

MUSICNET

ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE

26/05/2025

V.2



Valentina Rozo Gonzalez

Paula Valentina López Cubillos

Historial de cambios

Versión	Fecha	Sección del documento modificada	Descripción de cambios	Responsable(S)
1.0	2024-11-10	Sección 1: Introducción, Alcance, Propósito, Definiciones, Apreciación Global	Redacción inicial del documento, incluyendo el propósito, alcance, definiciones, y la estructura general del SRS.	Valentina López
2.0	2025-13-05	Sección 2, 3 4	Se ajustaron todas las secciones	Valentina Rozo Valentina López

Tabla 1: Historial de cambios

Contenido

Historial de cambios	2
1. Introducción.....	5
1.1. Propósito.....	5
1.2. Alcance	5
1.3. Definiciones, Acrónimos, y Abreviaciones.....	6
1.4. Referencias	7
1.5. Apreciación Global.....	7
2. Descripción global	7
2.1 Perspectiva del Producto	7
2.1.1 Interfaces con el sistema	8
2.1.2 Interfaces con el usuario.....	8
2.1.3 Interfaces con el Hardware	9
2.1.4 Interfaces con el Software	9
2.1.5 Interfaces de Comunicación	10
2.1.6 Restricciones de Memoria	10
2.1.7 Operaciones	10
2.1.8 Requerimientos de Adaptación del Sitio	11
2.2 Funciones del Producto	11
2.3 Características del Usuario.....	12
2.4 Restricciones	13
2.5 Modelo del Dominio	14
2.6 Suposiciones y Dependencias	16
2.7 Distribución de Requerimientos	17
3. Requerimientos específicos	18
3.1 Requerimientos de Interfaces Externas	19
3.1.1 Interfaces con el Usuario	19
3.1.2 Interfaces con el Hardware	20
3.1.3 Interfaces con el Software	20
3.1.4 Interfaces de Comunicaciones	21
3.2 Características del Producto de Software	21
3.2.1 Funcionalidad 1: Jugar	22
3.2.2 Cruce de funcionalidades	22

3.3	Requerimientos de Desempeño	22
3.4	Restricciones De Diseño.....	23
3.5	Atributos del Sistema de Software (No funcionales)	25
3.5.1	Confiabilidad	25
3.5.2	Disponibilidad	25
3.5.3	Mantenibilidad	26
3.5.4	Portabilidad	26
4.	Proceso de ingeniería de requerimientos	27
5.	Proceso de verificación	28

Lista de Tablas

Tabla 1:	Historial de cambios.....	2
Tabla 2:	Definiciones	6
Tabla 3	Métricas	13
Tabla 4:	Modelo del Dominio 1	14
Tabla 5:	Modelo del Dominio 2	15
Tabla 6:	Modelo del Dominio 3	15
Tabla 7:	Modelo del Dominio 4	15
Tabla 8:	Modelo del Dominio 5	15
Tabla 9:	Modelo del Dominio 6	15
Tabla 10:	Modelo del Dominio 7	16
Tabla 11:	Distribución de requerimiento.....	18
Tabla 12:	Requerimientos específicos.....	19
Tabla 13:	Interfaces con el software.....	21
Tabla 14:	Características del producto de software	22
Tabla 15:	Funcionalidad 1	22
Tabla 17:	Cruce funcionalidades.....	22

1. Introducción

1.1. Propósito

El propósito de este documento es especificar los requerimientos del sistema **MusicNet**, una aplicación web multijugador a la práctica del piano. Este documento está dirigido a desarrolladores, supervisores del proyecto, evaluadores académicos y usuarios interesados en la validación técnica del sistema, y proporciona una descripción detallada de las funcionalidades y características que debe tener la aplicación.

Product Name: MusicNet

Product Description: Un juego multijugador en línea para practicar piano. La aplicación permite que los jugadores toquen notas musicales en una escala asignada y, a través de un sistema de plataformas, los avatares representados por notas musicales saltan de una plataforma a otra.

Audiencia:

- **Desarrolladores:** Encargados de implementar las funcionalidades del sistema de acuerdo con los requerimientos.
- **Directores de Tesis:** Aseguran que el sistema cumpla con los objetivos pedagógicos y técnicos del proyecto.
- **Usuarios Finales:** Estudiantes y profesores que utilizarán la aplicación como herramienta de enseñanza musical.

Este documento es clave para alinear las expectativas de todos los involucrados y asegurar que **MusicNet** cumpla con los objetivos propuestos y funcione de manera eficiente y efectiva.

1.2. Alcance

El alcance de este documento cubre el desarrollo de la aplicación web **MusicNet**, detallando sus requerimientos funcionales y no funcionales. **MusicNet** está diseñado para permitir que dos jugadores interactúen en tiempo real, tocando notas musicales en función de una escala asignada, con el objetivo de que sus avatares representados por notas musicales se desplacen a través de plataformas.

Las funcionalidades principales incluyen:

- Modos de juego estático y progresivo.
- Detección de notas en tiempo real utilizando el módulo PitchDetector con AubioJS.
- Interacción multijugador.
- Implementación de técnicas para compensar la pérdida de paquetes y reducir la latencia en redes variables.

Este sistema es fundamental para mejorar la enseñanza del piano en línea, haciendo el aprendizaje más dinámico y accesible para estudiantes de diversas edades. Además,

contribuye al avance de la tecnología en la enseñanza musical, integrando innovaciones en la detección de sonidos y sincronización en línea.

1.3. Definiciones, Acrónimos, y Abreviaciones

A continuación, se presentan las definiciones clave y acrónimos utilizados en este documento:

Término	Definición
PARCnet-IS²	Sistema predictivo de ocultamiento de pérdida de paquetes
EDA	Event-Driven Architecture
DSP	Digital Signal Processing
Jitter	Variación del retardo en la red
SRS	Software Requirement Specification: Documento que define los requisitos del software.
UI	User Interface: Interfaz gráfica con la que interactúan los usuarios.
Latency	Retraso en la transmisión de datos entre el cliente y el servidor, o entre dispositivos en la red.
PitchDetector	Módulo utilizado para la detección de frecuencias musicales y conversión a notas musicales.
AubioJS	Biblioteca JavaScript para el análisis de audio, utilizada en la detección de notas musicales en tiempo real.
Latency Compensation	Técnicas utilizadas para mitigar el impacto de la latencia en la experiencia del usuario.
MusicNet	Nombre del proyecto de aplicación web multijugador para la enseñanza del piano.
Multiplayer	Modo de juego que permite la participación simultánea de dos o más jugadores.
Octava	Rango de notas musicales en una escala, agrupadas en intervalos de frecuencia.
Note Mapping	Proceso de asignar notas musicales detectadas a elementos visuales en el juego.
Packet Loss	Pérdida de paquetes de datos durante la transmisión a través de la red, que puede afectar la sincronización del juego.
Music Packet Loss Concealment (MPLC)	Técnicas para compensar la pérdida de datos de audio en tiempo real, siguiendo el estándar del IEEE-IS ² 2024.
Avatar	Representación gráfica del jugador en el entorno del juego.
Synchronization Module	Componente encargado de alinear el tiempo de las acciones de los jugadores para asegurar una experiencia sincronizada.
WebSocket	Protocolo de comunicación que permite la transmisión bidireccional de datos en tiempo real entre cliente y servidor.
WebRTC	Protocolo que permite la transmisión de audio en tiempo real entre navegadores y dispositivos sin necesidad de plugins.
Real-Time Feedback	Retroalimentación proporcionada al usuario inmediatamente después de realizar una acción, como tocar una nota en el piano.

