



**GRUPO 2**

*INTEGRANTES:*

*BRUNO BARNES Y GONZALO TORRES*

# **SOFTNOTE**

**PRÁCTICA PROFESIONAL 3**

*PROFESORA:*

*PATRICIA LITOVICIUS*



## ÍNDICE

1. Descripción del Proyecto
  - 1.1. Contexto
  - 1.2. Problema
  - 1.3. Tipo de sistema
  - 1.4. Actores principales
  - 1.5. FODA
2. Objetivos
  - 2.1. Objetivo General: ¿Qué necesidad resuelve el sistema?
  - 2.2. Objetivos Específicos: Metas concretas del sistema
3. Alcance
  - 3.1. Funcionalidades incluidas en esta versión (Alcance esperado)
  - 3.2. Limitaciones (qué no se cubrirá en el prototipo)
4. Análisis de Usuarios y Experiencia de Usuario
  - 4.1. Actores Principales del Sistema SoftNote
  - 4.2. User Persona
  - 4.3. Mockup
  - 4.4. Principios Gestalt, Nielsen y UX
  - 4.5. Cuadro de Trazabilidad
5. Casos de Uso
  - 5.1. Descripción del Diagrama de Casos de Uso
6. Especificación de Requerimientos
  - 6.1. Requerimientos funcionales
  - 6.2. Requerimientos no funcionales
  - 6.3. Requisitos de Arquitectura de SoftNote
  - 6.4. Modelo Cliente-Servidor (Roles)
  - 6.5. Cliente (Front-end)
  - 6.6. Servidor (Back-end)
  - 6.7. Base de Datos
  - 6.8. Posibles Servicios Externos (APIs y Nube)
7. Recursos Disponibles
  - 7.1. Recursos Humanos
  - 7.2. Recursos Materiales y Tecnológicos
8. Plan de Trabajo
9. Gestión de Riesgos
  - 9.1. Riesgos Técnicos
  - 9.2. Riesgos Organizacionales
10. Diseño del sistema
  - 10.1. Descripción del Diagrama de Despliegue
    - 10.1.1. Diagrama de Componentes y Despliegue
  - 10.2. Descripción del Diagrama de Casos de Uso
    - 10.2.1. Diagrama de Casos de Uso
  - 10.3. Descripción del Diagrama de Actividad
    - 10.3.1. Diagrama de Actividad
  - 10.4. Descripción del Diagrama de Secuencia
    - 10.4.1. Diagrama de Secuencia
  - 10.5. Descripción del Diagrama de Clases
    - 10.5.1. Diagrama de Clases
  - 10.6. Interfaces de Usuario (UI)
11. Conclusiones finales



## 1) Descripción del Proyecto

### 1.1) Contexto

El caso sucede en el ámbito del aprendizaje musical para todo público. Hoy en día existen aplicaciones que ayudan a detectar acordes, pero la mayoría están pensadas para músicos con conocimientos previos, dejando afuera a principiantes y personas que solo buscan acercarse a la música de manera simple y divertida.

### 1.2) Problema

Muchas personas sienten que aprender música es difícil o inaccesible. Los principiantes se frustran al no entender acordes o teoría, y los aficionados abandonan por falta de motivación. Las apps actuales suelen ser técnicas, poco amigables o limitadas. Se necesita una herramienta que combine detección automática de acordes con recursos interactivos, educativos y lúdicos para que cualquier persona pueda aprender y practicar música sin importar su edad o nivel.

### 1.3) Tipo de Sistema

Desarrollar una aplicación móvil cliente-servidor utilizando una API REST que facilite el aprendizaje musical a través de la detección de acordes en tiempo real, pero que también ofrezca afinador, ejercicios, videos, grabación, reconocimiento musical y elementos de mecánicas de juegos. De esta manera, SoftNote busca ser una plataforma inclusiva, accesible y motivadora para personas de todas las edades y niveles musicales.

### 1.4) Actores principales

Los principales usuarios de SoftNote son personas que quieren aprender y practicar música:

- Principiantes: Niños y jóvenes que están empezando, así como adultos sin conocimientos previos que buscan aprender de forma recreativa.
- Músicos amateurs: Estudiantes y aficionados que buscan una herramienta para practicar y mejorar sus habilidades.
- Profesores de música: Buscan una herramienta didáctica para usar en sus clases.



- Público en general: Personas que simplemente quieren disfrutar de la música de una manera más sencilla y accesible.

### 1.5) FODA

#### Fortalezas

- Innovador: detección automática de acordes por audio.
- Amplia variedad de funcionalidades (afinador, ejercicios, juegos).
- Adaptado a todo público: desde principiantes hasta músicos avanzados.
- Uso de mecánicas de juegos para motivar la práctica diaria.

#### Oportunidades

- Mercado amplio: estudiantes, aficionados y profesores.
- Poca competencia que combine detección con educación y gamificación.
- Posibilidad de integraciones con plataformas educativas o de música.
- Aumento del interés por apps de aprendizaje lúdico (Duolingo, Kahoot, etc.).

#### Debilidades

- Desarrollo complejo (reconocimiento de audio, IA y gamificación).
- Puede requerir mucho tiempo y recursos técnicos.
- Riesgo de sobrecarga de funciones si no se priorizan bien.
- Necesidad de contenido constante (videos, ejercicios) para mantener interés.

#### Debilidades

- Desarrollo complejo (reconocimiento de audio, IA y gamificación).
- Puede requerir mucho tiempo y recursos técnicos.
- Riesgo de sobrecarga de funciones si no se priorizan bien.
- Necesidad de contenido constante (videos, ejercicios) para mantener interés.



## 2) Objetivos

### 2.1) Objetivo General: ¿Qué necesidad resuelve el sistema?

El objetivo principal de SoftNote es crear una aplicación que facilite el aprendizaje musical y resuelva la necesidad de una herramienta inclusiva, accesible y motivadora.

Resuelve la necesidad de ofrecer una plataforma que combine la detección automática de acordes con recursos interactivos, educativos y lúdicos, permitiendo que cualquier persona pueda aprender y practicar música, sin importar su edad o nivel de conocimiento previo.

### 2.2) Objetivos Específicos: Metas concretas del sistema

Los objetivos específicos del sistema se traducen en las funcionalidades clave que SoftNote busca ofrecer:

- Desarrollar una aplicación cliente-servidor que realice la detección de acordes en tiempo real a partir de audios o canciones cargadas.
- Ofrecer un afinador integrado.
- Proporcionar ejercicios interactivos para practicar el oído, los cambios de acordes y el ritmo.
- Incluir videos educativos por nivel.
- Implementar un grabador de audio con análisis de desempeño y feedback.
- Desarrollar una función de reconocimiento musical estilo Shazam (canción, artista, letra y acordes).
- Incorporar elementos de gamificación (rachas, logros, puntos, minijuegos) para motivar la práctica.
- Permitir la visualización de diagramas de acordes según el instrumento seleccionado (guitarra, piano, ukelele, etc.).
- Ofrecer la exportación a PDF de los resultados y la opción de compartir



### 3) Alcance

#### 3.1) Funcionalidades incluidas en esta versión (Alcance esperado)

El proyecto busca desarrollar una aplicación cliente-servidor para facilitar el aprendizaje musical. Las funcionalidades incluidas son:

- Detección de acordes en tiempo real a partir de audios o canciones.
- Visualización con diagramas de acordes según el instrumento seleccionado.
- Exportación a PDF y opción de compartir los resultados.
- Un afinador integrado.
- Ejercicios interactivos para practicar el oído, los cambios de acordes y el ritmo.
- Vídeos educativos clasificados por nivel.
- Un grabador con análisis de desempeño y feedback.
- Reconocimiento musical (estilo Shazam) que identifica canción, artista, letra y acordes.
- Mecánicas de juegos (gamificación) como rachas, logros, puntos y minijuegos para motivar la práctica.

