

# A-e-i!: Herramienta para monitorear el deterioro del habla en pacientes con enfermedades neurodegenerativas

María Belén Rivera<sup>1</sup> y Sofía Rodríguez<sup>2</sup>

<sup>1</sup> Fac. de Ingeniería, Universidad Nacional de La Pampa  
riveramb@ing.unlpam.edu.ar

<sup>2</sup> Fac. de Ingeniería, Universidad Nacional de La Pampa  
rodriguezsofia@gmail.com

**Resumen.** En algunas demencias neurodegenerativas, las personas podrían presentar una disminución de habilidades lingüísticas que son difíciles de cuantificar con pruebas estandarizadas. El monitoreo y análisis del habla en estos pacientes puede proporcionar información valiosa sobre la evolución de su discurso hablado. Dentro de los tratamientos, se incluye a la lectura en voz alta pero, rara vez, es aplicada sistemáticamente como parte integral de la terapia y poco se sabe respecto a su eficacia. Este trabajo presenta el desarrollo de una aplicación denominada “A-e-i!”, que utiliza funciones de servicios alojados en servidores en la nube para transcribir audios de lecturas de pacientes con deterioro neurodegenerativo derivado en afasia progresiva primaria. Con las transcripciones generadas, se pretende monitorear los cambios en el discurso a través del tiempo de dichos pacientes. Para realizar este monitoreo, se diseñaron métricas que permiten medir distintos aspectos de la fluidez de la lectura. Y sirven también, para una posterior evaluación, a partir de indicadores que interpretarán los valores medidos en el tiempo. De esta manera, se espera constituir un acervo de datos que ayuden a mejorar las técnicas para el tratamiento de la lectura en voz alta, en pacientes con deterioro neurodegenerativo derivado en afasia progresiva primaria.

**Palabras claves:** Deterioro Neurodegenerativo · Procesamiento Lenguaje Natural · Computación en la Nube.

## 1 Introducción

La demencia incluye un conjunto de síntomas que afectan la memoria, el pensamiento, la ejecución y las habilidades sociales, y son lo suficientemente graves como para interferir en la vida diaria de la persona afectada. Las enfermedades neurodegenerativas son actualmente la causa más frecuente de demencia. La detección y caracterización de las alteraciones del lenguaje desempeña un papel cada vez más importante en la identificación y diagnóstico de muchas enfermedades neurodegenerativas, como por ejemplo, Afasia Primaria Progresiva (APP). Estos déficits del lenguaje se manifiestan por lo general, en etapas tempranas, sin mayores indicadores más que alguna latencia en el habla. Por lo que,

un análisis de la evolución del deterioro del lenguaje puede ser considerado como parte de la evaluación cognitiva de los pacientes afectados por alguna variedad de las enfermedades neurodegenerativas [1].

Actualmente, la mayoría de las evaluaciones cognitivas estándares que se realizan a estos pacientes<sup>3</sup> permiten conocer ciertos indicadores en relación a distintas dimensiones, como la independencia del paciente, su grado de expresividad, sus emociones, su capacidad para asociar y reconocer imágenes, y secuencias de ejecución de distintas tareas, por mencionar solo algunos de los enfoques que tienen los diagnósticos. Pero, no se refieren específicamente a evaluar y monitorear el deterioro del habla. Si bien, tanto la lectura en voz alta como el diálogo continuo, son prescritos por los profesionales como parte del tratamiento; a la fecha, no cuentan los mismos con herramientas que les permitan elaborar análisis cuantitativos permanentes de estas habilidades. Eso posibilitaría contrastar muchos casos y abordar conclusiones a fin de mejorar las terapias ofrecidas. Aún más, usualmente, no es objetivo de las terapias, monitorear la lectura oral [2]. El tratamiento de las demencias neurodegenerativas sigue siendo sintomático o de soporte [2], pero con escasas herramientas para monitorear el deterioro del habla.

Por lo expuesto, se evidencia la necesidad de contar con herramientas que permitan un monitoreo y posterior evaluación del lenguaje oral, principalmente, el producido con la lectura. Los especialistas médicos podrían indicar como tratamiento terapéutico, la grabación de la lectura en voz alta y se beneficiarían si contaran con esos datos del paciente obtenidos desde su propio hogar. En este sentido, la grabación de las voces de los pacientes constituye un flujo importante de datos, que se puede recopilar de forma discreta y económica [3].

