****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERÍA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *JumpQuest EduVenture Game***

Curso: *DISEÑO Y CREACIÓN DE VIDEOJUEGOS*

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

***Zevallos Purca Justin Zinedine (2020066924)***

***Anahua Coaquira, Mayner Gonzalo (2020067145)***

***Erick Javier SALINAS CONDORI (2020069046)***

**Tacna – Perú**

***2024***

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

Sistema *JumpQuest EduVenture Game*

Documento de Arquitectura de Software

Versión *1.0*

| CONTROL DE VERSIONES | | | | | |
| --- | --- | --- | --- | --- | --- |
| Versión | Hecha por | Revisada por | Aprobada por | Fecha | Motivo |
| 1.0 | MPV | ELV | ARV | 10/10/2020 | Versión Original |

ÍNDICE GENERAL

**Contenido**

[***1.***](#_heading=h.1fob9te) ***INTRODUCCIÓN 5***

[**1.1.**](#_heading=h.3znysh7) **Propósito (Diagrama 4+1) 5**

[**1.2.**](#_heading=h.2et92p0) **Alcance 5**

[**1.3.**](#_heading=h.tyjcwt) **Definición, siglas y abreviaturas 5**

[**1.4.**](#_heading=h.3dy6vkm) **Organización del documento 5**

[***2.***](#_heading=h.1t3h5sf) ***OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS 5***

[2.1.1.](#_heading=h.4d34og8) Requerimientos Funcionales 5

[2.1.2.](#_heading=h.17dp8vu) Requerimientos No Funcionales – Atributos de Calidad 5

[***3.***](#_heading=h.3rdcrjn) ***REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA 6***

[**3.1.**](#_heading=h.lnxbz9) **Vista de Caso de uso 6**

[3.1.1.](#_heading=h.35nkun2) Diagramas de Casos de uso 6

[**3.2.**](#_heading=h.1ksv4uv) **Vista Lógica 6**

[3.2.1.](#_heading=h.2jxsxqh) Diagrama de Subsistemas (paquetes) 7

[3.2.2.](#_heading=h.z337ya) Diagrama de Secuencia (vista de diseño) 7

[3.2.3.](#_heading=h.3j2qqm3) Diagrama de Colaboración (vista de diseño) 7

[3.2.4.](#_heading=h.1y810tw) Diagrama de Objetos 7

[3.2.5.](#_heading=h.4i7ojhp) Diagrama de Clases 7

[3.2.6.](#_heading=h.2xcytpi) Diagrama de Base de datos (relacional o no relacional) 7

[**3.3.**](#_heading=h.1ci93xb) **Vista de Implementación (vista de desarrollo) 7**

[3.3.1.](#_heading=h.3whwml4) Diagrama de arquitectura software (paquetes) 7

[3.3.2.](#_heading=h.2bn6wsx) Diagrama de arquitectura del sistema (Diagrama de componentes) 7

[**3.4.**](#_heading=h.qsh70q) **Vista de procesos 7**

[3.4.1.](#_heading=h.3as4poj) Diagrama de Procesos del sistema (diagrama de actividad) 8

[**3.5.**](#_heading=h.1pxezwc) **Vista de Despliegue (vista física) 8**

[3.5.1.](#_heading=h.49x2ik5) Diagrama de despliegue 8

[***4.***](#_heading=h.2p2csry) ***ATRIBUTOS DE CALIDAD DEL SOFTWARE 8***

[**Escenario de Funcionalidad 8**](#_heading=h.147n2zr)

[**Escenario de Usabilidad 8**](#_heading=h.3o7alnk)

[**Escenario de confiabilidad 9**](#_heading=h.23ckvvd)

[**Escenario de rendimiento 9**](#_heading=h.ihv636)

[**Escenario de mantenibilidad 9**](#_heading=h.32hioqz)

[**Otros Escenarios 9**](#_heading=h.1hmsyys)

1. **INTRODUCCIÓN**
   1. Propósito (Diagrama 4+1)

El propósito de este documento es proporcionar una visión global y resumida de la arquitectura del sistema para el proyecto JumpQuest EduVenture Game. Este juego educativo en 3D, desarrollado con Unity, combina elementos de parkour con preguntas matemáticas para crear una experiencia de aprendizaje interactiva y divertida. El diseño del sistema está influenciado por los requisitos funcionales y no funcionales, con un enfoque en la eficiencia y la portabilidad del juego.