#### 3.2) Limitaciones (qué no se cubrirá en el prototipo)

El alcance del prototipo no incluye las siguientes funcionalidades:

- La aplicación no reemplaza clases presenciales.
- La aplicación no ofrece repertorios prearmados.



## 4) Análisis de Usuarios y Experiencia de Usuario (UX)

### 4.1) Actores Principales del Sistema SoftNote

#### 1. Usuarios de la Aplicación (Actores Humanos)

Estos perfiles interactúan directamente con las funcionalidades de aprendizaje y práctica musical.

- Principiantes
  - Definición: Niños y jóvenes que recién empiezan en la música, o adultos sin conocimientos previos que buscan aprender de forma recreativa.
  - Objetivos: Aprender de forma simple y divertida, practicar canciones fáciles, y ser motivados con juegos/recompensas.
  - Frustraciones: Les frustra no entender acordes o teoría, y pueden abandonar por falta de motivación.
- Estudiantes y Músicos Amateurs
  - Definición: Estudiantes y aficionados que buscan una herramienta para practicar y mejorar sus habilidades.
  - Objetivos: Practicar canciones, mejorar el oído, las progresiones de acordes, y obtener métricas de práctica.
  - Necesidades: Precisión en la detección de acordes y poder exportar los resultados.
- Profesores de Música
  - Definición: Buscan una herramienta didáctica para usar en sus clases.
  - Objetivos: Usar la app como recurso didáctico y compartir acordes/exportaciones con sus alumnos.
  - Necesidades: Simplicidad y rapidez de uso durante las clases.
- Público General
  - Definición: Personas que simplemente quieren disfrutar de la música de una manera más sencilla y accesible.



## 2. Administrador del Sistema (Actor No Humano)

Este perfil está a cargo de las funciones de gestión y mantenimiento de la plataforma.

- Definición: Es el encargado del soporte y mantenimiento del sistema y de gestionar el contenido

### 4.2) User Persona

Nombre	Emma Perez
Edad / ocupación	12 años, estudiante de secundaria
Objetivos	Aprender guitarra de forma divertida, sin teoría difícil
Necesidades / frustraciones	Quiere practicar canciones fáciles, se frustra si no entiende acordes, busca juegos/recompensas para motivar
Contexto de uso	Celular con auriculares, 3-4 veces por semana en su casa

Nombre	Gonzalo Perez
Edad / ocupación	22 años, universitario
Objetivos	Practicar canciones complejas, mejorar oído y progresiones

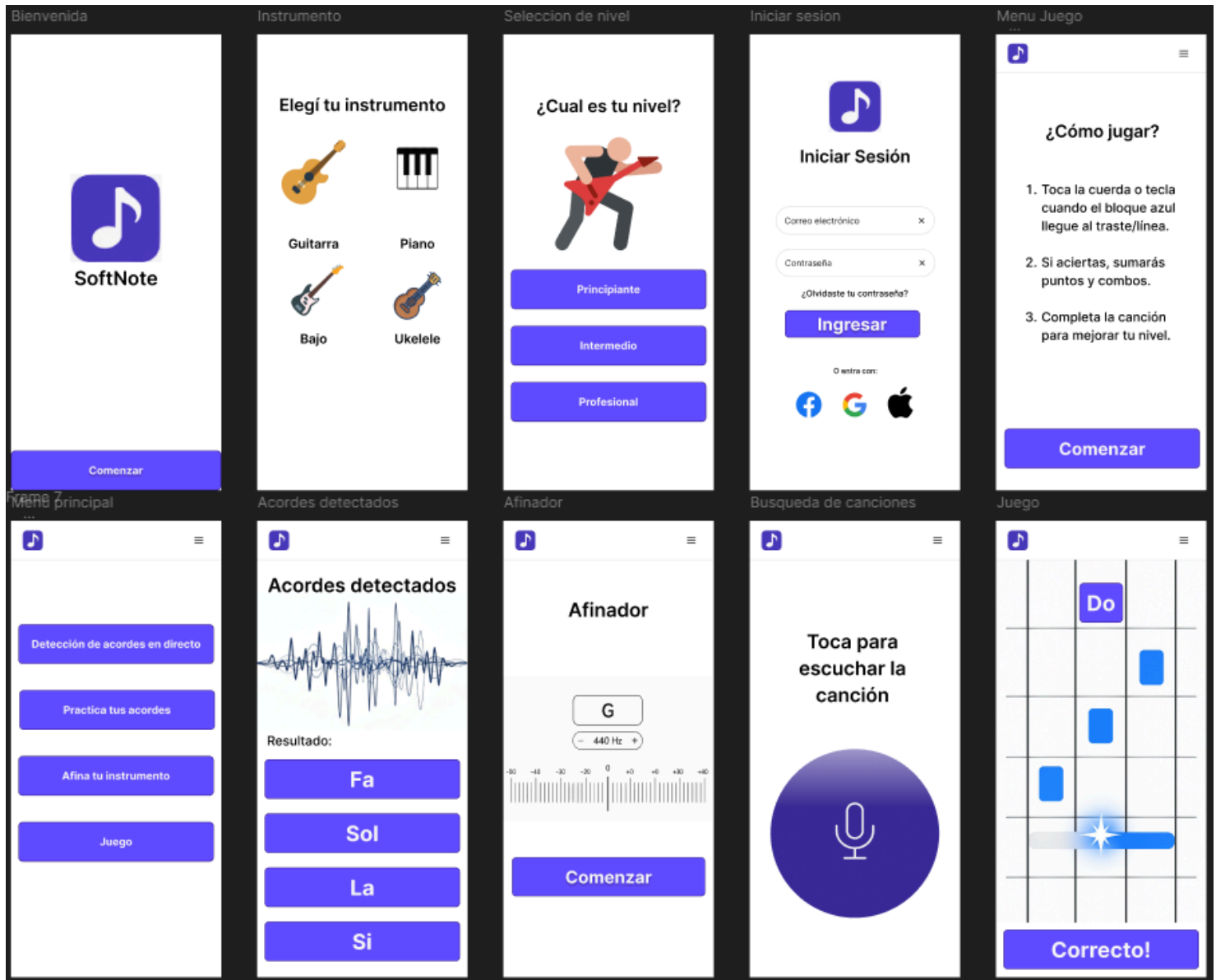




Necesidades / frustraciones	Precisión en la detección de acordes, exportar resultados, métricas de práctica, frustración si falla en acordes avanzados
Contexto de uso	Notebook y celular, diariamente en su casa o sala de ensayo

Nombre	Bruno Perez
Edad / ocupación	35 años, profesor de piano y teoría musical
Objetivos	Usar la app como recurso didáctico y compartir acordes/exportaciones con alumnos
Necesidades / frustraciones	Simplicidad y rapidez en clase, frustración si la app es confusa o consume mucho tiempo
Contexto de uso	Tablet o notebook en clases presenciales, 2-3 veces por semana

## 4.3) Mockup





#### 4.4) Principios Gestalt, Nielsen y UX

En el diseño del prototipo de SoftNote se definieron ocho pantallas principales: bienvenida, selección de instrumento, selección de nivel, inicio de sesión, menú principal, detección de acordes, afinador y búsqueda de canciones. La navegación se planteó de forma lineal y clara, permitiendo al usuario avanzar desde la personalización inicial hacia el uso completo de las funciones de la aplicación.

