# Grabadora de Voz Inteligente

Juan José Muñoz Noda, Ovido Navarro Pazos y Jesús Armando Padrón

October 4, 2024

**Abstract**

En este reporte se presenta el desarrollo de una grabadora inteligente capaz de procesar la voz, transcribir el contenido hablado y detectar las emociones de los hablantes utilizando varios modelos preentrenados de aprendizaje automático. La motivación principal del proyecto es mejorar la accesibilidad y el análisis de las interacciones verbales en diversos contextos. Se discuten los modelos utilizados, la arquitectura del sistema, los experimentos realizados y los resultados obtenidos. Además, se abordan las implicaciones éticas del uso de esta tecnología y se sugieren futuras mejoras y aplicaciones potenciales.

# Contents

# Chapter 1

# Introducción

El trabajo con sonido siempre ha sido una tarea desafiante para la computación. Desde los primeros días de la humanidad, el sonido ha jugado un papel crucial en el desarrollo de la cultura, la música y diversos aspectos de la vida en sociedad. Con el avance progresivo de la computación y el desarrollo de hardware más potente, las técnicas de procesamiento de sonido han evolucionado significativamente, logrando resultados cada vez más complejos y precisos.

La computación, desde sus inicios, ha buscado maneras de capturar, analizar y manipular el sonido de formas innovadoras. Inicialmente, estos esfuerzos se centraron en aplicaciones simples, pero con el tiempo, las capacidades tecnológicas han permitido abordar problemas mucho más sofisticados. La mejora en los algoritmos de procesamiento de señal y el incremento en la capacidad de almacenamiento y procesamiento han sido fundamentales para estos avances.

Paralelamente, el desarrollo de la inteligencia artificial (IA) y el aprendizaje automático (ML) ha abierto nuevas fronteras en el procesamiento de sonido. Estas tecnologías han permitido la creación de modelos que pueden aprender y adaptarse a patrones complejos en los datos de audio, facilitando tareas que antes eran inalcanzables.

## 1.1 Motivación

Gracias a la evolución de las redes neuronales y los modelos de aprendizaje automático (ML), ha surgido la idea de combinar múltiples modelos para crear una grabadora inteligente. Esta grabadora no solo captura audio, sino que también ofrece funcionalidades avanzadas como la identificación del hablante, la detección de emociones en su voz y la transcripción automática de audio a texto.

## 1.2 Problemática

La parte más importante de este proyecto es lograr integrar todos los modelos utilizados para crear un flujo de los procesos necesitados para el procesamiento de los audios. Tambien es necesario ajustar los modelos externos que son utilizados en estos procesos. La recoleccion de los audios necesarios para el entrenamiento del modelo creado por los integrantes del proyecto, la modificación, procesamiento y etiquetado de los audios.

# Chapter 2

# Objetivos generales y específicos

## 2.1 Objetivos Generales

Como objetivos generales se han planteado los siguientes:

1. **La correcta recolección de los audios para los conjuntos de entrenamiento y prueba.** Este objetivo implica la recopilación de muestras de audio de alta calidad, asegurando que se obtenga una diversidad adecuada de voces y entonaciones. Se presta especial atención a la limpieza y preprocesamiento de los datos para garantizar que sean representativos y útiles para el entrenamiento y la evaluación del modelo.

2. **El entrenamiento del modelo para la identificación de los hablantes.** Este objetivo se centra en el desarrollo y ajuste de un modelo de aprendizaje automático capaz de identificar con precisión a diferentes hablantes. Involucra la selección de algoritmos adecuados, la configuración de parámetros óptimos y la realización de múltiples iteraciones de entrenamiento para mejorar la precisión del modelo.

3. **La correcta separación de los hablantes en audios distintos para su posterior procesamiento.** Se busca desarrollar técnicas efectivas para separar las voces de diferentes hablantes cuando están presentes en una misma grabación. Esto incluye el uso de algoritmos de separación de fuentes y técnicas de segmentación de audio para asegurar que cada hablante pueda ser procesado individualmente.

4. **El reentrenamiento de los modelos externos para ajustarlos a las necesidades de este proyecto.** Este objetivo implica tomar modelos preentrenados existentes y ajustarlos para que se adapten mejor a los requisitos específicos del proyecto. Esto incluye la personalización del modelo para mejorar la detección de emociones en el contexto del proyecto y optimizar su rendimiento en las condiciones de audio particulares que se esperan enfrentar.

