INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

ESCOM

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Trabajo Terminal

“Melody Unmix: Aplicación web para separación de pistas y descomposición musical”
2025 - XXXX

Presentan

Caballero Perdomo Axel Lennyn
Hernández Pérez Diego Francisco
Jiménez Morales Marco Antonio
Martínez Pérez Raúl

Directores

Dra. Úrsula Samantha Morales Rodríguez
Dr. Andrés García Floriano

Capítulo 1. Introducción

En la actualidad, músicos y productores enfrentan el desafío de manipular pistas de audio de manera precisa para estudiar o analizar las composiciones musicales, esto dificulta que sea posible identificar con claridad los distintos instrumentos en una canción. La separación de instrumentos en una canción es un proceso complejo y generalmente requiere herramientas costosas y conocimientos técnicos avanzados. La aplicación busca ofrecer una solución accesible utilizando IA y procesamiento de señales, la cual permita a los usuarios separar los distintos instrumentos de una pista de audio. Esta herramienta facilitará el trabajo de quienes se dedican a la música, desde el análisis y aprendizaje hasta la producción y mejora de sus composiciones.

1.1 Antecedentes

La música ha sido una forma de expresión artística y cultural fundamental a lo largo de la historia. Con el avance de la tecnología, la producción musical ha evolucionado considerablemente, permitiendo la separación de pistas de audio de manera más eficiente. La

capacidad de aislar y manipular pistas individuales dentro de una canción ha sido de gran interés tanto para productores musicales como para investigadores de inteligencia artificial, músicos aficionados que solo quieren tocar con el acompañamiento de sus bandas favoritas y sin dejar alado a aquellos profesores de música que buscan una nueva forma de enseñar.

Tradicionalmente, la separación de pistas requería acceso a los archivos originales en formatos multipista o el uso de hardware costoso en estudios de grabación profesionales. Sin embargo, con la llegada de modelos avanzados de inteligencia artificial, como Spleeter desarrollado por Deezer (Hennequin, Khlif, Voituret, & Moussallam, 2020), ha sido posible separar elementos individuales en archivos de audio estéreo estándar con resultados de alta calidad.

Este avance ha tenido aplicaciones en múltiples áreas, como la educación musical, donde los estudiantes pueden aislar instrumentos para mejorar su comprensión y técnica; en la producción musical, para crear remixes o mejorar la calidad de grabaciones antiguas; y en la investigación de inteligencia artificial aplicada al sonido, donde se buscan nuevas técnicas para mejorar la separación de fuentes de audio.

1.2 Problemática

Uno de los principales problemas que enfrentan músicos, DJs y productores es la dificultad de acceder a pistas separadas dentro de una canción completa. Muchas veces, las grabaciones originales no están disponibles, lo que hace complejo el proceso de remix, remasterización o estudio detallado de los elementos de una pista.

El problema se intensifica con la creciente tendencia del uso de la inteligencia artificial en la producción musical, donde se busca aplicar técnicas de machine learning para remezclar o restaurar audios de manera automatizada. Sin herramientas accesibles, la manipulación de pistas de audio es un proceso costoso y técnico, limitado a estudios profesionales con equipos y software de alto nivel.

Además, la gran cantidad de archivos de audio con diferentes formatos, calidades y fuentes hace que la separación de pistas en un entorno no controlado sea un desafío. La compresión de audio, la calidad de la grabación original y el tipo de mezcla afectan significativamente la precisión de los resultados. (Jansson, y otros, 2017)

1.3 Solución Propuesta

Para abordar la problemática identificada, se plantea el desarrollo de **Melody Unmix**, una plataforma web basada en inteligencia artificial que permitirá a los usuarios cargar archivos de audio y video para la separación automática de sus pistas.

La solución se enfocará en los siguientes aspectos clave:

1. **Desarrollo con tecnologías accesibles:**
    - Implementación de Django para el desarrollo del back-end y React.js para el front-end.
    - Uso de modelos de deep learning preentrenados, como **Spleeter** de Deezer o **Demucs** para realizar la separación de fuentes de audio con alta fidelidad.
    - Procesamiento de datos en la nube para garantizar eficiencia sin comprometer la capacidad de los dispositivos del usuario final.
2. **Flujo de trabajo de la aplicación:**
    - El usuario podrá acceder a la plataforma web a través de su navegador.
    - Se permitirá la carga de archivos de audio/video.
    - El sistema ejecutará un modelo de inteligencia artificial que separará los distintos componentes del audio (voz, percusión, instrumentación, etc.).
    - Una vez finalizado el proceso, el usuario podrá descargar cada una de las pistas separadas.
3. **Interfaz y experiencia de usuario:**

- Diseño de un sistema responsive para la interfaz gráfica de la aplicacion web.
- Compatibilidad con dispositivos móviles y computadoras.
- Retroalimentación en tiempo real sobre el progreso del proceso de separación.

4. **Enfoque en privacidad y seguridad:**
    - Implementación de medidas de seguridad como cifrado de datos, autenticación segura y eliminación automática de archivos después de un tiempo determinado.
    - Cumplimiento de normativas internacionales en protección de datos.
5. **Escalabilidad y rendimiento:**
    - Uso de tecnologías en la nube para procesamiento eficiente de pistas de audio sin depender del hardware del usuario.
    - Escalabilidad del servicio para manejar una alta demanda de solicitudes.

En resumen, **Melody Unmix** busca democratizar el acceso a herramientas avanzadas de separación de pistas mediante un enfoque basado en inteligencia artificial, garantizando un entorno seguro, rápido y accesible para la comunidad musical y de investigación.

1.4 Justificación

El problema central que aborda este proyecto es la dificultad que enfrentan músicos, productores y docentes para separar los diferentes instrumentos de una pista de audio de manera precisa y accesible. Actualmente, las soluciones existentes suelen ser costosas, complejas o requieren conocimientos técnicos avanzados. La inteligencia artificial ha sido una de las mejores soluciones ante distintas actividades en los últimos años, gracias a eso ayudaría bastante el uso de modelos y técnicas implementadas con dicha tecnología. La solución propuesta consiste en desarrollar una página web que utilice inteligencia artificial y procesamiento de señales para permitir la separación eficiente de los instrumentos en una canción, proporcionando una herramienta accesible y fácil de usar.

El proyecto ofrece las siguientes contribuciones importantes:

- Originalidad: Aunque existen herramientas como Spleeter, nuestro proyecto innova al ofrecer una solución web centrada en la simplicidad y accesibilidad para el usuario final. A diferencia de las opciones más técnicas y especializadas, la plataforma integrará algoritmos optimizados para lograr una mayor precisión en la separación de instrumentos, facilitando la experiencia de los usuarios menos experimentados y permitiendo su uso sin necesidad de instalación ni configuración avanzada.
- Mejora a lo ya existente: Este proyecto no solo busca replicar las funcionalidades existentes, sino que también pretende optimizar y mejorar la separación de pistas al

enfocarse en la precisión y la rapidez de los algoritmos utilizados. Además, a diferencia de herramientas comerciales de alto costo, la plataforma será accesible desde cualquier navegador, eliminando la necesidad de software especializado y mejorando la integración de múltiples fuentes de audio. Se propone un enfoque más inclusivo, permitiendo a usuarios novatos y avanzados acceder a tecnologías de IA de forma sencilla, desde cualquier dispositivo con conexión a internet.

