



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

Proyecto *Astroneer Parkour*

Curso: *Diseño y Creación de Videojuegos*

Docente: *Mag. Patrick José Cuadros Quiroga*

Integrantes:

Condori Loayza, Helbert Andres

2020067571

Amaya Torres, Josue Israel

2020067149

Mamani Lima, Erick Mauricio

2020066321

**Tacna – Perú
2024**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
1.0	HCL	HCL	HCL, EML, JAT	22/06/2024	Versión Original

Sistema *Astronner Parkour* **Documento de Arquitectura de Software**

Versión {1.0}

1. INTRODUCCIÓN	4
1.1. Propósito (Diagrama 4+1)	4
1.2. Alcance	4
1.3. Definición, siglas y abreviaturas	4
1.4. Organización del documento	4
2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS	4
2.1. Priorización de requerimientos	4
2.2. Restricciones	5
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	5
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	11
4.1. Escenario de Funcionalidad	11
4.2. Escenario de Usabilidad	11
4.3. Escenario de confiabilidad	11
4.4. Escenario de rendimiento	12

1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)

El propósito de este documento es describir la arquitectura del sistema del videojuego Space Escape Games, que combina acción, estrategia y educación en un entorno espacial realista y desafiante. El diagrama 4+1 se utiliza para representar la arquitectura del sistema desde diferentes perspectivas: vista de caso de uso, vista lógica, vista de implementación, vista de procesos y vista de despliegue.

1.2. Alcance

Este documento cubre la arquitectura del sistema del videojuego Space Escape Games, incluyendo sus componentes, relaciones y restricciones. El alcance del proyecto incluye el desarrollo del juego para PC y consolas, con un enfoque en la ciencia y la tecnología.

1.3. Definición, siglas y abreviaturas

SEG: Space Escape Games

PC: Personal Computer

UML: Unified Modeling Language

1.4. Organización del documento

Este documento está organizado en secciones que cubren la introducción, objetivos y restricciones arquitectónicas, representación de la arquitectura del sistema, atributos de calidad del software y conclusiones.

2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS

2.1. Priorización de requerimientos

2.1.1. Requerimientos Funcionales

ID	Requerimiento	Prioridad
RF01	El sistema debe permitir la generación de preguntas.	Alta
RF02	El sistema debe permitir el movimiento por parte del jugador.	Alta
RF03	El sistema debe incluir desafíos ambientales que afecten al jugador.	Alta
RF04	El sistema debe tener un sistema de vidas	Media
RF05	El sistema debe permitir el manejo de niveles	Media

2.1.2. Requerimientos No Funcionales – Atributos de Calidad

ID	Requerimiento	Prioridad
RNF01	El sistema debe ser compatible con plataformas PC y consolas.	Alta
RNF02	El sistema debe tener tiempos de carga menores a 5 segundos.	Alta
RNF03	El sistema debe soportar el cuestionario.	Media
RNF04	El sistema debe tener una interfaz de usuario intuitiva y fácil de usar.	Media

