****

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto Pinball Math**

Curso: Diseño y Creación de Videojuegos

Docente: Ing. Patrick Jose Cuadros Quiroga

Integrantes:

* Corrales Solis, Moisés Alessandro (2020067579).
* Lizárraga Pomareda, Sergio Pedro (2020066921).
* Arenas Paz Soldan, Miguel Jesus (2017059282).

**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 *{Nombre del Sistema}*

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 |

INDICE 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 ARQUITECTONICAS 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. Viabilidad del Sistema

Consideramos que el proyecto resultará viable porque es sencillo de llevar a múltiples plataformas para que esté al alcance de todos los usuarios, volviéndose fácil de mantener y muy rentable de producir.

* 1. Alcance

El alcance de nuestro proyecto será en todo el Perú y usaremos el motor de Unity y Visual Studio 2022 para el desarrollo del videojuego.

* 1. Descripción del problema

La problemática que este proyecto planea resolver es la falta de motivación de los jóvenes para aprender matemáticas. Pinball Math tiene como objetivo desarrollar un juego educativo que sea a la vez atractivo y desafiante, manteniendo el interés del jugador sin aburrirse o desanimar. Al integrar la resolución de ecuaciones matemáticas en un entorno lúdico, el juego buscará motivar a los jugadores a mejorar sus habilidades matemáticas de manera divertida y efectiva.

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

[Establezca las prioridades de los requerimientos y las restricciones del proyecto)

* 1. Priorización de requerimientos

1. Cuadro de Requerimientos funcionales Inicial:

| **ID** | **NOMBRE** | **DESCRIPCIÓN** | **PRIORIDAD** |
| --- | --- | --- | --- |
| RF01 | Ver opciones del juego | El sistema debe mostrar las opciones o controles que dispone el usuario. | Alta |
| RF02 | Lanzar Bola | El usuario deberá poder lanzar la bola cuando el usuario presione una tecla. | Alta |
| RF03 | Mover Palancas | El usuario deberá mover las palancas para golpear la bola. | Alta |
| RF04 | Sumar y restar números | El sistema deberá sumar o restar un número al resultado de la operación | Alta |

### Requerimientos Funcionales

1. Cuadro de Requerimientos funcionales Final:

| **ID** | **NOMBRE** | **DESCRIPCIÓN** | **PRIORIDAD** |
| --- | --- | --- | --- |
| RF01 | Ver opciones del juego | El sistema debe mostrar las opciones o controles que dispone el usuario. | Alta |
| RF02 | Ver detalles del juego | El sistema permitirá ver a los usuarios, la cantidad de bolas que tienen, el tiempo, el puntaje, el nivel, la operación matemática, etc. | Alta |
| RF03 | Lanzar Bola | El usuario deberá poder lanzar la bola cuando el usuario presione una tecla. | Alta |
| RF04 | Mover Palancas | El usuario deberá mover las palancas para golpear la bola. | Alta |
| RF05 | Sumar y restar números | El sistema deberá sumar o restar un número al resultado de la operación | Alta |

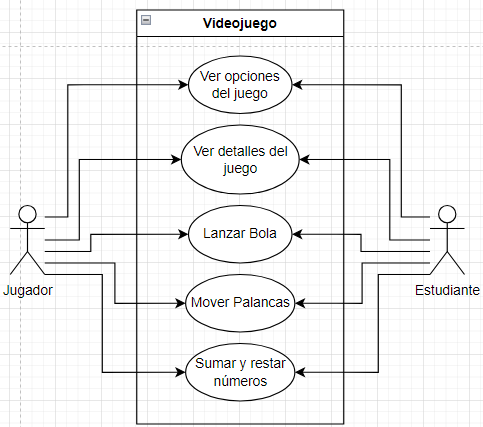
### Requerimientos No Funcionales – Atributos de Calidad

