

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

Sofía Florencia Rodríguez

Universidad Nacional de La Pampa, Facultad de Ingeniería

Resumen En algunas demencias neurodegenerativas, las personas pueden 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 las pruebas y 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 (aunque es altamente prescripta) respecto a su eficacia. Este trabajo presenta el desarrollo de una aplicación web, 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 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. Desde el punto de vista tecnológico, se combinaron algoritmos que transcriben discurso oral a texto y de reconocimiento de partes de un discurso.

Palabras clave: Deterioro Neurodegenerativo · Lenguaje Natural · Computación en la Nube.

1. Introducción

Actualmente, la demencia es causada con mayor frecuencia por enfermedades neurodegenerativas. Una persona con demencia experimenta una variedad de síntomas que son lo suficientemente graves como para impedir sus actividades cotidianas y pueden afectar la memoria, el razonamiento, la ejecución y las habilidades sociales. Alrededor del 20% de ellas son consideradas demencias frontotemporales [4]. La demencia frontotemporal (DFT) es una enfermedad neurodegenerativa que afecta especialmente a los lóbulos frontales y temporales del cerebro. Éstas son las áreas que están vinculadas al lenguaje, el comportamiento y la personalidad. La mayoría de las veces, se desconoce el motivo de estas alteraciones, aunque las anomalías genéticas pueden desempeñar un papel importante. Dependiendo qué sección del cerebro esté lesionada, aparecerán

diferentes signos y síntomas: algunos pacientes con demencia frontotemporal experimentan cambios de personalidad notables, siendo impulsivos, emocionalmente fríos o socialmente inadecuados, mientras que otros pierden la capacidad de hablar correctamente. De acuerdo a los síntomas predominantes de la DFT, se distinguen tres síndromes clínicos principales: la DFT variante conductual, la demencia semántica y la afasia primaria progresiva no fluente [3]. En las últimas dos, se sufre el deterioro o pérdida del habla, ocasionando los siguientes problemas: [2]:

- Dificultad creciente para usar y entender el lenguaje escrito y hablado.
- Dificultad para nombrar cosas, posiblemente reemplazando una palabra específica con una palabra más general.
- Desconocimiento del significado de las palabras.
- Tener un habla vacilante que puede sonar telegráfica.
- Cometer errores en la construcción de frases.

La Afasia Progresiva Primaria (APP) es el término utilizado para describir las dificultades del lenguaje que empeoran gradualmente y, por lo tanto, afectan la comunicación de quienes la padecen. Puede tener un impacto en la comprensión del lenguaje, el habla y la escritura. La detección y caracterización de las alteraciones del lenguaje juega un papel cada vez más importante en la identificación y diagnóstico de muchas enfermedades neurodegenerativas. Estos déficits del lenguaje se manifiestan por lo general, en etapas tempranas, sin mayores indicadores más que alguna latencia en el habla. Un análisis de la evolución del deterioro del lenguaje puede ser considerado como parte integral de la evaluación cognitiva de los pacientes afectados por alguna variedad de esta enfermedad neurodegenerativa [1].

1.1. Problema detectado

Muchas enfermedades neurodegenerativas, se identifican y diagnostican cada vez más mediante la detección y caracterización de alteraciones del lenguaje. Al evaluar las capacidades cognitivas de un paciente que tiene alguna forma de esta enfermedad neurodegenerativa, es importante considerar cómo ha cambiado el deterioro del lenguaje con el tiempo.

Actualmente, la mayoría de las evaluaciones cognitivas estándares que se realizan a estos pacientes, permiten conocer ciertos indicadores en relación a distintas dimensiones, como la independencia, la expresividad, las emociones, la identificación de 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 en función de patrones léxicos de acuerdo a la patología detectada. Y si bien, tanto la lectura en voz alta como el diálogo continuo, son indicados como “el mejor remedio”, actualmente no cuentan los profesionales con análisis cuantitativos permanente de estas habilidades a fines de contrastar muchos casos y abordar conclusiones que les permitan mejorar las terapias ofrecidas a estos pacientes. Aún más, usualmente, no es objetivo de las terapias mejorar la lectura oral [3]. Poco se analiza en pacientes con APP y DFT el discurso sintáctico, morfosintáctico, léxico

semántico, así como la fonética, por mencionar algunos aspectos lingüísticos que caracterizan el lenguaje. El tratamiento de esta enfermedad neurodegenerativa sigue siendo sintomático o de soporte [3], pero con escasas herramientas para monitorear el deterioro del habla.