Tabla 2: Definiciones

1.4. Referencias

1.5. Apreciación Global

Este documento está estructurado en varias secciones que detallan los requerimientos funcionales y no funcionales de **MusicNet**, así como la metodología de desarrollo y validación. A continuación, se presenta una guía para la estructura del documento:

- **Sección 1: Introducción:** Expone el propósito del documento, el alcance de la aplicación y las definiciones clave.
- **Sección 2: Descripción General:** Presenta una descripción del sistema, incluyendo su arquitectura y funcionalidades principales.
- **Sección 3: Requerimientos Funcionales:** Detalla las funcionalidades que debe ofrecer la aplicación.
- **Sección 4: Requerimientos No Funcionales:** Define los requerimientos de rendimiento y otros aspectos no funcionales.
- **Sección 5: Proceso de Verificación:** Describe cómo se llevará a cabo la validación y verificación de los requerimientos.

2. Descripción global

Esta sección tiene como propósito brindar una visión general del producto, proporcionando información contextual y de fondo que facilite la comprensión del sistema MusicNet y su entorno, sin detallar los requerimientos formales. La descripción presentada aquí se desarrollará en mayor detalle en la siguiente sección de requerimientos específicos.

2.1 Perspectiva del Producto

MusicNet es una aplicación web multijugador-orientada a la enseñanza del piano, concebida como una evolución del prototipo académico JumpApp. Aunque JumpApp demostró ser eficaz en aumentar el compromiso de los estudiantes, su rendimiento se ve comprometido en entornos con latencia elevada o pérdida de paquetes, dificultando la sincronización musical en tiempo real. MusicNet nace como una solución integral a estas limitaciones, ofreciendo una experiencia más robusta, precisa y adaptable a condiciones de red variables.

El núcleo de MusicNet es su sistema de sincronización musical entre jugadores, diseñado para preservar la coherencia temporal incluso en escenarios de red desfavorables. Esto es especialmente relevante en la enseñanza del piano, donde la precisión rítmica es fundamental para el desarrollo auditivo y técnico de los estudiantes.

La plataforma incorpora tecnologías avanzadas como:

- Detección de notas en tiempo real mediante algoritmos de procesamiento de audio.
- Sincronización multijugador precisa basada en WebRTC y mecanismos de compensación.
- Mitigación de pérdida de datos usando modelos predictivos como PARCNet.

- Interfaz interactiva gamificada desarrollada con Phaser para mejorar la motivación del estudiante.

MusicNet se ha construido desde cero como una plataforma escalable y modular, con cada uno de sus componentes —detección de tono, lógica multijugador, compensación de red, visualización y sincronización— desacoplado para facilitar su evolución e integración futura.

Gracias a este enfoque, MusicNet no solo mejora la experiencia individual de aprendizaje, sino que habilita prácticas grupales coordinadas en tiempo real, transformando el aprendizaje en línea del piano en una experiencia colaborativa, fluida y estimulante.

2.1.1 Interfaces con el sistema

MusicNet interactúa con múltiples componentes internos y externos que permiten ofrecer una experiencia de usuario fluida, colaborativa y musicalmente sincronizada. Estas interfaces definen la forma en que el sistema se comunica con servicios, módulos y tecnologías externas:

- **Servidor de Señalización (WebRTC):** MusicNet utiliza conexión peer-to-peer entre jugadores se establece mediante el intercambio de ofertas, respuestas e ICE candidates, utilizando Socket.IO sobre WebSockets. La comunicación en tiempo real basada en eventos se realiza a través de canales como offer, answer, candidate y disconnect. El servidor está alojado en Render.com.
- **API de Compensación de Pérdida de Paquetes:** El propósito es mitigar la pérdida de datos en transmisiones de audio bajo redes inestables usando un modelo de predicción basado en PARCNet. Se implementa como una REST API desplegada en Render.com, que recibe solicitudes HTTP con audio y metadatos de red, y responde en formato JSON con el audio reconstruido o instrucciones de corrección.
- **Web del Cliente:** Ofrece una interfaz gráfica para que los usuarios accedan a funciones como crear/salir de salas, interactuar con el piano y ver participantes. Está desarrollada con HTML, CSS y JavaScript, utilizando Phaser.js y Tone.js, y se despliega como un sitio estático en Surge.sh, accesible desde cualquier navegador moderno sin instalación.
- **Módulo PitchDetector (AubioJS):** MusicNet utiliza AubioJS para la detección de frecuencias en tiempo real. Esta integración permite convertir los sonidos capturados desde el dispositivo del usuario en notas musicales.
- **Servicios de Red (TCP/UDP):** El sistema utiliza protocolos de comunicación para la sincronización de datos y la transmisión en tiempo real entre los jugadores. TCP se emplea para la transmisión segura de datos críticos y UDP para minimizar la latencia en el envío de eventos de audio.

2.1.2 Interfaces con el usuario

MusicNet ofrece una interfaz gráfica de usuario (GUI) diseñada para ser intuitiva y centrada en la experiencia del jugador. Las interfaces incluyen:

- **Pantalla de Inicio:** Permite a los usuarios elegir la modalidad de juego entre un solo jugador o multijugador
- **Configuración:** Permite a los usuarios ajustar parámetros del juego, como la selección de la escala musical, el nivel de dificultad y las preferencias de audio antes de iniciar la partida.
- **Crear/ Unirse a Salas:** En la modalidad multijugador se crearán salas donde debe unirse los jugadores para sincronizar las partidas.
- **Pantalla de Juego:** Está muestra el progreso del oponente, avatares y el marcador en tiempo real. Los indicadores visuales de ritmo ayudan a los usuarios a tocar las notas con precisión.
- **Panel de Resultados:** Al final de cada partida, se muestra un resumen del rendimiento del jugador, incluyendo estadísticas como la precisión de las notas tocadas, el número de errores y la racha de aciertos.

2.1.3 Interfaces con el Hardware

MusicNet requiere el uso de hardware específico para ofrecer una experiencia de juego óptima. Las interfaces de hardware incluyen:

- **Dispositivos del Usuario:** La aplicación debe ejecutarse en computadoras personales con al menos 8 GB de RAM, un procesador de 2.4 GHz y soporte para gráficos en tiempo real. Los jugadores pueden utilizar teclados MIDI como entrada para la detección de notas.
- **Micrófono:** Para capturar el sonido del piano o teclado del usuario, se requiere un micrófono de calidad media-alta. La detección precisa de notas depende de la calidad del audio capturado.
- **Tarjeta de red (LAN/WLAN):** El dispositivo debe contar con soporte para el protocolo **TCP/IP**, permitiendo la conectividad necesaria para el funcionamiento multijugador y la comunicación en tiempo real.
- **Pantalla:** Se requiere una resolución mínima de **1024x768 píxeles** para asegurar una visualización adecuada de la interfaz gráfica de usuario y los elementos musicales interactivos.
- **Tarjeta de sonido:** Es indispensable para la reproducción auditiva precisa de notas, sonidos y pistas musicales. Debe ser compatible con los estándares de audio del sistema operativo.

2.1.4 Interfaces con el Software

MusicNet depende de diversas interfaces de software para integrar sus componentes y ofrecer una experiencia de usuario fluida y en tiempo real. Estas interfaces permiten la ejecución del juego en el navegador, la comunicación entre usuarios, y el procesamiento y reproducción de audio musical. Estas incluyen:

- **Navegador Web:** MusicNet se ejecuta en navegadores compatibles con HTML5, JavaScript y Web Audio API, como Google Chrome, Mozilla Firefox y Safari.

- **Librerías de Audio:** MusicNet utiliza **AubioJS** para la detección de frecuencias y **AudioSynth** para la generación de retroalimentación sonora durante el juego.
- **Motor de Juego:** MusicNet utiliza **Phaser 3** como framework gráfico principal para la renderización de elementos interactivos y visuales del juego.
- **Comunicación en Tiempo Real:** MusicNet utiliza **WebRTC** y **WebSocket** para establecer canales de comunicación directa entre jugadores y con el servidor, habilitando funcionalidades multijugador y sincronización de audio.