## 2.2 Objetivos Específicos

Como objetivos específicos se han planteado los siguientes:

1. Entrenar los modelos para que tengan una precisión lo más acertada posible en las interacciones fuera de los conjuntos de prueba y entrenamiento

2. Que el modelo sea capaz de identificar los hablantes correctamente

3. Lograr crear un flujo de trabajo con todos los modelos utilizados para garantizar la armonía entre estos

4. Devolver con presición el audio transcrito, el nombre del hablante y la emocion predominante en el audio.

Siguiento estos objetivos se ha logrado crear una aplicacion de python en la cual es posible grabar una conversación la cual se procesa y se devuelven los resultados obtenidos de la misma con una presición aceptable y adecuada para este proyecto. Todo esto gracias a los modelos preentrenados y los entrenados por los integrantes del equipo.

# Chapter 3

# Estado del arte

En esta sección se hace un recuento sobre la literatura acerca de los temas utilizados en este proyecto.

## 3.1 Separación de Hablantes

En el contexto de la separación de hablantes, se realizó una investigación exhaustiva que reveló la existencia de varios modelos capaces de llevar a cabo esta tarea con éxito. Entre los modelos evaluados se encuentran pyannote.audio, Kaldi y SpeechBrain. Todos estos modelos ofrecen un rendimiento sólido y cumplen con los requisitos para la tarea propuesta. Sin embargo, se optó por utilizar pyannote.audio por varias razones clave.

Primero, pyannote.audio se destaca por su alta precisión en la diarización de hablantes. Al estar construido sobre PyTorch, tiene la ventaja de soportar el uso de GPUs, lo que permite procesar grandes volúmenes de datos y mejorar significativamente la velocidad de ejecución. Este modelo no solo maneja eficazmente los cambios entre hablantes, sino que también es capaz de detectar solapamientos de voces, lo que resulta crucial en escenarios donde varias personas hablan al mismo tiempo. Además, cuenta con una comunidad activa y en crecimiento, lo que garantiza actualizaciones constantes y soporte para las últimas técnicas en tareas de procesamiento de voz.

En comparación, Kaldi es un modelo más tradicional, basado en técnicas estadísticas como los GMM (Gaussian Mixture Models) y el PLDA (Probabilistic Linear Discriminant Analysis), lo que lo hace más lento en relación a pyannote.audio y SpeechBrain. Aunque Kaldi admite x-vectors, su implementación tiende a ser más costosa computacionalmente. Además, la complejidad de su uso es considerablemente mayor, ya que requiere una comprensión avanzada del procesamiento de voz y su instalación y configuración pueden resultar tediosas, lo que aumenta el tiempo y esfuerzo necesarios para trabajar con este modelo.

Por otro lado, SpeechBrain comparte varias fortalezas con pyannote.audio, ya que también está basado en PyTorch y es compatible con GPUs, lo que lo hace igual de rápido. Además, al ser open source y contar con una comunidad activa, SpeechBrain es una excelente opción para tareas de investigación y académicas. Sin embargo, la decisión de optar por pyannote.audio sobre SpeechBrain se debe a que el equipo de trabajo ya está más familiarizado con pyannote.audio, lo que facilita su implementación y optimización en el proyecto.

Aunque SpeechBrain es igualmente poderoso y preciso, pyannote.audio fue preferido por su facilidad de uso, rapidez y precisión en el manejo de casos complejos como la

detección de solapamientos de hablantes, y por la capacidad del equipo de aprovechar al máximo sus características debido a su experiencia previa con este modelo

## 3.2 Clasificación de Hablantes

En el campo de la separación de hablantes, existen varias bibliotecas y modelos que han demostrado ser extremadamente eficientes y precisos. Entre los más destacados se encuentra x-vectors, un modelo robusto que utiliza redes neuronales profundas para generar embeddings de hablantes. Estos embeddings se proyectan en un espacio de alta dimensionalidad que permite separar las características únicas de cada hablante, lo que lo convierte en una de las técnicas más utilizadas para tareas de clasificación de voz en grandes sistemas.

Por otro lado, los d-vectors son una variante más simple de los x-vectors. Aunque ambos se basan en redes neuronales, los d-vectors se centran en tareas más sencillas o sistemas en tiempo real, donde la complejidad y el costo computacional son factores importantes. A pesar de ser menos complejos, siguen ofreciendo un rendimiento respetable en la clasificación de hablantes.

Otra arquitectura destacada es ResNet, que ha sido adaptada para tareas de clasificación de hablantes. ResNet se basa en redes neuronales profundas con bloques residuales, lo que le permite extraer embeddings que capturan las características más relevantes de los audios. Este enfoque es altamente eficaz para crear representaciones robustas de los hablantes, lo que facilita su diferenciación en escenarios complejos.

A pesar de la disponibilidad de estos modelos altamente sofisticados, el equipo de trabajo ha decidido desarrollar un modelo propio como parte de la implementación del proyecto. Esta decisión responde a los objetivos de la asignatura, que requieren el entrenamiento de un modelo personalizado o el reentrenamiento de un modelo preentrenado. Esto no solo permitirá un entendimiento más profundo de las técnicas de procesamiento de audio y machine learning, sino que también permitirá la personalización del modelo para adaptarlo a los requisitos específicos del proyecto.