2. Solución Propuesta

El presente trabajo presenta el desarrollo de una aplicación web, denominada “A-e-i!”, de arquitectura distribuida, que utiliza servicios ubicados en servidores en la nube para transcribir audios de lecturas pacientes con deterioro neurodegenerativo derivado en APP. 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.

Desde el punto de vista tecnológico, se combinaron algoritmos de aprendizaje automático o *machine learning* como los relacionados a la transcripción de discurso oral a texto o *speech-to-text* y de reconocimiento de partes de un discurso dentro de un texto (que permiten comparar y detectar omisiones, similitudes y nuevas ocurrencias de palabras comparando con un texto original). La herramienta se implementó bajo una arquitectura REST distribuida que consume servicios de reconocimiento del discurso, alojados en plataformas en la nube y se propone un despliegue de la misma usando servicios de infraestructura también en la nube. El almacenamiento de los resultados en una base de datos permitirá una evaluación posterior basada en indicadores que interpretarán los valores medidos a lo largo del tiempo.

El propósito final de la herramienta “A-e-i!”, es asistir a los profesionales fonoaudiólogos que atienden pacientes con deterioro neurodegenerativo, en esta instancia, enfocados en aquellos con DFT con consecuencias de APP. Así, el principal beneficio, desde la perspectiva de los profesionales de la salud, es el de facilitarles el monitoreo del tratamiento de estos pacientes en relación al deterioro del discurso hablado. Y poder arribar a conclusiones más precisas respecto a esta afectación.

Básicamente, lo que se pretende con “A-e-i!” es que sea una herramienta tanto para el paciente (en supervisión de un acompañante) y al profesional que lo atiende (fonoaudiólogos). El paciente, tiene asignada la lectura de algún texto, acordado con el profesional y procede a grabar su lectura, los días indicados. Cuando termina la grabación, la aplicación asocia ese audio a la cuenta del profesional que lo atiende, se realiza la transcripción del audio a texto y el consecuente procesamiento del lenguaje natural. Se compara con el texto base de referencia, el cual está almacenado y se miden distintos atributos partir de las métricas diseñadas. Finalmente, el profesional dispone de los resultados de la medición, en forma de gráficos, donde podrá ir viendo 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).

Asimismo, se pretende optimizar los tiempos de los profesionales de la salud y permitir hacer controles rigurosos ya que si la tarea de grabar los audios, transcribirlos y analizarlos debiera realizarlos el mismo profesional o alguna persona, en el centro de salud, llevaría un tiempo considerable por lo que no solamente se incurriría en costos mayores sino que también se deberían realizar controles con poca frecuencia. Con esta aplicación es posible que el paciente grave los audios desde su hogar, con la compañía y en el momento del día que desee. La transcripción y el análisis del texto se realizan de forma automática y solamente es facultad del profesional, acceder a los resultados generales y elaborar una conclusión a partir de ellos.

3. Desarrollo de la Solución

La primera tarea que se llevó a cabo fue la de realizar un relevamiento de tecnologías actuales de procesamiento sintáctico y semántico y análisis del lenguaje natural en la nube. Así, se decidió utilizar *SpaCy* [5], que es una librería de procesamiento de lenguaje natural.

A la vez que se realizó la comparativa de tecnologías, se realizó una revisión de la literatura para determinar trabajos relacionados con el seguimiento y la evaluación del lenguaje en pacientes con APP. Este paso fue importante para analizar la viabilidad del desarrollo de la aplicación.

Una vez exploradas las tecnologías de voz y procesamiento de lenguaje natural y analizado el estado del arte, se continuó con el diseño de la arquitectura distribuida de microservicios en la que se estructura la aplicación A-e-i!. Para esto, se implementaron los microservicios que dan soporte al procesamiento de lenguaje natural y reconocimiento de voz y se diseñó e implementó una aplicación web prototipo que consuma los servicios web de la arquitectura distribuida diseñada. Una vez terminado el desarrollo se llevó a cabo una prueba de concepto con un paciente real.

3.1. Arquitectura de A-e-i!

La arquitectura de la aplicación se compone de cuatro componentes, cuyo esquema se ilustra en la Figura 1.

- Componente *Frontend*: Para implementar la interfaz de usuario del navegador se utilizó Next.js ¹ junto a Chakra UI ². Next.js es un framework frontend basado en React para crear interfaces de usuario interactivas. Brinda elementos básicos para crear aplicaciones web rápidas. Next.js propone un marco de desarrollo para resolver los requisitos comunes de las aplicaciones, como el enrutamiento, la obtención de datos, las integraciones.