- Vinculación con los usuarios potenciales: Los principales beneficiarios de este proyecto son músicos, productores, educadores y aficionados a la música que desean separar y analizar canciones por instrumentos. El sistema proporcionará una herramienta útil para aquellos que buscan estudiar composiciones musicales, realizar remixes, o mejorar la calidad de grabaciones mediante la manipulación de sus componentes individuales. Asimismo, puede ser de gran utilidad en entornos educativos, donde los estudiantes podrán analizar canciones de forma más profunda y práctica.
- Complejidad: El proyecto involucra la integración de múltiples áreas de conocimiento, como inteligencia artificial, procesamiento digital de señales, y desarrollo web, lo que eleva su nivel de complejidad. Se requerirá la implementación de redes neuronales o modelos de aprendizaje profundo para realizar la separación de instrumentos de manera precisa, lo cual implica la necesidad de procesamiento intensivo y optimización de algoritmos. Además, la interfaz web debe estar diseñada para manejar grandes archivos de audio sin comprometer la experiencia del usuario, integrando procesamiento en servidores o tecnologías en la nube, lo que refleja la calidad y profundidad del proyecto, demostrando un dominio avanzado en varias disciplinas.

El proyecto es viable dentro de los tiempos y recursos disponibles para las fases de TT1 y TT2. Se estima que el desarrollo de los algoritmos y la plataforma web puede completarse en los plazos estipulados, utilizando tecnologías open-source y servidores en la nube de bajo costo para pruebas y despliegue. Los costos estarán principalmente asociados con la infraestructura web y el uso de herramientas de procesamiento de señales, pero se mantendrán bajos gracias al uso de soluciones gratuitas y de código abierto. El alcance incluye la separación de varios instrumentos principales y la creación de una interfaz web amigable y funcional, lo cual es factible dentro del tiempo y los recursos disponibles.

Este proyecto representa un aporte significativo en la mejora de herramientas ya existentes, con un enfoque en la simplicidad, accesibilidad y precisión, adaptándose tanto a usuarios profesionales como a aficionados que buscan explorar y manipular música de manera creativa y educativa.

1.5 Objetivos

Desarrollar una aplicación web que, mediante el uso de inteligencia artificial y procesamiento de señales, permita a los usuarios cargar una pista de audio y separar de manera eficiente y precisa los diferentes instrumentos que la componen, con el propósito de facilitar el análisis y estudio musical en un entorno accesible para músicos, productores, aficionados y docentes.

1.5.1 Objetivos Específicos

Convertir la señal de audio usando cálculos de frecuencias de distintos instrumentos, e ingresar esta información a una base de datos para realizar un correcto análisis.

Desarrollar un modelo computacional basado en modelos de inteligencia artificial, como lo son los modelos basados en Redes Neuronales, con el fin de capturar las relaciones complejas que se tienen en las frecuencias, así como capturar dependencias en el audio.

Desarrollar una interfaz web que permita a los usuarios cargar archivos de audio en diferentes formatos, interactuar con las pistas separadas y descargar los resultados.

Validar la precisión y eficiencia del sistema mediante pruebas de procesamiento de señales y realizar ajustes basados en métricas de funcionalidad proporcionadas por los mismos usuarios para la mejora continua.

1.6 Alcance y Limitaciones

Este proyecto tiene como objetivo el desarrollo de una aplicación web basada en modelos de inteligencia artificial y procesamiento de señales para la separación de instrumentos en pistas de audio. La aplicación permitirá a músicos, productores, estudiantes de música e investigadores analizar y manipular composiciones de forma más accesible y precisa.

Los alcances determinarán lo que se lograra, especificando las funcionalidades, características y entregables que se desarrollarán. En contraste, las limitaciones se refieren a los factores que podrían restringir el desarrollo del proyecto, tales como los recursos disponibles, el tiempo, el presupuesto, la tecnología y otros posibles obstáculos que pudieran afectar la ejecución y los resultados. Se realizará un enfoque en las funcionalidades clave que se planean implementar, los resultados deseados y los beneficios que se espera ofrecer a los usuarios, Mientras que las limitaciones abordaran las restricciones y desafíos identificados, como lo son las limitaciones técnicas o los recursos disponibles para completar el proyecto.

1.6.1 Alcance

- Carga y almacenamiento de archivos de audio: El sistema permitirá que los usuarios podrán subir pistas en formatos como MP3 y WAV, las cuales serán almacenadas en la nube.
- Procesamiento de audio con técnicas de IA: Se aplicarán algoritmos de aprendizaje automático para segmentar cada instrumento en pistas separadas.
- Interfaz web: Desarrollada con React, Bootstrap, HTML, CSS y JavaScript, permitirá gestionar archivos, visualizar la separación de instrumentos y descargar las pistas procesadas.
- Base de datos estructurada: Se implementará una base de datos SQL para metadatos y registros de procesamiento, complementada con MongoDB para datos más complejos como espectrogramas y transformadas de Fourier.

- Optimización del procesamiento de señales: Se utilizarán técnicas como FFT, STFT y MFCC para mejorar la precisión en la separación de sonidos.
- Exportación de resultados: Una vez separado el audio, los usuarios podrán descargar los archivos individuales en formatos estándar.

1.6.2 Limitaciones

- Calidad dependiente de los modelos de IA: La precisión de la separación puede verse afectada por la complejidad de la mezcla, el ruido y la calidad del modelo utilizado.
- Requerimientos computacionales: El procesamiento de señales con IA demanda un alto poder de cómputo, por lo que el sistema dependerá de infraestructura en la nube o GPUs potentes.
- Latencia en el procesamiento: La separación de pistas no será en tiempo real, el tiempo de procesamiento dependerá del tamaño y la complejidad del archivo de audio.
- Formatos de archivo limitados: Inicialmente, solo se soportarán formatos comunes como MP3 y WAV, dejando de lado formatos menos utilizados.
- Ruido y artefactos en la separación: Algunos instrumentos pueden no separarse de forma completamente limpia debido a la superposición de frecuencias en la mezcla original.
- Capacidad de almacenamiento y costos: Si el número de usuarios crece, los costos asociados al almacenamiento en la nube y al procesamiento pueden volverse un desafío.

1.7 Contribuciones

Melody Unmix contribuirá significativamente a la producción musical, el procesamiento de señales y la accesibilidad tecnológica al proporcionar una herramienta basada en inteligencia artificial que permitirá la separación de instrumentos en pistas de audio sin necesidad de software costoso o conocimientos avanzados. Facilitará en gran medida el análisis y manipulación de composiciones musicales, beneficiando a músicos, productores, investigadores y estudiantes de música al optimizar la extracción de características y mejorar la clasificación de instrumentos mediante modelos híbridos de aprendizaje automático y transformadas matemáticas.

Se fomentará la idea de normalizar el acceso a tecnologías avanzadas al ofrecer una plataforma web intuitiva, integrando bases de datos SQL y NoSQL junto con almacenamiento en la nube para gestionar grandes volúmenes de datos de manera eficiente.

La implementación de este proyecto conlleva obtener nuevas oportunidades para hacer innovaciones en distintos ámbitos, como lo es en la producción musical, la composición, la investigación del procesamiento de señales o el mismo desarrollo y entrenamiento de modelos de inteligencia artificial aplicados a la manipulación del sonido.

1.8 Metodología

Para el desarrollo de la aplicación web Melody Unmix, se empleará la metodología ágil SCRUM (SCRUM, 2024), la cual permite la adaptabilidad y mejora continua a lo largo del proyecto. SCRUM es especialmente adecuada para proyectos de software complejos, ya que facilita el desarrollo iterativo mediante entregas parciales y revisiones constantes. Esto permitirá optimizar el tiempo, recursos y calidad del producto final (ver ilustración siguiente).