2.1.5 Interfaces de Comunicación

La comunicación en MusicNet es esencial para garantizar una experiencia multijugador fluida y sincronizada, especialmente durante la reproducción musical en tiempo real. El sistema hace uso de varios protocolos de red y configuraciones de conexión que permiten minimizar la latencia y mantener la estabilidad del juego. El sistema utiliza:

- **Protocolo UDP:** Utilizado para la transmisión de eventos de audio en tiempo real. Este protocolo reduce la latencia y mejora la sincronización entre jugadores durante la ejecución musical.
- **Protocolo TCP:** Empleado para la transferencia de datos críticos como el estado del juego, puntaje y configuración inicial.
- **WebSocket / WebRTC:** Implementados sobre TCP y UDP, respectivamente, para habilitar canales persistentes de comunicación en tiempo real entre los clientes y el servidor.
- **Tipo de Red:** Se recomienda una red de tipo **LAN o WLAN** con un ancho de banda mínimo de **10 Mbps** para asegurar una conexión estable y con baja latencia.
- **Puertos de Comunicación:** Se manejan puertos dinámicos con el navegador, las conexiones WebRTC/WebSocket son gestionadas automáticamente por el navegador, por lo que no requieren configuración manual por parte del usuario.

2.1.6 Restricciones de Memoria

MusicNet requiere ciertos recursos de hardware a nivel de memoria y procesamiento para garantizar una ejecución estable y en tiempo real, tanto en el lado del cliente como del servidor.

- **Memoria RAM:** Se requiere un mínimo de **4 GB de RAM** para permitir la ejecución simultánea del navegador web, los módulos de detección de audio, procesamiento gráfico y comunicación en red. Se recomienda cerrar otras aplicaciones para maximizar el rendimiento.
- **Procesador (CPU):** El sistema debe ejecutarse en un procesador **dual-core o superior**, con una frecuencia mínima recomendada de **2.0 GHz**, para asegurar una respuesta fluida en tareas de análisis de audio y renderizada visual.

2.1.7 Operaciones

MusicNet incluye varias operaciones críticas para garantizar la experiencia de juego:

- **Iniciar Partida:** Los usuarios configuran los parámetros del juego y esperan la conexión de otro jugador para comenzar una partida multijugador.
- **Detección y Procesamiento de Notas:** El sistema detecta las notas tocadas por el usuario en tiempo real, las convierte en datos de juego y las envía al servidor para sincronización.
- **Sincronización en Tiempo Real:** MusicNet garantiza que los eventos de juego, como los movimientos de los avatares, sean consistentes y sincronizados entre ambos jugadores, independientemente de las variaciones en la calidad de la red.

2.1.8 Requerimientos de Adaptación del Sitio

MusicNet está diseñado para adaptarse a diferentes entornos y configuraciones de red:

- **Resolución de Pantalla:** La interfaz de usuario se adapta automáticamente a diferentes resoluciones de pantalla, con un mínimo de 1280x720 píxeles para una experiencia visual óptima.
- **Entornos de Red:** La aplicación está optimizada para funcionar tanto en redes LAN (Red de Área Local) como en redes WAN (Red de Área Amplia), ajustando dinámicamente el envío de datos según la calidad de la conexión.
- **Compatibilidad Multiplataforma:** MusicNet es compatible con sistemas operativos Windows, macOS y Linux, siempre que se utilice un navegador moderno compatible.

2.2 Funciones del Producto

MusicNet es una aplicación multijugador-orientada a la práctica del piano, la cual incorpora un conjunto de funciones que permiten a los usuarios aprender y practicar notas musicales en tiempo real, manteniendo la precisión incluso en entornos de red con latencia o pérdida de paquetes. Las funciones principales del producto se pueden describir de la siguiente manera:

1. **Captura de Notas en Tiempo Real:** MusicNet utiliza el módulo PitchDetector basado en AubioJS para capturar las notas que los jugadores tocan en sus dispositivos (pianos). Esta función permite detectar la frecuencia del sonido en tiempo real, convertirla en una nota musical, y hacer que el avatar del jugador avance en el juego si la nota es correcta.
2. **Sincronización de Jugadores en Tiempo Real:** El producto asegura que ambos jugadores, en un entorno multijugador, vean y respondan a la misma secuencia de notas a tiempo, independientemente de las fluctuaciones de la red. Para lograr esto, se aplican técnicas de compensación de pérdida de paquetes derivadas del IEEE-IS² 2024 Music Packet Loss Concealment Challenge, lo que permite mantener la fluidez y sincronización de la competencia.
3. **Modos de Juego:** MusicNet incluye dos modalidades principales de juego:
 - **Estática:** En este modo, los jugadores pueden entrenar tocando en una escala específica basada en los modos de la escala mayor. El jugador debe tocar las notas correctas para evitar que su avatar caiga de las plataformas.

- **Progresiva:** En este modo, la dificultad aumenta a medida que se cambia la escala modal en cada nivel, ofreciendo un desafío adicional conforme se avanza en el juego.
4. **Interfaz Multijugador con Marcador:** La interfaz del juego está diseñada para ofrecer una experiencia visualmente clara y dinámica. Cada jugador puede ver su propio desempeño, mientras un marcador en la pantalla muestra el progreso de ambos competidores en tiempo real. Este marcador resalta las notas acertadas por cada jugador y fomenta una competencia sana, permitiendo a los usuarios concentrarse en mejorar su precisión musical sin distracciones adicionales.
 5. **Indicadores Rítmicos y Visuales** Para ayudar a los jugadores a seguir el ritmo, la interfaz incluye indicadores visuales que marcan los pulsos del compás en cuatro cuartos. Estos indicadores proporcionan una guía para tocar las notas a tiempo, mejorando la precisión rítmica de los jugadores.
 6. **Comparación de Puntuaciones:** MusicNet permite comparar el desempeño de ambos jugadores en tiempo real. Las puntuaciones se muestran en un panel dedicado, que incluye la cantidad de notas correctas, los errores y la racha de aciertos de cada jugador, ofreciendo una retroalimentación inmediata.
 7. **Configuración del Juego:** Los jugadores tienen acceso a un panel de control desde el cual pueden ajustar diferentes parámetros antes de empezar una partida. Estos ajustes incluyen la elección de la escala musical, el rango de octavas, y el nivel de dificultad, personalizando la experiencia de acuerdo con el nivel de habilidad de cada jugador.
 8. **Compensación de Latencia:** Para asegurar una experiencia de juego fluida, MusicNet incorpora técnicas avanzadas para la compensación de latencia. El sistema monitorea la calidad de la red y ajusta dinámicamente el comportamiento del juego para mitigar los efectos de la latencia, asegurando que el progreso de ambos jugadores esté alineado incluso en condiciones de red desfavorables.
 9. **Efectos Visuales y Auditivos:** El juego utiliza efectos visuales dinámicos, como el cambio de colores de fondo o de las plataformas al avanzar de nivel, para mejorar la experiencia inmersiva del usuario. Además, se utilizan sonidos generados por la biblioteca AudioSynth para proporcionar retroalimentación auditiva cuando se tocan notas correctamente.

2.3 Características del Usuario

Jugador

- **Privilegios:** Acceso a todas las funciones de juego, incluyendo la posibilidad de competir en tiempo real con otros jugadores. No tiene permisos para modificar configuraciones del sistema ni gestionar usuarios.
- **Rol:** Competidor principal en MusicNet. El objetivo es tocar las notas correctas en su piano mientras compite con otros jugadores en tiempo real.