En cuanto a los principios de la Gestalt, se aplicó proximidad agrupando botones relacionados (por ejemplo, opciones de inicio de sesión), semejanza al usar colores e íconos consistentes en cada función, figura-fondo para destacar información clave como los acordes

detectados, continuidad en el flujo de navegación entre pantallas y simetría en la distribución de los elementos para dar equilibrio visual.

Respecto a las heurísticas de Nielsen, se consideró la visibilidad del estado del sistema mediante indicadores claros durante el análisis de acordes o el uso del afinador; consistencia y estándares al mantener el mismo estilo de botones e íconos; prevención de errores con confirmaciones en acciones críticas como salir o borrar; control y libertad del usuario ofreciendo siempre la opción de volver atrás; y reconocimiento antes que recuerdo mediante íconos intuitivos que reducen la necesidad de memorizar pasos.

Estas decisiones buscan mejorar la experiencia de usuario (UX) garantizando simplicidad, claridad y confianza en el uso de la app, y al mismo tiempo fortalecer la interfaz (UI) con un diseño minimalista y profesional, orientado a que los usuarios de distintos niveles musicales puedan interactuar sin dificultades.

<https://www.figma.com/design/TJE1RIMm5BIY4ro5RBxgzN/Softnote?node-id=0-1&t=e42n0D9u4YcOMkV>  
[K-1](#)



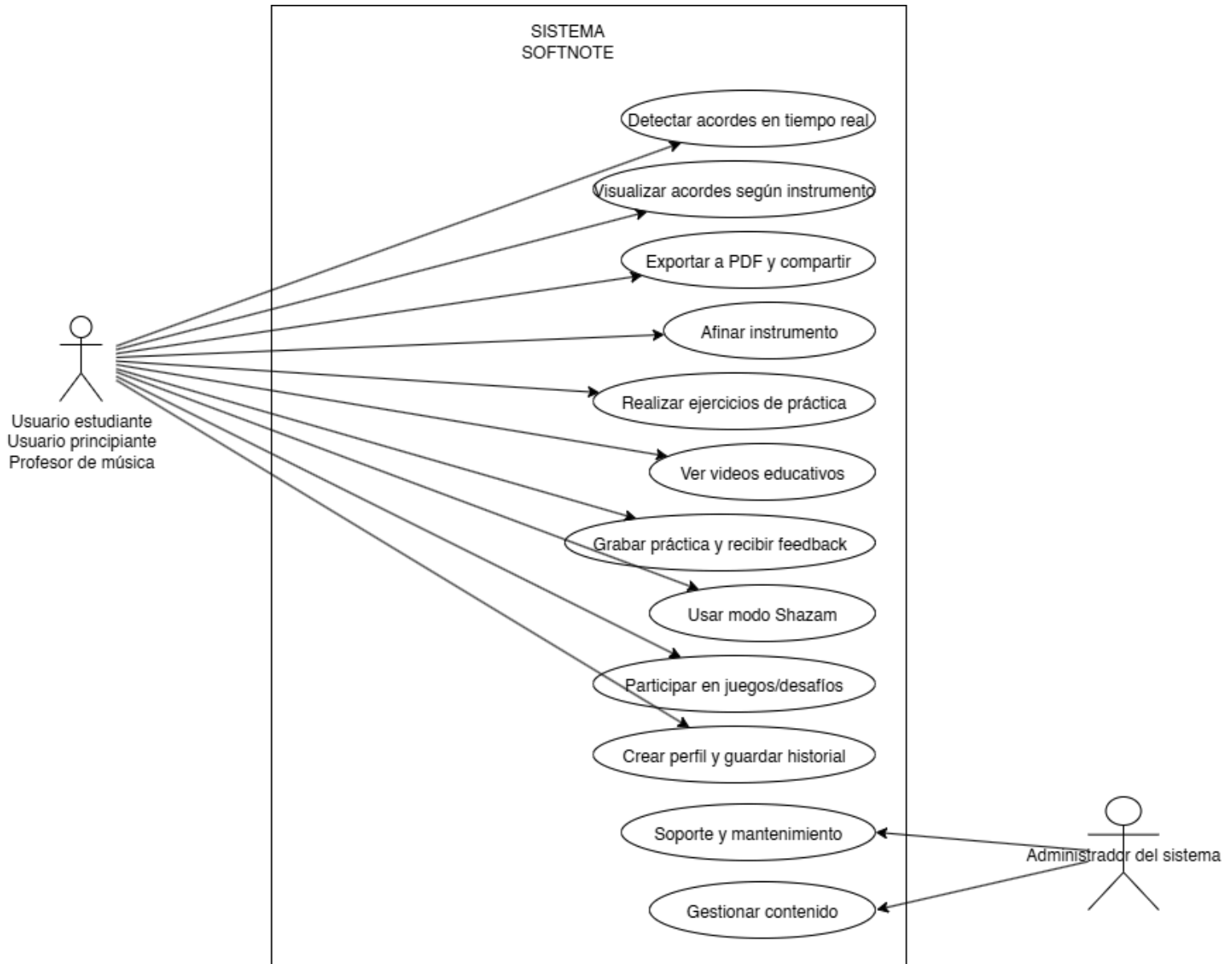
#### 4.5) Cuadro de Trazabilidad

<b>Actor</b>	<b>Caso de Uso</b>	<b>Actividad del Sistema</b>	<b>Pantalla UX Asociada</b>
Usuario (todos)	Registrarse / Iniciar sesión	Validar credenciales y cargar perfil	Pantalla “Login / Registro”
Usuario principiante	Seleccionar instrumento	Guardar preferencia de instrumento (guitarra, piano, etc.)	Pantalla “Selección de Instrumento”
Usuario estudiante	Seleccionar nivel	Configurar dificultad según nivel elegido	Pantalla “Selección de Nivel”
Usuario profesor	Navegar menú principal	Mostrar accesos a funciones principales	Pantalla “Menú Principal”
Usuario (todos)	Practicar acordes	Mostrar acordes, guiar digitación y evaluar precisión	Pantalla “Acordes”
Usuario (todos)	Afinar instrumento	Captar audio y mostrar desviación respecto a nota estándar	Pantalla “Afinador”



Usuario (todos)	Buscar canciones	Reconocer canción o mostrar resultados según búsqueda	Pantalla “Búsqueda de Canciones”
Usuario (todos)	Jugar para practicar música	Mostrar interfaz de juego rítmico, detectar aciertos y fallos	Pantalla “Juego Interactivo”

## 5) Casos de uso



### 5.1) Descripción del Diagrama de Casos de Uso

El diagrama de casos de uso de SoftNote representa las interacciones principales entre los usuarios del sistema y las funcionalidades centrales de la aplicación. Se identifican dos actores principales: el



Estudiante/Músico, que utiliza la app para aprender, afinar su instrumento y practicar, y el Administrador, que gestiona contenidos internos del sistema. El diagrama describe las acciones clave del usuario, como ingresar a la aplicación, afinar su instrumento, detectar acordes desde audio, realizar ejercicios interactivos, acceder a videos educativos y utilizar el grabador con análisis de desempeño. También se incluyen extensiones como la exportación a PDF y el reconocimiento avanzado de canciones.

