

---

# SISTEMAS DISTRIBUIDOS I: DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS (SRS)

---

Proyecto: Asymetric Puzzle



## **INTEGRANTES:**

Julia Alejandra Rodríguez Abud - Líder de proyecto y Diseñador de software

Carlos Cárdenas Ruiz– Desarrollador de software

Gustavo Alfonso Orozco Velázquez – Analista de Software

## **STAKEHOLDERS:**

Dr. Félix Francisco Ramos Corchado



16 DE ABRIL DE 2021  
MERCURIO SYSTEMS  
CINVESTAV Unidad Guadalajara

## Prefacio

Se espera que este documento proporcione la información suficiente para el diseño y la implementación de una aplicación distribuida que utiliza el modelo cliente servidor para proporcionar un servicio en un videojuego cooperativo enfocado en la colaboración y el trabajo conjunto entre usuarios para la solución de problemas dentro de un modelo de vista isométrica.

Este documento será utilizado principalmente por:

- Clientes del sistema.
- Ingeniero administrador de proyecto.
- Ingeniero de implementación del sistema.
- Ingeniero de pruebas del sistema.
- Ingeniero de mantenimiento del sistema.

HISTORIAL DEL DOCUMENTO			
Fecha	Versión	Comentarios	Autor
28 marzo 2021	0.1	Versión inicial.	Gustavo Orozco
28 marzo 2021	0.1	Revisada por el equipo.	Julia Abud
15 abril 2021	1.0	Agregado de especificaciones.	Gustavo Orozco
15 abril 2021	1.0	Revisada por el equipo.	Julia Abud
16 abril 2021	1.0	Versión final.	Dr. Félix Ramos

## **Contenido**

<b><i>Prefacio</i></b> .....	<b>1</b>
<b><i>Contenido</i></b> .....	<b>2</b>
<b><i>Introducción</i></b> .....	<b>3</b>
Resumen del documento .....	3
<b><i>1. Propósito</i></b> .....	<b>4</b>
1.1 Alcance del proyecto .....	4
<b><i>2. Referencias:</i></b> .....	<b>4</b>
<b><i>3. Definiciones</i></b> .....	<b>5</b>
<b><i>4. Análisis del sistema</i></b> .....	<b>6</b>
4.1 Entorno del sistema: .....	6
4.2 Definición de los requerimientos del usuario .....	6
4.3 Trazabilidad.....	7
<b><i>5. Especificación de los requerimientos del sistema</i></b> .....	<b>8</b>
5.1 Requerimientos funcionales.....	8
5.2 Especificación de requerimientos funcionales.....	10
5.3 Dependencias .....	14
5.4 Prioridades de desarrollo .....	15
<b><i>6. Requerimientos no funcionales</i></b> .....	<b>16</b>
6.1 Modelos del sistema propuesto .....	17
6.2 Arquitectura del sistema propuesto .....	18
6.3 Requerimientos de dominio .....	19
<b><i>7. Evolución del sistema</i></b> .....	<b>20</b>
<b><i>8. Apéndices</i></b> .....	<b>21</b>

## **Introducción**

Con el surgimiento de la necesidad de unir un conjunto de fuerzas para el trabajo colaborativo dentro de un mundo conectado, se demuestra una mayor capacidad para tratar problemas que afectan a los agentes dentro de un entorno de simulación dentro de un sistema distribuido.

A lo cual se solicitó el desarrollo de una aplicación a nuestra casa desarrolladora para modelar un entorno multijugador donde los usuarios pueden colaborar e interactuar con su entorno para solucionar problemas, resolver acertijos en conjunto y avanzar a un lugar destino. Esto con el soporte de puertos y conexión permanente a un servidor web para realizar consultas RPC mediante un middleware para la gestión de peticiones al servidor.

## **Resumen del documento**

En los siguientes capítulos, se realiza una descripción general sobre el análisis de especificación de requerimientos del cliente, este documento ofrece una descripción de los requerimientos funcionales, no funcionales y de dominio. Utilizado para establecer contexto para la especificación de requisitos técnicos a ser implementados de acuerdo con las necesidades del sistema.