2.2. Restricciones

Plataforma: PC y consolas

Tecnología: UML y herramientas de desarrollo de videojuegos

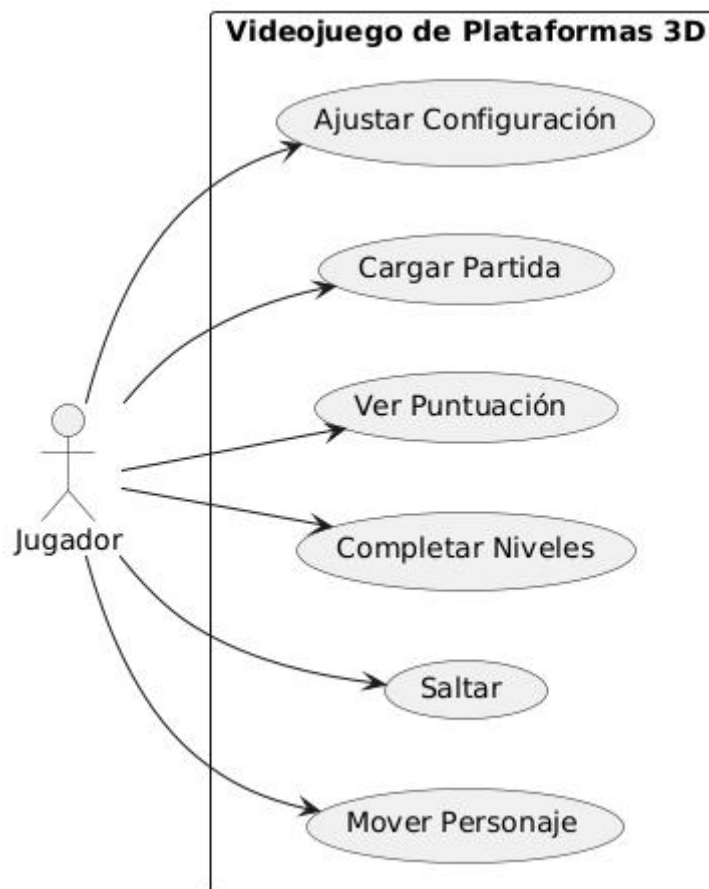
Presupuesto: \$500,000

Tiempo de desarrollo: 6 meses

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