## 3.3 Predecir Emoción del Hablante

En el campo de la detección de emociones, existen diversas herramientas y enfoques que han demostrado ser útiles para abordar esta tarea. Entre las más destacadas, se encuentran los modelos basados en redes neuronales, que han evolucionado significativamente en su capacidad para procesar señales de audio y extraer representaciones significativas. Las redes neuronales convolucionales (CNNs) son especialmente eficaces porque pueden aprender representaciones jerárquicas a partir de los datos. En el contexto del procesamiento de audio, las CNNs suelen trabajar con espectrogramas, que son representaciones visuales de la energía de las frecuencias a lo largo del tiempo. Gracias a su capacidad para detectar patrones espaciales y temporales en estos espectrogramas, las CNNs logran capturar características importantes de la señal de voz que son fundamentales para la identificación de emociones, lo que las convierte en una opción poderosa para tareas como la predicción de emociones en segmentos de audio cortos a medianos.

Las redes neuronales recurrentes (RNNs), y en particular su variante Long Short-Term Memory (LSTM), se destacan por su capacidad para manejar datos secuenciales, lo cual es clave en el análisis de audio donde las emociones evolucionan a lo largo del

tiempo. Las RNNs y LSTMs son útiles para capturar la dependencia temporal en los datos de audio, analizando cómo las emociones pueden cambiar en diferentes partes de una conversación o discurso. Aunque en el contexto de este proyecto no son las más eficientes debido a la naturaleza de los audios más cortos, en el futuro podrían ser útiles cuando se procesen audios más largos y cargados, lo que permitiría una evaluación más detallada de las emociones en tiempo prolongado.

Finalmente, también existen modelos preentrenados que ofrecen una solución práctica para el análisis de emociones. Uno de los más destacados es wav2vec, que ha sido entrenado con grandes volúmenes de datos y, gracias a su enfoque de aprendizaje por representación, puede ajustarse fácilmente a tareas específicas como la detección de emociones con un pequeño ajuste adicional. Este enfoque permite aprovechar las ventajas de un modelo previamente entrenado en un contexto más general y adaptarlo a las necesidades concretas del proyecto, ahorrando tiempo de entrenamiento y mejorando el rendimiento en la tarea específica de predicción de emociones.

## 3.4 Transcripción de Audio

Sobre la transcripción de audio a texto, contamos con varios modelos que destacan por su eficiencia, algunos de los cuales ya hemos mencionado, y otros que, aunque no se han discutido en detalle, también son muy competitivos. Uno de los más avanzados hasta el momento es Whisper, desarrollado por OpenAI. Este modelo ha sido entrenado con una enorme cantidad de datos multilingües, lo que le permite no solo transcribir audio con gran precisión, sino también detectar automáticamente el idioma en el que se habla. Gracias a su robustez en la transcripción de audios con ruido de fondo o interferencias, Whisper será el modelo que utilizaremos para esta tarea, ya que ha demostrado ser extremadamente fiable en escenarios difíciles.

Otro modelo destacado es Wav2Vec 2.0, desarrollado por Meta, que también ofrece una transcripción precisa incluso cuando no se dispone de una gran cantidad de datos etiquetados. Su arquitectura autosupervisada le permite aprender eficientemente representaciones de audio y adaptarse a diferentes tareas con poca supervisión, lo que lo convierte en una excelente opción para la transcripción en situaciones donde los datos etiquetados son escasos.

En cuanto a Kaldi, aunque es un modelo tradicional ampliamente utilizado en el ámbito académico y en proyectos de investigación avanzada, como se mencionó anteriormente, su uso requiere personal con experiencia en procesamiento de voz y es más complejo de configurar. Aunque es muy flexible y potente, no es la opción más adecuada para este proyecto debido a su curva de aprendizaje.

Inicialmente, consideramos usar Google Speech-to-Text debido a su precisión y eficiencia, pero finalmente decidimos cambiar a Whisper. A pesar de que la solución de Google ofrecía un rendimiento excelente, encontramos un inconveniente clave: requiere una conexión constante a Internet, lo que limita su uso en escenarios donde la conectividad no es estable o garantizada. Esta limitación nos llevó a optar por Whisper, que, al ser un modelo open-source y ejecutarse localmente, nos proporciona mayor control y flexibilidad en el entorno de desarrollo sin depender de conexiones externas.