* 1. Alcance

El proyecto consiste en el desarrollo de un juego de plataformas utilizando Unity 3D. El alcance incluye el desarrollo completo del juego, integrando niveles, personajes, mecánicas de juego (como saltos, obstáculos y recolección de objetos), y elementos visuales y sonoros que aseguren una experiencia de juego atractiva y dinámica. Se utilizarán tecnologías avanzadas para la programación, diseño gráfico y modelado 3D, garantizando la compatibilidad con múltiples plataformas de dispositivos. Se llevarán a cabo pruebas exhaustivas para garantizar la jugabilidad, detectar y corregir errores, y optimizar el rendimiento del juego en diferentes dispositivos. Además, se creará documentación técnica y de usuario final, incluyendo manuales y guías de juego. El proyecto se alinea con una visión estratégica que considera análisis de factibilidad económica, financiera, social, ambiental y legal, asegurando su viabilidad y éxito a largo plazo en el competitivo mercado de videojuegos.

* 1. Definición, siglas y abreviaturas

| **Sigla/Abreviatura** | **Definición** |
| --- | --- |
| AI | Artificial Intelligence (Inteligencia Artificial) |
| API | Application Programming Interface (Interfaz de Programación de Aplicaciones) |
| CPU | Central Processing Unit (Unidad Central de Procesamiento) |
| FPS | Frames Per Second (Cuadros por Segundo) |
| GUI | Graphical User Interface (Interfaz Gráfica de Usuario) |
| RAM | Random Access Memory (Memoria de Acceso Aleatorio) |
| SRS | Software Requirements Specification (Especificación de Requisitos de Software) |
| SAD | System Architecture Document (Documento de Arquitectura del Sistema) |
| UI | User Interface (Interfaz de Usuario) |
| UX | User Experience (Experiencia de Usuario) |

* 1. Organización del documento

Este documento está estructurado de la siguiente manera para proporcionar una visión clara y detallada de la arquitectura del software del proyecto **JumpQuest EduVenture Game**.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS**

Desarrollar un juego de plataformas en 3D utilizando Unity, enfocado en proporcionar una experiencia interactiva y educativa para los jugadores, integrando desafíos matemáticos como parte del gameplay para promover el aprendizaje interactivo de manera divertida y accesible.

* 1. Priorización de requerimientos

| **ID** | **Nombre de Requerimiento** | **Descripción** |
| --- | --- | --- |
| RF01 | Control de personaje | El sistema debe permitir al usuario controlar un personaje en un entorno de parkour. |
| RF02 | Implementación de trampas | El sistema debe implementar trampas como púas y zonas de caída. |
| RF03 | Preguntas matemáticas | El sistema debe mostrar preguntas matemáticas al usuario si cae del mapa. |
| RF04 | Registro de progreso | El sistema debe registrar el progreso del usuario, incluyendo los puntos ganados por recoger monedas y corazones. |
| RB05 | Respuesta correcta | El usuario debe responder correctamente a las preguntas para poder continuar el juego. |
| RB06 | Selección aleatoria de preguntas | Las preguntas matemáticas se seleccionan de manera aleatoria cada vez que el usuario cae del mapa. |
| RB07 | Sistema de checkpoints | El sistema debe implementar un sistema de checkpoints para que el usuario pueda reanudar desde un punto guardado. |