Este trabajo presenta el desarrollo de un prototipo<sup>4</sup> de aplicación web con arquitectura REST [4] distribuida, denominada “A-e-i!” que utiliza funciones de servicios de reconocimiento del discurso, alojados en servidores en la nube, para transcribir audios de lecturas de pacientes con deterioro neurodegenerativo afectando el habla. Con las transcripciones generadas, se pretende monitorear los cambios en el discurso a través del tiempo de dichos pacientes. Para realizar este monitoreo, se diseñaron métricas que permiten medir distintos aspectos en relación a la fluidez de la lectura. Y que, además, sirven para una posterior evaluación, a partir de indicadores que interpretarán los valores medidos en el tiempo. Con este prototipo de herramienta, el profesional médico dispone de los resultados de la medición en forma de gráficos, y podrá observar cómo evolucionan en el tiempo, distintas dimensiones, como por ejemplo, la duración de la lectura, que puede ser un indicador de la latencia en el habla.

Es preciso resaltar que el alcance de este trabajo no contempla el diseño de los indicadores, que implicaría evaluar las medidas obtenidas considerando umbrales de decisión y poder interpretar los valores medidos graficados. Esto,

<sup>3</sup> Entre las más habituales se destacan: *Boston Diagnostic Aphasia Examination* (BDAE), *Western Aphasia Battery*.

<sup>4</sup> Se considera un prototipo pues en esta etapa, su grado de desarrollo no permite utilizarse en ambiente productivo.

debe constituir una tarea en conjunto con profesionales médicos. La Figura 1 ilustra el esquema de la propuesta de A-e-i! como herramienta para monitorear el deterioro del habla en pacientes con enfermedades neurodegenerativas.



**Fig. 1.** Esquema del funcionamiento de A-e-i!.

El trabajo se organiza de la siguiente manera. La Sección 2 introduce la arquitectura de A-e-i! y su descripción, desde el punto de vista de las tecnologías empleadas. La Sección 3 presenta el diseño de las métricas para ser calculadas en cada transcripción de la lectura. La Sección 4 discute algunos trabajos relacionados que fueron estudiados para comprender el tema y robustecer la propuesta presentada. La Sección 5 presenta el caso de estudio con una paciente real, que sirvió como referencia y principal motivación del trabajo y línea de investigación. Finalmente, la Sección 6 aborda las conclusiones y trabajos futuros.

## 2 Arquitectura de A-e-i!

Desde el punto de vista tecnológico, A-e-i! tiene una arquitectura distribuida en la nube de servicios RESTful. La Figura 2 presenta la vista de componentes y conectores de la arquitectura [5]. Cada elemento de una vista de componentes y conectores representa una pieza que tiene presencia en tiempo de ejecución, como procesos, servidores, clientes o fuentes de datos y estas piezas están conectadas entre sí por distintos tipos de interacción.

En la arquitectura diseñada, los elementos que conforman esta vista se corresponden con los siguientes módulos:

- Módulo *Frontend*: Para implementar la interfaz de usuario se utilizó Next.js<sup>5</sup> junto a Chakra UI<sup>6</sup>. Next.js es un framework *frontend* basado en React para crear interfaces de usuario interactivas.
- Módulo *Backend*: La lógica de la aplicación está diseñada como servicios web RESTful (conocidas en la terminología informática como APIs). Comprende

<sup>5</sup> <https://nextjs.org/>. Fecha acceso, junio 2024.

<sup>6</sup> <https://v2.chakra-ui.com/>. Fecha acceso, junio 2024

la API de autenticación, la API que gestiona la información y tareas de los profesionales médicos y la API que gestiona los audios de los pacientes. Para el desarrollo de los servicios se utilizó el módulo de Python conocido como Flask<sup>7</sup>.

- Módulo de persistencia: El almacenamiento de los datos de A-e-i! se diseñó como un servicio en la nube de *Google Cloud* y utilizando una base de datos MySQL.
- Módulo de procesamiento de lenguaje natural: Para el análisis de los audios y la transcripción de los mismos, se diseñaron componentes que utilizan los servicios cognitivos *Azure Speech Recognition* de *Microsoft Azure* que permiten transcribir voz a texto.

Para el procesamiento de la transcripción obtenida se usó reconocimiento de partes de un discurso con Spacy<sup>8</sup>. Spacy es una biblioteca de procesamiento del lenguaje natural en el lenguaje Python. Está diseñada para ser rápida y eficiente, especialmente en tareas de procesamiento de texto como tokenización, lematización, reconocimiento de entidades nombradas (conocido por su sigla en inglés NER, de *Named Entity Recognition*), análisis sintáctico (*parsing*), entre otros. Los resultados del procesamiento de la transcripción se almacenan en la base de datos, que es luego recuperada para cada paciente.