# Chapter 4

# Propuestas de solución

Para resolver esta problemática es necesario dividir el problema en partes:

## 4.1 Separación de Hablantes

El primer paso es, del audio que se quiere procesar, lograr separar los hablantes. Esta tarea es crucial, ya que permite el análisis individual de cada voz, facilitando posteriores procesos como la transcripción y la detección de emociones. Para lograr esto, se ha utilizado un modelo preentrenado de la biblioteca de Python `pyannote.audio`. Este modelo ha sido seleccionado debido a su alta precisión en la separación de hablantes en diversas condiciones de grabación. En este paso se separa el audio principal en segmentos de audios los cuales pasaran a ser procesados mas tarde

El modelo de `pyannote.audio` utiliza técnicas avanzadas de procesamiento de señales y aprendizaje profundo para identificar y separar las diferentes voces presentes en una grabación. Sin embargo, la calidad del audio juega un papel fundamental en la efectividad de este modelo. Audios con ruido de fondo, grabaciones de baja calidad o situaciones donde los hablantes se superponen pueden presentar desafíos significativos. En tales casos, el modelo puede cometer errores, como confundir las voces o no separar adecuadamente a los hablantes.

## 4.2 Identificar Hablante

Después de separar los hablantes, y asumiendo que dicha separación ha sido efectiva, el siguiente paso es identificar a la persona que está hablando. Para lograr esto, se ha creado un conjunto de audios de los integrantes del equipo, los cuales fueron utilizados para entrenar y evaluar el modelo de identificación de hablantes.

Este conjunto de datos se ha dividido en dos subconjuntos: uno de entrenamiento y otro de prueba, siguiendo una proporción del 70 y 30 respectivamente. Esta división permite entrenar el modelo con una cantidad significativa de datos mientras se reserva una porción para evaluar su desempeño y generalización a nuevos datos no vistos durante el entrenamiento.

El modelo de identificación de hablantes fue desarrollado desde cero por los integrantes del equipo, empleando técnicas de aprendizaje supervisado. Se utilizaron diversas bibliotecas de Python especializadas en la creación y entrenamiento de redes neuronales, como TensorFlow y scikit-learn (sklearn). Estas herramientas proporcionaron los métodos

necesarios para construir y optimizar un modelo de aprendizaje profundo capaz de reconocer a los hablantes con alta precisión.

El proceso de desarrollo del modelo incluyó varias etapas clave:

1. **Preprocesamiento de Datos**: Los audios recopilados fueron normalizados y preprocesados para asegurar una calidad consistente y eliminar ruido. Esto incluye técnicas de reducción de ruido y normalización del volumen.

2. **Extracción de Características**: Se extrajeron características relevantes del audio, como coeficientes cepstrales de frecuencia mel (MFCCs), que capturan la información esencial para diferenciar las voces de los hablantes.

3. **Construcción del Modelo**: Se diseñó una red neuronal utilizando TensorFlow, adaptada específicamente para la tarea de identificación de hablantes, donde los parametros de entrada son las caracteristicas extraidas previamente. La arquitectura del modelo se optimizó para balancear precisión y eficiencia computacional.

4. **Entrenamiento del Modelo**: Utilizando el subconjunto de entrenamiento, el modelo fue entrenado iterativamente, ajustando sus parámetros para minimizar el error de predicción.

5. **Evaluación y Validación**: El subconjunto de prueba se utilizó para evaluar el desempeño del modelo, asegurando que pueda generalizar bien a nuevos datos.

## 4.3 Identificar Emoción en el Hablante

Al finalizar la identificación del hablante, nuestro objetivo es predecir su estado emocional basándonos en sus grabaciones de audio. Para abordar esta tarea, hemos empleado el modelo Wav2Vec2, una arquitectura avanzada diseñada inicialmente para la transcripción automática de voz. Este modelo, entrenado en una gran cantidad de datos de habla, captura tanto las características acústicas como lingüísticas de manera robusta.

Aplicamos Fine Tuning al Wav2Vec2 para adaptarlo específicamente a nuestro problema de clasificación emocional. Definimos seis etiquetas que representan las emociones clave que deseamos identificar en el hablante, incluyendo tristeza, ira, disgusto, miedo, felicidad y neutralidad. El proceso de Fine Tuning implica ajustar los pesos del modelo en función de nuestro conjunto de datos etiquetado, permitiendo que el modelo aprenda y se especialice en la clasificación precisa de estas emociones.

Para este propósito, utilizamos un conjunto de datos suplementario obtenido de fuentes en línea. Aunque este conjunto de datos está en inglés, su diversidad y calidad han demostrado ser altamente efectivas durante nuestros experimentos. Esta decisión se basa en la capacidad del modelo para generalizar patrones emocionales a partir de diferentes contextos lingüísticos y acústicos, mejorando así la precisión y la capacidad de generalización del modelo en escenarios prácticos.

