos o inconsistencias en la cuenta regresiva.

**Estado:** Aprobado.

**Atributos Evaluados:** Funcionalidad, Rendimiento.

**Métricas:**

- Tiempo de respuesta: 30 milisegundos.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Grabación de video mostrando el temporizador en tiempo real.

**Vídeo de YouTube:** <https://youtu.be/rHocAmIC-sY?si=OofYHLs4Ai9b6eLf>

## Pruebas de Integración

Las pruebas de integración se centran en verificar la correcta interacción y comunicación entre diferentes módulos o componentes del juego. Se busca asegurar que, al trabajar en conjunto, los sistemas se comportan como se espera y no introducen errores.

PI-001: Integración Compra de Materiales y Gestión de Presupuesto/Inventario

**Objetivo:** Validar que la compra de materiales reduce el presupuesto y actualiza el inventario de manera correcta.

**Método:**

- Inicializar con presupuesto e inventario conocidos.
- Comprar materiales desde la tienda.
- Verificar reducción del presupuesto y aumento de inventario.
- Intentar comprar sin presupuesto suficiente.

**Resultado:** El sistema descuenta correctamente el monto y actualiza el inventario. Las compras no válidas son rechazadas con mensaje de error.

**Estado:** Aprobado

**Atributos evaluados:** Rendimiento, Seguridad, Integridad de módulos.

**Métricas:**

- Tiempo de respuesta: 280 ms .
  - Estado (Métrica): Cumple totalmente.
- Vulnerabilidades detectadas: 0.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Video con flujo completo de compra y actualización del inventario.

**Video de YouTube:** [https://www.youtube.com/watch?v=5VOJn\\_SKBDk&](https://www.youtube.com/watch?v=5VOJn_SKBDk&)

PI-002: Integración Fabricación de Producto y Gestión de Inventario/Presupuesto

**Objetivo:** Confirmar que la fabricación de un producto consume los materiales correctos del inventario, ajusta el presupuesto y añade el producto al inventario de salida.

**Método:**

- Iniciar con materiales suficientes en inventario y presupuesto definido.
- Fabricar un producto desde la estación correspondiente.
- Verificar disminución de materiales, reducción de presupuesto y aparición del producto fabricado.
- Intentar fabricar sin materiales suficientes.

**Resultado:** El sistema consumió correctamente los materiales, ajustó el presupuesto y añadió el producto fabricado al inventario. Se bloqueó la fabricación al no contar con los insumos necesarios.

**Estado:** Aprobado

**Atributos evaluados:** Funcionalidad, Integridad lógica.

**Métricas:**

- Tiempo de respuesta: 310 ms.



- Estado (Métrica): Cumple parcialmente.
- Vulnerabilidades detectadas: 0
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Video mostrando las interfaces antes y después del proceso de fabricación.

**Video de YouTube:** [https://www.youtube.com/watch?v=J7tE-d\\_jZaU](https://www.youtube.com/watch?v=J7tE-d_jZaU)

PI-003: Integración Sistema de Penalización y Manejo de Errores

**Objetivo:** Validar que los errores en la producción activan penalizaciones y que la estación de manejo de errores permite recuperar parcialmente los recursos utilizados.

**Método:**

- Cometer un error en la fabricación para activar la penalización.
- Usar la estación de manejo de errores.
- Verificar recuperación del 50% de materiales y decremento del contador de uso.
- Intentar usar la estación más de 3 veces.

**Resultado:** El sistema penalizó al jugador correctamente, y la estación de errores devolvió el 50% de materiales usados hasta el límite de tres usos.

**Estado:** Aprobado

**Atributos evaluados:** Funcionalidad, Lógica de control.

**Métricas:**

- Tiempo de respuesta: 290 ms.
  - Estado (Métrica): Cumple totalmente.
- Vulnerabilidades detectadas: 0
- Estado (Métrica): Cumple totalmente.

**Evidencia:** Video del flujo completo desde el error, la penalización y la recuperación de recursos.

**Video de YouTube:** [https://www.youtube.com/watch?v=J7tE-d\\_jZaU](https://www.youtube.com/watch?v=J7tE-d_jZaU)

#### PI-004: Integración Guardado de Partida y Carga de Estado del Jugador

**Objetivo:** Asegurar que el sistema de guardado almacena y recupera el estado del jugador, incluyendo nivel, presupuesto, inventario y tiempo restante.