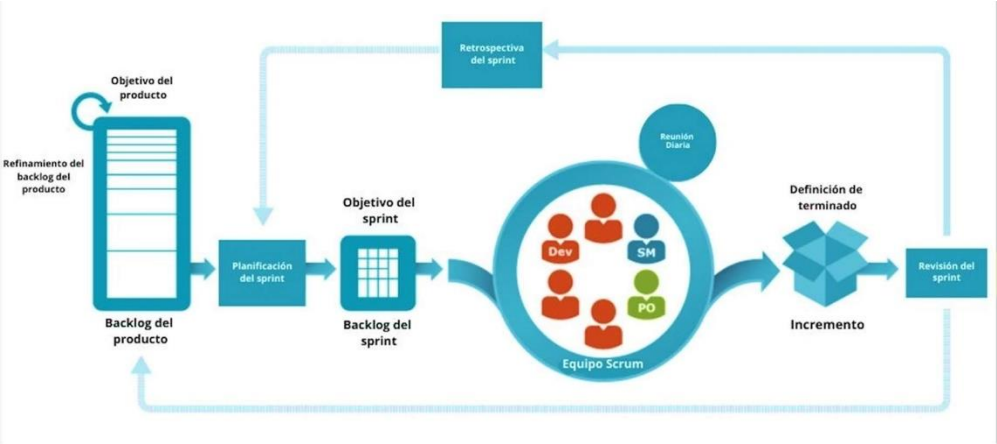


Figura #1. Proceso Metodología SCRUM

Roles en el Equipo SCRUM:

1. Product Owner (PO): Marco Antonio Jiménez Morales.
2. Scrum Master: Diego Francisco Hernández Pérez.
3. Equipo de Desarrollo: Axel Lennyn Caballero Perdomo, Diego Francisco Hernández Pérez, Marco Antonio Jiménez Morales y Raúl Martínez Pérez.

Funciones del Product Owner y Scrum Master:

- Product Owner (PO): Es el responsable de definir la visión del producto y priorizar las funcionalidades en el backlog del producto. Se encarga de la comunicación entre el equipo de desarrollo y las partes interesadas, asegurando que los requerimientos del usuario se implementen correctamente.
- Scrum Master: Actúa como facilitador del equipo, asegurándose de que se sigan las prácticas de SCRUM. Su función es eliminar impedimentos, fomentar la colaboración y mejorar continuamente el proceso de trabajo.

Cada sprint tendrá una duración de dos a tres semanas, con revisiones periódicas para evaluar el progreso y realizar mejoras. El proceso SCRUM se compone de las siguientes etapas:

1. Backlog del Producto:

- Contiene todas las características, requisitos y mejoras que se desean implementar en el producto.
- Se actualiza constantemente mediante el refinamiento del backlog del producto.

2. Planificación del Sprint:

   - Se seleccionan las tareas del backlog del producto que se trabajarán en el sprint.
   - Se definen objetivos claros y alcanzables para el sprint.

3. Backlog del Sprint:

   - Conjunto de tareas seleccionadas para ser desarrolladas dentro del sprint.
   - Se desglosan en actividades más pequeñas y manejables para el equipo de desarrollo.

4. Equipo SCRUM:

   - Compuesto por el Scrum Master, el Product Owner y los desarrolladores.
   - Participa en reuniones diarias para revisar el progreso y resolver problemas.

5. Reunión Diaria (Daily Scrum):

   - Se realiza cada día del sprint para discutir el avance, obstáculos y próximos pasos.
   - Permite detectar y resolver problemas de forma temprana.

6. Definición de Terminado:

   - Se establecen los criterios de finalización para cada funcionalidad.
   - Garantiza que el producto entregado sea funcional y cumpla con los estándares de calidad.

7. Incremento:

   - Representa el conjunto de funcionalidades completadas en un sprint.
   - Debe ser un producto potencialmente entregable y funcional.

8. Revisión del Sprint:

   - Se presentan los avances a las partes interesadas.
   - Se recoge retroalimentación para mejorar futuras iteraciones.

9. Retrospectiva del Sprint:

   - Se analiza el sprint realizado y se identifican oportunidades de mejora.
   - Se ajustan los procesos para optimizar el desempeño del equipo.

1.9 Estructura del reporte

En el **capítulo 1**, se introduce la problemática que enfrentan los músicos, productores y aficionados en la manipulación y análisis de pistas musicales debido a la falta de herramientas accesibles y eficientes. Se presentan los antecedentes y la justificación del proyecto, destacando la importancia de la inteligencia artificial en la separación de pistas. Además, se establecen los objetivos del proyecto y se delimita su alcance.

En el **capítulo 2**, se revisan los trabajos similares existentes en el ámbito de la separación de pistas musicales. Se analizan las herramientas y modelos de inteligencia artificial empleados en la industria, como Spleeter, iZotope RX y Lalal.ai, identificando sus ventajas y limitaciones. Se destaca cómo la inteligencia artificial ha transformado la producción musical, permitiendo la separación de pistas con mayor precisión y eficiencia, pero también planteando desafíos en la calidad del audio resultante y la accesibilidad de estas herramientas.

En el **capítulo 3**, se desarrolla el marco teórico del proyecto, abordando los fundamentos del procesamiento de señales en la música, la representación de la señal de audio en los dominios del tiempo y la frecuencia, y las principales técnicas de transformación y análisis espectral, como la FFT, STFT y los coeficientes MFCC. También se profundiza en la aplicación de modelos de aprendizaje profundo, incluyendo redes neuronales convolucionales (CNN), redes recurrentes (RNN) y modelos de Markov ocultos (HMM) en la separación de instrumentos. Asimismo, se discuten las tecnologías web, bases de datos y herramientas necesarias para la implementación de la aplicación.

En el **capítulo 4**, se detallan los requerimientos funcionales y no funcionales del sistema, estableciendo las funcionalidades clave, como la carga y procesamiento de archivos de audio, la visualización y descarga de pistas separadas, y la gestión de usuarios. Se analizan las herramientas y lenguajes de programación seleccionados, incluyendo Django, React.js, TensorFlow y MongoDB, evaluando su capacidad técnica y operativa. Además, se examinan los riesgos técnicos, legales y económicos, así como la viabilidad del sistema a largo plazo.

Capítulo 2. Estado del arte

En los últimos años la tecnología ha evolucionado significativamente, impulsando avances en diversos sectores, entre ellos, la industria musical. El desarrollo de nuevas herramientas y técnicas ha permitido a músicos y productores a mejores sus procesos de creación, análisis y producción de audio. La separación de instrumentos dentro de una canción sigue siendo un

desafío, y esto es debido a que requiere de conocimientos especializados y de un software elevado en cuanto a precio.

Se explorarán los fundamentos necesarios para comprender la problemática y la solución propuesta, en la cual se abordan conceptos clave relacionados con el procesamiento de señales de audio, inteligencia artificial aplicada a la música y los métodos utilizados para la separación de fuentes sonoras. Además, se revisan los métodos existentes y sus limitaciones, proporcionando el contexto necesario para el desarrollo de una herramienta accesible que facilite la manipulación y análisis de pistas de audio.

2.1 Spleeter (Deezer)

Uno de los softwares más reconocidos para la separación de pistas musicales es Spleeter, desarrollado por Deezer. Este software utiliza redes neuronales profundas para dividir una canción en diferentes *stems* o pistas de instrumentos (Hennequin & Khlif, 2020). Es un proyecto de código abierto ampliamente adoptado en la comunidad musical. Spleeter ofrece modelos preentrenados que permiten la separación de voces y acompañamiento, además de opciones para dividir las pistas en cuatro o cinco componentes, incluyendo voces, batería, bajo, piano y otros sonidos. Se puede utilizar como una biblioteca de Python o como una herramienta independiente desde la línea de comandos (Ubunlog, 2024).