## 4.4 Transcripción del audio

Una vez identificado al hablante y evaluado su estado emocional, el siguiente paso es convertir el audio correspondiente en texto. Para este propósito, utilizamos la biblioteca

de Python Google Speech Recognition, la cual se basa en un avanzado servicio de reconocimiento de voz desarrollado por Google. Este servicio permite la conversión instantánea del habla en texto en tiempo real, facilitando la automatización de procesos que requieren transcripción precisa y eficiente.

El proceso implica iterar a través de todos los audios previamente separados y aplicar este servicio para obtener la transcripción de cada archivo de audio individual. Google Speech Recognition emplea modelos y algoritmos optimizados, entrenados con grandes volúmenes de datos, lo que asegura una alta precisión en la conversión, incluso en entornos con diferentes acentos y condiciones acústicas.

# Chapter 5

# Experimentación y resultados

Como parte del proceso de experimentacion se ha sacado el promedio de las caracteristicas de audio de los integrantes del equipo reflejado en la siguiente imagen. Tener en cuenta que los graficos siguientes se han promediado de un conjunto de mas de 700 audios. Al menos 200 audios por cada integrante:

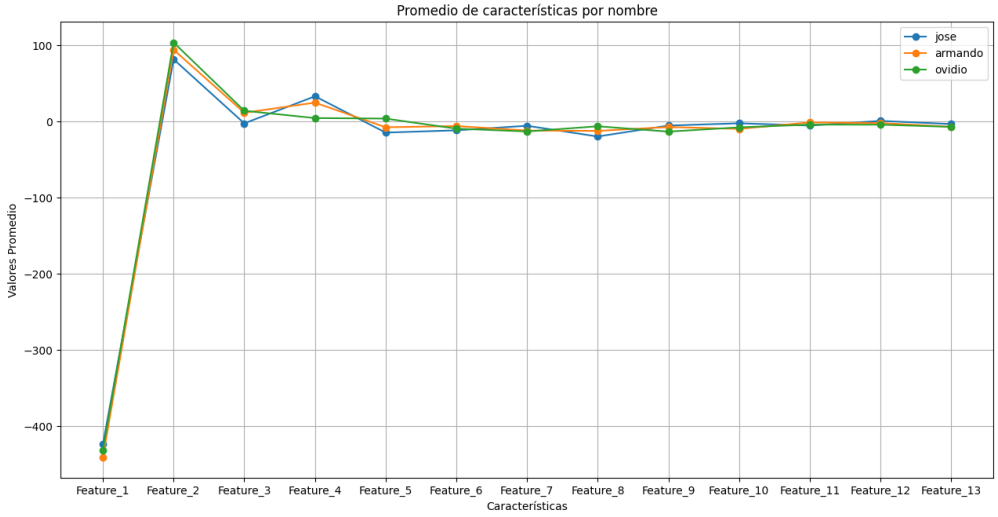



Figure 5.1: Caracteristicas de los hablantes

Mapa de correlacion entre las caracteristicas:

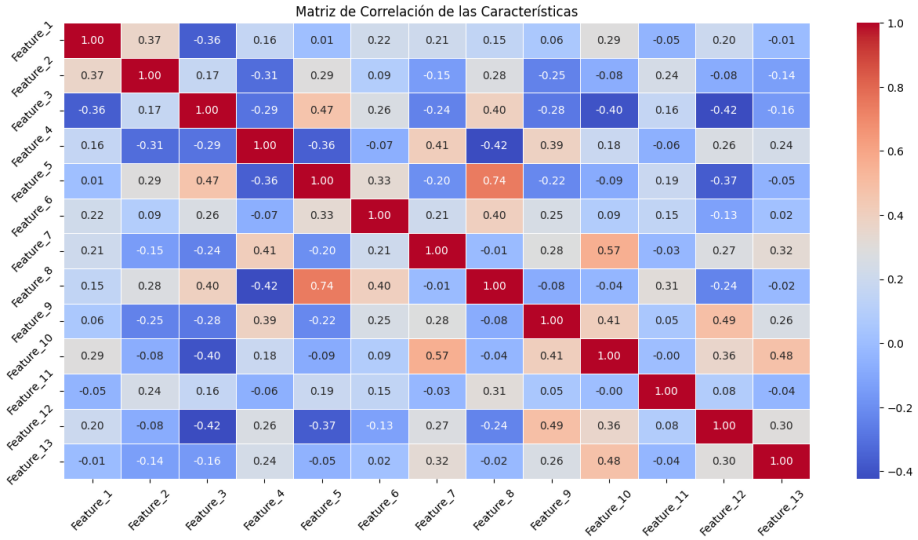



Figure 5.2: Correlación de las características

## 5.1 Experimentos

Para la realización de los experimentos se han utilizado 10 audios de 1min cada 1 en los cuales intervienen los diferentes integrantes del equipo de forma intermitente:

Audio 1: Aquí el modelo fue capaz de detectar a todos los hablantes con gran precisión sin equivocarse. Además detecta las emociones con una presición similar aunque comete algunos fallos. En la fase de transcripción se desempeña bastante bien, pero confunde algunas palabras, ya que el dialecto utilizado en el habla puede ser complejo

Audio 2: En este audio se puede notar una sucesion de errores como se esperaba ya que los hablantes superponen las voces y el modelo de separacion de los audios comete errores y los demas modelos arrastran este error haciendo predicciones erróneas tanto del hablante como de los sentimientos.