### Requerimientos Funcionales

| **ID** | **Nombre de Requerimiento** | **Descripción** |
| --- | --- | --- |
| RF01 | Control de personaje | El sistema debe permitir al usuario controlar un personaje en un entorno de parkour. |
| RF02 | Implementación de trampas | El sistema debe implementar trampas como púas y zonas de caída. |
| RF03 | Preguntas matemáticas | El sistema debe mostrar preguntas matemáticas al usuario si cae del mapa. |
| RF04 | Registro de progreso | El sistema debe registrar el progreso del usuario, incluyendo los puntos ganados por recoger monedas y corazones. |
| RF05 | Guardar y cargar progreso | El sistema debe permitir guardar y cargar el progreso del juego. |

### 

### Requerimientos No Funcionales – Atributos de Calidad

| **ID** | **Nombre de Requerimiento** | **Descripción** |
| --- | --- | --- |
| RNF01 | Accesibilidad por edad | El sistema debe ser accesible para usuarios de edades 8-15 años. |
| RNF02 | Especificaciones mínimas | El sistema debe funcionar en dispositivos con especificaciones mínimas de 4GB de RAM y procesador de 2.0 GHz. |
| RNF03 | Tiempo de carga | El tiempo de carga del juego no debe exceder los 10 segundos. |
| RNF04 | Interfaz intuitiva | La interfaz debe ser intuitiva y fácil de navegar para usuarios jóvenes. |

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso

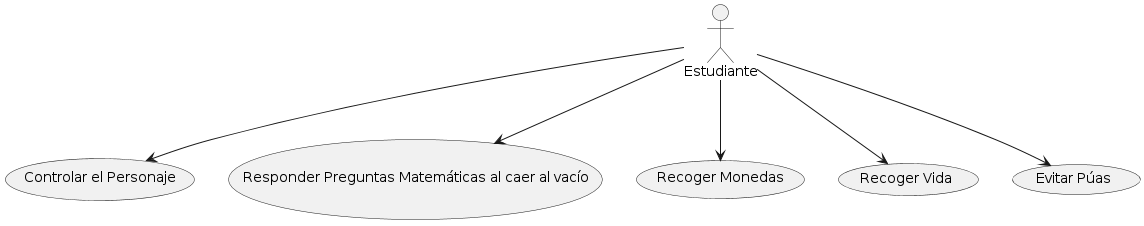
**Control del Personaje:**

El estudiante controla al personaje principal usando las teclas de flecha o WASD para moverse, saltar y esquivar obstáculos en el nivel de parkour.

**Responder Preguntas Matemáticas:**

Si el estudiante se cae del mapa, el sistema presenta una pregunta matemática. El estudiante debe responder correctamente para continuar desde el último checkpoint.