**Método:**

- Jugar parcialmente un nivel.
- Guardar la partida desde el menú.
- Cerrar el juego y reiniciarlo.
- Verificar carga exacta del estado anterior.
- Probar cierre abrupto y recuperación automática.

**Resultado:** El sistema guardó y restauró correctamente el estado del juego.

**Estado:** Aprobado con observaciones

**Atributos evaluados:** Persistencia, Seguridad.

**Métricas:**

- Tiempo de carga: 290 segundos.
  - Estado (Métrica): Cumple totalmente.
- Tasa de recuperación automática: 100%.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Video con guardado manual y carga exitosa. Registro de logs.

**Video de YouTube:** <https://www.youtube.com/watch?v=tfFTaHPgdDs>

#### PI-005: Integración Personalización de Avatar y Perfil de Usuario

**Objetivo:** Validar que los cambios en la apariencia del avatar se guardan correctamente y se mantienen entre sesiones.

**Método:**

- Modificar color, vestimenta y accesorios del avatar.
- Salir y volver a entrar al perfil del usuario.
- Verificar persistencia de los cambios.

**Resultado:** La personalización fue almacenada correctamente y persistió en futuras sesiones, sin errores ni pérdidas de datos.

**Estado:** Aprobado

**Atributos evaluados:** Usabilidad, Persistencia de datos.

**Métricas:**

- Tiempo de respuesta: 260 ms.
  - Estado (Métrica): Cumple totalmente.
- Vulnerabilidades detectadas: 0.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Capturas de pantalla antes y después de la personalización. Video del flujo.

**Video de YouTube:** <https://www.youtube.com/watch?v=SKginEeFlx4>

PI-006: Integración Diseño de Niveles y Movimiento del Personaje

**Objetivo:** Verificar que el personaje se mueve correctamente en los niveles preconfigurados y respeta las colisiones y áreas delimitadas.

**Método:**

- Cargar un nivel con obstáculos y estaciones.
- Mover el personaje por todo el mapa.
- Verificar que no atraviesa objetos y que responde a las teclas correctamente.

**Resultado:** El movimiento del personaje fue fluido. Las colisiones funcionaron correctamente sin glitches ni desplazamientos inesperados.

**Estado:** Aprobado

**Atributos evaluados:** Usabilidad, Rendimiento.

**Métricas:**

- FPS promedio: 62 FPS.
  - Estado (Métrica): Cumple totalmente.

- Latencia de entrada: 95 ms.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:**

Video de jugabilidad con desplazamiento del personaje y colisiones.

**Video de YouTube:** [https://www.youtube.com/watch?v=q\\_iwkbMG1Ac](https://www.youtube.com/watch?v=q_iwkbMG1Ac)

## Pruebas de Sistema

### PS-001: Flujo Completo del Juego (De Principio a Fin)

**Objetivo:** Verificar que el juego puede ejecutarse completamente, desde el inicio hasta la finalización de niveles, integrando todas las mecánicas sin errores.

**Método:**

- Iniciar sesión con un nuevo perfil.
- Completar el tutorial inicial.
- Jugar al menos un nivel preconfigurado realizando: compra de materiales, fabricación de productos, gestión de presupuesto e inventario, entrega de productos, uso del sistema de penalización, guardado/carga de partida y validación de desempeño.
- Verificar que todos los módulos funcionen durante el flujo completo.

**Resultado:** El juego se ejecutó de principio a fin sin errores críticos. Todas las funcionalidades respondieron según lo esperado. Se validó el guardado y la carga en medio del nivel, así como la correcta progresión de dificultad.

**Estado:** Aprobado

**Atributos evaluados:** Funcionalidad, Integración, Usabilidad.

**Métricas:**

- Tiempo promedio por nivel: 3 min 42 s.
  - Estado (Métrica): Cumple totalmente.
- Errores detectados: 0.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:**

Video completo de una sesión de juego desde el inicio hasta la validación del nivel.