## 1. Propósito

Este documento describe los requerimientos de software del sistema para el desarrollo de un videojuego cooperativo multijugador en vista isométrica con el objetivo principal de utilizar los recursos y capacidades del jugador para resolver acertijos en conjunto sobre problemas dentro de un entorno de simulación.

### 1.1 Alcance del proyecto

La aplicación se desarrolla utilizando tecnologías que pueden ejecutarse y adaptarse en dispositivos con acceso a internet con el uso de una arquitectura cliente servidor que soporte la conexión e interacción de tres jugadores dentro de un videojuego de no suma cero donde los usuarios colaboran para un bien común con otros usuarios, fomentando la relación de compañerismo y el uso de capacidades y recursos en un entorno de simulación.

Alcance:	<p>Este documento de requerimientos de software es la base del desarrollo de software del proyecto. Describe los siguientes tópicos:</p> <ul style="list-style-type: none"><li>- Requerimientos del usuario.</li><li>- Requerimientos del sistema.</li><li>- Requerimientos funcionales.</li><li>- Requerimientos no funcionales.</li><li>- Requerimientos de dominio.</li></ul>
----------	--

## 2. Referencias:

IEEE Estándar 830-1998: Práctica recomendada de IEEE para especificaciones de requisitos de software. IEEE, 1998.

### 3. Definiciones

*Web Socket:* Es un protocolo de comunicación full-dúplex de una sola conexión permanente, que permite realizar Stream de mensajes y transferencia de contenido en tiempo real.

*Agente:* Un sistema cooperativo está formado por un conjunto de agentes (jugadores) comunicándose y que trabajan para lograr un objetivo global o un trabajo común.

*RPC:* Llamadas a procedimientos remotos realizadas al servidor.

*RMI:* Comunicación entre objetos distribuidos, extensión del modelo de programación de objetos (POO), de invocación local de un método conocido hacia invocación remota.

*Servicio:* Web service consumido mediante la comunicación durante la ejecución de la aplicación.

*Modelo Cliente servidor:* La arquitectura cliente-servidor es un modelo de diseño de software en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

*Middleware:* Módulos “plug and play” que se pueden apilar arbitrariamente en cualquier orden y provén cierta funcionalidad.

*Node.js:* Es un controlador de tareas que permite a JavaScript ejecutarse del lado del servidor es un entorno de programación construido sobre el motor Chrome V8.

*Treads:* Los hilos son accesos al sistema de administración de memoria de un dispositivo para la gestión del consumo y la prioridad de ejecución de una operación que se ejecuta durante el procesamiento de información de la aplicación.

*Usuario:* Persona (jugador) que ingresa al sistema.

*No suma cero:* Sistema desarrollado en base a la teoría de juegos cooperativos donde todos los participantes de la partida colaboran entre sí para obtener resultados en igualdad de condiciones, dando como resultado que todos ganen o todos pierdan.

*Unity:* Software para el desarrollo de proyectos en entornos gráficos, interfaz de usuario, game object, componentes, capas y escenas; Así como recursos Assets y prefabs para la construcción y parametrización de videojuegos.