### Diagramas de Casos de uso

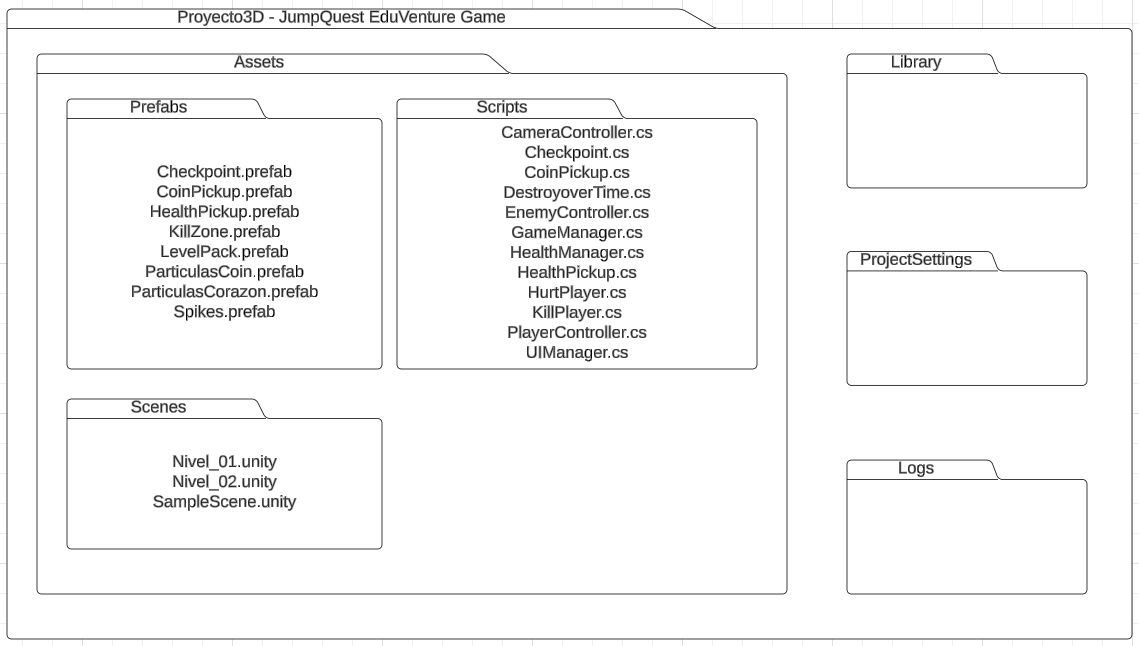


* 1. Vista Lógica

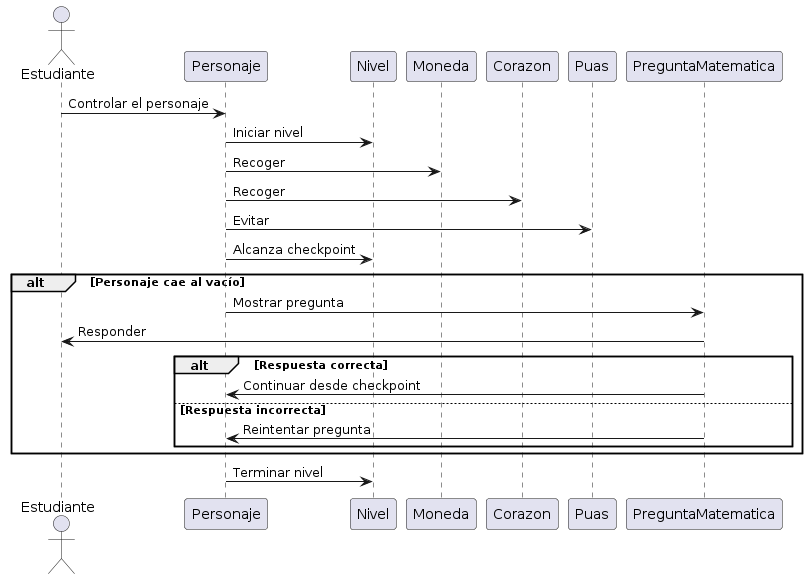
### 

La vista lógica se encarga de representar los requerimientos funcionales del sistema. Esta sección describe las partes del diseño del modelo significativas para la arquitectura, tales como subsistemas y paquetes. En el contexto del proyecto JumpQuest EduVenture Game, esta vista se centra en los componentes del juego, cómo interactúan entre sí y con el usuario, y cómo están organizados dentro del sistema.

