

**UNIVERSIDAD PRIVADA DE TACNA**

**FACULTAD DE INGENIERIA**

**Escuela Profesional de Ingeniería de Sistemas**

**Proyecto *TruckMonster***

Curso: *Diseño y Creación de Videojuegos*

Docente: *Ing. Patrick Jose Cuadros Quiroga*

Integrantes:

1. Cano Sucso, Anthony Alexander
2. Jarro Cachi, Jose Luis
3. Rivera mendoaza, Jhonny
4. Valverde Zamora, jean Pier Elias

**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 TruckMonster-Driving Knowledge

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. Propósito (Diagrama 4+1)

El propósito de este informe es presentar una visión global y resumida de la arquitectura del sistema de un videojuego en realidad virtual (VR) desarrollado en Unity, enfocado en la preparación para el examen de conducir. Los objetivos generales del diseño incluyen proporcionar una experiencia inmersiva y efectiva para los usuarios, combinando simulación de manejo y preguntas teóricas del examen de conducir.

* 1. Alcance

El documento se centrará en el desarrollo de la vista lógica del framework del videojuego en realidad virtual. Se incluirán los aspectos fundamentales del resto de las vistas, como la vista de desarrollo y la vista de despliegue, omitiendo aquellas que no se consideren pertinentes, como la vista de procesos. Se detallarán las características principales del sistema, su arquitectura lógica, y los componentes clave que conforman el videojuego.

* 1. Definición, siglas y abreviaturas

**VR**: Virtual Reality (Realidad Virtual)

**SDK**: Software Development Kit (Kit de Desarrollo de Software)

**Unity**: Motor de desarrollo de videojuegos utilizado para crear el sistema.

**Oculus Rift**: Dispositivo de realidad virtual.

**HTC Vive**: Dispositivo de realidad virtual.

**SRS**: Software Requirements Specification (Especificación de Requisitos del Software)

**SAD**: Software Architecture Document (Documento de Arquitectura del Software)

**GUI**: Graphical User Interface (Interfaz Gráfica de Usuario)

**FPS**: Frames Per Second (Cuadros por Segundo)

* 1. Organización del documento

# **OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS**

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

* 1. Priorización de requerimientos

### Requerimientos Funcionales

##### Requerimientos Funcionales

1. **Simulación de Manejo en Realidad Virtual**
   * Crear un entorno de manejo en VR que imite las condiciones reales de conducción.
   * Integración de controles intuitivos para simular la experiencia de conducir un vehículo.
2. **Sistema de Preguntas del Examen Teórico**
   * Implementar un sistema de preguntas que simule el examen teórico de conducir.
   * Mostrar preguntas en la pista de manejo y permitir respuestas interactivas.
3. **Evaluación de Desempeño**
   * Evaluar el desempeño del usuario tanto en la conducción como en la resolución de preguntas teóricas.
   * Proporcionar retroalimentación inmediata y recomendaciones de mejora.
4. **Progreso del Usuario**
   * Registrar y rastrear el progreso del usuario a lo largo de las sesiones de entrenamiento.
   * Permitir al usuario revisar su historial de desempeño y resultados.
5. **Interfaz de Usuario (UI) Intuitiva**
   * Diseñar una interfaz de usuario que sea fácil de usar y que facilite la navegación por las distintas funcionalidades del juego.

### Requerimientos No Funcionales – Atributos de Calidad

**Rendimiento**

* El sistema debe funcionar de manera fluida con un mínimo de 60 FPS para garantizar una experiencia inmersiva.

**Escalabilidad**

* La arquitectura del sistema debe permitir la incorporación de nuevos niveles, preguntas y funcionalidades sin comprometer el rendimiento.

**Seguridad**

* Proteger los datos del usuario y garantizar la privacidad de la información.

**Usabilidad**

* La interfaz y las interacciones deben ser intuitivas y fáciles de entender para usuarios de todas las edades.

**Compatibilidad**

* El juego debe ser compatible con múltiples dispositivos de VR como Oculus Rift y HTC Vive.
  1. Restricciones

**Presupuesto Limitado**

* El desarrollo del sistema debe mantenerse dentro del presupuesto asignado de S/. 33,400.

**Tiempo de Desarrollo**

* El proyecto debe ser completado en un plazo de 6 meses, según el cronograma establecido.