**Video de YouTube:** <https://www.youtube.com/watch?v=WJPTSTfHJ-8>

### PS-002: Rendimiento General y Fluidez de Juego

**Objetivo:** Evaluar el rendimiento general del sistema, incluyendo FPS, tiempos de carga y latencia de entrada, bajo condiciones normales y exigentes.

**Método:**

- Ejecutar el juego en dos equipos con especificaciones distintas (mínimas y recomendadas).
- Medir el FPS promedio y mínimo durante una sesión completa.
- Evaluar tiempos de carga entre pantallas y niveles.
- Medir la latencia de respuesta a acciones clave (movimiento, interacción).

**Resultado:** El rendimiento fue estable en ambas configuraciones. No se detectaron caídas críticas de FPS ni latencia perceptible. Los tiempos de carga fueron adecuados incluso bajo carga visual intensa.

**Estado:** Aprobado

**Atributos evaluados:** Rendimiento, Usabilidad.

**Métricas:**

- FPS promedio (recomendado): 64 FPS.
- FPS mínimo (mínimas): 34 FPS.
- Latencia de entrada: 87 ms.
- Tiempos de carga entre niveles: 2.6 s.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Video con overlay de métricas activas y sesión de juego en ambas máquinas.

**Video de YouTube:** <https://www.youtube.com/watch?v=WJPTSTfHJ-8>

PS-003: Compatibilidad y Adaptación a Entornos Variados

**Objetivo:** Verificar que *ProfitQuest* se ejecuta correctamente en los sistemas operativos objetivo (Windows, macOS, Linux) y en distintas resoluciones de pantalla.

**Método:**

- Ejecutar el juego en Windows 10, macOS Ventura y Ubuntu 22.04.
- Cambiar la resolución de pantalla entre 1280x720 y 1920x1080.
- Verificar escalado de interfaz y gráficos, y la funcionalidad de controles (teclado, ratón, gamepad).

**Resultado:** El juego corrió sin fallos en los tres sistemas operativos. La interfaz y los gráficos se adaptaron correctamente. Los controles respondieron de forma consistente con todos los dispositivos probados.

**Estado:** Aprobado

**Atributos evaluados:** Compatibilidad, Usabilidad.

**Métricas:**

- Tiempo de inicio por plataforma:
  - Windows: 3.1 s
  - macOS: 3.5 s
  - Linux: 3.2 s
- Errores de escalado detectados: 0.
  - Estado (Métrica): Cumple totalmente.

**Evidencia:** Video de pruebas multiplataforma con capturas de pantalla en distintas resoluciones.

**Video de YouTube:** <https://www.youtube.com/watch?v=WJPTSTfHJ-8>

Pruebas de aceptación de usuario

PA-001: Usabilidad y Claridad General del Juego

**Objetivo:** Evaluar si los usuarios finales pueden navegar por el juego, comprender las mecánicas y completar tareas clave sin asistencia, y si los mensajes del sistema son claros.

**Método:**

- Se reclutó un grupo de estudiantes sin formación económica y docentes.  
Cada participante debió completar el tutorial inicial, fabricar tres productos, comprar materiales, usar la estación de reciclaje, personalizar su avatar y consultar la tabla de clasificación.
- Se midió el tiempo para completar cada tarea y si requirieron ayuda.

**Resultado:** La mayoría de los usuarios (91%) completaron las tareas sin asistencia. Los mensajes fueron claros y la navegación intuitiva. Algunos sugirieron mejorar la visibilidad de ciertos botones en el HUD.

**Estado:** Aprobado

**Atributos evaluados:** Usabilidad, Claridad de la interfaz.

**Métricas:**

- Tasa de éxito sin asistencia: 91%.
  - Estado (Métrica): Cumple totalmente.
- Tiempo promedio por tarea clave: 1 min 12 s.
  - Estado (Métrica): Cumple totalmente.

PA-002: Relevancia Educativa y Efectividad del Aprendizaje

**Objetivo:** Determinar si *ProfitQuest* enseña efectivamente los conceptos de gestión de costos y recursos, y si las mecánicas favorecen el aprendizaje.

**Método:**

- Luego de completar un nivel, se aplicó un breve cuestionario a los participantes.
- Se consultó sobre la claridad de los objetivos, la retroalimentación del sistema y la dificultad progresiva.
- Se pidió identificar al menos un concepto aprendido y cómo fue reforzado por la mecánica del juego.