¹ <https://nextjs.org/>

² <https://chakra-ui.com/>

- Componente *Backend*: La lógica de la aplicación está diseñada como servicios web RESTful para que el cliente (navegador que despliega el *frontend*) obtenga la información necesaria para ser consultada o modificada por el usuario. Para el desarrollo de estos servicios se utilizó el módulo de Python conocido como Flask, que es un microframework puesto que no incluye un ORM (*Object-Relational Mapping*) ni otras funcionalidades como el ruteo o un motor de plantillas. Se optó por este microframework puesto que solo se requería implementar la API REST; por esto, un framework más robusto como Django, sería innecesario.
- Base de datos: Para el almacenamiento de los datos de A-e-i! se eligió el servicio de *Cloud SQL* por lo que se debió configurar una cuenta académica en *Google Cloud* y proceder a instalar una base de datos de tipo SQL, MySQL.
- Componente de Reconocimiento automático de voz: Es el programa que recibe la transcripción del audio de una lectura y lo analiza comparando con un texto original. Los resultados del análisis se almacenan en la base de datos. Para esto se utilizó un servicio en la nube llamado *Azure Speech Recognition*.

Por otro lado, el despliegue de la aplicación fue realizado por el servicio de computación sin servidor o *serverless* que provee la plataforma de Google Cloud de Google App Engine denominado Google Cloud Run. Este servicio de la nube es del tipo *Container as a Service*, llamado así porque el usuario no tiene que desplegar ningún orquestador de contenedores. La elección fue impulsada principalmente por el hecho de que el primer millón de peticiones es gratis. Es por eso que, en un principio y debido a la cantidad de usuarios activos que pretende tener una aplicación desarrollada, el coste del uso de *Google Cloud Platform* para el despliegue recaería en la base de datos únicamente. Con esta arquitectura distribuida, tanto el despliegue de la aplicación, la gestión de la base de datos y los servicios de reconocimiento automático de voz, no requieren de esfuerzos computacionales gracias a la implementación de los mismos bajo los servicios de la computación en la nube.

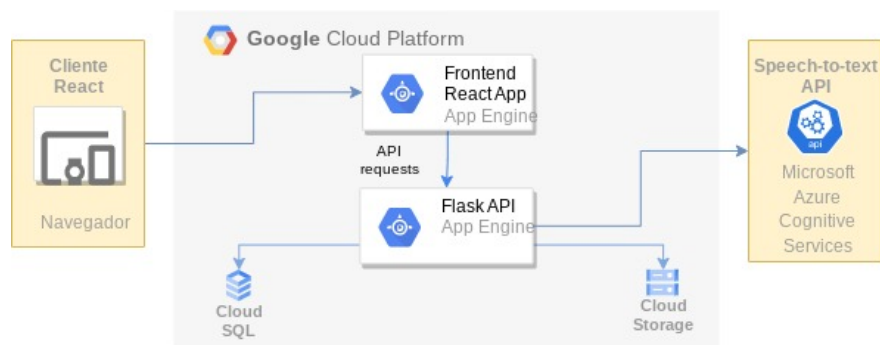


Figura 1. Esquema de la arquitectura de la solución desarrollada.

3.2. Diseño de Casos de Uso

El modelo de casos de uso sirve para representar los requerimientos funcionales del sistema. Un caso de uso especifica el comportamiento de un sistema o de una parte del mismo porque describe las secuencias de acciones que debe llevar a cabo un actor para poder lograr el cumplimiento de una tarea [6]. Los requerimientos funcionales para la aplicación que se desarrolló, ilustrados en el modelo de casos de uso de la Figura 2, son los siguientes:

1. El usuario profesional deberá poder consultar la información de un paciente registrado en el sistema (edades, patologías diagnosticadas, etc.)
2. El usuario profesional debe poder actualizar la información de los pacientes.
3. El usuario profesional debe poder gestionar su propia información.
4. Los usuarios pacientes deben poder grabar sus audios de lecturas.
5. El sistema permitirá la transcripción de audio a texto.
6. El sistema permitirá el análisis del texto proveniente de los audios mediante el uso de procesamiento de lenguaje natural devolviendo resultados tanto numéricos como categóricos.
7. El usuario profesional debe poder visualizar los resultados del análisis de los archivos de audio de los pacientes en formato texto o de forma gráfica, idealmente éstos deberían permitir interacciones.
8. El usuario profesional debe poder filtrar los resultados según el indicador que se quiera controlar (por ejemplo, mostrar solo la evolución de la cantidad de palabras enunciadas correctamente).
9. El sistema debe contar con un módulo de autenticación para tener el control de los usuarios y para proteger la información almacenada.