Este diagrama resume el alcance funcional del sistema desde la perspectiva del usuario final y permite visualizar de manera clara cómo cada actor se vincula con las funciones principales, sirviendo como base para el diseño posterior de interfaces, arquitectura y planificación del desarrollo.

Nombre	Detección de acordes en tiempo real
Actores	Usuario (principiante, estudiante, profesor), Sistema SoftNote
Descripción	El usuario selecciona un audio (desde micrófono, archivo o link) y el sistema procesa la canción mostrando los acordes detectados en tiempo real.
Precondiciones	El usuario debe tener acceso al sistema y cargar un audio válido
Flujo principal	El usuario abre la opción “Detección de acordes”.  Elige fuente de audio (archivo, micrófono o link).  El sistema procesa el audio.  El sistema muestra los acordes detectados en tiempo real con diagramas según instrumento elegido.
Postcondiciones	El usuario visualiza los acordes y puede exportar el resultado.



Excepciones	<p>Si el archivo no es válido → el sistema muestra un mensaje de error.</p> <p>Si no se detectan acordes → el sistema ofrece volver a intentar con otro audio.</p>
-------------	--

## 6) Especificación de Requerimientos

### 6.1) Requerimientos funcionales

El sistema debe permitir detectar acordes en tiempo real a partir de canciones o audios cargados.

El sistema debe mostrar diagramas de acordes adaptados al instrumento seleccionado (guitarra, piano, ukelele, etc.).

El sistema debe ofrecer ejercicios interactivos para practicar oído, ritmo y progresiones de acordes.

### 6.2) Requerimientos no funcionales

El sistema debe ser compatible con dispositivos móviles y navegadores web modernos (Android, iOS, Chrome, Edge).

El sistema debe ser fácil de usar (interfaz intuitiva, pensada para principiantes).

El sistema debe garantizar tiempos de respuesta menores a 3 segundos al detectar acordes en canciones cortas.

### 6.3) Requisitos de Arquitectura de SoftNote

SoftNote está planeado como una aplicación con un modelo cliente-servidor, que requiere capacidades de procesamiento intensivo de audio, lo que influye en los roles de cada componente y en la necesidad de servicios externos, especialmente de Inteligencia Artificial (IA).





#### 6.4) Modelo Cliente-Servidor (Roles)

El sistema opera con dos componentes principales (Cliente y Servidor) que se comunican, y una Base de Datos para persistencia.

#### 6.5) Cliente (Front-end)

Es la interfaz de usuario. Debe ser compatible con dispositivos móviles (Android, iOS) y navegadores web. Sus funciones incluyen:

- Captura de audio (micrófono o archivo).
- Envío de audio y peticiones al Servidor.
- Visualización de acordes, diagramas, ejercicios y resultados de juegos.
- Gestión de la UX (navegación, consistencia, visibilidad del estado).

#### 6.6) Servidor (Back-end)

- Es el motor del sistema, encargado del procesamiento pesado. Sus funciones incluyen:
- Procesamiento de audio para detección de acordes en tiempo real.
- Lógica de negocio (gamificación, análisis de desempeño, ejercicios).
- Gestión de la seguridad (validación de credenciales).
- Coordinación con servicios externos (IA, Shazam-like).

#### 6.7) Base de Datos

Almacena la información persistente. Sus funciones incluyen:

- Perfiles de usuario (preferencias, historial, rachas, logros).
- Contenido del sistema (videos educativos, ejercicios, datos de afinador).

#### 6.8) Posibles Servicios Externos (APIs y Nube)

Debido a la complejidad del reconocimiento de audio y la necesidad de contenido constante, SoftNote probablemente requerirá integración con servicios externos:



- Servicios de Detección de Audio / IA
  - Se necesitan para la detección automática de acordes.
  - Dada la complejidad del desarrollo (reconocimiento de audio e IA), se puede considerar una API de terceros o una infraestructura en la nube (como Google AI Platform o AWS SageMaker) para alojar el modelo de Machine Learning que realiza el análisis de acordes.
- API de Reconocimiento Musical (Shazam-like)
  - El sistema busca tener una funcionalidad de reconocimiento de canción, artista, letra y acordes estilo Shazam.
  - Esto requeriría la integración con una API de terceros especializada en el reconocimiento de huellas dactilares de audio.
- Servicios de Contenido/Media Streaming
  - Para la provisión de videos educativos, podría integrarse con plataformas de alojamiento de videos (como YouTube API o Vimeo) o un servicio de Content Delivery Network (CDN) para asegurar la carga rápida y la baja latencia.
- Servicios de Almacenamiento y Exportación
  - Para la funcionalidad de exportación a PDF, se podría usar un servicio en la nube para generar y almacenar temporalmente los documentos.

## 7) Recursos Disponibles

### 7.1) Recursos Humanos

El equipo del proyecto está conformado por dos integrantes, quienes asumieron roles múltiples debido al tamaño reducido del grupo. La distribución de responsabilidades se organizó de la siguiente manera:

- Analista Funcional: Encargado de la definición de requerimientos, análisis de usuarios, elaboración de casos de uso y documentación técnica.



- Diseñador UX/UI: Responsable del diseño de wireframes, prototipos en Figma, lineamientos de accesibilidad y experiencia del usuario.
- Programador / Arquitecto Técnico: A cargo de la definición de la arquitectura cliente-servidor, modelado de datos, diagramas UML y evaluación de viabilidad técnica.
- Tester: Elaboración de pruebas preliminares de navegabilidad y validación del prototipo interactivo.
- Coordinación General: Ambos integrantes comparten el rol de coordinación y control del avance, cumplimientos y ajustes del proyecto.

## 7.2) Recursos Materiales y Tecnológicos

### Software utilizado:

- Figma – Diseño de interfaces, prototipos navegables y sistemas visuales.
- Draw.io / Diagrams.net – Elaboración de diagramas UML (casos de uso, clases, secuencia, actividades).
- Google Docs – Documentación colaborativa del proyecto.
- Google Sheets – Construcción del cronograma y diagrama de Gantt.
- OBS Studio – Grabación del video de presentación final.
- Canva – Creación de presentaciones visuales para la exposición.
- Visual Studio Code o editor equivalente – Simulación de estructura del sistema y análisis técnico.
- GitHub (opcional para futuras versiones) – Control de versiones y colaboración.

### Hardware:

- Notebooks personales con capacidad suficiente para edición de prototipos y trabajos colaborativos.
- Dispositivos móviles (Android) utilizados como referencia en el diseño de pantallas y navegación.

### Servicios complementarios:

- Google Drive para almacenamiento y sincronización de archivos.
- Plataformas de videollamadas (Discord/Meet) para coordinación del equipo.