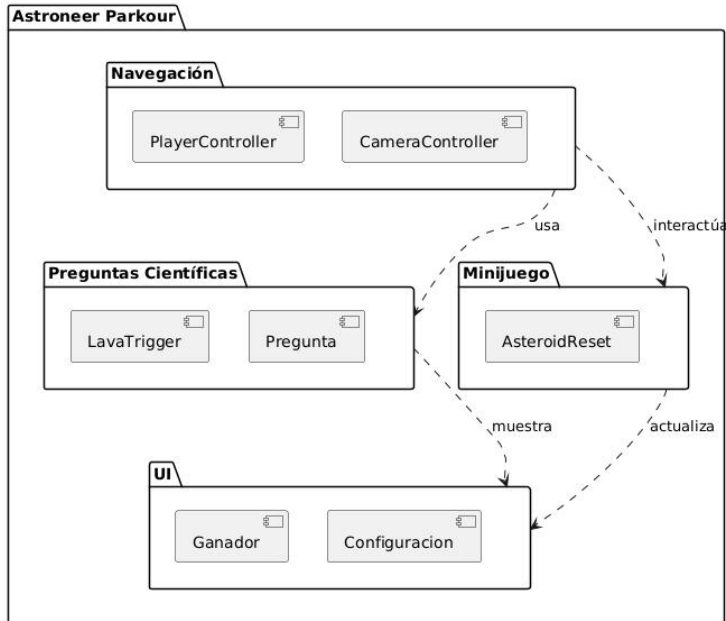
3.1. Vista de Caso de uso

3.1.1. Diagramas de Casos de uso

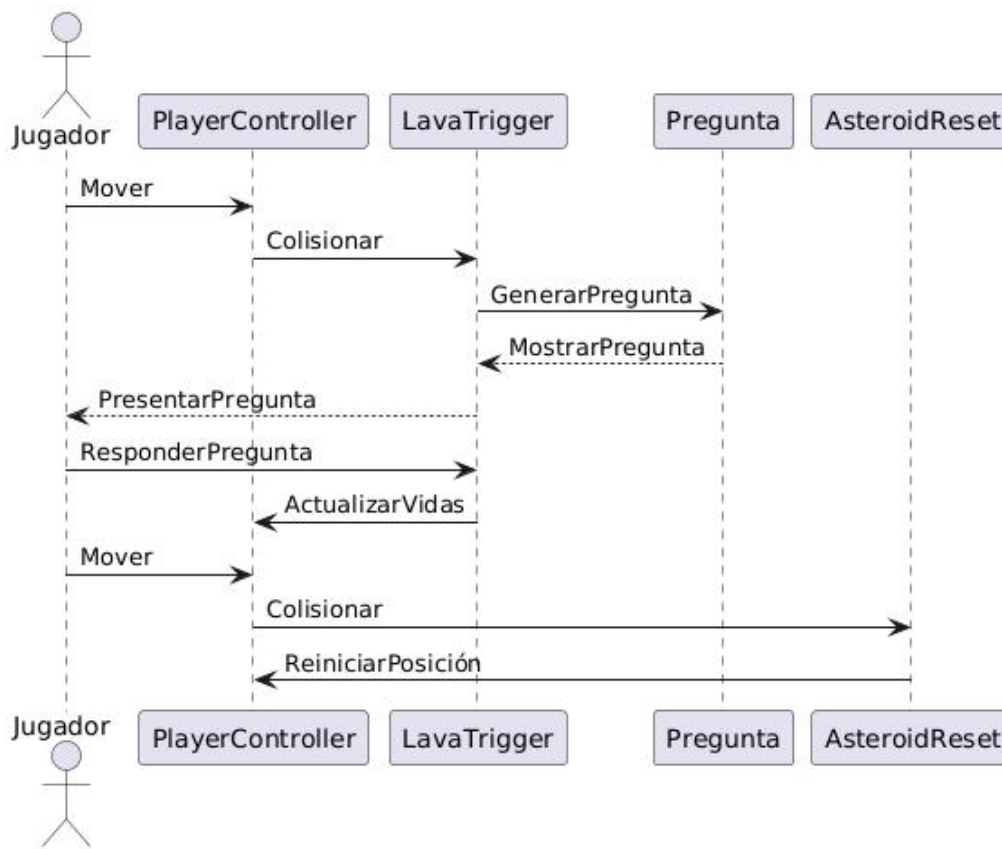


3.2. Vista Lógica

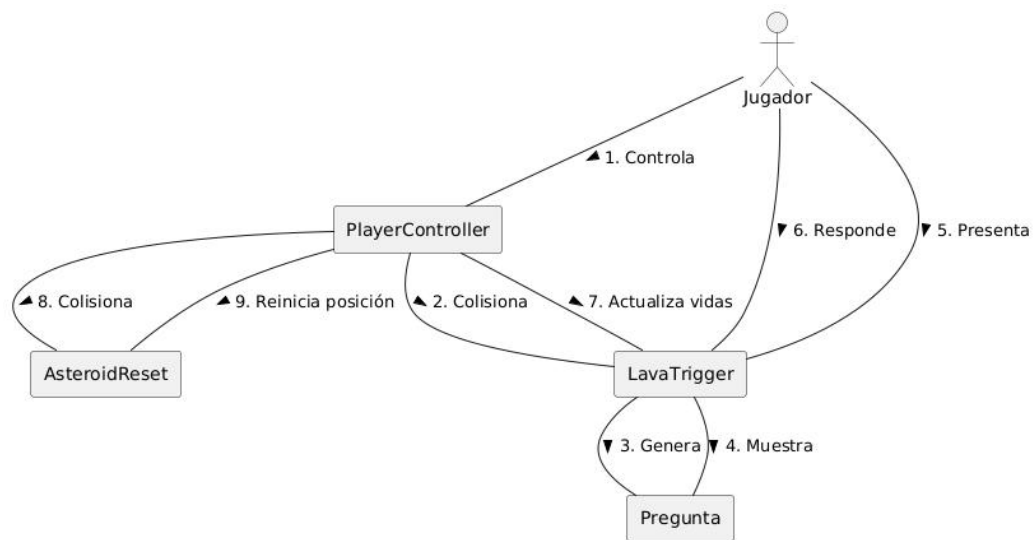
3.2.1. Diagrama de Subsistemas (paquetes)