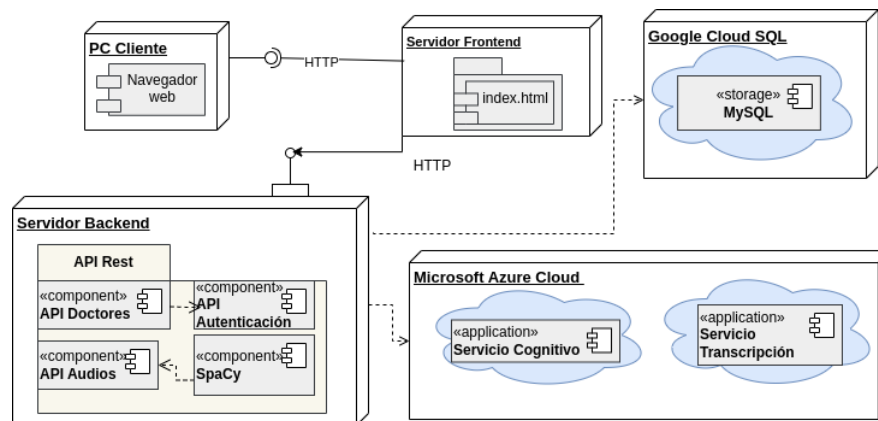


Fig. 2. Arquitectura de A-e-i!. Vista de componente y conector.

## 2.1 Módulo de Procesamiento del Lenguaje Natural

Los componentes que procesan el audio de las grabaciones de las lecturas del paciente forman parte del sistema de reconocimiento de voz implementado uti-

<sup>7</sup> <https://flask.palletsprojects.com/en/3.0.x/>. Fecha acceso, junio 2024

<sup>8</sup> <https://spacy.io/models>. Fecha acceso, junio 2024.

lizando el servicio *Azure Speech Recognition* de *Microsoft Azure Cognitive Services* [6]. Para implementar el algoritmo de transcripción de audio, se utilizó el kit de desarrollo de software (*software development kit*, SDK) de voz de Azure para lenguaje Python. Luego, para poder analizar un audio en particular, se creó una configuración de audio o *AudioConfig*. La función de transcripción recibe como parámetro el audio y esta instancia de configuración puede conectarse a un reconocedor de discurso o *SpeechRecognizer* que es quien inicia el proceso de transcripción,

Una vez creada la instancia de *SpeechRecognizer* se debe configurar, o bien el reconocimiento de una sola captura o el reconocimiento continuo. El primero, reconoce una expresión única que se determina mediante la escucha de un silencio que marca un final o hasta que se procesa un máximo de 15 segundos de audio. El reconocimiento continuo se utiliza en los casos de audios de más de 15 segundos, pues permite establecer cuándo detener el reconocimiento. Este tipo de reconocimiento se ajusta mejor a las necesidades de la herramienta A-e-i! porque se espera que la duración de los audios sea mayor a 15 segundos, dada la naturaleza elentecida del discurso de los pacientes. Para marcar el fin del reconocimiento, se utilizó una variable booleana, denominada “fin”, e inicializada en *false* y otra variable de tipo lista, para almacenar los resultados parciales de la transcripción. También, se conectó una función *callback*, denominada *stop*, encargada de detener el reconocimiento continuo al recibir un evento de cancelación o detenimiento.

Otro componente desarrollado como parte del sistema de reconocimiento de voz, es el que permite reducir el ruido. El motor de reconocimiento de discurso tiene que distinguir entre el ruido del ambiente en que se graba cada audio y la voz del paciente, lo que le dificulta su tarea de identificar palabras. Para implementar este componente, se programó el algoritmo mostrado en el Código 1.1, que utiliza la biblioteca *noisereducer* de Python que proporciona una alternativa conveniente para reducir el ruido en señales de audio, utilizando técnicas avanzadas de procesamiento de señales. Esta biblioteca se integró con *librosa*<sup>9</sup>, que también es una biblioteca popular en Python utilizada principalmente para análisis y procesamiento de señales de audio, especialmente en el contexto de música y sonido. La utilización de *noisereducer* en el algoritmo de reducción de ruido permitió utilizar el método de estimación de ruido por banda de frecuencia y filtrado adaptativo.<sup>10</sup>

```

1 import numpy as np
2 import librosa
3 from scipy.io import wavfile
4 import noisereducer as nr
5 def reducir_ruido(filename):
6     rate, data = wavfile.read(filename)
7     audio_modificado = nr.reduce_noise(y=data, sr=rate)

```

<sup>9</sup> <https://librosa.org/doc/latest/index.html>. Fecha acceso, junio 2024.