Spleeter Deezer es un motor de separación de fuentes de audio basado en inteligencia artificial desarrollado por el equipo de I+D de Deezer, diseñado específicamente para separar la voz de los sonidos de los instrumentos musicales de una canción. Sin embargo, es posible que todavía estés confundido acerca de cómo usar Spleeter, aunque está diseñado como una herramienta profesional de separación de pistas, la experiencia de usuario poco amigable y la característica difícil de instalar han bloqueado tu interés en investigar.

Spleeter es una herramienta genial que puede separar diferentes pistas de un único archivo de audio, como canciones, podcasts, películas o juegos. Puede extraer mágicamente las voces y los instrumentos de cualquier música o fuente de audio y separarlos del resto del sonido.

Después de la depuración y el entrenamiento previo, Spleeter Deezer creará tres modos diferentes para que Deezer le ayude a separar el audio en consecuencia.

• 2 tallos: Voz + todos los instrumentos

• 4 tallos: Voz + Batería + Bajo + Otros

• 5 tallos: Voz + Batería + Bajo + Piano + Otros

2.2 iZotope RX

Es un software profesional utilizado en la postproducción de sonido, que incluye herramientas avanzadas para la restauración de audio y la separación de *stems*. Emplea algoritmos sofisticados para procesar y limpiar el audio, permitiendo la eliminación de ruido,

la reparación de diálogos y la restauración de grabaciones antiguas o deterioradas. Su capacidad para la edición detallada y la separación de elementos individuales en una pista lo convierte en una solución poderosa para la industria del cine, la televisión y la música profesional. Aunque es conocido por su alta calidad y precisión, su costo elevado lo sitúa principalmente en el ámbito profesional (iZotope, 2024).

Con tecnología de aprendizaje automático, el conjunto completo de herramientas de RX aborda todo, desde los problemas de audio más comunes hasta los rescates sonoros más complicados, para música, posproducción de audio y creación de contenido.

RX está disponible como una aplicación de edición de audio independiente que incluye un conjunto de complementos de software para usar con estaciones de trabajo de audio digital (DAW).

Elimina en tiempo real el ruido y la reverberación de diálogos o voces con la tecnología puntera de aprendizaje automático del nuevo Dialogue Isolate. Monta fácilmente declaraciones compuestas por varios fragmentos con el contorno de diálogo mejorado, independientemente de cuántas tomas tengas que integrar para una lectura perfecta.

Figura #2. Interfaz de iZotopee

2.3 Lalal.ai

Es un servicio web que permite a los usuarios cargar pistas de audio y separarlas en componentes como voces e instrumentos, utilizando redes neuronales para lograr una separación precisa. A través de su interfaz intuitiva, los usuarios pueden aislar elementos como batería, bajo, piano, guitarra eléctrica, guitarra acústica y sintetizador, sin pérdida significativa de calidad. Lalal.ai es accesible para usuarios sin conocimientos técnicos, eliminando la necesidad de instalación de software y ofreciendo una opción rápida y eficiente para la separación de pistas musicales. Su enfoque en la simplicidad y su precisión lo han convertido en una herramienta popular entre músicos y creadores de contenido (Lalal.ai, 2024).

Lalal.ai se creó con el objetivo de facilitar el trabajo con audio y video para músicos, productores de sonido, ingenieros musicales, blogueros de video, transmisores, transcriptores, traductores, periodistas y muchos otros profesionales y creativos.

En 2020 se trabajó en una red neuronal única llamada **Rocknet** con el uso de 20TB de datos entrenados para extraer pistas instrumentales y de voz de canciones. En 2021 se creó Cassiopea, la solución de próxima generación superior a Rocknet que proporcionaba mejores resultados de división con mucho menos artefactos audios.

Empezando como un servicio de separación de dos stems, LALAL.AI creció significativamente en 2021. Además de vocal e instrumental, el servicio fue mejorado con la capacidad de extraer instrumentos musicales - baterías, bajo, guitarra acústica, guitarra eléctrica, piano, y sintetizador. Como el resultado de esta actualización, LALAL.AI convertió en el primer separador de 8 stems en el mundo.

Figura #3: Lalal.AI

2.4 Music Information Retrieval (MIR) Community

Esta comunidad de investigación se dedica a mejorar la manera en que las máquinas procesan y entienden la música. Sus contribuciones han sido fundamentales para el desarrollo de aplicaciones de análisis musical, incluyendo la clasificación de géneros, la recomendación de canciones y la separación de instrumentos. Estas innovaciones han facilitado la comprensión y manipulación de la música mediante sistemas automatizados (ISMIR, 2024).

La **recuperación de información musical** (MIR) es un área de investigación interdisciplinaria en rápido crecimiento que abarca diversas disciplinas, como la informática, la recuperación de información, la musicología, la teoría musical, la ingeniería de audio, el

procesamiento de señales digitales, la ciencia cognitiva, la bibliotecología, la edición y el derecho.

Su objetivo principal es desarrollar métodos para gestionar colecciones de material musical con fines de conservación, acceso, investigación y otros usos. En este sentido, la MIR guarda similitudes con la bibliotecología tradicional, ya que las bibliotecas han desempeñado un papel fundamental en el desarrollo de colecciones musicales.

El interés por aplicar técnicas de recuperación automática de información (IR) a la música se remonta a la década de 1960. Sin embargo, la MIR ha cobrado mayor relevancia en los últimos años debido al auge de las colecciones digitales de música en red. Factores como el desarrollo de tecnologías de compresión, como el MP3, la aparición de servicios en línea como Napster, los avances en el reconocimiento óptico de música (OMR) y la reducción de costos en almacenamiento digital y ancho de banda han impulsado su crecimiento.

En este contexto, la MIR está estrechamente relacionada con las bibliotecas digitales, ya que ambas buscan optimizar el acceso y la organización de grandes volúmenes de contenido musical en formatos digitales.

Los investigadores en **Recuperación de Información Musical** (MIR) buscan desarrollar nuevas técnicas para gestionar el creciente volumen de música digital disponible. Su motivación principal radica en la necesidad de mejorar los métodos de recuperación de información ante la expansión acelerada de colecciones musicales en formato digital (Durey et al., 2001; Hoos et al., 2001; Kornstädt, 2001; Yang, 2001).

A pesar de este interés, se ha dedicado poco esfuerzo a evaluar con precisión el volumen de música digital disponible, su tasa de crecimiento o su comparación con factores como el costo del ancho de banda, el almacenamiento y la potencia de procesamiento. Asimismo, investigaciones enfocadas en la escalabilidad de las técnicas existentes, como la de Jang et al. (2001), siguen siendo relativamente escasas.

Esto sugiere que el enfoque de la MIR no se limita únicamente a la gestión del creciente volumen de datos, sino que también busca abordar problemas más amplios y fundamentales en el acceso, organización y análisis de la música digital.

2.5 Tesis “Procesamiento de Audio Digital para la Clasificación de Sonidos Urbanos” (ITESO, 2023)

Este proyecto del ITESO busca clasificar sonidos ambientales en entornos urbanos utilizando inteligencia artificial. Para ello, emplea técnicas como los coeficientes cepstrales de Mel (*Mel-Frequency Cepstral Coefficients* o MFCC) para convertir el audio en representaciones numéricas, y una red neuronal para identificar sonidos como motores y ruido de construcción. El modelo se entrena con el conjunto de datos *UrbanSound8k*, mejorando la precisión en la

clasificación mediante estrategias como *data augmentation* y técnicas de regularización como *Dropout*.