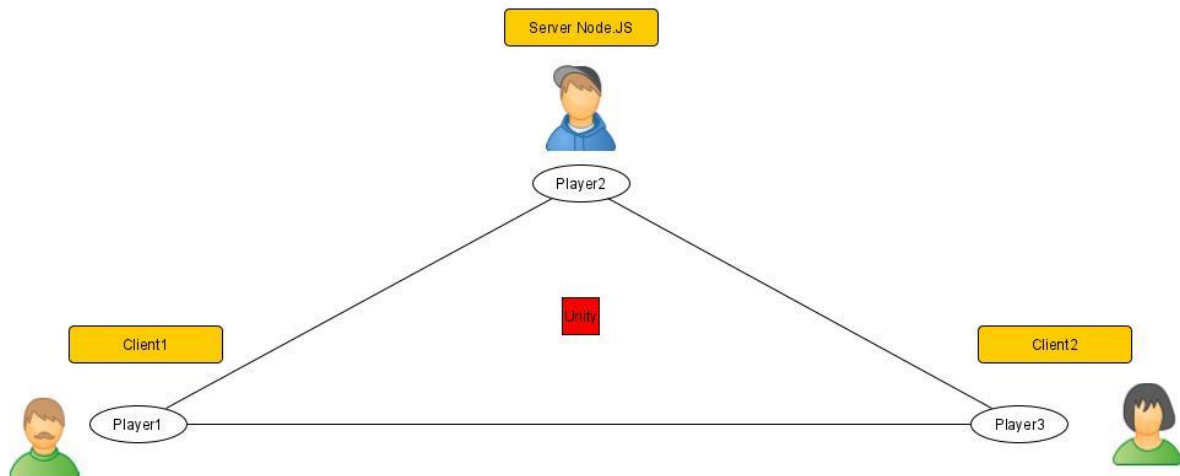
*Especificación de requisitos de software (SRS):* Un documento que describe las funciones de un sistema propuesto y las restricciones bajo las cuales debe operar.

*Parte interesada (Stakeholder):* Cualquier persona interesada en el proyecto que no sea desarrollador.

*Lobby:* Sala de espera en una sesión multijugador donde se realiza la interconexión de usuarios dentro de la aplicación.

#### 4. Análisis del sistema

Diagrama de caso de uso:



##### 4.1 Entorno del sistema:

Plataforma web desarrollada en Unity con servicio a componentes mediante un middleware que permite la interacción con soporte multijugador en un entorno de simulación para la solución de problemas mediante el uso de recursos y fomentar el comportamiento colaborativo simulado dentro de un entorno.

##### 4.2 Definición de los requerimientos del usuario

Con la necesidad de fomentar el trabajo colaborativo como un método para la integración de capacidades y recursos que permitan resolver problemas en un entorno donde se requiere expresar e incentivar a compartir en un esfuerzo conjunto, maneras para desempeñar mejor ciertas actividades y superar retos como parte de la experiencia de juego; el jugador puede interactuar con otros jugadores dentro de un nivel de juego con el objetivo de superar diversas pruebas en conjunto utilizando las herramientas habilitadas a su disposición en un tiempo de simulación.

#### 4.3 Trazabilidad

Matriz de trazabilidad de requisitos:

Requisitos del negocio	Requisito del usuario	Requisito del sistema
El sistema soporta llamadas RPC al servidor por parte del cliente.	El sistema requiere soportar a tres jugadores para iniciar la partida multijugador.	El sistema debe administrar los componentes y dispositivos externos mediante un middleware que sincronice la partida.
El sistema permite un acceso a las funciones multijugador utilizando un servidor que soporte el servicio mediante una conexión segura.	El sistema deberá soportar a los usuarios conectados al servidor y resolver peticiones en caso de desconexión del cliente.	El sistema debe proporcionar las herramientas y recursos disponibles dentro de la partida para cada jugador.
El sistema deberá proporcionar un servicio con web services para su funcionamiento distribuido.	El sistema deberá ser intuitivo, accesible al usuario mediante parámetros de interfaz.	El sistema gestiona la transferencia de datos.
Se utilizará la plataforma Unity para proporcionar el entorno de simulación con conexión permanente a un servidor externo.	El usuario que crea la partida se convierte en el coordinador de la sesión.	El sistema debe obtener información y consultar el estado actual de la partida.



## 5. Especificación de los requerimientos del sistema

Esta sección describe los casos de uso para cada uno de los subsistemas. El usuario es el actor principal en este sistema.

### 5.1 Requerimientos funcionales

- El sistema soporta conexiones y web services al servidor por parte del cliente.

#### **SRS\_ID:** SRS\_ER-F:01

**Descripción:** Antes de que se pueda iniciar este caso de uso, el usuario ya ha accedido a la aplicación:

- El sistema muestra la interfaz “lobby” multijugador para interconectar jugadores.
  - El sistema carga los componentes para ejecutar llamadas a la aplicación mediante RPC al servidor.
  - El sistema realiza una conexión a un servidor disponible para soportar a tres jugadores activos.
- El sistema permite un acceso a las funciones multijugador utilizando un servidor que soporte el servicio.