- **Nivel de Estudios o Experiencia Técnica:** No se requiere experiencia técnica previa, pero es recomendable que los jugadores tengan nociones básicas del manejo del piano.
- **Frecuencia de Uso:** Los jugadores pueden ser usuarios frecuentes, que entrenan regularmente, u ocasionales, que participan en competiciones esporádicas.

2.4 Restricciones

Restricciones Generales

Las reglas que rigen la aplicación MusicNet están diseñadas para garantizar su buen funcionamiento y facilidad de uso para los usuarios. Algunas de las restricciones generales son:

- **Tolerancia a Fallos:** El sistema debe ser capaz de manejar fallos temporales, como la pérdida de conexión de los jugadores durante una partida, sin que se interrumpa el progreso del juego. Para esto, se utilizarán PARCnet una técnica de compensación de pérdida de paquetes.
- **Reglas de la Aplicación:** Las reglas del juego deben estar claramente definidas para evitar conflictos entre jugadores y para garantizar una experiencia de usuario justa. Estas reglas estarán integradas en el código.
- **Sin Persistencia del Progreso:** MusicNet no almacena el estado del juego o el progreso del jugador entre sesiones. Toda la lógica del sistema opera en sesiones temporales.

Restricciones de Conectividad y Red

- **Conexión a Internet Requerida:** MusicNet necesita una conexión a Internet estable. No es funcional en modo offline.
- **Dependencia del Micrófono del Navegador:** La aplicación requiere acceso a la API de micrófono del navegador. La falta de permisos bloquea funcionalidades clave como la detección de notas musicales.
- **Requisitos de Red en Tiempo Real:** Para asegurar la experiencia multijugador y el análisis de audio sincronizado, deben cumplirse las siguientes métricas de red:

Métrica	Valor Máximo Permitido
Latencia	< 100 ms
Pérdida de paquetes tolerada	Hasta 10%
Jitter	< 30 ms
Diferencia de sincronización entre jugadores	≤ 50 ms

Tabla 3 Métricas

Restricciones de Software

El software que soporta MusicNet tiene varias restricciones impuestas por las tecnologías utilizadas y los requisitos legales:

- **Licencias de Software:** El sistema utilizará tecnologías únicamente herramientas open-source.

- **Procesamiento de Audio:** La detección y reconstrucción de notas musicales perdidas se realiza con AubioJS en el cliente y con PARCNet en Python como componente de soporte en servidores externos o de desarrollo.
- **Entorno de Ejecución:** El sistema depende de la ejecución en navegadores modernos compatibles con HTML5, JavaScript, Web Audio API y WebRTC/WebSocket.
- **Lenguaje de Programación:** Todo el desarrollo de MusicNet, tanto cliente como servidor de señalización, se realizó en JavaScript. La lógica del juego está implementada con Phaser3 y se ejecuta en un único archivo HTML.

Restricciones de Hardware

La ejecución de MusicNet también estará sujeta a los requerimientos de hardware, tanto para el servidor que aloje la aplicación como para los dispositivos de los usuarios finales:

- **Requisitos del Cliente:** La aplicación debe ejecutarse en dispositivos con capacidad de procesamiento para ejecutar gráficos y procesamiento de audio en tiempo real. Se recomienda un mínimo de 8 GB de RAM, un procesador de 2.4 GHz y un adaptador de red con velocidades de transferencia de al menos 100 Mbps.
- **Resolución de Pantalla:** Los usuarios deben contar con una resolución mínima de pantalla de 1280x720 para garantizar una correcta visualización de la interfaz del juego y las notas musicales.
- **Velocidad del Adaptador de Red:** Para minimizar la latencia y garantizar una buena experiencia multijugador, se recomienda que los jugadores utilicen conexiones a internet con una velocidad mínima de 10 Mbps.
- **Micrófono:** Para una detección precisa de las notas musicales se sugiere un micrófono de calidad media-alta

Restricciones Adicionales

Las siguientes restricciones también deben ser tomadas en cuenta, pues limitan las opciones de diseño del sistema:

- **Limitaciones de Hardware:** Dado que MusicNet procesará audio en tiempo real, los dispositivos utilizados por los jugadores deben tener capacidad de procesamiento de señales de audio sin retrasos perceptibles.
- **Requisitos de Confiabilidad:** El sistema debe garantizar una alta disponibilidad y confiabilidad, lo que significa que debe estar preparado para funcionar de manera continua sin interrupciones inesperadas.

2.5 Modelo del Dominio

ID	Elemento del Dominio	Jugador
Descripción	Usuario que participa en una partida de MusicNet.	
Objetivo	Representar a los usuarios que participan en una partida del juego.	

Tabla 4: Modelo del Dominio 1

ID	Elemento del Dominio	Partida
----	----------------------	---------

Descripción	Sesión de juego donde dos jugadores participan simultáneamente.	
Atributos	Nombre	Descripción
	Jugador1	Identificación del primer jugador
	Jugador2	Identificación del segundo jugador
	Estado	Estado actual de la partida
Objetivo	Controlar y registrar cada sesión de juego.	

Tabla 5: Modelo del Dominio 2

ID	Elemento del Dominio	Nota Musical
Descripción	Nota musical detectada por el sistema.	
Atributos	Nombre	Descripción
	NombreNota	Nombre de la nota (e.g., C, D, E)
	Frecuencia	Frecuencia de la nota en Hz
	TiempoDetectado	Tiempo de detección de la nota
Objetivo	Capturar las notas musicales tocadas.	

Tabla 6: Modelo del Dominio 3

ID	Elemento del Dominio	Avatar
Descripción	Representación visual del jugador.	
Atributos	Nombre	Descripción
	Posición	Posición actual del avatar en el juego
	Color	Color del avatar
	Estado	Estado del avatar (activo, inactivo)
Objetivo	Visualizar el progreso del jugador.	

Tabla 7: Modelo del Dominio 4

ID	Elemento del Dominio	Sincronización
Descripción	Controla el ajuste de tiempos entre los jugadores para sincronizar la partida.	
Atributos	Nombre	Descripción
	Latencia	Tiempo de retraso detectado
	AjusteTiempo	Indica si se aplicó ajuste para sincronización
Objetivo	Asegurar la sincronización y coherencia temporal entre los jugadores.	

Tabla 8: Modelo del Dominio 5

ID	Elemento del Dominio	Estadística
Descripción	Almacena los datos del rendimiento del jugador durante la partida.	
Atributos	Nombre	Descripción
	Puntuación	Puntos obtenidos por el jugador
	NotasCorrectas	Número de notas acertadas
	NotasIncorrectas	Número de notas falladas
Objetivo	Evaluar y proporcionar retroalimentación del rendimiento del jugador.	

Tabla 9: Modelo del Dominio 6

ID	Elemento del Dominio	ConfiguraciónJuego
-----------	-----------------------------	---------------------------

Descripción	Preferencias establecidas para cada partida, como la escala musical y nivel de dificultad.	
Atributos	Nombre	Descripción
	Escala	Tipo de escala musical (e.g., Mayor, Menor)
	NivelDificultad	Dificultad seleccionada para la partida
	OctavasSeleccionadas	Número de octavas a utilizar
Objetivo	Personalizar la experiencia de juego según las preferencias del jugador.	

Tabla 10: Modelo del Dominio 7

2.6 Suposiciones y Dependencias

Suposiciones

A continuación, se enumeran las suposiciones clave que afectan los requerimientos de la aplicación **MusicNet**. Estas suposiciones deben cumplirse para garantizar que el sistema funcione como se espera, y cualquier cambio o incumplimiento podría tener un impacto significativo en el desarrollo del proyecto:

1. **Conexión a Internet:** Los usuarios y el servidor necesitarán acceso a una conexión a internet estable para garantizar que el juego multijugador funcione sin interrupciones. La velocidad mínima recomendada es de 10 Mbps para una experiencia fluida.
2. **Cumplimiento de las especificaciones de hardware de la sección 2.4:** Se asume que los jugadores usarán dispositivos que cumplan con los requisitos mínimos de hardware especificados anteriormente, como la capacidad de procesamiento, resolución de pantalla y velocidad del adaptador de red.
3. **Que el cliente no haga cambios significativos en los requerimientos:** Se espera que los requerimientos acordados inicialmente con el cliente permanezcan estables a lo largo del desarrollo. Cambios sustanciales en las funcionalidades o estructura del sistema podrían afectar los tiempos de entrega y la viabilidad del proyecto.