Capítulo 3. Marco Teórico

La música es uno de los aspectos más importantes en la vida cotidiana y ha evolucionado a la par de los avances tecnológicos. Dentro de este desarrollo, una de las principales problemáticas en el ámbito musical es la dificultad de identificar y separar los distintos instrumentos dentro de una composición. Esta limitación afecta tanto a músicos y productores como a investigadores que buscan analizar estructuras sonoras con mayor precisión. En la actualidad, las soluciones disponibles requieren herramientas especializadas con un costo elevado además de conocimientos avanzados en procesamiento de señales, lo que limita su accesibilidad para distintas personas.

Ante esta problemática, este proyecto propone el desarrollo de una herramienta basada en inteligencia artificial (IA) y aprendizaje automático (ML) para la separación de instrumentos en pistas de audio. Mediante el uso de modelos de aprendizaje automático y procesamiento de señales, se busca ofrecer una solución accesible que permita a músicos y productores realizar un análisis más profundo y manipular el contenido de una canción de manera precisa y eficiente.

El presente marco teórico aborda los fundamentos que sustentan esta propuesta, desde la estructura y características de los instrumentos musicales hasta las tecnologías y técnicas de procesamiento de señales que permitirán su implementación.

3.1 Procesamiento de señales en la música.

El procesamiento de señales es una disciplina que permite la manipulación y análisis de datos sonoros con el objetivo de obtener información relevante sobre un sonido en específico. En el contexto musical, esto implica el análisis de frecuencias, amplitudes solo por mencionar algunas características de las ondas sonoras que conforman una pista musical.

Cada instrumento emite sonidos con un rango de frecuencias característico. Por ejemplo, el rango de un piano varía aproximadamente entre 27.5 Hz y 4,186 Hz, mientras que la voz humana se encuentra entre 85 Hz y 255 Hz, dependiendo del género y tipo de voz. La capacidad de identificar y separar estos sonidos es fundamental para la segmentación y clasificación de pistas de audio, tarea que se vuelve compleja debido a la superposición de señales que se encuentran en una mezcla musical.

En el contexto de este proyecto, se utilizarán diversas técnicas para segmentar, filtrar y extraer características de las pistas musicales con el objetivo de identificar y separar los distintos instrumentos.

3.1.1 Representación de la Señal de Audio

- **Dominio del tiempo:** La señal se representa como una función de la amplitud en función del tiempo. Es la forma en que las ondas de sonido se capturan y reproducen en sistemas de grabación y reproducción.
- **Dominio de la frecuencia:** Se obtiene al aplicar la Transformada de Fourier, lo que permite analizar los componentes frecuenciales de la señal. Esto es útil para identificar las características espectrales de cada instrumento.

3.1.2 Adquisición y Preprocesamiento de la Señal

Antes de aplicar cualquier técnica avanzada de procesamiento de señales, es necesario adquirir la señal de audio y prepararla para su análisis.

- **Conversión a un formato de trabajo**: Los archivos de audio suelen estar en formatos como WAV, MP3 o FLAC. Se deben convertir a una estructura de datos que facilite su procesamiento con Python.
- **Muestreo y cuantización**: La señal de audio analógica se convierte en digital mediante la frecuencia de muestreo y la resolución en bits.
- **Normalización**: Se ajustan los valores de amplitud de la señal para evitar saturaciones o pérdidas de información.

3.1.3 Transformación y Análisis de Frecuencia

El sonido se compone de diferentes frecuencias que varían en el tiempo. Para analizar y separar estos componentes, se utilizan transformadas matemáticas como:

Transformada de Fourier (FFT - Fast Fourier Transform)

Permite descomponer una señal en sus componentes de frecuencia, facilitando la identificación de los instrumentos según sus rangos espectrales.

Transformada de Short-Time Fourier Transform (STFT)

La FFT analiza la señal en el dominio de la frecuencia, pero no proporciona información sobre cómo varían esas frecuencias en el tiempo. Para ello, se usa la **STFT**, que divide la señal en pequeñas ventanas temporales y aplica FFT a cada una.

Transformada Wavelet

A diferencia de la STFT, que usa ventanas de tamaño fijo, la **Wavelet Transform** adapta la resolución para analizar mejor señales con variaciones rápidas o lentas.

3.1.4 Filtrado y Extracción de Características

Para separar instrumentos, es necesario eliminar frecuencias no deseadas y extraer características clave:

Filtrado de Frecuencias

Los filtros ayudan a aislar las frecuencias características de cada instrumento.

- **Filtro pasa bajos (Low-pass):** Permite frecuencias bajas (ej. bajos y bombos).
- **Filtro pasa altos (High-pass):** Permite frecuencias altas (ej. platillos, voces).
- **Filtro pasa banda (Band-pass):** Aísla un rango específico de frecuencias.

Extracción de Características

- *MFCCs (Mel-Frequency Cepstral Coefficients):* Simulan la percepción del oído humano y ayudan en la clasificación de instrumentos y voces.
- *Zero Crossing Rate (ZCR):* Indica la cantidad de veces que la señal cruza el eje cero, útil para distinguir sonidos suaves de sonidos más percusivos.

3.1.5 Post-Procesamiento y Generación de Pistas Separadas

Una vez que los modelos han identificado y separado los instrumentos, es necesario realizar un post-procesamiento para mejorar la calidad de las pistas individuales. Algunas de las técnicas utilizadas incluyen:

- Normalización de Audio: Ajuste de niveles de volumen para mantener una calidad de sonido homogénea.
- Interpolación y Relleno de Espacios Perdidos: Aplicación de técnicas para recuperar partes de la señal que pudieron haber sido eliminadas durante el filtrado.
- Reducción de Artefactos: Eliminación de ruidos y errores generados por el procesamiento de la señal.

Una vez implementando dichas herramientas para lograr el procesamiento de señales, podemos iniciar con la parte de la inteligencia artificial del sistema

3.2 Inteligencia Artificial y Aprendizaje Automático Aplicado a la Música

El uso de inteligencia artificial en el ámbito musical ha permitido avances significativos en la identificación y manipulación de pistas de audio. Una vez extraídas las características relevantes, y con la señal transformada se pueden emplear modelos de separación basados en inteligencia artificial.

En conjunto con el aprendizaje automático, también tenemos el aprendizaje supervisado, la cual es una técnica ampliamente utilizada que se basa en el entrenamiento de modelos mediante conjuntos de datos etiquetados. En este enfoque, cada dato de entrada está asociado a una etiqueta que representa su salida esperada. El objetivo principal es que el modelo aprenda una función que relacione las características de los datos con sus respectivas etiquetas, lo que le permite hacer predicciones precisas en nuevos conjuntos de datos.

3.2.1 Separación Basada en Redes Neuronales

- Redes Neuronales Convolucionales (CNNs): Utilizadas para analizar espectrogramas y aprender patrones asociados a diferentes instrumentos.
- Redes Recurrentes (RNNs) y LSTM: Capturan la evolución temporal de los sonidos en una pista musical.
- Modelos de Mezcla Latente (Latent Source Separation Models): Aprenden representaciones ocultas de los sonidos individuales en una mezcla.

3.2.2 Métodos Basados en Probabilidad y Estadística

Algoritmos como **Hidden Markov Models (HMMs)** y **Naïve Bayes** pueden ser utilizados para modelar la probabilidad de que una determinada frecuencia pertenezca a un instrumento en particular.

Para este proyecto, se utilizarán bibliotecas de Python como TensorFlow, PyTorch y Scikit-learn, las cuales facilitan la implementación de modelos predictivos para el procesamiento y segmentación de audio. Estas herramientas permitirán entrenar modelos que identifiquen patrones en las señales de sonido y realicen una separación precisa de los distintos instrumentos en una pista musical.