#### **SRS\_ID:** SRS\_ER-F:02

**Descripción:** Antes de que se pueda iniciar este caso de uso, el usuario ya ha accedido e iniciado sesión a la aplicación. El usuario entra y crea una partida.

- El sistema detecta los componentes y dispositivos para el control de la aplicación mediante el middleware.
- El sistema establece los parámetros de juego y carga los archivos y páginas de servicio como respuesta RPC del servidor para generar la partida.
- El sistema establece la sincronización a un servidor y web services para responder a las acciones del jugador.
- El sistema configura los puertos y controladores externos al middleware.

- El sistema establece una conexión a un servidor.

**SRS\_ID:** SRS\_ER-F:03

**Descripción:** Antes de que se pueda iniciar este caso de uso, el usuario ya ha accedido a la aplicación.

- El usuario ingresa en el sistema.
  - El sistema calcula los jugadores conectados.
  - El sistema proporciona su nombre de jugador.
  - El usuario inicia la partida generada por el sistema.
  - El usuario interactúa con objetos y otros jugadores en una partida.
- El usuario debe entrar al lobby dentro del sistema para validar la sesión y desplegar el sistema con la información del usuario.

**SRS\_ID:** SRS\_ER-F:04

**Descripción:** Antes de que se pueda iniciar este caso de uso, el usuario ya ha accedido a la aplicación.

- El usuario ingresa al sistema y se conecta a un servidor.
- El usuario introduce el nombre dentro del inicio de sesión.
- El sistema valida el nombre de usuario.

## 5.2 Especificación de requerimientos funcionales

A continuación, se describen los objetivos de trazabilidad sobre la especificación del cliente:

- Soporte de llamadas al servidor y objetos de la aplicación.

SRS_ID:	SRS_ER-F:01
Prioridad:	Alta.
Estabilidad:	Alta.
Función:	El sistema soporta llamadas RPC y RMI al servidor por parte del cliente.
Descripción:	Permite recibir los componentes necesarios dentro del dispositivo precisando un correcto funcionamiento del sistema.
Entradas:	La ejecución de la aplicación introduce a establecer una comunicación full dúplex entre los jugadores conectados, el usuario podrá ingresar a una partida multijugador, generar una partida mediante parámetros y realizar ajustes de accesibilidad.
Procedencia:	Detección de la sesión para interconectar al jugador y cargar los datos en el sistema. Otros datos son consultados al servidor mediante llamadas RPC.
Salida:	Converge una conexión permanente del usuario al sistema, consumiendo los servicios disponibles para generar una partida en línea.
Destino:	Interfaz del usuario.
Efectos colaterales:	Ninguno.
Dependencia:	SRS_ER-F:04

- El sistema configura las funciones multijugador utilizando un servidor para consumir el servicio.

SRS_ID:	SRS_ER-F:02
Prioridad:	Alta.
Estabilidad:	Alta.
Función:	El sistema permite un acceso a las funciones multijugador utilizando un servidor que soporte el servicio.
Descripción:	Cuando el usuario entre en el sistema, se configuran los dispositivos, se cargan los componentes necesarios, se establece conexión y se conecta al usuario con un servidor distribuido.
Procedencia:	Se establece conexión síncrona mediante el servidor distribuido.
Salida:	Muestra otros usuarios conectados disponibles para establecer una partida mediante una interfaz.
Destino:	Interfaz de usuario.
Precondición:	El usuario entrar en el sistema para iniciar sesión.
Postcondición:	El sistema establece una conexión permanente entre jugadores conectados para iniciar una partida.
Efectos colaterales:	El servidor no está disponible; en tal caso se envía un mensaje de error.
Dependencia:	SRS_ER-F:01, SRS_ER-F:04

- El sistema establece una conexión a un servidor.