Dependencias

Las dependencias del sistema son factores externos que influyen directamente en el rendimiento y la estabilidad de **MusicNet**. Estos factores incluyen tanto aspectos técnicos como componentes reutilizados de otros proyectos. Algunas de las principales dependencias son:

1. **Velocidad de la red:** El rendimiento de **MusicNet** depende en gran medida de la velocidad y estabilidad de la conexión a internet de los usuarios. Para asegurar tiempos de respuesta adecuados en el juego multijugador, se requiere una conexión mínima de 10 Mbps para cada jugador. Problemas de latencia o pérdida de paquetes podrían afectar negativamente la experiencia del usuario.

2. **Cambio de alguna de las reglas que rigen la aplicación:** Si se realiza alguna modificación a las reglas del juego establecidas, el sistema deberá ser actualizado para reflejar dichos cambios. Esto puede incluir ajustes en la detección de notas musicales, modificación de los objetivos del juego, o cambios en la lógica de puntuación, lo cual podría implicar trabajo adicional de desarrollo y pruebas.
3. **Factores externos como componentes de software reutilizados:** El sistema reutiliza componentes como AubioJS para la detección de notas musicales. Si alguno de estos componentes externos deja de recibir soporte o se actualiza con cambios que no son compatibles, podría ser necesario realizar ajustes en el código del sistema para mantener su funcionalidad.

2.7 Distribución de Requerimientos

En esta sección, se realiza un mapeo de los requerimientos específicos de MusicNet en relación con las funciones clave de la aplicación, así como con los componentes y módulos definidos inicialmente. Esta distribución facilita la localización de los requerimientos en los diferentes módulos del sistema, lo cual es esencial para tareas de validación, verificación y actualización. La estructura definida en esta sección debe ser coherente con la arquitectura detallada en la sección 3 de este documento.

En el caso de MusicNet, los módulos principales incluyen el Servidor, Jugador, Sincronización de Audio, Interfaz de Usuario (GUI) y Base de Datos. Estos módulos se relacionan con varias funcionalidades, como el juego en tiempo real, la detección de frecuencias y conversión a notas (usando el módulo PitchDetector con AubioJS), la sincronización de eventos entre jugadores y estadísticas. Cada funcionalidad es asignada a uno o varios módulos que la soportan. A continuación, se detallan algunos ejemplos de distribución de requerimientos por módulo:

1. Módulo Servidor:

- Gestiona la lógica de juego en tiempo real y la sincronización de eventos entre jugadores.
- Administra la compensación de pérdida de paquetes mediante técnicas del IEEE-IS² 2024 Music Packet Loss Concealment Challenge.

2. Módulo Jugador:

- Controla la interacción del jugador con el piano y la detección de frecuencias musicales, convirtiéndolas en notas que se reflejan en el juego.
- Envía y recibe datos sobre la posición del avatar en el juego.

3. Módulo de Sincronización de Audio:

- Garantiza que las notas detectadas estén alineadas con los tiempos de juego y se traduzcan en movimientos precisos de los avatares.
- Implementa algoritmos de latencia y ajuste de sincronización en redes con conexiones variables.

4. Interfaz de Usuario (GUI):

- Visualiza en tiempo real el progreso de cada jugador, las plataformas y los avatares.
- Muestra los marcadores y las estadísticas de cada partida.

A continuación, se presenta un cuadro resumen (similar a la Ilustración 14 mostrada en la imagen proporcionada) que ayuda a visualizar la asignación de requerimientos en función de la funcionalidad y el módulo correspondiente. Los módulos y casos de uso de **MusicNet** están distribuidos de la siguiente forma:

Funcionalidad	Módulo	Casos de Uso
Juego Multijugador	Servidor, Jugador, Sincronización de Audio, GUI	Detectar notas, mover avatar, sincronizar eventos
Visualización de Progreso	GUI	Mostrar estadísticas, actualizar marcador
Sincronización y Ajustes	Sincronización de Audio, Servidor	Ajustar latencia, compensar pérdida de paquetes

Tabla 11: Distribución de requerimiento

Requerimientos Planificados para Futuras Versiones

Para futuras versiones de **MusicNet**, se planean los siguientes requerimientos:

1. **Modo de Juego Avanzado:** Añadir modos de dificultad progresiva para que los jugadores puedan enfrentar desafíos de mayor complejidad en la interpretación musical.
2. **Ampliación del Sistema de Avatares:** Permitir que los jugadores personalicen sus avatares con diferentes elementos visuales que representen sus logros y nivel de habilidad.
3. **Sistema de Torneos:** Implementar la funcionalidad de organización de torneos y tablas de clasificación para fomentar la competitividad.
4. **Análisis y Reportes de Rendimiento:** Generar informes detallados sobre el desempeño del jugador, brindándole recomendaciones para mejorar su precisión y rapidez.
5. **Personalización de Juego:** Permitir que el jugador elija una canción que desee practicar

3. Requerimientos específicos

La sección de Requerimientos Específicos es fundamental para el diseño e implementación de MusicNet, ya que permite una comunicación clara entre los desarrolladores y los futuros usuarios de la aplicación. Aquí se especifican los requisitos necesarios para transformar las necesidades del usuario en elementos técnicos y funcionales que orienten la construcción de la aplicación. Cada requerimiento se describe con detalle, garantizando que sea comprensible tanto para desarrolladores como para el cliente.

A continuación, se presentan los requerimientos funcionales y no funcionales, así como las interfaces externas requeridas por el sistema. Cada requerimiento cumple con las características de ser atómico, correcto, no ambiguo, completo, consistente, importante, verificable, modificable, trazable, asociado a versión, no redundante, y preciso.

#	Tipo de Requerimiento	Casos de Uso Asociados	Descripción	Razón	Autor	Criterio de Medición	Prioridad	Módulo Asociado	Versión	Fecha
1	Funcional	CU01	El sistema debe permitir a los jugadores ingresar a una partida y seleccionar un instrumento virtual (piano).	Permitir la selección de instrumentos para el juego.	Equipo de desarrollo	Inspección de la funcionalidad	Alta	Módulo de Selección de Instrumento	1.0	7/11/20
2	Funcional	CU02	El sistema debe detectar y sincronizar las notas tocadas en tiempo real mediante el módulo PitchDetector y AubioJS.	Sincronizar las notas para mejorar la precisión en el juego.	Equipo de desarrollo	Prueba de sincronización en diferentes conexiones de red	Alta	Módulo de Detección de Notas	1.0	7/11/20
3	Funcional	CU03	El sistema debe mostrar las plataformas y el progreso del avatar del jugador en tiempo real.	Visualizar el desempeño en el juego para ambos jugadores.	Equipo de desarrollo	Inspección visual de las plataformas y progreso del avatar	Alta	Módulo de Visualización	1.0	7/11/20
4	No Funcional	N/A	La aplicación debe compensar la pérdida de paquetes utilizando técnicas del IEEE-IS ² 2024 Music Packet Loss Concealment Challenge.	Mejorar la experiencia de juego en redes con pérdida de paquetes.	Equipo de desarrollo	Análisis de rendimiento bajo condiciones de pérdida de paquetes	Alta	Módulo de Compensación de Red	1.0	7/11/20
5	Funcional	CU04	La aplicación debe mostrar un marcador que refleje el puntaje de ambos jugadores en tiempo real.	Facilitar la competencia entre jugadores y visualizar el progreso.	Equipo de desarrollo	Verificación de la actualización en tiempo real del marcador	Media	Módulo de Marcador	1.0	7/11/20

Tabla 12: Requerimientos específicos

3.1 Requerimientos de Interfaces Externas

Los requerimientos de interfaces externas de MusicNet están diseñados para asegurar una interacción fluida entre los usuarios y la aplicación. Las interfaces especificadas permiten una comunicación clara y efectiva con el sistema, a través de la disposición de los elementos en pantalla y la integración con periféricos del dispositivo.