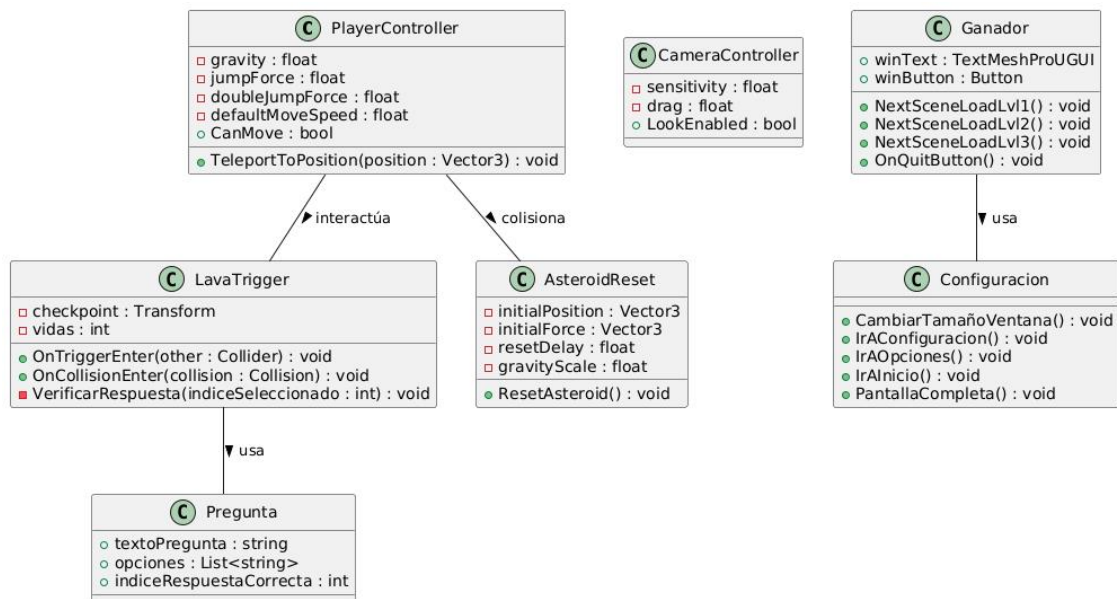
3.2.2. Diagrama de Secuencia (vista de diseño)



3.2.3. Diagrama de Colaboración (vista de diseño)

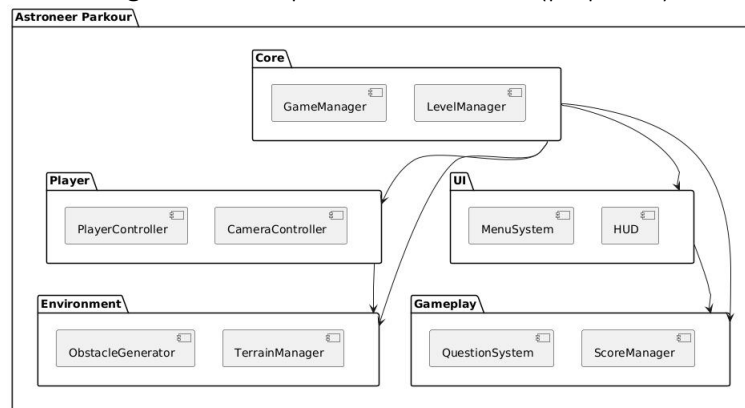


3.2.4. Diagrama de Clases

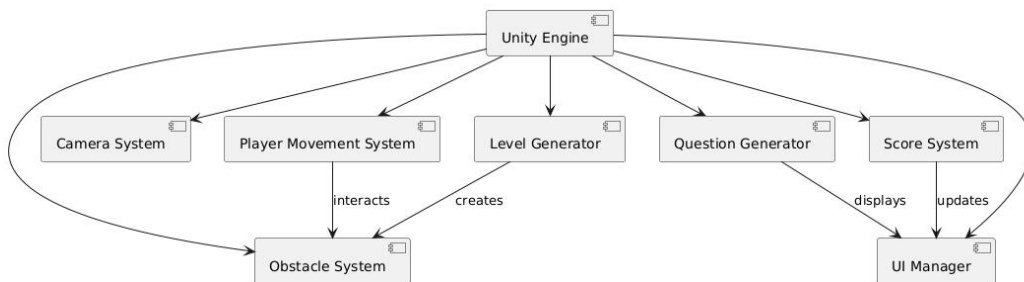


3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)