1. Cuadro de Requerimientos No funcionales:

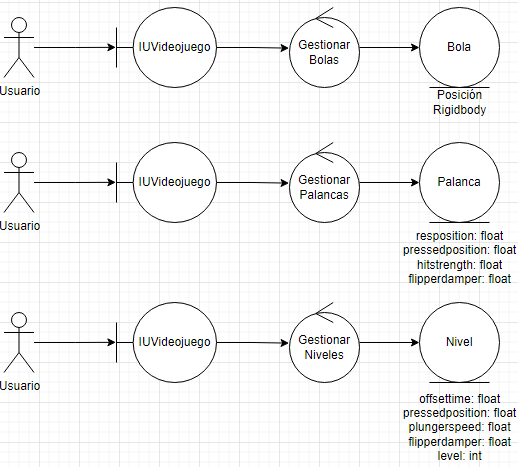
| **ID** | **NOMBRE** | **DESCRIPCIÓN** | **PRIORIDAD** |
| --- | --- | --- | --- |
| RNF01 | Eficiencia | Todo proceso o funcionalidad de la aplicación deberá de tener un tiempo de respuesta de 10 segundos máximo. | Alta |
| RNF02 | Usabilidad | El software mostrará una interfaz dinámica para que el usuario pueda adecuarse al sistema fácilmente. | Alta |
| RNF03 | Disponibilidad | La aplicación deberá funcionar al 90% en caso de error o no esté disponible, notificará con un mensaje de error con el servicio. | Alta |
| RNF04 | Escalabilidad | La aplicación mantendrá su buen funcionamiento y rendimiento tras los cambios o crecimiento que pueda aplicarse a lo largo de su ciclo de vida. | Alta |

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso



### 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.]*

### 

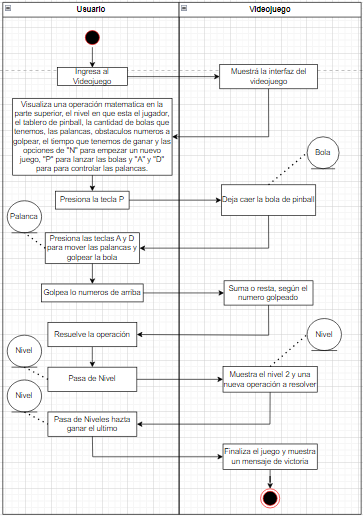
### Diagrama de Subsistemas (paquetes)

| **Número** | **Caso de Uso** | **Narrativa** |
| --- | --- | --- |
| CU-01 | Ver opciones del juego | 1. El usuario ingresa al videojuego. 2. El sistema mostrará, en la parte lateral izquierda las opciones de: "N" para empezar un nuevo juego, "P" para lanzar las bolas y "A" y "D" para controlar las palancas. |
| CU-02 | Ver detalles del juego | 1. El usuario ingresa al videojuego. 2. El sistema mostrará el tiempo que dispone el usuario debajo de las opciones, el nivel en que está arriba de las opciones, la operación a resolver en la parte central superior, el puntaje del jugador en la parte lateral derecha y el resultado de la operación arriba del puntaje. |
| CU-03 | Lanzar Bola | 1. El usuario ingresa al videojuego. 2. El sistema mostrará las opciones. 3. El usuario presiona la tecla “P”. 4. El sistema mostrará o lanzará la bola en el tablero. |
| CU-04 | Mover Palancas | 1. El usuario ingresa al videojuego. 2. El sistema mostrará las opciones. 3. El usuario presiona las teclas “A” y “D”. 4. El sistema mostrará el movimiento de las palancas |
| CU-05 | Sumar y restar números | 1. El usuario ingresa al videojuego. 2. El sistema mostrará la operación matemática. 3. El jugador golpea los números de arriba con la bola para llegar a la respuesta de la operación 4. El sistema mostrará cómo se suma o resta el resultado de la operación hasta resolver la operación |