SRS_ID:	SRS_ER-F:03
Prioridad:	Alta.
Estabilidad:	Alta.
Función:	Establecer conexión a un servidor remoto.
Descripción:	El usuario debe entrar al lobby dentro del sistema para validar la sesión y desplegar el sistema con la información del usuario.
Procedencia:	Escuchar la conexión del usuario al conectarse al servidor, realizar paginación de información requerida para almacenar en la memoria local. Otros datos son consultados mediante web services.
Salida:	El sistema calcula los jugadores conectados, carga de información y realizar operaciones permanentes durante la sesión del jugador.
Destino:	Interfaz de usuario.
Precondición:	Las indicaciones no están pensadas para soportar más usuarios de los establecidos en el documento.
Efectos colaterales:	El usuario no puede entrar a un servidor. Fallo de conexión.
Dependencia:	SRS_ER-F:01, SRS_ER-F:02, SRS_ER-F:04

- El usuario debe ejecutar la aplicación en el sistema para validar la sesión y desplegar la información del nivel de la partida dentro del sistema.

SRS_ID:	SRS_ER-F:04
Prioridad:	Alta.
Estabilidad:	Alta.
Función:	El usuario es cualquier persona en general que desee utilizar la aplicación.
Descripción:	El usuario se registra en el sistema, sus datos son utilizados anónimamente para generar estadísticas durante el desarrollo de la partida.
Procedencia:	El usuario ingresa a la aplicación por primera vez e ingresa su nombre de jugador en el sistema.
Salida:	Registro de usuario exitoso: El usuario accede a a la partida. Registro de usuario fallido: el sistema detecta un nombre de usuario ya registrado en el sistema, solicita ingresar los datos nuevamente.
Destino:	Aplicación del lado del cliente del sistema.
Acción:	1.- El usuario ingresa a la aplicación, ingresa el nombre de jugador. 2.- Se genera un identificador único de usuario, el sistema captura las acciones registradas por el usuario en el servidor. 3- El sistema reacciona al comportamiento del usuario dentro de la partida.
Postcondición:	El sistema muestra la interfaz de usuario.
Efectos colaterales:	El usuario ya ha entrado con el mismo nombre de jugador.
Dependencia:	SRS_ER-F:01

### 5.3 Dependencias

Matriz de dependencia:

	SRS_ER-F:01	SRS_ER-F:02	SRS_ER-F:03	SRS_ER-F:04
ER-F:01				D
ER-F:02	D			D
ER-F:03	D	D		D
ER-F:04	D			

#### 5.4 Prioridades de desarrollo

Cada prioridad está señalada en los Sprints elaborado como plan de trabajo por el administrador de proyecto en base al análisis de requerimientos y analizando la evaluación de riesgos.

La prioridad (P) se asigna de acuerdo con el nivel de dependencia y estabilidad del sistema para que este sea funcional y cumpla con los requerimientos críticos.

##### P1: Alta - Alta

SRS_ER-F:03:	El sistema establece una conexión a un servidor.
Prioridad:	Alta.
Estabilidad:	Alta.
Dependencia:	SRS_ER-F:01, SRS_ER-F:02, SRS_ER-F:04

##### P2: Alta - Alta

SRS_ER-F:02	El sistema configura las funciones multijugador utilizando un servidor para consumir el servicio.
Prioridad:	Alta.
Estabilidad:	Alta.
Dependencia:	SRS_ER-F:01, SRS_ER-F:04

##### P3: Alta - Alta

SRS_ER-F:04	El usuario debe ejecutar la aplicación en el sistema para validar la sesión y desplegar la información del nivel de la partida dentro del sistema.
Prioridad:	Alta.
Estabilidad:	Alta.
Dependencia:	SRS_ER-F:01

##### P4: Media - Alta

SRS_ER-F:01	Soporte de llamadas al servidor y objetos de la aplicación.
Prioridad:	Alta.
Estabilidad:	Alta.
Dependencia:	SRS_ER-F:04