3.2.3 Herramientas de Python

Para este proyecto, se utilizarán diversas bibliotecas de Python:

- **TensorFlow y PyTorch:** Frameworks de aprendizaje profundo para el entrenamiento de modelos de separación de audio.
- **Scikit-learn:** Librería para implementar modelos de aprendizaje automático clásicos.
- **Librosa y SciPy:** Herramientas para la lectura y procesamiento de archivos de audio.

3.3 Tecnologías Web

El desarrollo de la interfaz de usuario jugará un papel clave en la accesibilidad del sistema. Para ello, se emplearán tecnologías web como React, Bootstrap, HTML, CSS y JavaScript, con el objetivo de ofrecer una plataforma intuitiva y fácil de usar para los usuarios. La interfaz permitirá cargar archivos de audio, visualizar la separación de instrumentos y exportar los resultados de manera sencilla.

Para facilitar la accesibilidad del sistema, se desarrollará una interfaz web intuitiva utilizando las siguientes tecnologías:

- **React:** Framework de JavaScript para el desarrollo de interfaces dinámicas.

- **Bootstrap, HTML y CSS:** Herramientas para el diseño y estructura visual de la plataforma.
- **JavaScript:** Lenguaje de programación para la interacción del usuario con la aplicación.

La interfaz permitirá a los usuarios cargar archivos de audio, visualizar la separación de instrumentos y exportar los resultados de manera sencilla.

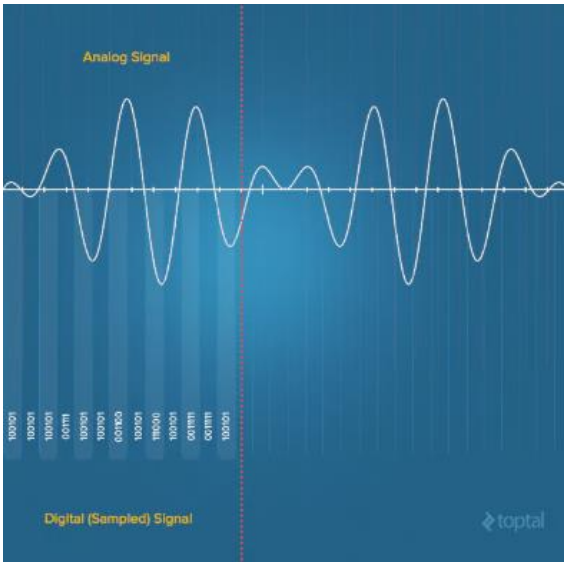
3.4 Bases de Datos para la Gestión de Información

El almacenamiento de datos en este proyecto será un aspecto fundamental para la gestión de archivos de audio, modelos entrenados y resultados generados. Se evaluará el uso de bases de datos relacionales y no relacionales para determinar cuál es la mejor opción en términos de rendimiento y escalabilidad. Entre las opciones consideradas se encuentran PostgreSQL, MySQL y MongoDB, dependiendo de los requerimientos específicos del sistema.

3.5 Algoritmos de Reconocimiento Musical

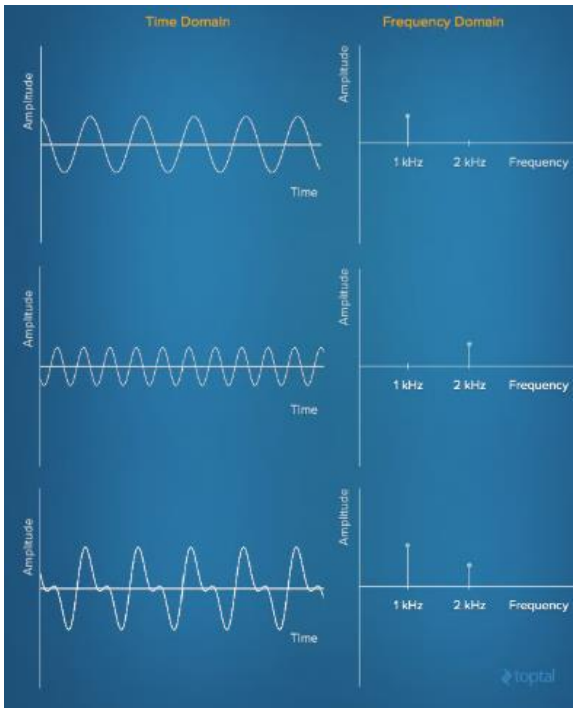
3.5.1 Shazam

El sonido es una vibración que se propaga como una ola mecánica de presión y se desplaza a través de un medio tal como el aire o el agua. Cuando esta vibración llega a nuestros oídos, particularmente en el tímpano, mueve pequeños huesos que transmiten las vibraciones a las células ciliadas bastante profundas en nuestro oído interno. Además, las pocas células ciliadas producen impulsos eléctricos que se transmiten a nuestro cerebro a través del nervio auditivo del oído.



Los dispositivos de grabación imitan este proceso mediante la presión de la onda de sonido para convertirla en una señal eléctrica. Una onda de sonido real en el aire es una señal continua de presión. En un micrófono, el primer componente eléctrico al encontrar esta señal se traduce en una señal de tensión analógica nuevamente, continua. Esta señal continua no es tan útil en el mundo digital, así que antes de ser procesada, debe ser traducida en una señal discreta que se pueda almacenar digitalmente. Esto se realiza mediante la captura de un valor digital que representa la amplitud de la señal.

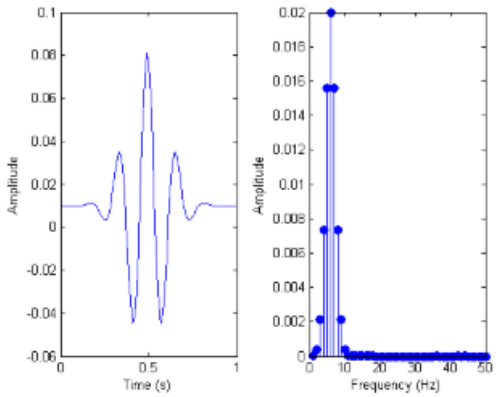
En particular, para captar todas las frecuencias que un ser humano pueda oír en una señal de audio, debemos muestrear la señal a una frecuencia doble que la del rango auditivo humano. El oído humano puede detectar frecuencias aproximadamente entre 20 Hz y 20.000 Hz. Como resultado, la mayoría de las veces el audio grabado tiene una velocidad de muestreo de 44.100 Hz. Esta es la frecuencia de muestreo de Compact Discs discos compactos y también es la más utilizada con MPEG-1 audio (VCD,SVCD, MP3).



es posible representar cualquier señal en el dominio del tiempo simplemente dando el conjunto de frecuencias, amplitudes y fases correspondientes a cada sinusoide que compone la señal. Esta representación de la señal es conocida como la frecuencia de dominio. En cierto modo, el dominio de la frecuencia actúa como un tipo de huella dactilar o firma para la señal de dominio de tiempo, proporcionando una representación estática de una señal dinámica.

Por lo tanto, necesitamos encontrar una manera de convertir nuestra señal desde el momento de dominio para el dominio de la frecuencia. Aquí lo llamamos la transformación Discreta de Fourier (DFT) para ayudar. La DFT es un método matemático para realizar Análisis de Fourier en un discreto (muestra) de la señal. Convierte una lista finita de muestras equidistantes de una función en la lista de los coeficientes de una combinación finita de complejas sinusoides, ordenadas por sus frecuencias, considerando si las sinusoides habían sido muestreadas en la misma proporción.

Señal antes y después de FFT:



Un desafortunado efecto colateral de FFT es que perdemos una gran cantidad de información acerca de la sincronización (aunque teóricamente esto puede ser evitado, el rendimiento de gastos generales es enorme) para una canción de 3 minutos, podemos ver todas las frecuencias y sus magnitudes, pero no tenemos una pista cuando aparecieron en la canción.