### Diagrama de Subsistemas (paquetes)

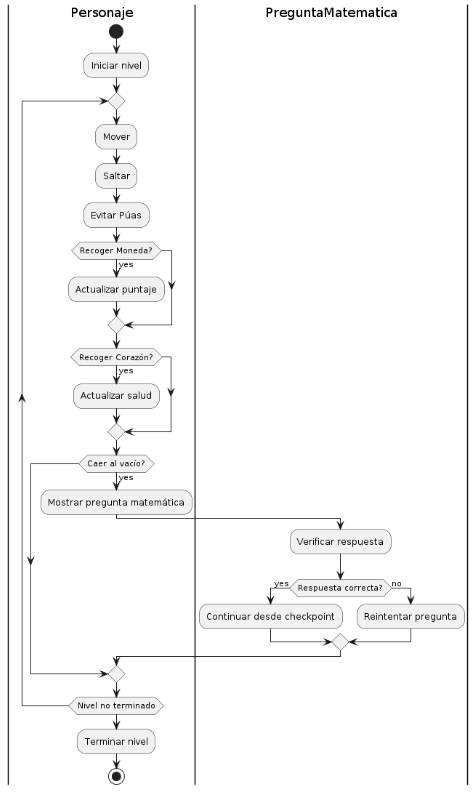


### Diagrama de Secuencia (vista de diseño)

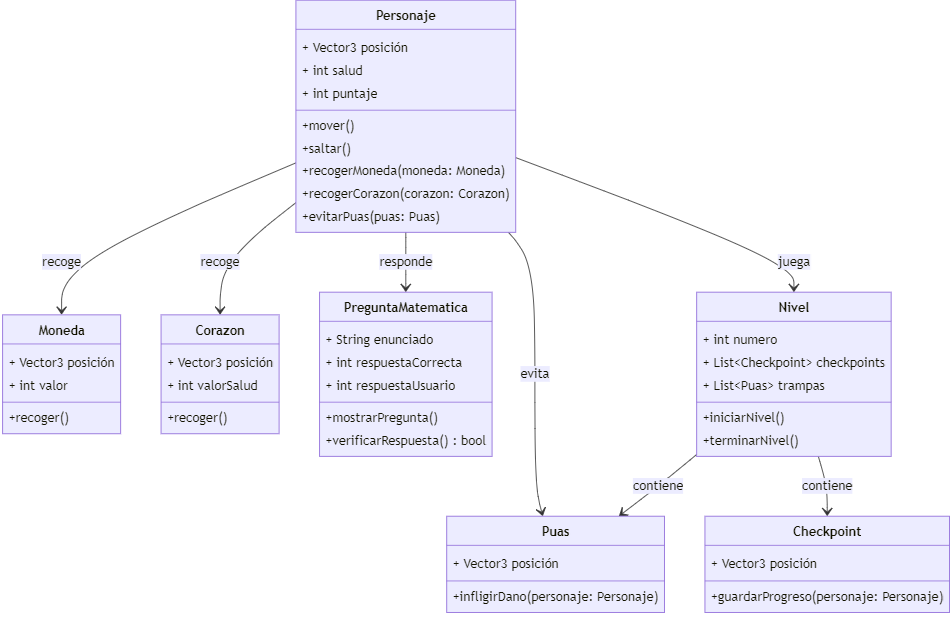


### Diagrama de Colaboración (vista de diseño)

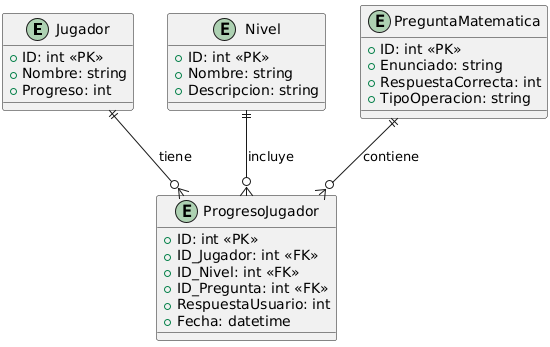