Audio 3: En este audio se exageran las emociones con el objetivo de poner a prueba el modelo encargado de esto. En los resultados se pudo observar que: la separacion de los hablantes fue exitosa, la identificacion de los mismos fue muy precisa exeptuando 2 casos en los que intercambia a los hablantes. La predicción de los sentimientos fue exitosa en todos los casos junto con la transcripción.

Audio 4: Este audio se puso a prueba el modelo de transcripcion, los hablantes dicen sus oraciones con rapidez. La separación de hablantes fue bastante existosa, exeptuando casos en los que los hablantes cambian con rapidez y no logra separa bien el momento exacto del cambio. La clasificación de los hablantes con buena presicion y la transcripcion tiene algunos errores pero todavia se asemeja bastante al audio.

Audio 5: En este, los hablantes actúan de forma pausada y respetando los tiempos del habla de los demás. La separacion, clasificación, sentimiento, transcripción con muy buena presición.

Audio 6: Los hablantes actuan de forma normal, Se aprecia una buena presición en la separación, la clasificación con errores menores, la deteccion sentimental con buena presición y la transcripción bastante bien.

Audio 7: Semejante al experimento 3, pero en este caso se equivoca detectando 1 emoción y 2 hablantes intercambiados. La transcripcion fue exitosa de igual forma

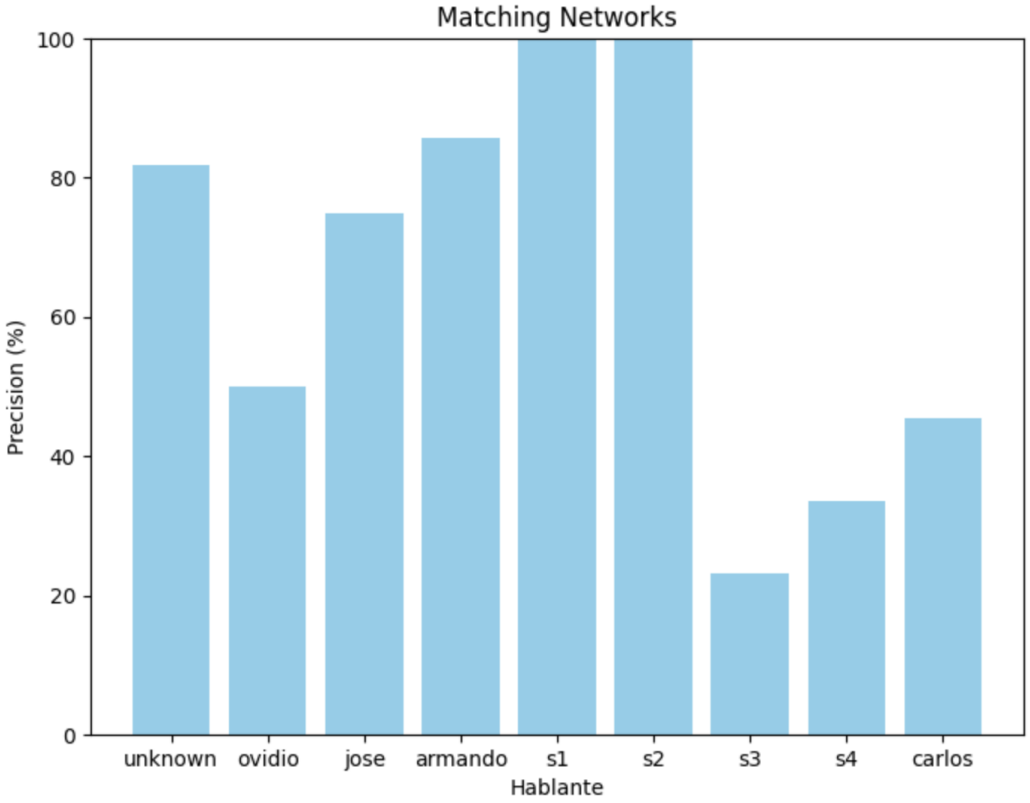
Audio 8: Similar al experimento 1, los hablantes actuan de forma natural y es capaz de reconocerlos a todos con gran precisión, buena identificacion de sentimientos y buena transcripción.