Es por eso que introducimos el tipo de ventana deslizante, o fragmento de datos, así como la transformación de esta parte de la información. El tamaño de cada fragmento puede determinarse de distintas formas. Por ejemplo, si queremos grabar el sonido en estéreo, con muestras de 16 bits, a 44.100 Hz, un segundo de ese sonido será de 44.100 muestras * 2 bytes * 2 canales ≈ 176 kB.

En el bucle interior estamos poniendo los datos de dominio de tiempo (las muestras) en un número complejo con parte imaginaria 0. En el bucle exterior, nos iteramos a través de todos los segmentos y realizamos un análisis FFT de cada uno.

Una vez que tengamos la información acerca de la frecuencia de la señal, podemos empezar a formar nuestra huella digital de la canción. Esta es la parte más importante de todo el proceso de reconocimiento de música de Shazam. El principal desafío es cómo distinguir, en el océano de las frecuencias capturadas, las frecuencias que son las más importantes. Intuitivamente, podemos buscar las frecuencias con mayor magnitud (comúnmente llamadas picos).

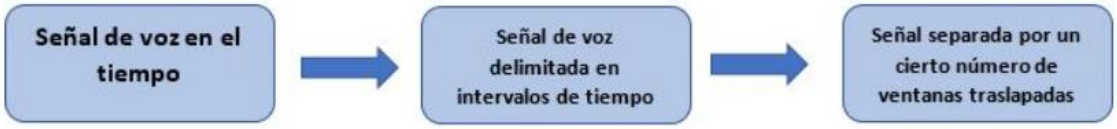
Ten en cuenta que debemos asumir que la grabación no está hecha en perfectas condiciones (por ejemplo, una “sala de sordos”) y como resultado se debe incluir un factor de aproximación. Un análisis del factor de aproximación debe ser tomado en serio y en un sistema real, el programa debe tener una opción para establecer este parámetro en función a las condiciones de la grabación.

Para facilitar la búsqueda, esta firma se convierte en la clave en una tabla de hashtag. El valor correspondiente es el tiempo en el que este conjunto de frecuencias apareció en la canción, junto con el ID de la canción (título de la canción y del artista).

3.5.2 Mel Ceptral Frecuency Coefficients MFSS

Los coeficientes obtenidos a partir de un proceso de filtrado conocido como Mel-Cepstral, son un conjunto de valores numéricos que resumen la información básica de las características que constituyen una señal de voz (Holmes & Holmes, 2001). El procedimiento para obtenerlos está basado en dos conceptos: El rango de frecuencias Mel y la separación de frecuencias por medio de Cepstrum.

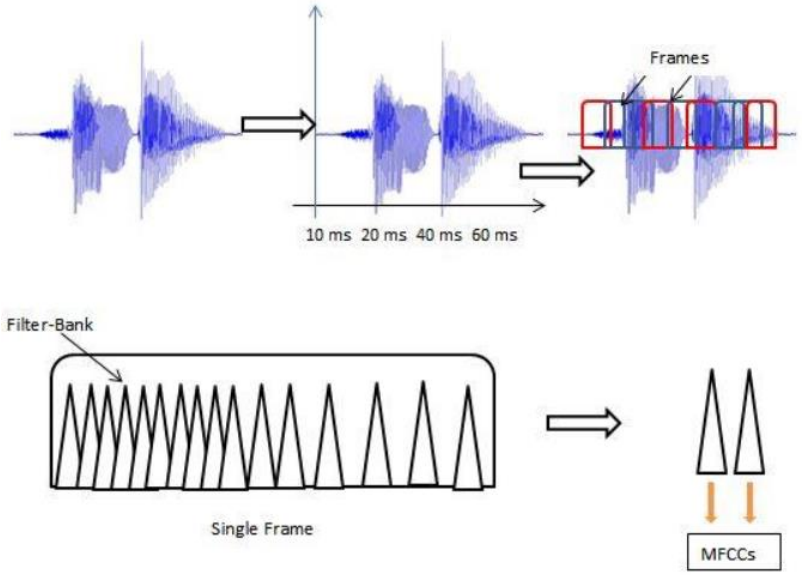
El rango de frecuencias Mel está basado en la reducción de frecuencias de la señal de voz teniendo como referencia el rango auditivo humano, es decir, aquellas frecuencias que se



pueden percibir más fácilmente. Por otro lado, Cepstrum es un concepto matemático que separa de la señal de voz en dos bandas de frecuencias baja y alta. La baja corresponde a los formantes de los fonemas producidos debido a las cavidades del tracto vocal y la banda alta es relativa a la excitación en las cuerdas vocales. Esta última es una señal periódica muy particular a los distintos fonemas independientemente de las variaciones en el tracto vocal.

1. Se hace pre-énfasis a la señal de voz, es decir se amplifican las altas frecuencias para facilitar el cálculo de las formantes con amplio contenido en el espectro alto.
2. Se aplica una ventana Hamming para obtener la frecuencia promedio en diferentes tramas o Generalmente se aplica una ventana de 20 ms a intervalos de 10 ms.
3. Se obtiene la DFT de cada frame.
4. Se aplica un banco de filtros a cada frame. De acuerdo con Davis y Mermelstein (Davis & Mermelstein, 1978) los filtros se distribuyen de manera no lineal de acuerdo a la escala Mel. Normalmente se utilizan 20 filtros. Los primeros 10 están linealmente distribuidos y los siguientes 10 crecen en forma logarítmica.
5. Se aplica la transformada Coseno Discrete, la cual es una variante de la FFT a la salida de cada filtro. Normalmente se obtienen de 10 a 12 coeficientes MFCC, pero el número es modificable por el usuario.

Los MFCC son una manera compacta de almacenar sonido. No son otra cosa más que números que revelan las diferentes amplitudes de la señal, pero no tienen en sí mismos energía acústica codificada.



3.5.3 Modelos de Markov Ocultos

Los modelos de Markov ocultos (HMM) son un tipo de modelo estadístico utilizado en el aprendizaje automático para describir sistemas que evolucionan con el tiempo.

- **Reconocimiento del habla**: Una de las aplicaciones más clásicas y exitosas de los HMM es en los sistemas de reconocimiento del habla. En el habla, las señales acústicas (observaciones) son generadas por la secuencia de fonemas o palabras pronunciadas (estados ocultos). Los HMM se utilizan para modelar las relaciones probabilísticas entre los fonemas y las características acústicas, lo que permite a los sistemas transcribir el lenguaje hablado en texto. Los sistemas modernos de reconocimiento del habla suelen utilizar modelos de aprendizaje profundo más complejos, pero los HMM desempeñaron un papel fundacional en este campo, y aún se utilizan en enfoques híbridos.
- Procesamiento del Lenguaje Natural (PLN): Para tareas como el etiquetado de partes del discurso, donde las palabras de una frase son observaciones y las etiquetas gramaticales subyacentes son estados ocultos. Puedes explorar más sobre el Procesamiento del Lenguaje Natural (PLN) y sus diversas aplicaciones en la IA.

El estado oculto actual sólo depende del estado oculto anterior, no de todo el historial de estados. Esta propiedad "sin memoria" simplifica el modelo y hace factible el cálculo. Por ejemplo, en la predicción meteorológica mediante un HMM, el tiempo de hoy (estado oculto) sólo depende del tiempo de ayer, no del tiempo de hace una semana.

La observación actual sólo depende del estado oculto actual, y es independiente de los estados ocultos pasados y de las observaciones pasadas dado el estado oculto actual. Siguiendo con

el ejemplo del tiempo, que veas llover hoy (observación) sólo depende del estado del tiempo de hoy (estado oculto, por ejemplo, "lluvioso", "soleado"), y no del estado del tiempo de ayer.