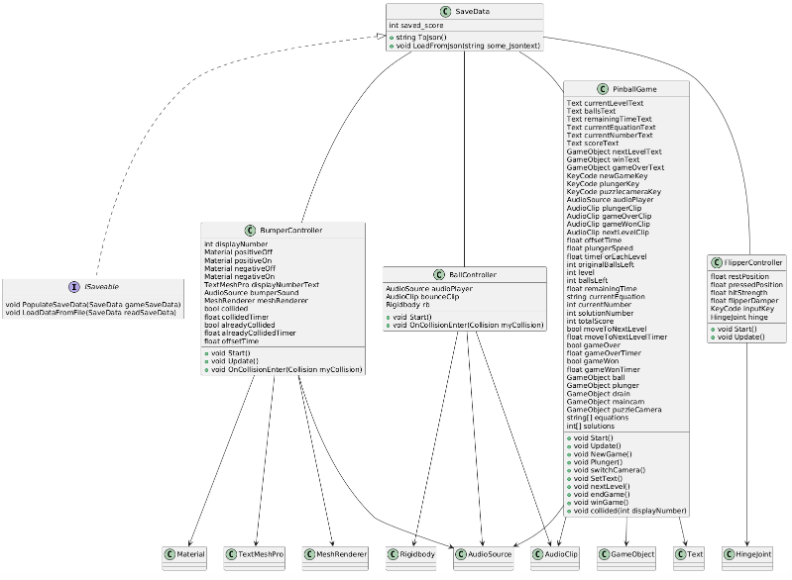
### Diagrama de Secuencia (vista de diseño)

### Diagrama de Colaboración (vista de diseño)

### Diagrama de Objetos



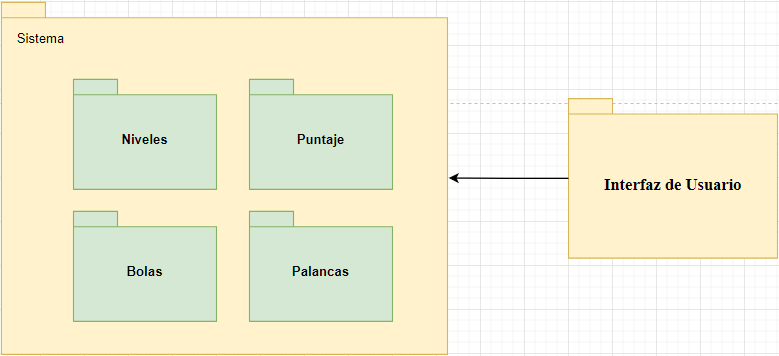
### Diagrama de Clases



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

* 1. Vista de Implementación (vista de desarrollo)

### Diagrama de arquitectura software (paquetes)

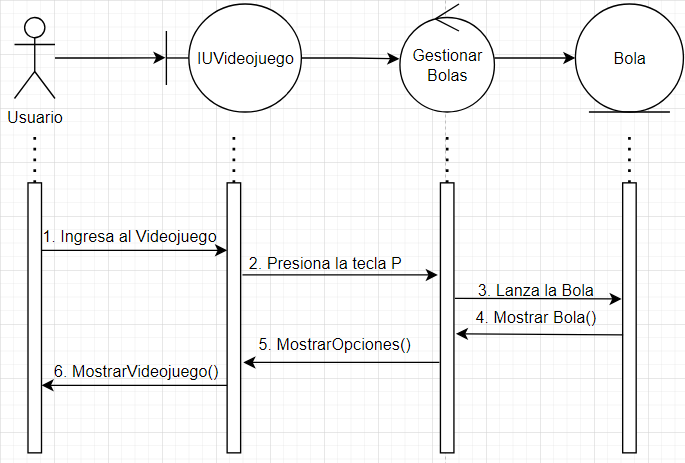


### Diagrama de arquitectura del sistema (Diagrama de componentes)



* 1. Vista de procesos

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



* 1. Vista de Despliegue (vista física)

### Diagrama de despliegue

**

*.*

**RECOMENDACIONES**

* Refinamiento Continuo: Mantener un enfoque iterativo en el desarrollo para mejorar continuamente la plataforma basándose en la retroalimentación de los usuarios y las tendencias cambiantes del mercado.
* Promoción y Marketing: Implementar estrategias de promoción y marketing efectivas para atraer nuevos usuarios y fomentar la participación en la plataforma, utilizando redes sociales y colaboraciones con socios locales.
* Pruebas de Usabilidad: Realizar pruebas con usuarios reales para evaluar la facilidad de uso y la efectividad de la plataforma, identificando áreas de mejora en la interfaz de usuario y la navegación.