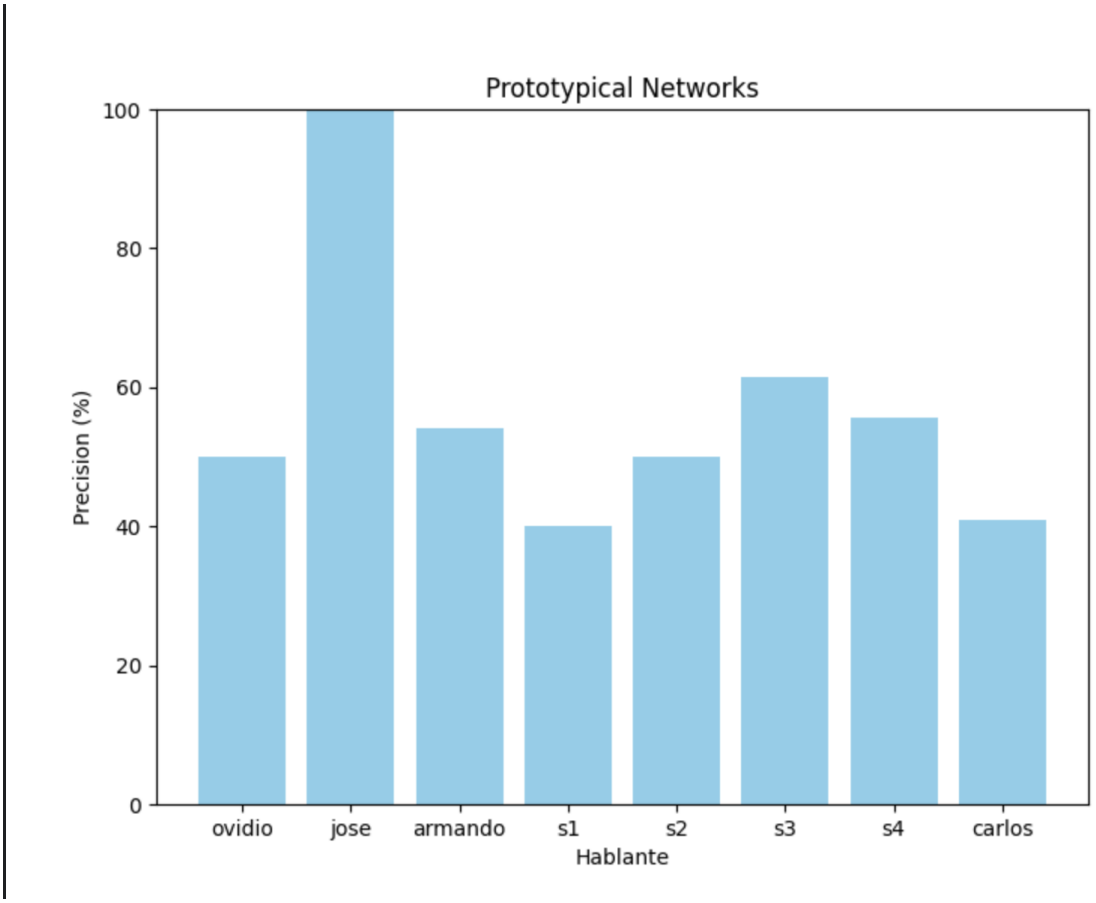
Audio 9: Los hablantes actuan de forma normal. Los diferentes modelos actuan correctamente.

Audio 10: Los hablantes actuan con sigilo como si no quisieran ser escuchados por los demas. El modelo de separacion actua con presición, la clasificacion con algunas deficiencias, la deteccion desentimientos con buena presición y la transcripcion muy buena.


Figure 5.3: Resumen de Experimentos

### 5.1.1 Identificación de nuevos hablantes






### 5.1.2 Resultados obtenidos

Se ovserva una mayor precisión en los resultados usando *Matching Networks* para la identificación de nuevos hablantes, esto se debe a que se manjeja mejor la variabilidad dentro de las clases al comparar directamente cada ejemplo de la clase con el ejemplo de prueba. En cambio, usando *Prototypical Networks* se pierde precisión cuando los ejemplos de una clase son muy diversos, ya que un solo prototipo, al ser el promedio de todos los *embeddings* de una clase no siempre representa adecuadamente la clase completa.

### 5.1.3 Transcripción de audios

Se compararon los resultados de las transcripciones de un conjunto de 100 audios utilizando Google Speech Recognition y el modelo small de Whisper. En el primer caso, se obtuvo un promedio de precisión del 96.09%, mientras que en el segundo, se alcanzó un 95.39%. Se optó por utilizar Whisper, ya que este no requiere una conexión a internet, a pesar de que su precisión es ligeramente menor, aunque no de manera significativa.