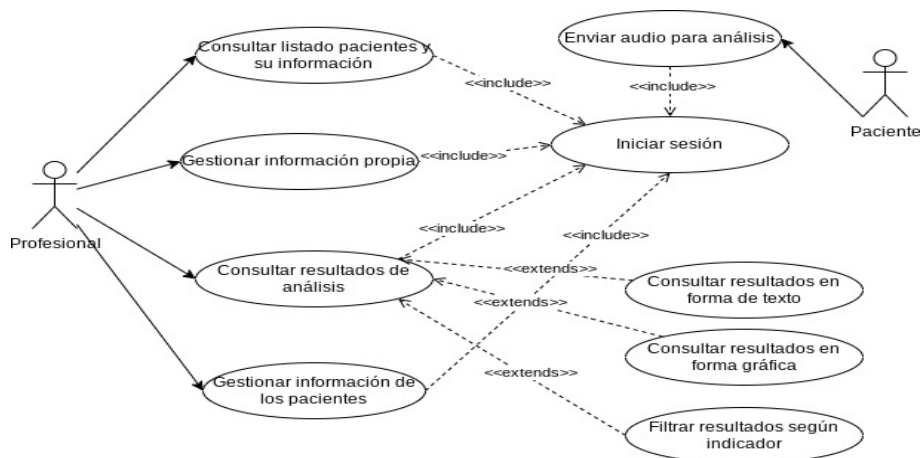


Figura 2. Modelo de Casos de Uso.

3.3. Transcripción de Discurso a Texto

Para la transcripción de los audios a texto es posible utilizar una red neuronal entrenada con audios de personas hispanohablantes. Pero, como esto conlleva muchos costos de tiempo y de procesamiento, se optó por analizar distintos servicios en la nube de *speech-to-text* que proveen la transcripción de audio a texto a través de modelos preentrenados. Como servicio de *Speech-to-text* se seleccionó el que provee Microsoft, denominado *Microsoft Azure Cognitive Services*. [7]. Para lograr la integración de este servicio con la API REST desarrollada en Flask, se llevaron a cabo los siguientes pasos:

1. Crear una suscripción en Azure.
2. Crear un recurso de voz en Azure Portal. ³
3. Obtener la clave y la región del recurso de voz asignado. Deben configurarse como variables de entorno bajo los nombres `SPEECH_KEY` y `SPEECH_REGION`.
4. Instalar Speech SDK.
5. Implementar el algoritmo de transcripción. Se deben configurar ciertos parámetros como por ejemplo el código del lenguaje.

Implementación del algoritmo de transcripción Para poder usar el servicio de transcripción de discurso de *Azure Cognitive Services* hay que importar el paquete *speech*.

```
import azure.cognitiveservices.speech as speechsdk
```

Como primer paso, se debe crear una instancia de configuración de voz (`speechconfig`) con una clave de suscripción y una región de servicio especificadas.

```
speech_key, service_region = "868eae0b1f0d4db2b743ed3eead1a9d4", "eastus"
speech_config = speechsdk.SpeechConfig(subscription=speech_key,
region=service_region)
speech_config.speech_recognition_language = languageCode
```

Luego se debe señalar el archivo de audio que se quiere transcribir creando una instancia de `AudioConfig`.

```
audio_input = speechsdk.AudioConfig(filename=audio_filename)
```

Para audios de no más de 15 segundos se puede usar el reconocimiento de una sola captura. Para audios más extensos, se debe usar el reconocimiento continuo. En este trabajo se optó por usar el reconocimiento continuo ya que lo más probable es que la duración de los archivos de audios supere los 15 segundos. El reconocedor de discurso continuo se define de la siguiente manera, con la configuración previamente establecida y el audio a transcribir como parámetros:

```
speech_recognizer = speechsdk.SpeechRecognizer(speech_config=speech_config,
audio_config=audio_input)
```

³ Un recurso de Azure es un elemento administrable que está disponible para su uso.

Por otro lado, se tiene que crear una variable booleana para administrar el estado del reconocimiento de voz. Esta variable señala si el reconocimiento ha sido finalizado, por lo que al principio se inicializa en “*False*”. El nombre de la variable será “*done*”.

```
done = False
speech_recognizer.start_continuous_recognition()
while not done:
    time.sleep(.5)
```