## 8) Plan de Trabajo

### 8.1) Descripción del Plan de Trabajo:

El diagrama de Gantt presentado resume la planificación del proyecto SoftNote a lo largo de nueve meses, siguiendo una secuencia lógica de etapas propias de un desarrollo real de software. En los primeros meses se abordaron las actividades de análisis, definición del problema, elaboración del FODA y relevamiento de requerimientos. Luego se avanzó con el diseño UX y UI, generando wireframes, mockups y el prototipo navegable.

En la mitad del proyecto se trabajó sobre la arquitectura del sistema, incluyendo modelado de base de datos, definición de la API y los diagramas UML correspondientes. Posteriormente se planificó el desarrollo dividido en dos grandes bloques: implementación de la aplicación móvil y construcción del servidor con su API. Finalmente, los últimos meses se destinaron a integración, pruebas de funcionamiento y usabilidad, documentación completa y preparación de la presentación final del proyecto.

Este cronograma permite visualizar de forma clara los hitos, la carga de trabajo de cada etapa y el orden lógico necesario para lograr un desarrollo organizado, eficiente y coherente con los objetivos del sistema



Tareas	MES 1	MES 2	MES 3	MES 4	MES 5	MES 6	MES 7	MES 8	MES 9
Relevamiento del problema									
Definición de alcance									
Wireframes Low-fi									
Pantallas Principales									
API Rest									
Módulo afinador									
Endpoints Juegos									
Pruebas de usabilidad									
Presentación final									

## 9) Gestión de Riesgos

Durante el desarrollo del proyecto se identificaron riesgos técnicos, organizacionales y de alcance. Para cada uno se definieron estrategias de mitigación con el objetivo de mantener la continuidad del proyecto y garantizar la calidad del producto final.



### 9.1) Riesgos Técnicos

- Fallas del servidor o API: caídas, errores de respuesta, baja disponibilidad.
- Reconocimiento de audio impreciso: detección incorrecta de acordes o interferencias.
- Compatibilidad entre dispositivos: diferencias en comportamiento entre Android/iOS.
- Seguridad de datos: riesgo de exposición o pérdida de información del usuario.

### 9.2) Riesgos Organizacionales

- Limitación de personal: al ser solo dos integrantes, las tareas requieren más tiempo.
- Sobrecarga de funcionalidades: riesgo de intentar abarcar más de lo posible en la primera versión.
- Retrasos en plazos: posibles demoras por la complejidad técnica del proyecto.

### Estrategias de Mitigación

- Implementación de pruebas frecuentes en API y servidor.
- Ajustes iterativos del algoritmo de detección de audio.
- Pruebas cruzadas en múltiples dispositivos.
- Aplicación de buenas prácticas de seguridad (HTTPS, roles, tokens).
- Mantener un enfoque MVP (Producto Mínimo Viable) para garantizar entregables funcionales.
- Planificación detallada mediante cronograma y trabajo distribuido.



## 10) Diseño del Sistema

### Arquitectura Cliente–Servidor

#### 10.1) Descripción del Diagrama de Despliegue

El diagrama de despliegue representa la infraestructura lógica del sistema SoftNote. En él se visualizan los tres componentes principales:

- Aplicación Móvil (Cliente): instalada en el dispositivo del usuario. Se encarga de la interfaz, la captura de audio y las acciones directas que inicia el usuario.
- Servidor / API REST: componente central que procesa las solicitudes entrantes, ejecuta funciones de detección de acordes, afinación, login, registro y manejo de datos.
- Base de Datos: repositorio donde se almacenan usuarios, configuraciones, progreso, historial de prácticas y resultados de ejercicios.

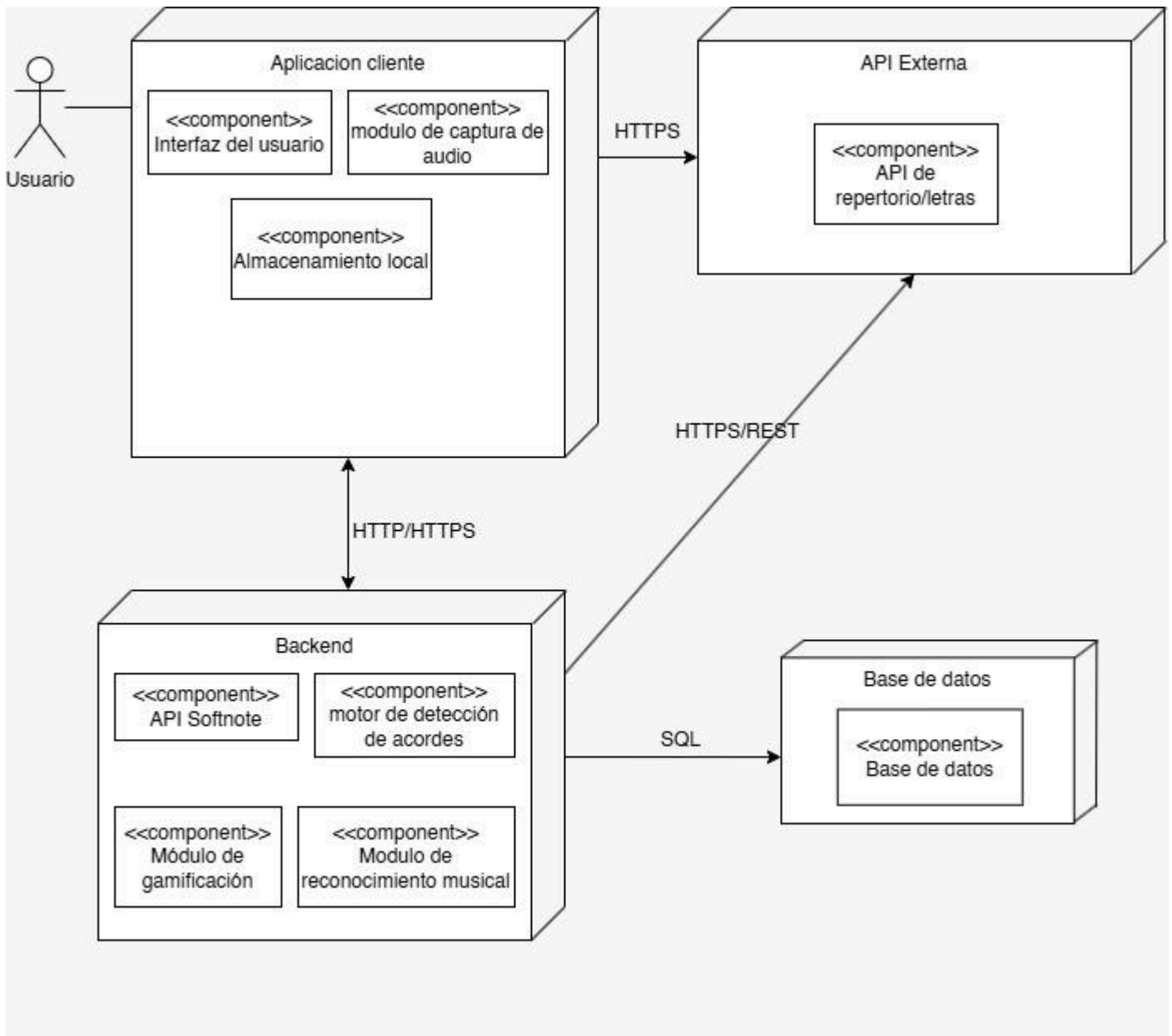
La comunicación entre cliente y servidor se realiza mediante HTTP/HTTPS, utilizando mensajes JSON, lo que garantiza compatibilidad, ligereza y estandarización.