# Chapter 6

# Discusión de los resultados

Como se ha podido apreciar gracias a la experimentacion con los diferentes modelos, los resultados que salen a la vista son evidentes

1. **La calidad del audio sí importa:** En los audios donde existe claridad entre los hablantes es más fácil la separacion de los hablantes y la clasificación de los mismos
2. **Escucha lo que dice tu compañero:** Los audios donde los hablantes superponen las voces causan problemas en la separación de hablantes y estos se acarrean a los demás modelos,
3. **Demuestra tus sentimientos:** Cuando en el audio procesado es facil identificar si una persona está feliz o alterada, el modelo de reconocimiento de emociones funciona de maravilla.
4. **Despacio...:** En los audios donde los hablantes van muy deprisa el modelo de transcripcion comete errores.

Estos experimentos no solo revelan las fortalezas del modelo en condiciones controladas y su capacidad para adaptarse a diversos estilos de habla, sino que también destacan áreas críticas para la mejora. Abordar estos desafíos mediante técnicas avanzadas y una diversificación del entrenamiento promete mejorar significativamente el rendimiento del modelo, haciéndolo más robusto y preciso para aplicaciones en el mundo real.

# Chapter 7

# Conclusiones y trabajo futuro

Tras la finalización de este proyecto, se ha identificado un amplio espectro de posibles aplicaciones tanto en ámbitos científicos como comerciales. Por ejemplo, la grabación de sesiones de psicoterapia para análisis posterior, el registro completo de procedimientos judiciales, el seguimiento de discusiones en grandes reuniones corporativas e incluso la captura de conversaciones informales durante entrevistas. Todos estos usos serían factibles mediante un reentrenamiento de los modelos específicamente adaptado a cada caso, con la inclusión de un nuevo conjunto de datos de entrenamiento que contenga grabaciones de los participantes.

Como trabajo futuro, se podrían integrar nuevas funcionalidades a la grabadora de voz, como el procesamiento de texto para detectar inconsistencias o mentiras en las grabaciones, y para identificar declaraciones contradictorias. Además, sería posible desarrollar una interfaz intuitiva que facilite la manipulación y configuración de la grabadora, mejorando así su accesibilidad para los usuarios finales.

Estas mejoras no solo podrían ampliar significativamente las aplicaciones prácticas de la tecnología desarrollada, sino también aumentar su utilidad en una variedad de contextos profesionales y personales, promoviendo una mayor transparencia y precisión en la captura y análisis de información vocal.

# Bibliography


[1] Bredin, H., Yin, R., Coria, J. M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., & Gill, M. (2020). Pyannote.Audio: Neural Building Blocks for Speaker Diarization. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7124-7128.

[2] Chen, Z., Han, B., Wang, S., & Qian, Y. (2023). Attention-based Encoder-Decoder Network for End-to-End Neural Speaker Diarization with Target Speaker Attractor. *ArXiv*, abs/2305.10704.

[3] Plaquet, A., & Bredin, H. (2023). Powerset multi-class cross entropy loss for neural speaker diarization. *ArXiv*, abs/2310.13025.

[4] Torfi, A., Nasrabadi, N. M., & Dawson, J. M. (2017). Text-Independent Speaker Verification Using 3D Convolutional Neural Networks. *2018 IEEE International Conference on Multimedia and Expo (ICME)*, 1-6.

[5] Selin, M., & Mathew, D. P. (2021). Text-independent Speaker Verification Using Hybrid Convolutional Neural Networks. *Webology*, 18, 756-766.

[6] Chiu, C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R. J., Rao, K., Gonina, K., Jaitly, N., Li, B., Chorowski, J., & Bacchiani, M. (2017). State-of-the-Art Speech Recognition with Sequence-to-Sequence Models. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4774-4778.

[7] Han, T., Zhang, Z., Ren, M., Dong, C., Jiang, X., & Zhuang, Q. (2023). Speech Emotion Recognition Based on Deep Residual Shrinkage Network. *Electronics*.

[8] Koluguri, N. R., Kriman, S., Zelenfroind, G., Majumdar, S., Rekesh, D., Noroozi, V., Balam, J., & Ginsburg, B. (2023). Investigating End-to-End ASR Architectures for Long Form Audio Transcription. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 13366-13370.

[9] Bain, M., Huh, J., Han, T., & Zisserman, A. (2023). WhisperX: Time-Accurate Speech Transcription of Long-Form Audio. *Interspeech*.

[10] Alsabhan, W. (2023). Human–Computer Interaction with a Real-Time Speech Emotion Recognition with Ensembling Techniques 1D Convolution Neural Network and Attention. *Sensors (Basel, Switzerland)*, 23.

[11] Baevski, A., Zhou, H., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. *ArXiv*, abs/2006.11477.