Por último, en el código hay que especificar cómo se conectan las devoluciones de llamadas a eventos.

```
""" Función que detiene el reconocimiento """
def stop(evt):
    speech_recognizer.stop_continuous_recognition()
    done = True

speech_recognizer.recognizing.connect(lambda evt: print(''))
# Señala los eventos que contienen los resultados intermedios del reconocimiento.
speech_recognizer.recognized.connect(lambda evt: full_transcription
.append(evt.result.text.lower()))
# Señala los eventos que contienen los resultados finales del reconocimiento,
se van agregando los resultados parciales al final de transcripción completa.
speech_recognizer.session_started.connect(lambda evt: print(''))
# Señala los eventos que indican el comienzo de una sesión de reconocimiento.
speech_recognizer.session_stopped.connect(lambda evt: print(''))
# Señala los eventos que indican el final de una sesión de reconocimiento.
# Detienen el reconocimiento de discurso
speech_recognizer.session_stopped.connect(stop)
speech_recognizer.canceled.connect(stop)
continuo si la sesión se detiene o se cancelan eventos
```

3.4. Procesamiento de Lenguaje Natural

El procesamiento de lenguaje natural, (conocido como NLP por sus siglas en inglés de *Natural Processing Language*), es un área intersección entre las ciencias de la computación, la inteligencia artificial y la lingüística computacional, que permite que se puede estructurar el lenguaje para analizarlo [8]. En este trabajo, la incorporación de NLP será de utilidad una vez obtenida la transcripción de los audios de los pacientes. De esta manera, es posible analizar estas transcripciones en función de las estructuras gramaticales del idioma español.

Para realizar el procesamiento de la transcripción obtenida se usó reconocimiento de partes de un discurso con *SpaCy*. *SpaCy* es un software comercial de código abierto publicado bajo la licencia MIT [5]. Brinda la posibilidad de realizar etiquetado, análisis sintáctico, reconocimiento de entidades nombradas,

clasificación de texto según su función en el discurso y aprendizaje multitarea con transformadores preentrenados. Además, permite generar diagramas de análisis sintáctico con el visualizador de dependencias llamado *displaCy*.

El modelo preentrenado que se descargó para realizar hacer el análisis en el idioma español se llama *es_core_news_sm*. La codificación “es” se refiere al idioma español, “core” significa que el modelo cumple tareas de propósito general, “news” o “noticias” es el tipo de género literario con el que fue entrenado y “sm” o “small” indica el tamaño del modelo. Cuando se quiere analizar un texto, conviene hacer un preprocesamiento de la entrada de texto eliminando las mayúsculas con la función `lower()` de Python.

Para procesar la entrada de texto proveniente de la transcripción se debe obtener un objeto `Doc`. Un `Doc` es una secuencia de objetos *token*. Luego, el resultado de llamar al objeto `nlp` con la cadena de texto como parámetro devolverá el `Doc` procesado que se necesita. Cada uno de esos objetos *token* tiene información sobre sí mismo y, entre los atributos que se puede consultar, se encuentran:

- *Text*: El texto de la palabra original.
- *Lemma*: La forma base de la palabra.
- *POS*: La etiqueta según la categorización de Universal POS Tag.
- *Dep*: Dependencia sintáctica, es decir, la relación entre tokens.
- *Forma*: La forma de la palabra: mayúsculas, puntuación, dígitos.
- *is_alpha*: Booleano que especifica si un *token* es un carácter alfa o no.
- *is_stop*: Booleano que especifica si un *token* es parte de la lista de las palabras más comunes del idioma.

A continuación, la Tabla 1 ejemplifica los atributos descriptos, con el resultado de analizar la oración “Es un día soleado”.

Tabla 1. Etiquetas de *Parts of Speech*

TEXT	LEMMA	POS	DEP	SHAPE	ALPHA	STOP
Es	ser	AUX	cop	xx	True	True
Un	uno	DET	det	xx	True	True
Día	día	NOUN	ROOT	xxx	True	True
Soleado	soleado	ADJ	amod	xxxx	True	False
S.	.	PUNCT	punct	.	False	False

La etiqueta que más interesa para analizar el contenido sintáctico del texto es la denominada POS. La etiqueta POS (acrónimo de *Part Of Speech* o Parte del discurso) explica cómo se usa una palabra en particular en una oración. Hay ocho posibilidades:

1. Sustantivo
2. Pronombre
3. Adjetivo
4. Verbo

5. Adverbio
6. Preposición
7. Conjunción
8. Interjección

3.5. Diseño de Métricas

Una vez hecha la transcripción del discurso y el análisis sintáctico del texto resultado de la transcripción, se aplicaron las métricas que permitieron medir algunos valores de ese texto. Las métricas arrojarán resultados que se grafican y, posteriormente, servirán para que los profesionales realicen la interpretación adecuada de esos resultados.