### Diagrama de Objetos



### Diagrama de Clases

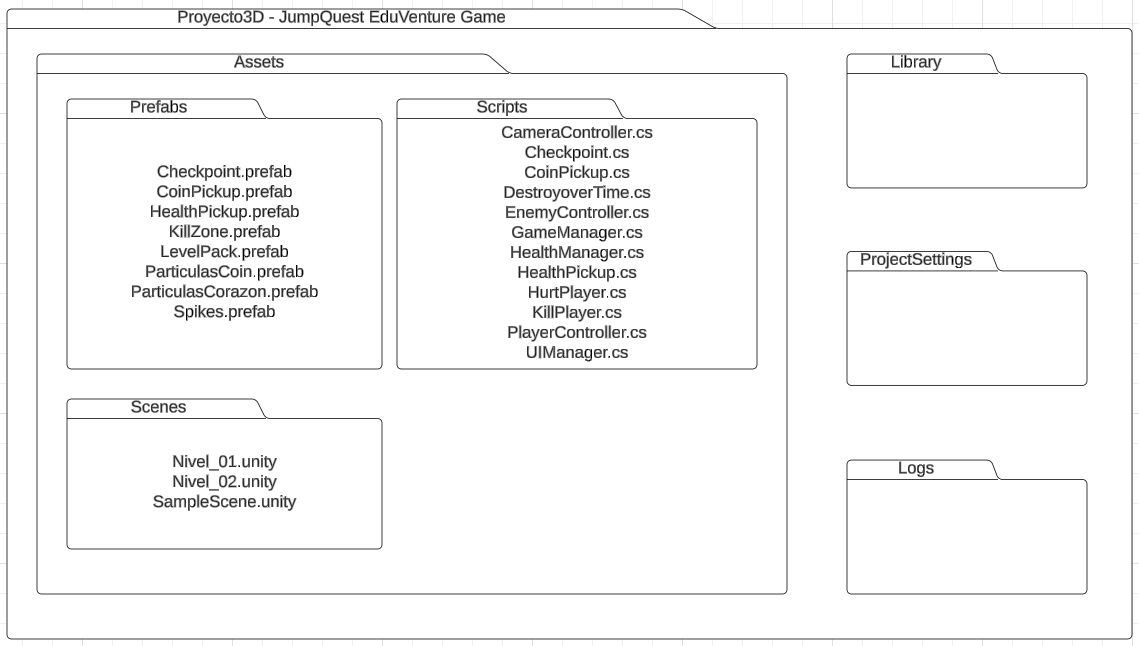


### Diagrama de Base de datos (relacional o no relacional)

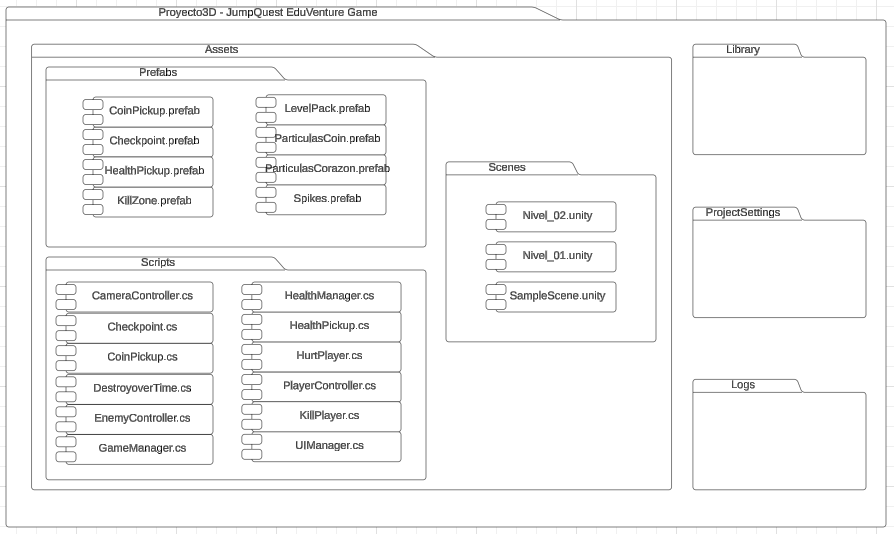


* 1. Vista de Implementación (vista de desarrollo)