<sup>10</sup> Los aspectos relacionados a mejorar este algoritmo, o la discusión sobre la elección de las librerías necesarias para implementarlo, escapan del objetivo del presente trabajo.

```
8 wavfile.write(filename, rate, audio_modificado)
```

**Código 1.1.** Algoritmo de reducción de ruido.

Por último, se mejoró la precisión del reconocimiento a partir del componente *PhraseListGrammar*<sup>11</sup> que permite definir y utilizar listas de frases específicas que el servicio de reconocimiento de voz debe reconocer con alta prioridad durante la transcripción. Una lista de frases es una lista de palabras o frases de las que se quiere ayudar a mejorar su reconocimiento ya que, luego de ser añadidas al reconocedor de discurso, aumenta la probabilidad de que éste las identifique [7]. Lo conveniente es agregar las palabras que son probables de ser pronunciadas. En esta instancia de avance preliminar de la aplicación, se optó por seleccionar 20 palabras que se corresponden con aquellas más frecuentes del texto de referencia utilizado como caso de estudio y se sabe que pueden llegar a ser pronunciadas por el paciente en su lectura. Para obtener estas palabras en el texto de referencia, se utilizó SpaCy [8], omitiendo los símbolos de puntuación, los espacios en blanco y las *stop words*<sup>12</sup>. Luego, se utilizó la colección *Counter* para obtener la cantidad de apariciones de cada palabra y ordenar la lista de forma descendente. El código 1.2 muestra el algoritmo implementado para agregar las palabras a la lista de frases de la instancia configurada en el reconocedor del discurso.

```
1 phrase_list_grammar = speechsdk.PhraseListGrammar.
2 from_recognizer(speech_recognizer)
3 lista_palabras_comunes = palabras_mas_comunes(
4     texto_original)
5 for palabra in lista_palabras_comunes:
6     phrase_list_grammar.addPhrase(palabra)
```

**Código 1.2.** Algoritmo para agregar una lista de palabras al reconocedor de discurso.

El procesamiento del lenguaje natural (conocido como NLP, siglas en inglés de *Natural Language Processing*) entra en escena una vez que se haya generado este reconocimiento de discurso; es decir, tan pronto el archivo de audio del paciente se convirtió en formato de texto. De esta manera, es posible conocer la gramática del texto transcripto, analizarlo en función de las estructuras gramaticales del idioma español y comparándolo con el texto original.

Para el procesamiento del lenguaje natural, se utilizó SpaCy. Para el idioma español hay modelos entrenados disponibles, sin embargo, una de las ideas detrás de SpaCy es que sus modelos no entrenados, que consisten en redes neuronales convolucionales, puedan usarse para entrenar un modelo en cualquier idioma [9]. En este trabajo se utilizó un modelo preentrenado, denominado *es\_core\_news\_sm*, que tiene una precisión del 98% en el reconocimiento de partes de un discurso.

<sup>11</sup> Es parte de Microsoft Speech SDK (*Speech Software Development Kit*)

<sup>12</sup> Las *stop words* son las palabras más comunes de un idioma que no aportan información sustantiva sin su contexto, por ejemplo, “y” o “yo”. Por lo general, se eliminan del texto para reducir el tiempo de procesamiento.

El reconocimiento de partes de un discurso se utiliza para clasificar las palabras según el conjunto al que pertenecen. En una oración se pueden identificar las partes del discurso, es decir, las funciones que ocupan cada palabra dentro del mismo, como por ejemplo “nombre propio”, “verbo”, “adjetivo”, entre otras. Estos modelos funcionan en dos fases. En la primera fase, se divide el texto en segmentos para clasificar cada uno. En la segunda fase, se clasifica cada segmento en categorías en forma de *tokens* [10]. Todo este proceso se lleva a cabo ignorando cuestiones de formato, como por ejemplo, la presencia de mayúsculas.

### 3 Diseño de Métricas

Finalmente, y dado que el objetivo de A-e-i! es brindar datos a profesionales médicos en relación al deterioro del habla en pacientes con enfermedades neurodegenerativas (como la APP), se diseñaron métricas que permitan medir distintos aspectos de las transcripciones realizadas a partir de los audios de las lecturas de estos pacientes. El diseño de las métricas, en esta instancia, no tuvo la opinión de expertos en el tema, sino que, estuvo basado en utilizar las librerías de NLP que permiten cuantificar las dimensiones que se han considerado relevantes para el propósito de la herramienta. Sin embargo, a futuro se prevé un trabajo interdisciplinario con profesionales de la salud y que sean éstos quienes indiquen los aspectos interesantes a medir para luego poder interpretar esas mediciones con fundamentos médicos.