Las métricas diseñadas para medir en el texto resultante, son las siguientes:

- Tiempo de duración del audio (#TDA): Es la duración en segundos de cada audio. Como cada paciente tiene un texto designado, se puede analizar el tiempo que se tarda en leerlo en voz alta y, posteriormente, analizar la fluidez de la lectura.
- Cantidad de adjetivos pronunciados correctos (#APC): Cuántos adjetivos se pronunciaron correctamente respecto al texto base de prueba.
- Cantidad de sustantivos pronunciados correctos (#SPC): Permite calcular cuántos sustantivos se pronunciaron correctamente respecto al total de sustantivos que tiene el texto base de prueba.
- Cantidad de verbos pronunciados correctos (#VPC): Permite calcular cuántos verbos se pronunciaron correctamente respecto del total de verbos que tiene el texto base de prueba.
- Cantidad de adjetivos pronunciados (correctos o incorrectos) (#AP): Permite calcular cuántos adjetivos se pronunciaron en total.
- Cantidad de sustantivos pronunciados (correctos o incorrectos) (#SP): Permite calcular cuántos sustantivos se pronunciaron en total.
- Cantidad de verbos pronunciados (correctos o incorrectos) (#VPC): Permite calcular cuántos verbos se pronunciaron en total.
- Cantidad de palabras pronunciadas correctas (#PPC): Permite calcular cuántas palabras correctas se pronunciaron en total sobre el total de palabras del texto base.
- Cantidad de palabras pronunciadas eliminadas (#PE): Permite calcular cuántas palabras que forman parte del texto base no fueron pronunciadas.
- Cantidad de palabras pronunciadas insertadas (#PI): 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): 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.
- Palabras del texto base (#P): Cantidad de palabras que forman parte del texto de referencia.
- Porcentaje de coincidencia (#PC): Es la relación entre el total de palabras pronunciadas correctas (#PPC) y el total de palabras del texto base (#P).

- Word Error Rate ($\#WER$): Es una métrica que se usa comúnmente para evaluar 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 de WER se deriva de la distancia de Levenshtein, cuyo algoritmo se detalla en la fórmula 1, 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.

Por ejemplo, si hay 32 palabras en total en un archivo de transcripción y entre esas palabras habladas, hay 14 errores (entre sustituciones, inserciones y eliminaciones), el WER de esa transcripción sería 0,379. Eso se redondea a 0.38, lo que hace que el WER sea del 38 por ciento.

4. Caso de estudio

A continuación, se describe el caso de estudio realizado que sirvió para poner en práctica el trabajo desarrollado. M.S. es una mujer, jubilada docente que hace unos años comenzó con problemas para hablar. Básicamente, su pensamiento iba más rápido que sus palabras. Luego de varios estudios, determinaron que estaba transitando APP y, las prueba de imágenes en el cerebro daban cuenta de una atrofia frontal-izquierda.

Dado su interés incesante por la lectura y, siendo ésta una práctica sugerida para mejorar las habilidades lingüísticas, le indicaron que practique lecturas en voz alta. Si bien al inicio (hace 4 años) M.S. pudo leer un libro entero en voz alta, la realidad es que la afasia, como su nombre lo indica, fue progresivamente avanzando, al punto que con el correr del tiempo, apenas podía llegar a leer algunos párrafos. Ante esta situación, los profesionales no tenían a disposición algo que le permitiera medir ese retroceso, más que la información aportada por los familiares que, claramente, era subjetiva (“lee más/lee menos”).

La prueba realizada con A-e-i! se desarrolló de la siguiente manera. Primero, se escogió un párrafo de un texto que era familiar para la paciente. Dado su avanzado deterioro neurocognitivo, el texto se seleccionó considerando que sea de lectura fácil y de historia conocida. Así se escogió los dos primeros párrafos de “La Hormiguita viajera” de Constancio C. Vigil. Luego, se diseñó la periodicidad de la prueba. Se estableció una frecuencia de dos días a la semana, a partir de diciembre 2022, siempre considerando primero, el estado emocional de la paciente, para no perturbarla. Por esta razón, la prueba no se ha realizado de manera regular.

El texto que se utilizó para que M.S. leyera es el siguiente: “*Esta hormiguita vive en un hormiguero que hay en el campo y tiene la entrada junto a una piedra grande todos los días sale en busca de lo que pueda ser útil pues es una hormiguita exploradora si encuentra algo que merezca aprovecharse vuelve inmediatamente al hormiguero para dar la noticia entonces salen y son guiadas por ella las obreras que trabajan en dividir y transportar los materiales cierto día encontró nuestra hormiguita una servilleta después de olerla palparla y morderla se dijo que era una hoja tan grande como no había otra y tan seca y desabrida como una piedra pero encima de la servilleta descubrió una montaña de olor apetitoso y de riquísimo gusto que bueno es esto exclamo emocionada*”

Un familiar acompañando a M.S grabó cada una de sus sesiones de lectura de ese texto de referencia. Cada audio se subió a la aplicación, donde se calculan todos los valores de las métricas descritas en la sección anterior.