**Compatibilidad Tecnológica**

* El sistema debe ser desarrollado utilizando Unity, y debe ser compatible con las plataformas de VR especificadas.

**Requerimientos Legales**

* El contenido del videojuego debe cumplir con las normativas legales y de seguridad vigentes para software educativo y de simulación.

**Accesibilidad**

* El juego debe ser accesible para usuarios con diferentes niveles de habilidad en VR y en el uso de tecnología en general.

# **REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA**

* 1. Vista de Caso de uso

### Diagramas de Casos de uso

* 1. Vista Lógica

### 

### 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)

### Diagrama de despliegue

# **ATRIBUTOS DE CALIDAD DEL SOFTWARE**

### **Escenario de Funcionalidad**

**Descripción:** La funcionalidad del software debe asegurar que todas las características esenciales estén presentes y funcionen como se espera.

**Escenario:** El usuario inicia la simulación de manejo y puede acceder a todas las funciones del sistema, como responder preguntas teóricas y recibir retroalimentación.

**Medida:**

* Todas las funciones del juego están accesibles y operativas en un 99% del tiempo.
* Las preguntas del examen teórico se despliegan correctamente en el 100% de los intentos.

### **Escenario de Usabilidad**

**Descripción:** La usabilidad del software se refiere a la facilidad con que los usuarios pueden aprender y utilizar el sistema.

**Escenario:** Un usuario novato sin experiencia previa en VR puede iniciar sesión, navegar por la interfaz del juego y comenzar la simulación de manejo en menos de 5 minutos.

**Medida:**

* El tiempo promedio para que un usuario nuevo complete su primera sesión es menor a 5 minutos.
* La tasa de satisfacción del usuario en las encuestas de usabilidad es superior al 85%.

### **Escenario de Confiabilidad**

**Descripción:** La confiabilidad del software implica que el sistema es robusto y funciona sin fallos bajo condiciones específicas.

**Escenario:** Durante una sesión de entrenamiento de 1 hora, el sistema no debe experimentar fallos críticos ni errores que requieran reiniciar el juego.

**Medida:**

* El sistema tiene un tiempo de inactividad menor al 1% durante el uso.
* La tasa de fallos críticos es menor a 1 por cada 100 horas de uso.

### **Escenario de Rendimiento**

**Descripción:** El rendimiento del software se refiere a la capacidad del sistema para responder de manera eficiente bajo diversas cargas de trabajo.

**Escenario:** Durante la simulación, el juego debe mantener una tasa de fotogramas por segundo (FPS) constante de al menos 60 FPS.

**Medida:**

* La tasa de FPS promedio durante las sesiones de juego es de 60 o superior.
* El tiempo de respuesta para cargar nuevas preguntas y escenarios es menor a 2 segundos.

### **Escenario de Mantenibilidad**

**Descripción:** La mantenibilidad del software abarca la facilidad con que el sistema puede ser modificado para corregir errores, mejorar el rendimiento o adaptarse a nuevos requerimientos.

**Escenario:** Los desarrolladores pueden implementar y probar nuevas preguntas del examen en menos de 2 horas sin afectar las funciones existentes.

**Medida:**

* El tiempo promedio para implementar y probar nuevos contenidos es menor a 2 horas.
* La documentación del código está completa y actualizada en un 100%.

### **Otros Escenarios**

**Seguridad:** **Descripción:** Proteger la información del usuario y evitar accesos no autorizados. **Escenario:** Los datos personales de los usuarios están encriptados y solo accesibles por el sistema. **Medida:**

* El sistema cumple con todas las normativas de protección de datos.
* No se han registrado brechas de seguridad.

**Escalabilidad:** **Descripción:** La capacidad del sistema para manejar un aumento en la carga de trabajo. **Escenario:** El sistema puede soportar hasta 100 usuarios simultáneos sin degradación del rendimiento. **Medida:**

* El sistema mantiene un rendimiento óptimo con hasta 100 usuarios concurrentes.
* La latencia promedio por usuario se mantiene por debajo de 50 ms.

**Compatibilidad:** **Descripción:** El software debe funcionar en múltiples dispositivos y plataformas de VR. **Escenario:** El juego se ejecuta sin problemas tanto en Oculus Rift como en HTC Vive. **Medida:**

* La tasa de compatibilidad en los dispositivos especificados es del 100%.
* Las pruebas de compatibilidad no muestran errores en diferentes dispositivos VR.