A continuación se mencionan los distintos algoritmos y fórmulas que se consideraron para el diseño de las métricas, las cuales se dividen en cinco categorías:

1. **Métricas que emplean el algoritmo de *Part-Of-Speech Tagging* (POS Tagging) de NLP:** Este algoritmo consiste en asignar a cada palabra en un texto una categoría gramatical específica, como sustantivo, verbo, adjetivo, etc. Las métricas que se diseñaron utilizando este algoritmo permiten comprender aspectos de la estructura gramatical y semántica de la transcripción del paciente y están basadas en el cálculo de cantidades. A futuro, se espera diseñar indicadores que interpreten el valor de estas métricas y permitan realizar, por ejemplo, análisis sintáctico, y abordar conclusiones en base a umbrales establecidos por profesionales médicos.
  - Cantidad de adjetivos pronunciados correctamente (**#APC**): Esta métrica mide cuántos adjetivos se pronunciaron correctamente respecto a los adjetivos del texto base de prueba.
  - Cantidad de adjetivos pronunciados (correctos o incorrectos) (**#AP**): Esta métrica permite calcular cuántos adjetivos se pronunciaron en total.
  - Cantidad de sustantivos pronunciados correctamente (**#SPC**): Esta métrica permite calcular cuántos sustantivos se pronunciaron correctamente respecto a los sustantivos del texto base de prueba.
  - Cantidad de sustantivos pronunciados (correctos o incorrectos) (**#SP**): Esta métrica permite calcular cuántos sustantivos se pronunciaron en total.

- Cantidad de verbos pronunciados correctamente ( $\#VPC$ ): Esta métrica permite calcular cuántos verbos se pronunciaron correctamente respecto del total de verbos que tiene el texto base de prueba.
- Cantidad de verbos pronunciados (correctos o incorrectos) ( $\#VP$ ): Esta métrica permite calcular cuántos verbos se pronunciaron en total.

2. **Métricas que emplean la medida WER (*Word Error Rate*):** La medida WER es una de las métricas más importantes y ampliamente utilizadas en los sistema de transcripción automática de voz a texto para medir qué tan precisamente el sistema ha transcrito los segmentos hablados. Así, las métricas que se diseñaron considerando esta medida WER están relacionadas a la cantidad de palabras omitidas, agregadas, sustituidas o pronunciadas correctamente. La valoración de los resultados de estas métricas se debe realizar con indicadores que interpreten dichos valores medidos y evalúen según criterios de satisfacción establecidos por profesionales entendidos en el tema.

- *Word Error Rate* ( $\#WER$ ): Es una métrica que se usa comúnmente para evaluar el rendimiento de un sistema de reconocimiento de voz o traducción automática. El análisis se hace teniendo en cuenta la longitud de la secuencia de palabras reconocidas comparada a la longitud de la secuencia de palabras correctas de referencia.

La fórmula 1 se usa para calcular la métrica WER. La misma deriva de la distancia de Levenshtein, trabajando a nivel de palabra en lugar de a nivel de fonema.

$$\#WER = (\#PS + \#PE + \#PI) / \#P \quad (1)$$

donde:

$\#PS$  es el número de sustituciones,

$\#PE$  es el número de eliminaciones,

$\#PI$  es el número de inserciones,

$\#P$  es el número de palabras en el texto base.

- Cantidad de palabras pronunciadas correctamente ( $\#PPC$ ): Esta métrica permite calcular cuántas palabras se pronunciaron correctamente sobre el total de palabras del texto base.
- Cantidad de palabras pronunciadas eliminadas ( $\#PE$ ): Esta métrica permite calcular cuántas palabras que forman parte del texto base, no fueron pronunciadas.
- Cantidad de palabras pronunciadas insertadas ( $\#PI$ ): Esta métrica permite calcular cuántas palabras fueron insertadas, es decir, cuántas palabras que no forman parte del texto base, fueron pronunciadas adicionalmente por el paciente.
- Cantidad de palabras pronunciadas sustituidas ( $\#PS$ ): Esta métrica permite calcular cuántas palabras fueron sustituidas, es decir, cuántas palabras que no forman parte del texto base, fueron pronunciadas en lugar de las palabras correctas.