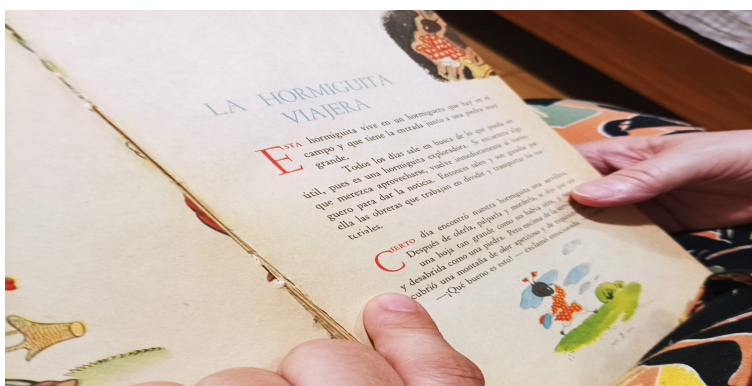


Figura 3. Lectura: “La hormiguita viajera”.

A continuación, se detallan los valores obtenidos para un audio cualquiera. En los días seleccionados para la prueba, se grabó solo un audio. En este caso, los valores medidos corresponden al 21 de diciembre de 2022.

- Fecha: Wed, 21 Dec 2022 00:00:00 GMT
- Transcripción: “*ta hormiguita vive en 1 hormiguero grande que hay y ni el campo y que tiene muchas entradas junto a un no piedra grande todos los dias sali ah bulgar comida del y lo que pueda suceder util pues una y pues lo oradora si encuentra algo kimmy diga aprovecharse vuelve al hormiguero para dar la noticia entonces si ni sabiendo y son guiadas por el tu 10 las obreras quita trabajan en dividir y y se para lo los cierto dia despues de ole la palpa la y morder en la si dijo que era una uh tan grande y pesada que no habia u otra igual mm no y y va igual y tan eh casa y de ah venir a como una piedra pero encima de la cera bicicletta 10 cubrio una monta a nyan de color apetitoso y de dia quisimos gusto que bueno es esto eh la amo*”
- Tiempo de duración del audio (#TDA): 6.019773 s,

- Cantidad de adjetivos pronunciados correctos (#APC): 10,
- Cantidad de sustantivos pronunciados correctos (#SPC): 21,
- Cantidad de verbos pronunciados correctos (#VPC): 23,
- Cantidad de adjetivos pronunciados (#AP): 12,
- Cantidad de sustantivos pronunciados (#SP): 23,
- Cantidad de verbos pronunciados (#VP): 19,
- Cantidad palabras pronunciadas correctas (#PPC): 75,
- Cantidad de palabras eliminadas (#PE): 2,
- Cantidad de palabras insertadas (#PI): 34,
- Cantidad de palabras sustituidas (#PS): 45,
- Porcentaje de Coincidencia (#PC): 61.475
- Cantidad de palabras del texto base (#P): 122
- WER: 0.664

Con cada grabación, la aplicación calcula el valor de todas las métricas diseñadas y grafica dicho valor en distintos gráficos. Así, el profesional que utilice A-e-i! ingresa a su cuenta de usuario, selecciona el paciente que quiere monitorear y puede ver la evolución en el tiempo, de las distintas dimensiones medidas. La Figura 4 muestra el gráfico de los valores medidos para la métrica cantidad de palabras correctas pronunciadas. De igual manera, se pueden seleccionar otras métricas y ver la evolución en el tiempo de los valores medidos. Además, en la Figura 5 se ilustra un gráfico de torta que es la información detallada para un día concreto, es decir, seleccionando un punto del gráfico xey, el gráfico de torta se actualiza con datos del día de la muestra seleccionada.

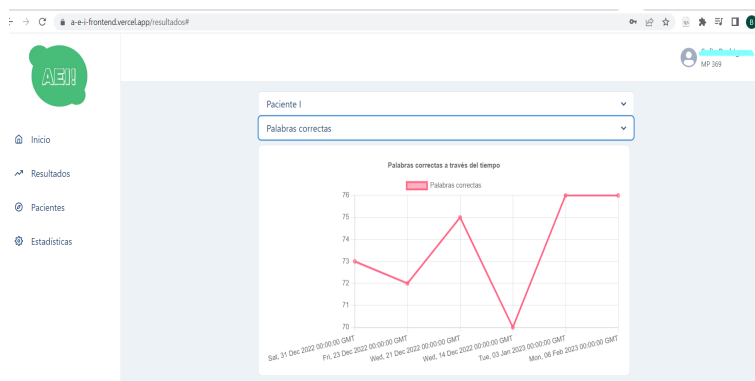


Figura 4. Cantidad de palabras correctas pronunciadas en todas las grabaciones realizadas.