**Resultado:** El 87% de los participantes identificó correctamente un concepto económico aprendido. La dificultad fue calificada como adecuada, y la retroalimentación del juego fue valorada como útil y clara.

**Estado:** Aprobado

**Atributos evaluados:** Valor educativo, Feedback pedagógico.

**Métricas:**

- Comprensión de conceptos clave: 87%.
  - Estado (Métrica): Cumple parcialmente.
- Percepción positiva del aprendizaje: 93%.
  - Estado (Métrica): Cumple totalmente.

PA-003: Experiencia General del Usuario y Satisfacción

**Objetivo:** Recopilar retroalimentación cualitativa sobre la experiencia de juego: estética, fluidez, motivación, y percepción del sistema de recompensas y penalizaciones.

**Método:**

- Al concluir la sesión, se aplicó una encuesta con escala Likert y preguntas abiertas.

- Se consultó sobre el movimiento del personaje, el atractivo visual del pixel art, la personalización del avatar, y la motivación generada por recompensas y clasificación.

**Resultado:** El 95% indicó estar satisfecho o muy satisfecho con la experiencia. El estilo pixel art fue percibido como atractivo. La tabla de clasificación motivó la mejora del desempeño.

**Estado:** Aprobado

**Atributos evaluados:** Satisfacción del usuario, Estética visual, Motivación.

**Métricas:**

- Satisfacción general: 95%.
  - Estado (Métrica): Cumple totalmente.
- Aceptación del estilo visual: 90%.
  - Estado (Métrica): Cumple totalmente.

## Recomendaciones por atributo de calidad

### Rendimiento

- Fortalezas: El juego mantiene un rendimiento fluido, con FPS estables y tiempos de carga menores a 3 segundos en todas las plataformas probadas.
- Recomendaciones:
  - Optimizar aún más la carga inicial del juego en sistemas con recursos mínimos.
  - Implementar una precarga de assets en segundo plano para niveles complejos.

### Seguridad

- Fortalezas: No se detectaron vulnerabilidades críticas en la gestión de datos guardados ni en las interacciones de usuario.
- Recomendaciones:
  - Validar las entradas de usuario también en el backend, especialmente en escenarios de guardado/carga para prevenir inyecciones o manipulaciones locales.
  - Implementar hashes o firmas para los archivos de guardado como medida antifraude si se planea uso competitivo o escalable.

### Usabilidad

- Fortalezas: Alta tasa de éxito sin asistencia (91%). La interfaz fue percibida como clara, y el estilo visual fue bien recibido.
- Recomendaciones:



- Mejorar la visibilidad de botones secundarios en el HUD (por ejemplo, reciclaje, tabla de clasificación).
- Incorporar accesibilidad básica (teclas configurables, escalado dinámico de texto).

#### Funcionalidad / Integración

- Fortalezas: Todos los flujos funcionales fueron ejecutados con éxito (compra, fabricación, penalización, guardado/carga, movimiento).

##### Recomendaciones:

- Consolidar logs de errores menores (por ejemplo, fallos en guardado automático tras cierre inesperado).
- Considerar agregar una notificación visual al guardar automáticamente.

#### Valor Educativo

- Fortalezas: La mayoría de usuarios logró identificar conceptos de economía y gestión de costos tras jugar.
- Recomendaciones:
  - Incluir glosarios interactivos u objetivos educativos visibles por nivel.
  - Permitir una revisión de desempeño posterior al nivel para fomentar la reflexión del usuario.

## Resumen por tipo de prueba

Tipo de Prueba	Casos Evaluados	Atributos Evaluados	Tasa de Aprobación (%)	Estado General
Pruebas Unitarias	5	Funcionalidad, Rendimiento, Usabilidad	95%	Cumple totalmente
Pruebas de Integración	6	Funcionalidad, Seguridad, Rendimiento	100%	Cumple totalmente
Pruebas de Sistema	3	Rendimiento, Compatibilidad, Integración	100%	Cumple totalmente
Pruebas de Aceptación	3	Usabilidad, Valor Educativo, Satisfacción	100%	Cumple totalmente