Capítulo 4. Análisis

4.1 Reglas de negocio

Las reglas de negocio en Melody Unmix establecen normas y restricciones bajo las cuales la aplicación web debe operar. Estas reglas garantizan la funcionalidad, protegen la integridad de los datos y mejoran la experiencia del usuario. Dentro de los cuales engloban cuatro parámetros esenciales:

Procesamiento de audio

- La aplicación web solo aceptará archivos de audio en formatos MP3 y WAV.
- Los archivos de audio no deben superar los 50 MB de tamaño.
- El tiempo de procesamiento de cada archivo debe estar entre 2 y 10 minutos, dependiendo de la carga del sistema.
- Cada usuario solo podrá procesar una pista a la vez.
- Los archivos procesados se almacenarán temporalmente por 30 días. Si el usuario no los descarga en ese periodo, serán eliminados automáticamente.

Gestión de usuarios

- El usuario debe registrarse con un correo electrónico válido y una contraseña segura.
- Para iniciar sesión, se validarán las credenciales mediante autenticación segura.
- Los usuarios pueden ver un historial de sus archivos procesados, pero no podrán acceder a archivos de otros usuarios.
- La plataforma permitirá la eliminación de cuenta bajo solicitud del usuario.
- Los usuarios podrán eliminar el procesamiento de una pista de audio de su historial.

Uso del servicio

- No existe una restricción que limite a los usuarios a la cantidad de pistas que desee procesar en un dia pero la disponibilidad está sujeta al tráfico mismo de la aplicación.
- Si el sistema detecta uso abusivo (como intentos automatizados de procesamiento, bots), la cuenta podrá ser suspendida temporal o indefinidamente.
- Los archivos subidos deben ser archivos de audio legítimos y no contener datos maliciosos.

Seguridad y privacidad

- Todos los archivos se procesan de manera segura y se eliminan automáticamente después del tiempo establecido.
- Los datos personales del usuario serán tratados de acuerdo con la política de privacidad de Melody Unmix.
- La plataforma implementará medidas de detección de malware para evitar la carga de archivos maliciosos.
- Los datos de usuario estarán cifrados y no podrán ser compartidos con terceros sin consentimiento.

Responsabilidad del usuario

- El usuario es responsable del contenido que sube a la plataforma y debe asegurarse de cumplir con las leyes de derechos de autor.
- Melody Unmix no se hace responsable por el uso indebido de los archivos generados por el sistema.
- En caso de detectar actividades que infrinjan estas políticas, el usuario será notificado y podría perder el acceso al servicio.

Modificaciones y actualizaciones de políticas

- Estas políticas pueden ser modificadas en cualquier momento para mejorar la seguridad y la experiencia de usuario.
- Los usuarios serán notificados de cambios importantes en las políticas de uso a través del sitio web o correo electrónico.

4.2 Requerimientos

4.2.1 Requerimientos funcionales

RF1. Procesamiento de Audio

El sistema debe permitir la carga de archivos de audio en formatos estándar como MP3 y WAV. Una vez subido el archivo, este debe procesarse en el servidor utilizando Spleeter y TensorFlow para separar las pistas individuales (voz, guitarra y bajo).

Las pistas generadas deben almacenarse temporalmente en el servidor, con una política de eliminación después de un tiempo determinado o una vez descargadas por el usuario.

RF2. Interfaz de Usuario (Front-end)

La interfaz debe permitir a los usuarios visualizar el estado del procesamiento de su canción mediante una barra de progreso o notificación en tiempo real cumpliendo con las prácticas de “design responsive”. Así como el manejo de errores en cualquier proceso del mismo.

Debe permitir la subida de archivos de manera óptima y evitando tecnicismos que puedan ser obstáculos para aquellas personas que no tengan conocimientos del área de sistemas. Una vez finalizado el procesamiento, el usuario debe poder descargar individualmente cada pista separada. Cabe recalcar que la aplicacion web rechazara cualquier archivo que no sea en el formato WAV y MP3.

RF3. Gestión de Usuarios

El sistema debe permitir registrar usuarios con datos básicos como nombre, correo electrónico y contraseña. Los usuarios deben poder iniciar sesión y cerrar sesión de manera segura. Se debe validar la autenticidad de los datos de acceso mediante un sistema de autenticación en Django.

El sistema debe almacenar y permitir a los usuarios consultar su historial de pistas procesadas, incluyendo la fecha de cada operación.

Los usuarios deben tener la opción de eliminar archivos de su historial para mantener la privacidad y liberar espacio.

RF4. Comunicación entre Front-end y Back-end

La comunicación entre el cliente (interfaz web) y el servidor debe realizarse mediante APIs REST desarrolladas con Django Rest Framework (DRF).

- Se deben implementar endpoints para:
- Subir archivos de audio.
- Consultar el estado del procesamiento.
- Descargar archivos procesados.
- Gestionar usuarios (registro, login, historial, eliminación de archivos).

- Las respuestas de las APIs deben estar optimizadas para un rendimiento eficiente y un menor consumo de ancho de banda.

4.2.2 Requerimientos no funcionales

RNF1. Rendimiento

La aplicación web debe ser capaz de realizar la separación de las pistas en un tiempo no máximo a 10 minutos, así como de soportar múltiples procesamientos de otros usuarios concurrentes sin verse afectado.

RNF2. Compatibilidad

La aplicación web debe funcionar en diferentes navegadores como son Google Chrome, Edge, Mozilla Firefox, Safari. Asi mismo, también deberá ser capaz de visualizarse y ejecutarse de forma correcta si la abrimos desde un móvil.

RNF3. Documentación

El código de la aplicación web debe estar documentado siguiendo buenas prácticas de programación con el fin de facilitar que se le pueda brindar actualización y mantenimiento.

RNF4. Seguridad

La aplicación debe encriptar las credenciales de cada uno de los usuarios. Asegurarse que los archivos que se carguen y almacenen no sean maliciosos, y al almacenarse estos deben estar seguros.

RNF5. Disponibilidad

La aplicación debe estar disponible el 99% del tiempo al mes, únicamente estará no disponible cuando se encuentre en actualización y mantenimiento. El tiempo que no estará disponible puede ser de hasta 4 horas máximo.

RNF6. Fiabilidad

La aplicación debe garantizar que el 99% de las solicitudes de separación realizadas se completen de forma exitosa.

4.3 Análisis de las herramientas a utilizar

4.3.1 Análisis de lenguajes y entornos de programación

4.4 Análisis de riesgos

Capitulo 5. Diseño

Bibliografía

https://www.deezer-techservices.com/solutions/spleeter/

https://www.tunefab.com/es/deezer/how-to-use-spleeter.html

https://www.izotope.com/en/products/rx.html

https://www.native-instruments.com/es/products/izotope/rx-11-advanced/?srsltid=AfmBOooEEpj9tcWjl4sh2vRLmKmg1sY86NkkGwnyRT7AuWyw0ySTwe38

https://www.lalal.ai/es/about/

https://ismir2002.ircam.fr/proceedings/02-FP07-3.pdf

Referencias de marco teórico:

Referencias de algoritmos de reconocimiento

Shazam: https://www.toptal.com/algorithms/shazam-reconocimiento-de-algoritmos-de-musica-huellas-dactilares-y-procesamiento

MFFC: https://francocarlos.com/2017/05/04/mel-cepstral-frequency-coefficients-mfcc/#:~:text=Los%20coeficientes%20obtenidos%20a%20partir,Holmes%20%26%20Holmes%2C%202001).

Markov: https://www.ultralytics.com/es/glossary/hidden-markov-model-hmm