3.1.1 Interfaces con el Usuario

MusicNet proporcionará una interfaz gráfica de usuario (GUI) intuitiva, que facilitará la navegación y el acceso a las funcionalidades del juego. Los requerimientos específicos para las interfaces de usuario son los siguientes:

- **Pantalla:** La aplicación se visualizará en una resolución mínima de 1024x768 píxeles, permitiendo una distribución clara y legible de los elementos. La pantalla estará dividida en dos secciones, cada una mostrando las plataformas de juego de un jugador.
- **Teclado y Mouse:** El sistema permitirá el uso del teclado para acciones básicas de navegación y configuración del juego. No obstante, la interacción principal será mediante la detección de notas en tiempo real, sin necesidad de otros dispositivos de entrada durante el juego.
- **Interfaz GUI:** La interfaz gráfica incluirá botones, menús y visualización de plataformas y avatares en movimiento. También habrá elementos visuales para el marcador y otras estadísticas del juego. Los elementos de la interfaz están organizados para facilitar la visualización y el control del juego.

- **Tarjeta Gráfica y Tarjeta de Red:** MusicNet requerirá una tarjeta gráfica básica compatible con navegadores modernos y una conexión a internet estable. La aplicación está optimizada para entornos de red variables y usa técnicas avanzadas para manejar la latencia y la pérdida de paquetes.

3.1.2 Interfaces con el Hardware

Dado que el sistema MusicNet debe tener la capacidad de compartir datos e información con otros dispositivos, como computadoras y móviles, es fundamental especificar cómo se realizará esta comunicación. Los componentes de software del sistema interactuarán con los elementos de hardware a través de distintas interfaces. Estas interfaces permitirán la conexión entre los dispositivos de los usuarios, asegurando una comunicación eficiente y sin interrupciones en el flujo de datos.

Las principales interfaces de hardware que se deben considerar incluyen:

- **Protocolos de Comunicación (TCP, UDP):** MusicNet utilizará los protocolos TCP (para garantizar una transmisión segura y confiable de datos) y UDP (para reducir la latencia en la transmisión de audio). Estos protocolos son esenciales para el envío y recepción de datos en tiempo real durante las partidas.
- **Puertos de Comunicación:** La aplicación debe tener habilitados puertos específicos que permitan la comunicación entre los diferentes dispositivos conectados en red. En el caso de aplicaciones distribuidas, estos puertos facilitan la transmisión de datos entre los servidores y los clientes.
- **Cables de Conexión (UTP):** En redes locales, se utilizarán cables de conexión UTP (Par trenzado sin apantallar) para conectar routers y switches con los dispositivos de los usuarios, permitiendo una conexión física estable y rápida en el caso de conexiones Ethernet.
- **Dispositivos de Red (Routers, Hubs, Switches):** Estos dispositivos son fundamentales para la infraestructura de red, ya que permiten conectar múltiples dispositivos en una red LAN (Local Area Network) o WAN (Wide Area Network). A través de routers, hubs y switches, se distribuirán los datos y se establecerán las conexiones entre los jugadores.

3.1.3 Interfaces con el Software

Para que MusicNet funcione adecuadamente, es necesario cumplir con ciertos requerimientos mínimos de software. Estos requisitos incluyen versiones específicas del sistema operativo, servidores de bases de datos y otras aplicaciones de soporte. La siguiente tabla describe los componentes de software necesarios para la correcta operación del sistema:

Componente de Software	Descripción	Requisitos Mínimos
Sistema Operativo	El sistema debe operar en entornos Windows, macOS o Linux para asegurar su compatibilidad con la mayoría de los dispositivos.	Windows 10 o superior, macOS 10.13+, Linux Ubuntu 18.04+

Servidor de Aplicación	Permite la ejecución de la lógica de negocio de MusicNet en una plataforma centralizada o distribuida.	Apache Tomcat 9+, o cualquier servidor compatible con Java
Navegador Web	La aplicación es accesible a través de navegadores compatibles con estándares modernos.	
Bibliotecas de Detección de Frecuencia	AubioJS es necesario para la detección en tiempo real de las notas musicales jugadas por los usuarios.	AubioJS versión compatible

Tabla 13: Interfaces con el software

3.1.4 Interfaces de Comunicaciones

Las interfaces de comunicación especifican los métodos y tecnologías que se utilizarán para interconectar las distintas instancias de MusicNet en diferentes dispositivos. Como la aplicación está diseñada para soportar partidas multijugador, estas interfaces deben incluir la infraestructura de red y los protocolos de comunicación necesarios:

- **Tipo de Red:** MusicNet funcionara en redes LAN (Red de Área Local) y WAN (Red de Área Amplia). Si bien las redes LAN proporcionan un entorno ideal para baja latencia en partidas locales, MusicNet estará especialmente diseñado para tolerar condiciones variables y fluctuantes en redes WAN.
- **Protocolos de Comunicación (TCP y UDP):** MusicNet utilizará principalmente el protocolo UDP para la transmisión de audio en tiempo real, debido a su baja latencia, lo cual es crucial para la sincronización de notas en el juego. También se utilizará el protocolo TCP para la transmisión de datos donde se requiere confiabilidad y para tareas no críticas en tiempo real, como el envío de datos de puntaje y estado de la partida.

3.2 Características del Producto de Software

Los requerimientos funcionales definen las acciones fundamentales que debe cumplir el sistema, procesando entradas y generando salidas específicas. A continuación, se presenta una estructura detallada de los requerimientos de MusicNet, organizados por funcionalidades clave del software.

ID	Requerimiento	Caso de Uso Asociado	Prioridad	Estado
RF-1	El sistema debe permitir al usuario iniciar una partida multijugador.	CU-1: Iniciar Partida	Alta	Activo
RF-2	El sistema debe detectar y procesar las notas musicales tocadas por el usuario en tiempo real.	CU-2: Detección de Notas	Alta	Activo
RF-3	El sistema debe mostrar el progreso y la puntuación en tiempo real en la interfaz del juego.	CU-3: Visualizar Progreso	Media	Activo

RF-4	El sistema debe proporcionar una interfaz donde los usuarios puedan consultar sus puntajes y rendimiento en partidas anteriores.	CU-4: Consultar Rendimiento	Media	Planificado
------	--	-----------------------------	-------	-------------

Tabla 14: Características del producto de software

3.2.1 Funcionalidad : Jugar

La funcionalidad de "Jugar" es el núcleo de MusicNet. Permite a los usuarios participar en partidas multijugador donde tocan notas en sus teclados o pianos, y el sistema detecta las notas en tiempo real para sincronizar la experiencia de juego.