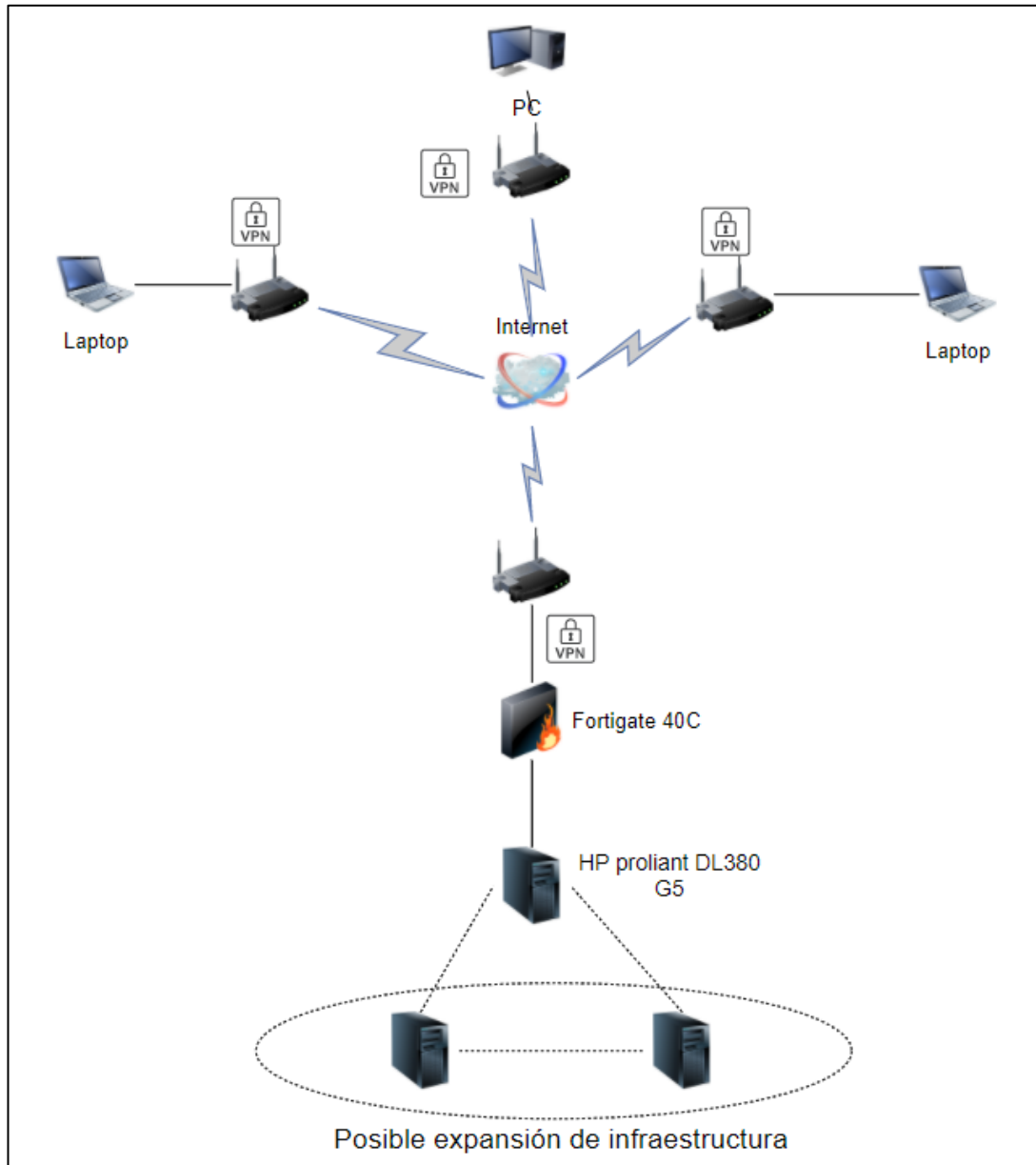


## **6. Requerimientos no funcionales**

- El sistema consulta una base de datos de registros que responde con la información del jugador y la partida.
- El sistema requiere un dispositivo con acceso a internet.
- El sistema consume el servicio mediante un web service para proporcionar información.
- El sistema consulta los registros actualizados de cada partida proporcionando la configuración predefinida para establecer los dispositivos y consulta los controladores correspondientes mediante la aplicación.
- El sistema deberá soportar gran cantidad de usuarios.
- El sistema deberá ser intuitivo mediante una interfaz de jugador accesible y configurable.
- El sistema deberá proporcionar una partida multijugador que soporte a tres usuarios como mínimo interconectados en un entorno compartido (partida).