#### Justificación de la Arquitectura

Se seleccionó una arquitectura cliente-servidor debido a que:

- Permite separar la lógica pesada (procesamiento de audio, manejo de datos) del dispositivo del usuario.
- Facilita la escalabilidad, ya que el servidor puede crecer independientemente de la app.
- Garantiza compatibilidad multiplataforma: Android, iOS o web pueden conectarse sin modificar la lógica del servidor.
- Mejora el mantenimiento, ya que las actualizaciones del servidor se reflejan automáticamente en todos los clientes.
- Es el estándar actual para aplicaciones móviles que requieren servicios en tiempo real.

### 10.1.1) Diagrama de Componentes y Despliegue







## Diagramas UML

A continuación se describen los diagramas UML desarrollados para modelar SoftNote:

### 10.2) Descripción del Diagrama de Casos de Uso

El diagrama de casos de uso muestra las interacciones principales entre los usuarios y el sistema.

Los actores incluyen:

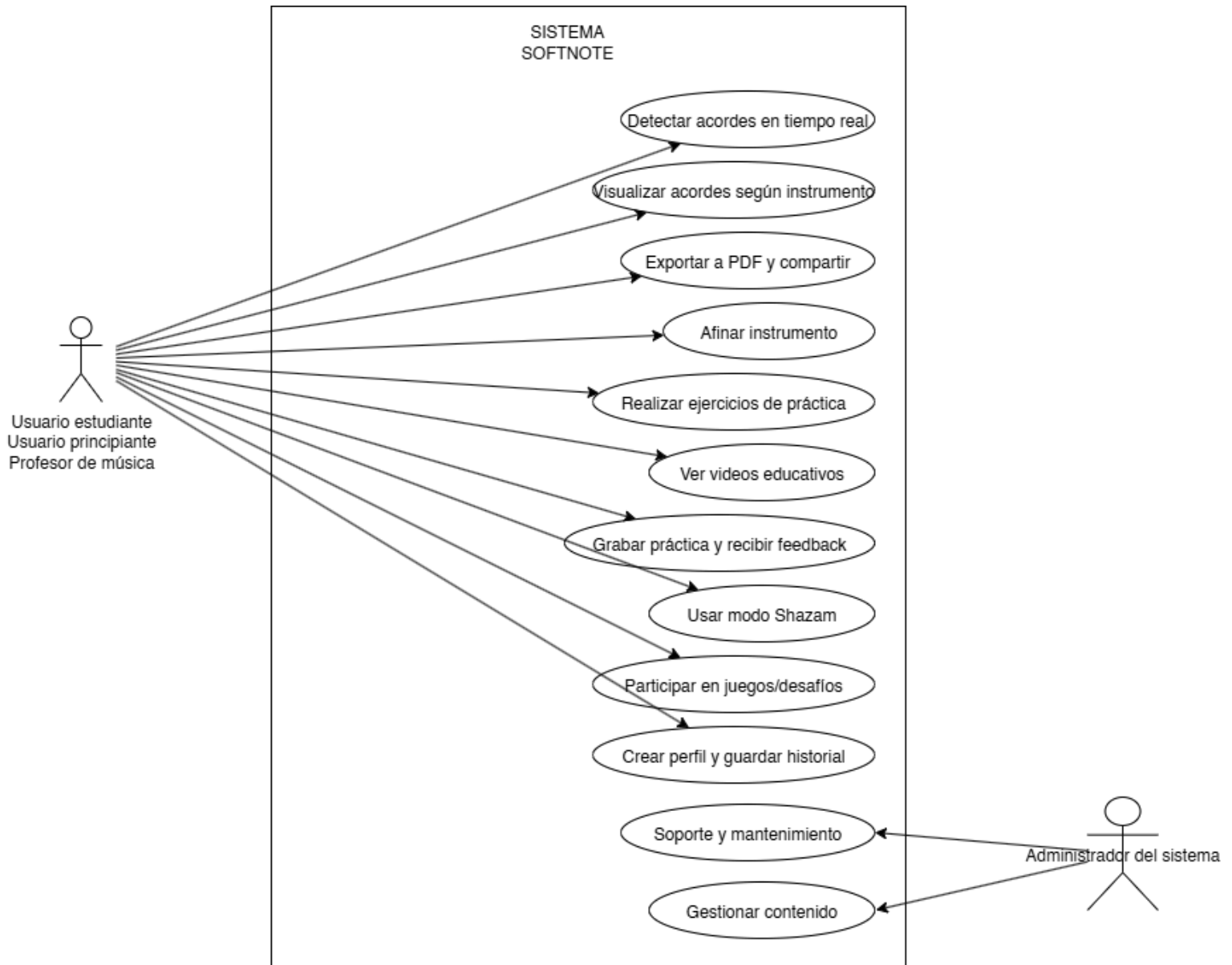
- Usuario registrado (alumno/músico)
- Profesor
- Administrador (si corresponde)

Los casos de uso principales son:

- Detectar acordes
- Afinar instrumento
- Realizar ejercicios/juegos
- Consultar acordes y progresiones
- Iniciar sesión / gestionar perfil
- Exportar resultados
- Grabar y analizar desempeño

Este diagrama permite visualizar el comportamiento esperado del sistema desde la perspectiva del usuario final.

### 10.2.1) Diagrama de Casos de Uso





### 10.3) Descripción del Diagrama de Actividad

El diagrama de actividad describe el flujo de acciones del sistema para una operación específica.

En el caso de SoftNote, se representó el proceso de:

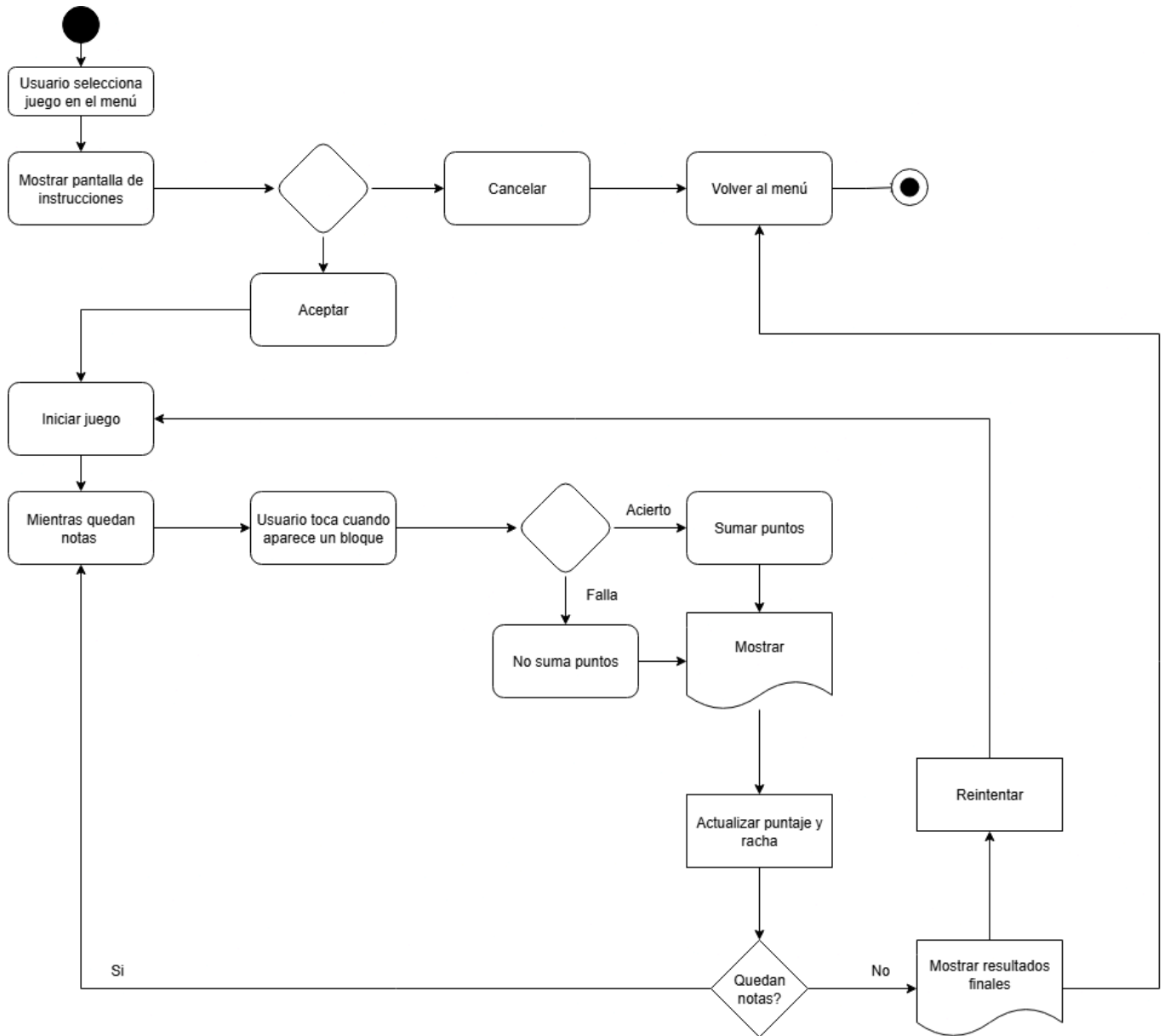
“Detección de acordes en tiempo real”

Incluye:

- Captura del audio
- Envío al servidor
- Procesamiento
- Identificación del acorde
- Devolución del resultado
- Visualización en pantalla

Este diagrama evidencia las decisiones, ramificaciones y pasos secuenciales que el sistema ejecuta, tanto del lado del cliente como del servidor.

### 10.3.1) Diagrama de Actividad





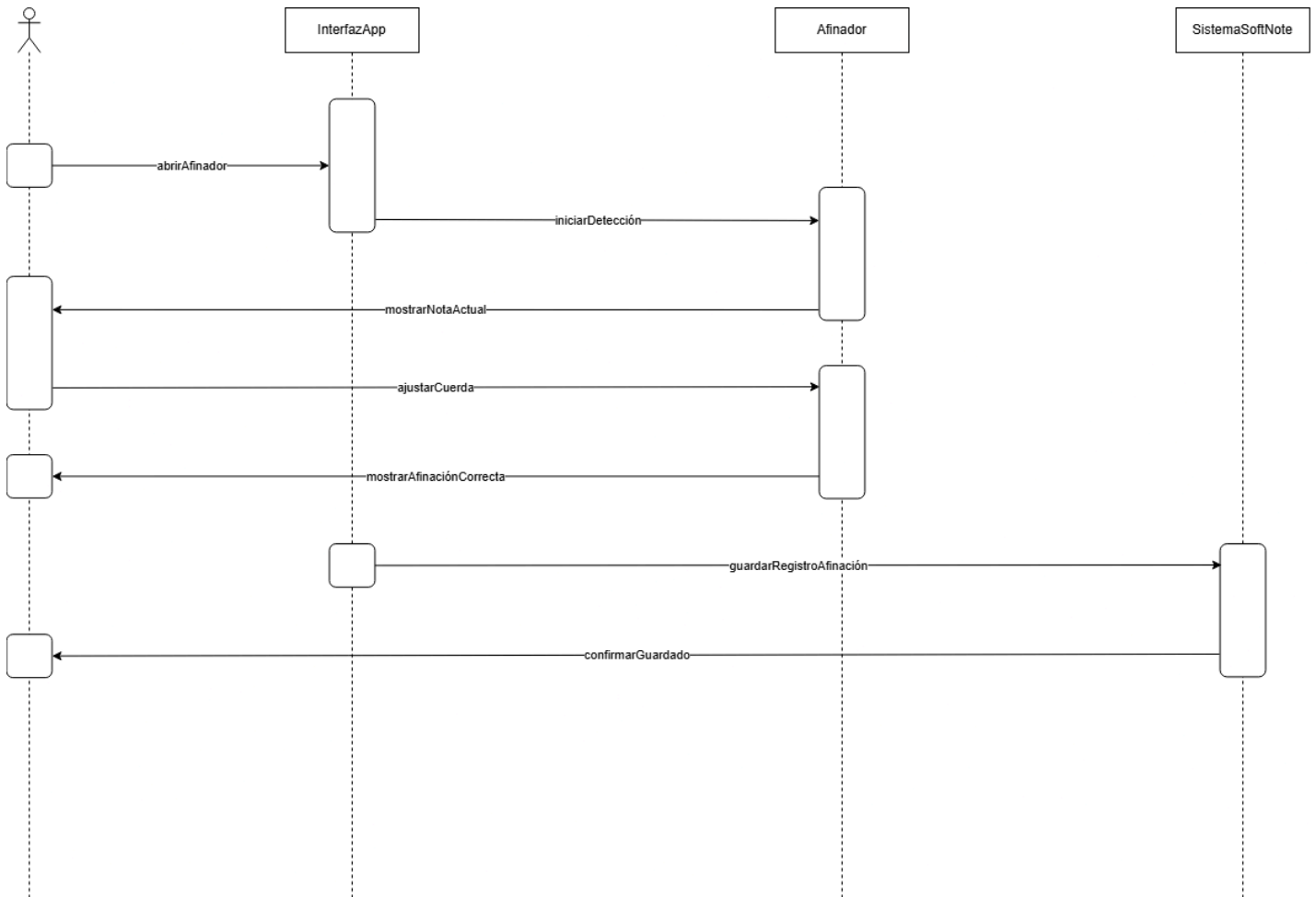
#### 10.4) Descripción del Diagrama de Secuencia

El diagrama de secuencia ilustra la comunicación temporal entre los componentes:

1. El usuario activa la función de detectar el acorde.
2. La app envía el audio o características procesadas al servidor.
3. La API REST ejecuta la lógica correspondiente.
4. Se consulta o actualiza la base de datos si es necesario.
5. El servidor retorna el acorde detectado y la información asociada.
6. La app lo muestra en pantalla en tiempo real.

Este diagrama permite comprender cómo viajan los datos y qué componente es responsable de cada acción.