### Diagrama de arquitectura software (paquetes)

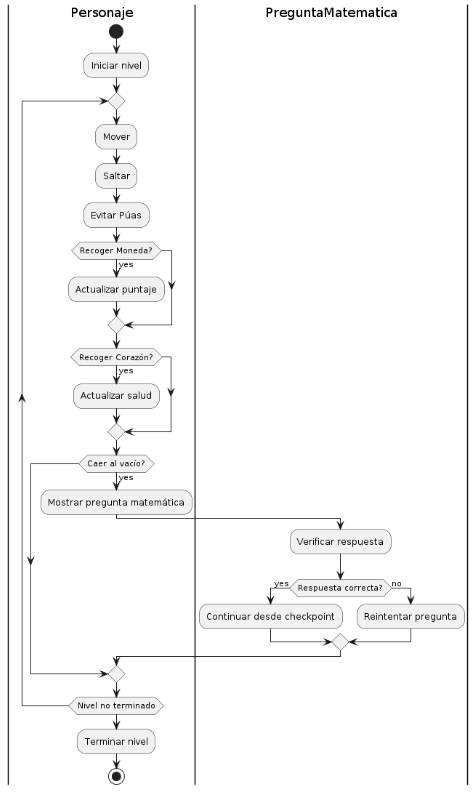


### Diagrama de arquitectura del sistema (Diagrama de componentes)



* 1. Vista de procesos

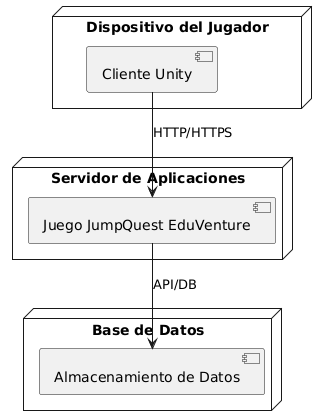
### Diagrama de Procesos del sistema (diagrama de actividad)



* 1. Vista de Despliegue (vista física)

La vista de despliegue despliega uno o más escenarios de distribución física del sistema sobre los cuales se ejecutará y hará el despliegue del mismo. Muestra la comunicación entre los diferentes nodos que componen los escenarios antes mencionados, así como el mapeo de los elementos de la Vista de Procesos en dichos nodos. En el contexto del proyecto JumpQuest EduVenture Game, esta vista describe cómo se distribuirá y ejecutará el sistema en un entorno físico.

### Diagrama de despliegue



# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

Los atributos de calidad son propiedades esenciales para garantizar la efectividad y satisfacción del usuario en un juego de plataformas desarrollado con Unity 3D. Aquí se detallan los principales escenarios de calidad relevantes:

**Escenario de Funcionalidad**

Evaluación de las capacidades y características del juego, incluyendo la jugabilidad, mecánicas de movimiento y respuesta a acciones del jugador. Asegurar que el juego cumpla con las expectativas de los jugadores en cuanto a funcionalidad y diversión.

**Escenario de Usabilidad**

Facilidad con la que los jugadores aprenden a jugar y manejan el juego. Incluye la eficiencia de la interfaz de usuario y la adaptación del juego a diferentes niveles de habilidad. Mejorar la experiencia del usuario a través de controles intuitivos, retroalimentación clara y diseño de niveles accesible.

**Escenario de confiabilidad**

Capacidad del juego para funcionar sin fallos ni interrupciones, manteniendo la estabilidad y consistencia en todas las condiciones de juego. Minimizar los errores y asegurar que el juego pueda ser jugado de manera confiable y consistente.

**Escenario de rendimiento**

Velocidad de respuesta del juego, tiempo de carga de niveles, y uso eficiente de recursos como CPU, GPU y memoria. Optimizar el rendimiento para proporcionar una experiencia de juego fluida y sin interrupciones.

**Escenario de mantenibilidad**

Facilidad con la que el juego puede ser modificado, actualizado y mantenido. Incluye la legibilidad del código, modularidad y documentación adecuada. Permitir el desarrollo continuo del juego y la incorporación de nuevas características sin comprometer la estabilidad ni la calidad.

**Otros Escenarios**

Evaluación de la capacidad del juego para manejar eventos intensivos como efectos visuales complejos o interacciones simultáneas de personajes.