### 6.1 Modelos del sistema propuesto

Los requerimientos funcionales validados proporcionan un concepto de alto nivel sobre el sistema.



## 6.2 Arquitectura del sistema propuesto

El sistema desarrollado utilizará la arquitectura de software Cliente Servidor con variantes a un enfoque orientado a eventos como patrón de diseño para desplegar la información en una aplicación distribuida multijugador. Características de la arquitectura del sistema:

Funcionalidad	Adecuación:	Mediante el tipo de dispositivo configurar los parámetros para mostrar el despliegue dentro de la interfaz de usuario.
	Exactitud:	El sistema realiza cálculos polinomiales y operaciones sobre matrices en un sistema de coordenadas para desplazarse en el entorno de simulación, los agentes realizan operaciones que debe responder en un dispositivo interconectado con un servidor que trabaja resolviendo peticiones en tiempo real.
	Interoperabilidad:	Utiliza la funcionalidad middleware del dispositivo para representar la localización de un vector y su desplazamiento en tiempo real. Así como consulta a servicios utilizando el patrón Cliente Servidor con una variante orientada a eventos.
	Seguridad:	Los datos son manejados como evento que trabajan en el controlador de acuerdo con las rutas proporcionadas por el web service y delimitado por la base de datos no relacional proporcionados por el servidor.
Fiabilidad	Tolerancia a fallos:	En caso de desconexión el sistema persiste inmediatamente para reconectarse a la partida en un tiempo límite. En caso de no encontrar jugadores conectados o esta sea de difícil acceso a una conexión estable con un jugador el sistema ejecuta excepciones.
	Recuperabilidad:	En caso de falla o error al no encontrar una partida estable, el sistema muestra un mensaje de error, guarda el error internamente, se refresca la interfaz y regresa al jugador al lobby. La partida se puede reanudar si existe algún jugador activo dentro de un nivel de juego.
Eficiencia	Desempeño:	El sistema será una aplicación con comunicación a web services y servicios del servidor ejecutando peticiones RPC óptimamente y en tiempo real mediante un middleware que resuelve los eventos en el dispositivo. Además, se consulta páginas de componentes requeridos de manera local.
	Utilización de recursos:	Consume objetos RMI mediante la aplicación para generar la partida proporcionando una conexión permanente, este puede ser

		guardado en el dispositivo de manera local y consultando información en el servidor. La base de datos utiliza una colección de datos de acceso rápido almacenado dentro del dispositivo.
Mantenimiento	Componentes:	<ul style="list-style-type: none"> <li>- Depende de web services.</li> <li>- Depende de servidores distribuidos.</li> <li>- Depende de componentes RMI.</li> <li>- Depende de un middleware que resuelva llamadas RCP a un servidor.</li> </ul>
Portabilidad	Adaptabilidad:	El sistema es responsivo adaptable a dispositivos tipos de resoluciones, tarjetas gráficas y componentes de desarrollo soportados como plug-ins y librerías C#.

### 6.3 Requerimientos de dominio

Se utilizará la nomenclatura de desarrollo Unity como referencia de dominio en el cual se realiza la estructura de este proyecto, para más información consultar la documentación en línea de Unity versión 4.17.b, esta licencia incluye la posibilidad de compartir un Unity Hub con el repositorio del proyecto que puede ser consultado para fines científicos y de Creative Common con permisos del autor/ diseñador de juego.

Para la conexión con el servidor se utiliza node.js para la gestión de los datos enviados mediante web services y con soporte middleware, para más información consultar el documento de diseño del sistema.

## **7. Evolución del sistema**

El sistema se ha implementado de acuerdo con el cronograma de actividades especificado en el plan maestro de gestión de proyecto, y en relación directa con el modelo general del proceso de diseño del proyecto de forma que se han realizado las actividades acordadas con los stakeholder implementando una aplicación con base en el análisis y la viabilidad de los requerimientos evaluados en este documento.

## **8. Apéndices**