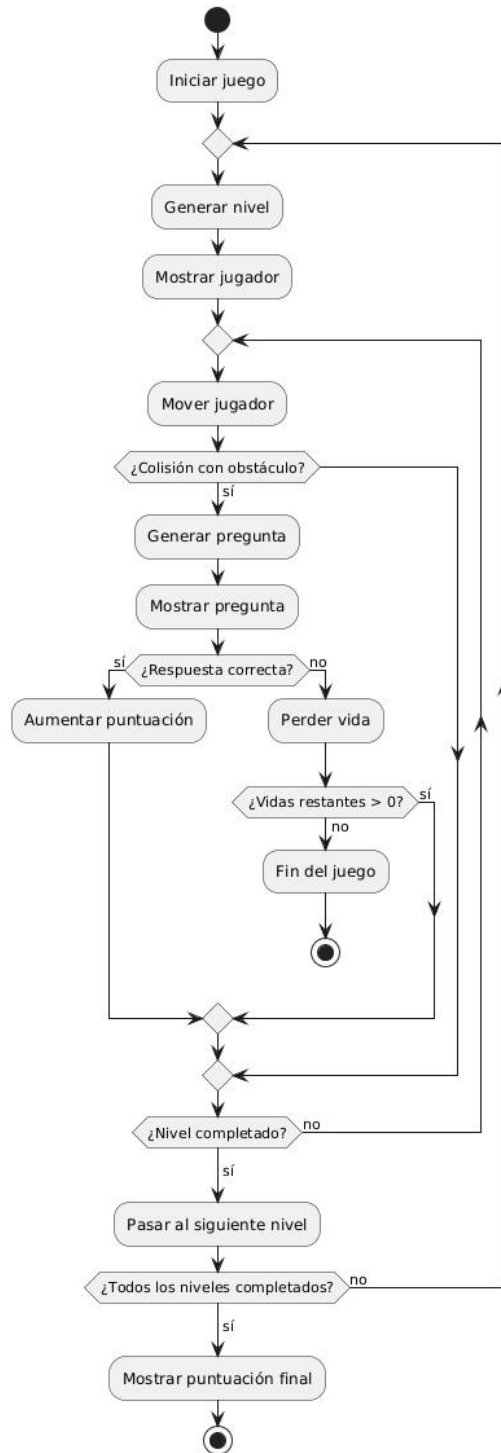


3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)



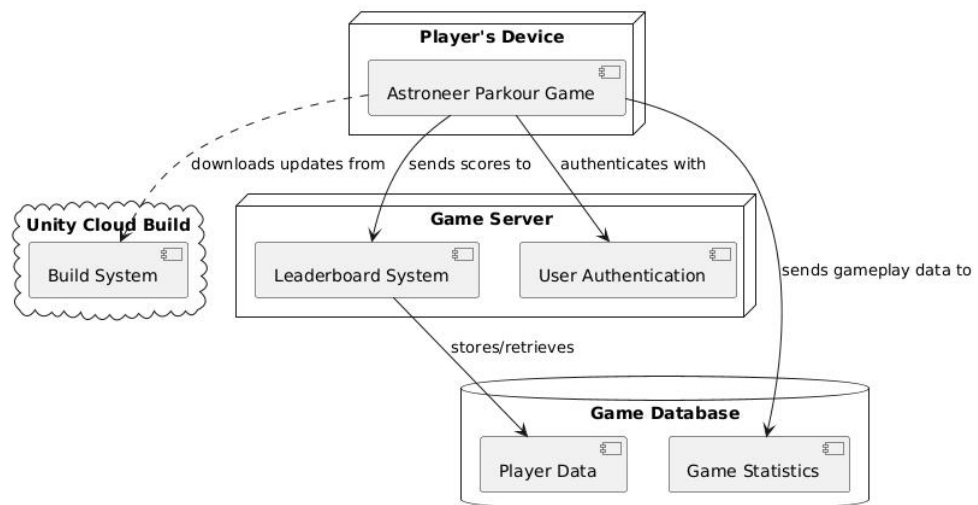
3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)