### 10.4.1) Diagrama de Secuencia





### 10.5) Descripción del Diagrama de Clases

El diagrama de clases modela la estructura interna del sistema.

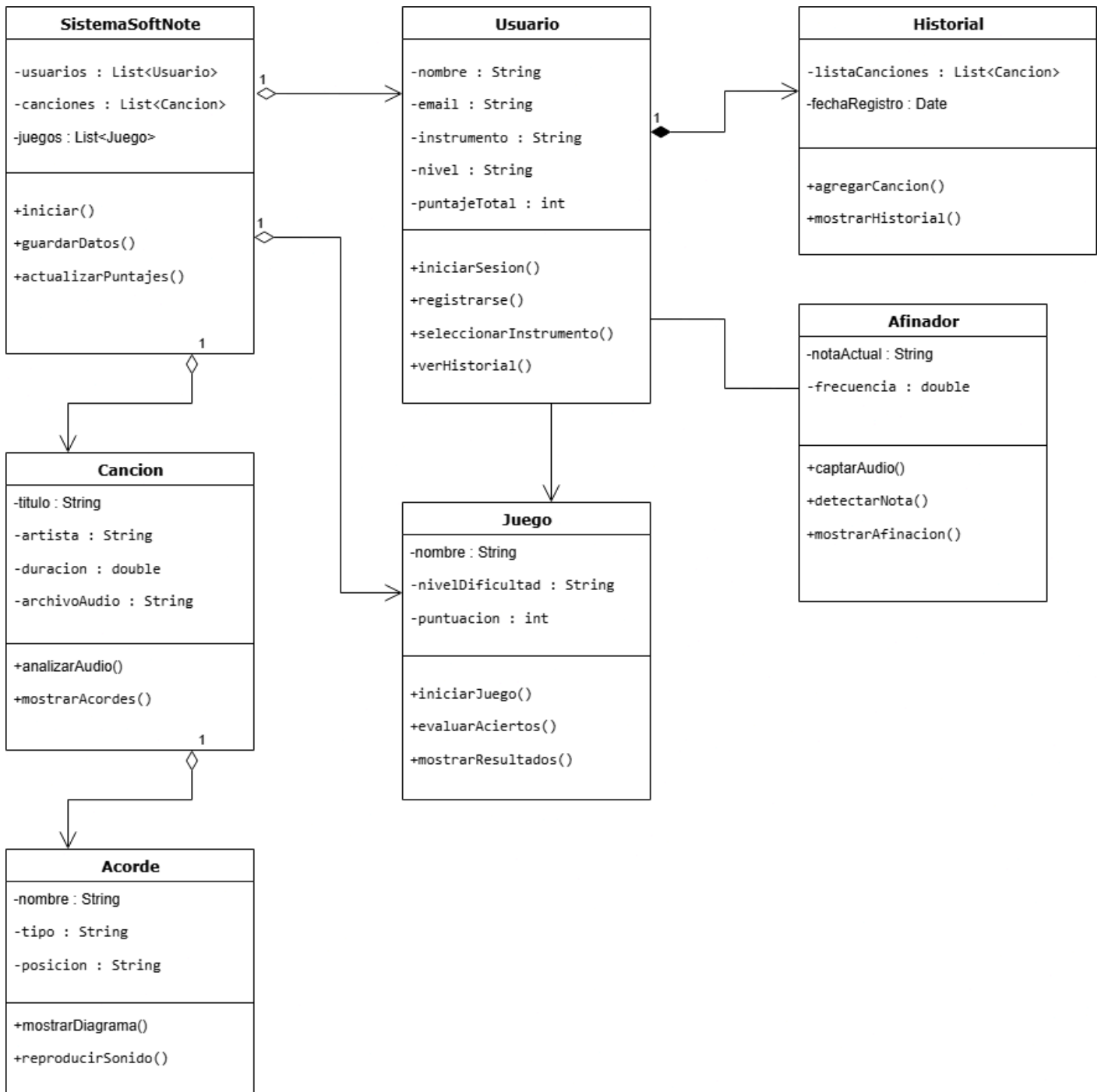
Incluye clases como:

- **Usuario**
- **Acorde**
- **Detección Audio**
- **Ejercicio/Juego**
- **Historial**
- **Exportación**
- **Afinador**

Cada clase define atributos, métodos y relaciones (asociación, agregación o herencia).

Este modelo sirve como base conceptual para una posterior implementación.

## 10.5.1) Diagrama de Clases







### 10.6) Interfaces de Usuario (UI)

El prototipo UI fue desarrollado en Figma y presenta un estilo minimalista, accesible y consistente.

Incluye las siguientes pantallas:

- Inicio
- Selección de instrumento y nivel
- Menú principal
- Detección de acordes
- Afinador
- Juegos y ejercicios
- Visualización de acordes
- Perfil del usuario
- Configuración
- Exportación y análisis

El objetivo del diseño es facilitar la navegación, reducir la carga cognitiva del usuario y mejorar la usabilidad general del sistema.





## Interfaces con Otros Sistemas

Si bien la versión actual no integra APIs externas, el diseño contempla posibles extensiones como:

- Servicios tipo Shazam para reconocimiento de canciones.
- APIs musicales (Spotify, Genius, Ultimate Guitar) para mostrar información adicional.
- Servicios de almacenamiento en la nube para grabaciones o datos de usuario.

Estas integraciones permiten ampliar las capacidades de SoftNote en versiones futuras sin afectar la arquitectura base.



## 11) Conclusiones finales

El desarrollo de SoftNote representó una experiencia integral que permitió aplicar de manera práctica los conceptos trabajados durante la materia. A lo largo del proyecto se abordaron todas las etapas del proceso de diseño y arquitectura de sistemas: análisis del problema, definición de requerimientos, modelado UML, planificación, diseño UI/UX y elaboración de un prototipo navegable. Este recorrido nos permitió reforzar habilidades de trabajo colaborativo, análisis crítico y comunicación técnica.

Entre las fortalezas del sistema se destacan la claridad conceptual de la propuesta, el enfoque en la experiencia del usuario, la definición precisa de la arquitectura cliente-servidor y la coherencia entre los distintos diagramas UML desarrollados. Asimismo, el prototipo demuestra un diseño ordenado, minimalista y escalable, preparado para evolucionar en futuras versiones.

Sin embargo, también identificamos debilidades y desafíos. El reconocimiento de audio constituye el componente más complejo del sistema y requeriría un tratamiento técnico más avanzado para lograr una precisión adecuada en entornos reales. Además, al ser un equipo reducido, el alcance del proyecto debió ajustarse a un MVP, dejando algunas funcionalidades para etapas posteriores.

En cuanto a mejoras futuras, se proyecta optimizar el motor de detección de acordes, integrar servicios externos de música, ampliar los ejercicios interactivos, mejorar el análisis del desempeño y extender la compatibilidad hacia más dispositivos. También se considera profundizar la seguridad del sistema y ampliar los roles de usuario, incorporando funciones específicas para profesores o academias.

En conclusión, SoftNote constituye una base sólida para un producto tecnológico real, con posibilidades de crecimiento y un diseño orientado a resolver necesidades concretas de músicos y estudiantes. El proceso no sólo permitió construir un prototipo funcional, sino también consolidar competencias técnicas y metodológicas fundamentales para futuros desarrollos profesionales.