3. **Métricas que permiten generar medidas directas e indirectas:** Estas métricas, como el porcentaje de palabras pronunciadas, se diseñaron contemplando la utilización de un algoritmo que convierte cada texto en una lista de objetos de tipo *string* y luego, el método *len()* de Python que devuelve la longitud de cada lista, lo que equivale a la cantidad de palabras de cada texto.
  - Cantidad de palabras del texto base (#P): Esta métrica calcula la cantidad de palabras que forman parte del texto de referencia.
  - Cantidad de palabras del texto de la transcripción (#T): Esta métrica calcula la cantidad de palabras que forman parte del texto leído por el paciente, luego de realizar la transcripción del audio.
  - Porcentaje de palabras pronunciadas (#PvsT): Relación entre el total de palabras de la transcripción (#T) y el total de palabras del texto base (#P).
  - Porcentaje de coincidencia (#PC): Es la relación entre el total de palabras pronunciadas correctamente (#PPC) y el total de palabras del texto base (#P).
  - Velocidad del discurso (#VD): Es la medida del número de palabras pronunciadas en un período de tiempo determinado. Es la relación entre las palabras del texto (#P) y la duración del audio (#TDA).
4. **Métricas que utilizan los algoritmos de cálculo de similitud con *SpaCy*:** Los algoritmos de cálculo de similitud se utilizan para comparar la similitud entre documentos, oraciones o palabras basadas en su significado semántico. Dado que Spacy proporciona métodos eficientes y precisos para calcular estas similitudes utilizando modelos pre-entrenados que capturan las representaciones vectoriales de las palabras y los contextos en los que aparecen, se diseñó una métrica indirecta que permita conocer el grado de similitud entre el texto original y el leído por el paciente.
  - Porcentaje de coincidencia según SpaCy (#CsS): se compararan dos objetos y se predice qué tan similares son a partir de la representación vectorial de cada palabra.
5. **Métricas para medir la duración del audio:** Esta métrica es calculada a partir de la propiedad *duration\_seconds* del objeto *AudioSegment* generado durante la ejecución del algoritmo de conversión de audio a texto.
  - Duración del audio (#TDA): Es la duración en segundos de cada audio. Como cada paciente tiene un texto asignado, esta métrica mide el tiempo que se tarda en leerlo en voz alta. Permitirá, a posterior y con el diseño de indicadores acordes que interpreten estas medidas, analizar la fluidez de la lectura.

## 4 Trabajos Relacionados

En [3] los autores muestran resultados sobre la efectividad de aplicar técnicas de *machine learning* para el diagnóstico automático de ciertas enfermedades

neurodegenerativas que alteran la producción del lenguaje y el discurso. Los resultados demuestran que se pueden predecir diagnósticos más precisos y prevén futuros estudios con grabaciones de mayor calidad que mejorarán los resultados.

El trabajo que se propone en [11] detecta parafasias fonémicas y neologismos a partir de ejemplos de discursos grabados desde *Aphasia Bank* [12]. Proponen un sistema de reconocimiento del discurso con modelos de lenguaje para transcribir automáticamente discursos de personas afásicas. Los resultados y análisis proporcionados en este trabajo ayudan a sentar las bases para el trabajo futuro, dirigido a la detección automática de parafasias.

El método ORLA (*Oral Reading for Language in Aphasia*) se describe en [2], [13] y fue desarrollado para mejorar las habilidades de comprensión en la lectura. El método provee a los pacientes, con práctica en lectura oral de oraciones y párrafos, en conjunto con la terapia clínica. Su autora, la Dra. Leora Cherney, ha contribuido al desarrollo de protocolos clínicos y pautas para la rehabilitación de la afasia y sus investigaciones han explorado diversas técnicas de tratamiento, como la terapia del lenguaje con restricciones, el entrenamiento de guiones y las pruebas rápidas cognitivo-lingüísticas [14].

Otro trabajo analizado fue el propuesto en [15] en el que se presenta un sistema para automáticamente detectar y clasificar categorías de errores en la producción de palabras característico de anomia del discurso conectado. Utilizan modelos estadísticos para identificar posibles errores y emplean un modelo de distancia fonológica para determinar similitud fonológica entre el enunciado del sujeto y un conjunto plausible de palabras intencionadas. Luego, aplican técnicas de *machine learning* para determinar en cuáles de las categorías cae un error de producción de la palabra.

Estos trabajos analizados y, especialmente los de Cherney, permiten avalar el estudio del análisis del discurso de pacientes con APP mediante la lectura en voz alta. En tanto que difieren, en términos generales, con la propuesta del presente trabajo, en los siguientes aspectos: 1) desarrollan técnicas y algoritmos para trabajar con bases fonéticas en inglés, las pruebas están realizadas en ese idioma, 2) requieren la asistencia conjunta del paciente y su terapeuta (fonoaudiólogo u otro especialista), 3) utilizan técnicas de grabación por medio de micrófonos o vídeos que luego transcriben, 4) centran su análisis en el discurso completo y no en cada palabra, 5) trabajan los aspectos fonológicos y análisis de fonemas.