3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

4.1. Escenario de Funcionalidad

El jugador inicia el juego y se encuentra en un entorno 3D espacial. Utilizando el PlayerController, puede moverse libremente por el nivel utilizando las teclas WASD y el ratón para controlar la cámara. A medida que avanza, se enfrenta a diversos obstáculos y desafíos ambientales, como asteroides en movimiento (gestionados por AsteroidReset). Cuando el jugador colisiona con un obstáculo o cae en una zona de "lava", el sistema (LavaTrigger) genera una pregunta científica aleatoria. El jugador debe responder correctamente para continuar. Si falla, pierde una vida. El sistema de vidas se actualiza en consecuencia. El jugador puede recolectar objetos a lo largo del nivel, lo que afecta su puntuación. Al completar un nivel, el sistema (Ganador) muestra una pantalla de victoria y ofrece la opción de pasar al siguiente nivel o volver al menú principal.

4.2. Escenario de Usabilidad

Al iniciar el juego, se presenta al jugador un menú principal intuitivo con opciones claras para iniciar el juego, ajustar la configuración o salir. Los controles se explican brevemente en una pantalla de tutorial accesible desde el menú principal. Durante el juego, la interfaz de usuario (UI) muestra claramente la puntuación actual, las vidas restantes y el nivel en curso. Las preguntas científicas se presentan en un formato legible y fácil de entender, con opciones de respuesta claramente diferenciadas. El sistema de movimiento y salto es receptivo y preciso, permitiendo a jugadores de todas las edades navegar fácilmente por los niveles. Los obstáculos y peligros están claramente señalizados para evitar confusiones. La configuración del juego (Configuracion) permite ajustar fácilmente opciones como el volumen del sonido, la resolución de pantalla y la dificultad del juego para adaptarse a las preferencias y capacidades de diferentes jugadores.

4.3. Escenario de confiabilidad

El sistema implementa un guardado automático regular para evitar la pérdida de progreso en caso de cierres inesperados. Los datos del jugador, incluyendo puntuaciones y niveles desbloqueados, se almacenan de forma segura y se sincronizan con servidores en la nube cuando es posible. El juego incluye un sistema de detección y prevención de trampas para mantener la integridad de las puntuaciones en línea. Además, se implementan medidas para prevenir la manipulación de las preguntas científicas o las respuestas. En caso de errores, el sistema registra información relevante para facilitar la depuración, sin comprometer datos sensibles del jugador. Se incluye una opción para que los jugadores reporten bugs o problemas directamente desde el juego.

4.4. Escenario de rendimiento

El juego está optimizado para funcionar a 60 FPS en PCs de gama media y consolas actuales. Se implementan técnicas de nivel de detalle (LOD) y culling para mantener un rendimiento estable incluso en niveles grandes o con muchos objetos en pantalla. Los tiempos de carga entre niveles se mantienen por debajo de los 5 segundos en la mayoría de las plataformas, gracias a la carga asíncrona de recursos y la optimización de assets. El sistema de física para los asteroides y el movimiento del jugador está optimizado para minimizar el impacto en la CPU, permitiendo que el juego funcione suavemente incluso en situaciones con muchos objetos en movimiento. La generación de preguntas y la lógica del juego están diseñadas para tener un impacto mínimo en el rendimiento, asegurando que no haya caídas de FPS notables durante las secuencias de preguntas o al actualizar la puntuación y las vidas.