ID	Requerimiento	Caso de Uso Asociado	Prioridad	Estado
RF-1	El sistema debe permitir que los usuarios inicien una partida multijugador sin necesidad de registrarse.	CU-1: Iniciar Partida	Alta	Activo
RF-2	El sistema debe detectar y procesar las notas tocadas en tiempo real por los usuarios.	CU-2: Detección de Notas	Alta	Activo
RF-3	El sistema debe representar las notas en la interfaz del juego como avatares que saltan de plataforma en plataforma.	CU-3: Representación de Notas	Media	Activo
RF-4	El sistema debe calcular y mostrar la puntuación basada en la precisión de las notas tocadas.	CU-3: Cálculo de Puntuación	Media	Activo
RF-5	El sistema debe mostrar el progreso de cada jugador en la interfaz de juego compartida.	CU-3: Visualizar Progreso	Media	Activo

Tabla 15: Funcionalidad 1

3.2.2 Cruce de funcionalidades

Estas son funcionalidades transversales necesarias para asegurar una experiencia de juego fluida y segura, especialmente en lo relacionado con la conexión y sincronización de los jugadores en tiempo real.

ID	Requerimiento	Caso de Uso Asociado	Prioridad	Estado
RF-8	El sistema debe asegurar una conexión estable entre los usuarios, aplicando técnicas para manejar la pérdida de paquetes.	CU-6: Conexión Estable	Alta	Activo
RF-9	El sistema debe permitir la sincronización de notas entre jugadores en tiempo real para mantener una experiencia de juego consistente.	CU-7: Sincronización de Notas	Alta	Activo

Tabla 16: Cruce funcionalidades

3.3 Requerimientos de Desempeño

Los requerimientos de desempeño del sistema MusicNet se dividen en dos categorías principales: requerimientos estáticos y requerimientos dinámicos. A continuación, se detallan los aspectos clave para cada uno de ellos.

Requerimientos Estáticos

1. El sistema debe contar con una arquitectura distribuida y escalable basada en tecnologías modernas, como WebRTC y Socket.IO, que permita la adaptación a escenarios con múltiples usuarios concurrentes. Además, se debe prever la realización de pruebas de estrés y rendimiento en futuras etapas para validar la capacidad del sistema de soportar cientos de dispositivos conectados simultáneamente.
2. La aplicación debe ser compatible con dispositivos que cuenten con al menos 8 GB de RAM y un procesador de 2.4 GHz, para garantizar el procesamiento en tiempo real de audio y gráficos sin pérdidas significativas de rendimiento.

Requerimientos Dinámicos

1. **Latencia:** El sistema debe garantizar una latencia máxima de 100 ms entre el momento en que el jugador toca una nota y la respuesta visual y auditiva correspondiente en el juego. Esto es crucial para mantener la precisión y sincronización en tiempo real durante el juego multijugador.
2. **Tasa de procesamiento de transacciones:** El sistema debe ser capaz de manejar un mínimo de 50 transacciones por segundo relacionadas con la sincronización de notas, la actualización de puntuaciones y la interacción de los jugadores, sin experimentar demoras perceptibles.
3. **Capacidad de carga en la red:** La red debe ser capaz de soportar una carga mínima de 100 MB por hora por jugador, asegurando que se pueda mantener la transmisión de datos de audio, video e interacciones entre jugadores de manera fluida.
4. **Disponibilidad:** El sistema debe tener una disponibilidad mínima del 99.9%, lo que garantiza que MusicNet esté operativo en todo momento, con un tiempo de inactividad máximo de 8 horas al año debido a mantenimiento o actualizaciones.

Criterios de Medición

Para verificar el cumplimiento de estos requerimientos de desempeño, se realizarán pruebas de estrés en las que se simularán las condiciones de carga máxima y las transacciones simultáneas de los usuarios. Los tiempos de respuesta se medirán a través de herramientas de monitoreo en tiempo real que registren la latencia y el rendimiento bajo diferentes condiciones de red. Además, se realizarán pruebas de disponibilidad para garantizar que el sistema cumpla con el objetivo de tiempo de inactividad.

3.4 Restricciones De Diseño

El diseño de MusicNet se encuentra sujeto a una serie de restricciones que derivan de las decisiones técnicas y las limitaciones inherentes al proyecto. A continuación, se detallan las principales restricciones de diseño, que afectan tanto a la arquitectura del sistema como a las tecnologías utilizadas.

a. Lenguajes de Programación

La aplicación MusicNet está desarrollada principalmente utilizando JavaScript, dado que se trata de una aplicación web y requiere integración con tecnologías de frontend como HTML

y CSS. Para la implementación de la lógica de juego y la interacción con las notas musicales en tiempo real, se utilizarán JavaScript y el framework AubioJS para la detección de frecuencias.

b. Herramientas CASE (Análisis y Diseño)

Se utilizarán herramientas CASE para el análisis y diseño del sistema, con el fin de asegurar que la estructura del software esté bien definida antes de su implementación. Las herramientas específicas incluirán:

- **PlantUML** para la creación de diagramas de clases y secuencia.
- **Lucidchart** o **Microsoft Visio** para diagramas de flujo de datos (DFD) y diagramas de arquitectura.

Estas herramientas permiten una representación visual clara de los componentes del sistema, facilitando la colaboración entre los desarrolladores y otras partes interesadas.

c. Restricciones de Hardware y Software

La aplicación MusicNet, al ser un juego en tiempo real multijugador, presenta ciertas restricciones en cuanto al hardware y software:

- **Requisitos de hardware:** Los jugadores deberán tener acceso a un dispositivo con pantalla, teclado (o piano MIDI compatible) y conexión a internet estable. En cuanto al hardware, se necesitará un procesador básico para ejecutar la aplicación de manera fluida y una tarjeta gráfica básica compatible con navegadores modernos.
- **Requisitos de software:** Se requiere que los usuarios utilicen navegadores compatibles con HTML5 y JavaScript (Chrome, Firefox, Edge, Safari). Además, la aplicación está diseñada para ser compatible con los sistemas operativos Windows 10, macOS 10.13+ y Linux Ubuntu 18.04+.
- **Conexión de red:** Para garantizar la experiencia multijugador sin interrupciones, se recomienda una conexión de red LAN para sesiones locales y WAN para sesiones en línea, usando TCP y UDP para la transmisión de datos.
- **Permisos del navegador:** Es necesario que el navegador web conceda permisos de acceso al **micrófono**, ya que MusicNet utilizara la entrada de audio en tiempo real para la detección y análisis de notas musicales interpretadas por el usuario.

d. Diseño de la Arquitectura Final del Sistema

La arquitectura de MusicNet se basará en un modelo distribuido híbrido, que combina cliente ligero en navegador con lógica de juego y procesamiento de audio, y un servidor de señalización para coordinar conexiones peer-to-peer mediante WebRTC. El sistema será optimizado para la sincronización en tiempo real de múltiples jugadores que interactúan musicalmente desde ubicaciones remotas. La detección y reconstrucción de notas utiliza algoritmos de análisis de audio y compensación de pérdida de paquetes. Los componentes principales son:

- **Cliente Web (Frontend):** Interfaz gráfica de usuario, detección de notas (pitch) usando micrófono, reproducción sonora (feedback al jugador), renderizado de escenas multijugador, establecimiento y gestión de conexión P2P (WebRTC).
- **Servidor de Señalización (Backend):** Coordinación inicial de conexiones WebRTC, gestión de salas de juego, enrutamiento de mensajes de control (ofertas, respuestas, ICE), asignación de roles (host / cliente), promoción automática del host si se desconecta.
- **Módulo de Reconstrucción de Audio (PARCNet):** Detección de pérdida de paquetes, predicción y reconstrucción de notas faltantes.

3.5 Atributos del Sistema de Software (No funcionales)

3.5.1 Confiabilidad

La confiabilidad en **MusicNet** está centrada en la disponibilidad de los datos y el correcto funcionamiento del sistema bajo condiciones normales de uso. Se considera que el sistema debe ser capaz de manejar múltiples conexiones concurrentes y recuperar su estado en caso de fallos.