En tanto que, A-e-i!, la propuesta de este trabajo se caracteriza por: 1) centrada en el idioma castellano, por tanto, las bases fonéticas es menor que las halladas para el idioma anglosajón, 2) parte de la premisa que una terapia de lectura en voz alta podría ser mejor si el paciente la realiza desde la propia comodidad de su casa, 3) centra su análisis en cada palabra de cierta lectura asignada al paciente, 4) no trabaja con el análisis de aspectos fonológicos y fonema debido a que, en esta instancia, no se tiene participación de profesionales en el tema.

## 5 Caso de Estudio

El caso de estudio consistió en trabajar con una paciente real (una mujer de 72 años, jubilada docente) con deterioro neurodegenerativo derivado en APP. Dado su interés por la lectura le indicaron que la practique en voz alta. Alrededor del 2020, la paciente podía leer un libro entero en voz alta. A la fecha actual (2024), apenas puede leer una oración.

### 5.1 Realización de Pruebas

La prueba realizada con A-e-i! para validar la implementación tecnológica y la utilidad de la idea, se desarrolló de la siguiente manera. Primero, se escogió un párrafo de un texto que era familiar para la paciente. Dado su avanzado deterioro neurodegenerativo, el texto se seleccionó considerando que sea de lectura fácil y con una historia conocida. Así se escogieron los dos primeros párrafos de “La hormiguita viajera” de Constancio C. Vigil, un libro que pertenece a la paciente desde su niñez. La Figura 3 retrata un momento de la prueba. Luego, se determinó la periodicidad de la prueba. Se estableció una frecuencia de dos días a la semana, a partir de diciembre 2022, siempre considerando el estado emocional de la paciente, para no perturbarla. Un familiar acompañando a la paciente, grabó cada una de sus sesiones de lectura del texto de referencia utilizando la aplicación A-e-i!.



**Fig. 3.** Lectura: “La hormiguita viajera”.

El caso de estudio se compone de un total de 8 audios de la paciente, que, con A-e-i! han sido transcritos y, el texto generado, analizado con el módulo que implementa el servicio de procesamiento de lenguaje natural de la herramienta. Como parte de dicho procesamiento, se calcularon las métricas diseñadas, cuyos valores obtenidos fueron graficados por la herramienta desarrollada. Estos gráficos no están a disposición de los pacientes, sino de los profesionales médicos que atiendan a los mismos y son usuarios del sistema.

## 5.2 Generación de Resultados

El profesional médico que utilice A-e-i! tiene a su disposición, para cada uno de sus pacientes que graba sus lecturas, el listado de métricas con las mediciones realizadas para cada día de lectura. Por ejemplo, para la métrica *velocidad del discurso* (#VD), la Figura 4 muestra la gráfica que genera A-e-i! con los valores obtenidos en el cálculo de la métrica para cada uno de los audios transcritos. Para la métrica *duración del audio* (#TDA), la Figura 5 ilustra los valores obtenidos en minutos, para cada uno de los días realizados de la prueba. Un posible análisis posterior de estos datos podría determinar si el habla del paciente en modo lector, incrementa las latencias con el correr del tiempo o si por el contrario, permanece estable.



Fig. 4. Valores de la métrica *velocidad del discurso*.



Fig. 5. Valores de la métrica *duración del audio*.

Como se ha resaltado en la Sección 3, en este trabajo no se considera la evaluación de los resultados y su valoración por carecer de una interpretación médica de los mismos. Por esta razón, A-e-i! solo grafica los resultados de las métricas, es decir, en esta primera etapa solo realiza la medición. El próximo paso es realizar la evaluación, diseñando indicadores que interpreten los valores medidos a partir de umbrales de aceptación que especifiquen los profesionales médicos, utilizando estrategias de medición y evaluación [16].

## 6 Conclusiones y Trabajo Futuro

Este trabajo presentó el desarrollo de un prototipo de una aplicación que permite procesar audios de lecturas de pacientes con enfermedades neurodegenerativas; transcribe, procesa dichos audios y calcula valores de interés (a partir de métricas) con el fin de ayudar a profesionales que atienden a estos pacientes, a monitorear la evolución del deterioro de su habla. La aplicación pretende ser una herramienta de asistencia a los facultativos médicos. Su propósito es que los ayude a controlar la terapia de lectura en voz alta de sus pacientes (que podrían estar padeciendo APP) y así, abordar conclusiones que sirvan, a futuro, para mejorar las prácticas a cada paciente y profundizar los estudios sobre cómo evoluciona el deterioro del habla. Sin embargo, en esta etapa inicial, la herramienta no brinda conclusiones ni el análisis de los resultados que se van obteniendo en el transcurso del tiempo que dure la terapia de lectura en voz alta para estos pacientes.

El trabajo constituye un avance inicial, pero pretende resaltar que la importancia de la terapia de leer en voz alta no alcanza, si no se mantiene un seguimiento por parte de los profesionales, del habla de sus pacientes, que progresivamente, se va perdiendo.

El procesamiento del lenguaje natural puede ser utilizado de diversas maneras para apoyar las terapias de enfermedades neurodegenerativas que afectan la capacidad de una persona para comunicarse y comprender el lenguaje. Sin embargo, no reemplaza la importancia de la terapia tradicional realizada por profesionales de la salud especializados en el tratamiento de estas enfermedades. Constituye una herramienta complementaria que ayuda a los terapeutas a personalizar el tratamiento y realizar un seguimiento objetivo del progreso del paciente a lo largo del tiempo.

Como trabajo futuro, se espera extender la tarea interdisciplinaria y trabajar conjuntamente con especialistas en el tema para: a) mejorar y ampliar las funcionalidades de la herramienta presentada, b) considerar otras enfermedades que afecten el habla, c) ampliar los casos de prueba con más pacientes, d) contrastar resultados de casos de pacientes sin afectación del habla contra aquellos que sí lo padecen y, finalmente, el aspecto prioritario, e) trabajar en el diseño de indicadores que interpreten las medidas calculadas al momento y poder realizar un análisis de los resultados generados. Desde el punto de vista tecnológico, se continuará con el estudio de servicios y algoritmos que mejoren la transcripción realizada y procesen en español el lenguaje natural.

## Referencias

1. Boschi, V., Catricala, E., Consonni, M., Chesi, C., Moro, A., Cappa, S.F. Connected speech in neurodegenerative language disorders: a review. *Frontiers in psychology* 8, 269, (2017)
2. Cherney, L.R.: Efficacy of oral reading in the treatment of two patients with chronic broca's aphasia. *Topics in Stroke Rehabilitation* 2(1), 57–67, (1995)
3. Peintner, B., Jarrold, W., Vergyri, D., Richey, C., Tempini, M. L. G., and Ogar, J. Learning diagnostic models using speech and language measures. En 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pág. 4648–4651. IEEE, (2008)
4. Fielding, R. T. Architectural styles and the design of network-based software architectures. University of California, Irvine, (2000)
5. Clements, P., Garlan, D., Little, R., Nord, R., and Stafford, J. Documenting software architectures: views and beyond. En 25th International Conference on Software Engineering, Proceedings, pág- 740–741, IEEE, (2003)
6. Microsoft: Speech to text, <https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text>. Último acceso: agosto 2024
7. Improve recognition accuracy with phrase list - Azure AI services | Microsoft Learn, <https://learn.microsoft.com/en-us/azure/ai-services/speech>. Último acceso: agosto 2024
8. Spacy. Models and languages. <https://spacy.io/usage/models>. Último acceso: agosto 2024
9. Honnibal, M. and Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing, (2017)
10. Costumero, R., García-Pedrero, Á., Gonzalo-Martín, C., Menasalvas, E., and Millan, S. Text analysis and information extraction from spanish written documents. En *Brain Informatics and Health: Int. Conference, Proceedings*, pág. 188–197, Springer, (2014)
11. Le, D., Licata, K., and Provost, E. M. Automatic paraphasia detection from aphasic speech: A preliminary study. En *Interspeech*, pág. 294–298, (2017)
12. MacWhinney, B., Fromm, D., Forbes, M., and Holland, A. Aphasiabank: Methods for studying discourse. *Aphasiology*, 25(11):1286–1307, (2011)
13. Cherney, L. R. Oral reading for language in aphasia: Impact of aphasia severity on cross-modal outcomes in chronic nonfluent aphasia. En *Seminars in speech and language*, vol. 31, pág. 042–051. Thieme Medical Publishers, (2010)
14. Cherney, L. R., Halper, A. S., Holland, A. L., and Cole, R. Computerized script training for aphasia: Preliminary results, (2008)
15. Adams, J., Bedrick, S., Fergadiotis, G., Gorman, K., and Van Santen, J. Target word prediction and paraphasia classification in spoken discourse. *BioNLP*, pág. 1–8, (2017)
16. Rivera, B., Becker, P., Olsina, L. Patrones de Estrategia para Proyectos de Medición, Evaluación y Mejora. In *CIbSE*, pag. 166, (2015)