- **Manejo de transacciones concurrentes:** Se garantiza que los jugadores puedan interactuar simultáneamente en el juego sin conflictos ni pérdida de datos, utilizando técnicas de sincronización de red.
- **Recuperación o tolerancia a fallos:** La aplicación estará equipada con mecanismos para manejar caídas inesperadas. Si un jugador se desconecta, la sesión se reestablecerá cuando se reconecte, recuperando el estado del juego y sin pérdida de progreso en las acciones.

Criterios de evaluación:

- El sistema debe garantizar un 99.9% de tiempo de actividad durante el periodo de uso en condiciones normales.
- La restauración de la sesión tras una desconexión no debe exceder los 5 segundos.

3.5.2 Disponibilidad

La disponibilidad se refiere al tiempo que la aplicación estará en funcionamiento sin necesidad de reiniciarla. Dado que **MusicNet** es un juego multijugador, debe funcionar sin interrupciones, con el objetivo de mantener el 100% de disponibilidad de los servidores de juego.

- **Aplicaciones externas:** La disponibilidad de la aplicación no dependerá de servicios externos como redes sociales o autenticaciones de terceros, pero el servicio de conexión a la red debe estar disponible siempre.
- **Comunicación:** El sistema debe mantener la comunicación entre los jugadores durante las sesiones, incluso con variaciones de latencia o pérdida de paquetes. Se utilizarán técnicas de compensación de pérdida de paquetes para garantizar la continuidad de la experiencia de juego.

- **Tiempo del Administrador:** Un administrador del sistema debe estar disponible durante las 24 horas del día para mantenimiento preventivo y resolución de incidencias críticas.

Criterios de evaluación:

- La disponibilidad del sistema será del 99.95% durante cada mes.
- Las actualizaciones del sistema no deben requerir más de 2 horas de inactividad en horarios programados.

3.5.3 Mantenibilidad

La mantenibilidad de **MusicNet** es fundamental para garantizar que el sistema pueda adaptarse a cambios futuros y correcciones sin comprometer la estabilidad.

- **Modularización:** El sistema estará dividido en módulos funcionales, como gestión de sesiones, puntuaciones y conexión en red, lo que permitirá actualizar y modificar partes del sistema sin afectar el resto de la aplicación.
- **Reutilización:** Se promoverá la reutilización de componentes para facilitar futuras expansiones del sistema (e.g., agregar nuevos niveles, instrumentos o modalidades de juego).
- **Documentación del código:** El código será documentado adecuadamente para que los desarrolladores futuros puedan entender y modificar el sistema sin dificultades.

Criterios de evaluación:

- El tiempo de implementación de nuevas funcionalidades no debe exceder las 3 semanas.
- El sistema debe contar con documentación de código al 100% para sus módulos principales.

3.5.4 Portabilidad

MusicNet debe ser capaz de ejecutarse en diferentes plataformas y adaptarse a diversas configuraciones de hardware y software, dado que se trata de una aplicación web multijugador.

- **Migración a diferentes sistemas operativos:** La aplicación estará basada en tecnologías web estándar (e.g., JavaScript, HTML5, CSS3) y podrá ser ejecutada en cualquier plataforma que soporte navegadores modernos como Chrome, Firefox y Safari.
- **Ambientes de ejecución:** Se implementará el sistema en Docker para garantizar que se pueda ejecutar en diferentes entornos de desarrollo, prueba y producción, independientemente de las configuraciones de hardware.
- **Lenguaje y compilador:** El sistema está desarrollado utilizando tecnologías basadas en JavaScript para el frontend y Java para el backend, lo que garantiza su portabilidad a diferentes servidores web.

Criterios de evaluación:

- La aplicación debe ser ejecutable en al menos tres sistemas operativos diferentes (Windows, Linux y macOS).
- La configuración de Docker debe ser capaz de desplegar el sistema en menos de 5 minutos.

4. Proceso de ingeniería de requerimientos

En la construcción de esta Especificación de Requerimientos de Software (SRS) para MusicNet, se siguió un proceso estructurado de Ingeniería de Requerimientos que incluye las siguientes etapas clave:

4.1 Elicitación de Requerimientos

Dado que el proyecto se desarrolla bajo la supervisión directa de los directores de tesis, la elicitación de requerimientos se realizó mediante entrevistas y reuniones periódicas con los mismos. Esto permitió capturar de manera precisa las necesidades y expectativas del sistema a desarrollar, evitando ambigüedades y alineando los objetivos con las capacidades técnicas del equipo.

Técnicas Utilizadas:

- **Reuniones:** Realizadas con los directores para definir los requerimientos iniciales y obtener retroalimentación continua.
- **Análisis de documentación previa:** Se revisaron proyectos similares, como JumpApp, para identificar mejoras y funcionalidades que añadan valor a MusicNet.
- **Prototipos y mockups:** Se presentaron prototipos de la interfaz de usuario para validar los requerimientos funcionales y de usabilidad.

4.2 Análisis y Priorización de Requerimientos

Una vez identificados los requerimientos, se realizó un análisis de factibilidad técnica y una priorización basada en su impacto y valor para el proyecto. Se utilizó la técnica **MoSCoW** (Must have, should have, could have, Won't have) para categorizar los requerimientos.

Resultado del Análisis:

- **Requerimientos esenciales:** Funcionalidades Core del sistema como la detección de notas en tiempo real, sincronización multijugador, y retroalimentación visual.
- **Requerimientos opcionales:** Mejoras adicionales como modos de juego avanzados o personalización del avatar.

4.3 Especificación y Documentación

Los requerimientos fueron documentados utilizando el formato de especificación IEEE 830, asegurando claridad y precisión. Cada requerimiento incluye una descripción detallada, criterios de aceptación y referencia a módulos específicos del sistema.

4.4 Control y Gestión de Cambios

Los cambios en los requerimientos se gestionan a través del proceso de control de requerimientos descrito en el SPMP. Cualquier ajuste o modificación debe ser validado por los directores de tesis antes de ser incorporado al documento SRS. La trazabilidad se mantiene utilizando una matriz de trazabilidad que permite rastrear cada requerimiento desde su origen hasta su implementación y prueba.

5. Proceso de verificación

El proceso de verificación y validación (V&V) aplicado a la especificación de requerimientos de MusicNet se desarrolló con el fin de garantizar que el SRS cumple con los estándares de calidad y que los requerimientos individuales reflejan correctamente las necesidades del sistema.

5.1 Verificación del Documento SRS

Para asegurar la calidad del documento, se aplicaron las siguientes actividades de verificación:

- **Revisión interna:** El documento fue revisado internamente por el equipo del proyecto, enfocado en detectar inconsistencias, ambigüedades y errores.
- **Lista de chequeo:** Se utilizó una lista de verificación basada en la norma IEEE 830 para evaluar la completitud y precisión de la especificación.

5.2 Verificación de Requerimientos Individuales

Cada requerimiento fue validado utilizando los siguientes métodos:

- **Inspección formal:** Los requerimientos se revisaron detalladamente para comprobar su claridad, viabilidad y ausencia de ambigüedades.
- **Criterios de aceptación:** Se definieron criterios de aceptación específicos para cada requerimiento, que servirán de referencia durante la fase de pruebas del sistema.
- **Prototipos de validación:** Algunos requerimientos, especialmente los relacionados con la interfaz de usuario y la detección de notas musicales, fueron validados mediante prototipos funcionales, permitiendo obtener retroalimentación inmediata.

5.3 Trazabilidad y Gestión de Cambios

Se ha establecido una matriz de trazabilidad para mantener un registro de cada requerimiento desde su identificación hasta su implementación y prueba. Cualquier modificación a los requerimientos debe ser aprobada por los directores de tesis y documentada adecuadamente, siguiendo el proceso de gestión de cambios definido en el SPMP.

Herramientas utilizadas:

- **Word:** Para la colaboración y edición del documento en tiempo real.
- **Matriz de trazabilidad:** Para asegurar la conexión entre los requerimientos, diseño e implementación del sistema.