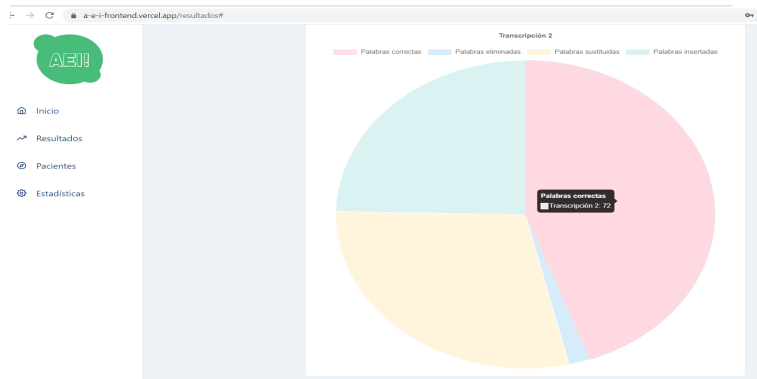


Figura 5. Valores medido para una grabación específica.

5. Conclusiones

Este trabajo fue desarrollado para aprobar la carrera de grado de Ingeniería en Sistemas. El mismo consistió en el diseño e implementación de una aplicación web con arquitectura distribuida de microservicios para dar soporte al procesamiento y análisis del lenguaje natural español de pacientes afásicos, utilizando tecnologías de computación en la nube para consumir servicios y desplegar la aplicación.

Desde el punto de vista de adquisición de conocimientos, realizar este trabajo permitió trabajar y aprender sobre los siguientes temas:

- Diseño e implementación de un microservicio de reconocimiento de voz para transcribir las muestras de lecturas realizadas.
- Diseño e implementación de un microservicio de clasificación de entidades de un texto para estructurar la información de una forma que facilite el análisis sintáctico y semántico.
- Tecnologías de computación en la nube.
- Enfermedades neurodegenerativas y su impacto en el habla.

Es un trabajo inicial, la aplicación es aún un prototipo que, con algunas pruebas con casos reales, ayudará a incrementar las funcionalidades y mejorar las actuales. La intención está enfocada en asistir a profesionales neurólogos, fonoaudiólogos y/o logopedas, para contribuir con el estudio y análisis de la evolución del deterioro del habla de sus pacientes con enfermedades neurodegenerativas que afecten las habilidades lingüísticas.

5.1. Trabajos Futuros

Como trabajo futuro desde el punto de vista tecnológico se prevé continuar con el estudio de tecnologías de procesamiento de lenguaje natural para mejorar la técnica empleada.

Desde la consideración de trabajo interdisciplinario, se prevé trabajar con fonoaudiólogos para que puedan aportar más casos y sumar pacientes para realizar pruebas y seguir validando la utilidad de A-e-i!. Además, junto con estos profesionales, se trabajará en el diseño de indicadores que interpreten los valores medidos y puedan darse recomendaciones o valoraciones sobre la evolución en un determinado período de tiempo. Al momento, la aplicación solo se limita a mostrar valores medidos provenientes de las lecturas, pero sin realizar ningún análisis sobre los mismos. Queda sujeto al profesional, la interpretación de los mismos.

También, a futuro sería interesante contar con otros tipos de enfermedades que afecten el habla, incluso, podría dar soporte a infantes que sufran retrasos en comenzar a hablar pero que no pueden monitorear cómo evoluciona su discurso. Este trabajo abre puertas para muchos otros trabajos interdisciplinarios.

Bibliografía

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. Buckingham Jr, H.W., Kertesz, A.: A linguistic analysis of fluent aphasia. *Brain and language* **1**(1), 43–61 (1974)
3. 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)
4. Henriksson, I., Laakso, K.: Book talk and aphasia: the power of a book. *International Journal of Language & Communication Disorders* **55**(1), 136–148 (2020)
5. Honnibal, M.: Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing — sentometrics research. <https://sentometrics-research.com/publication/72/>, (Accedido el 05/08/2023)
6. Jacobson, I., Booch, G., Rumbaugh, J.: El lenguaje unificado de modelado uml. Disponible en: JACOBSON, I (2000)
7. Microsoft: Speech to text (2023), <https://azure.microsoft.com/en-us/products/cognitive-services/speech-to-text>
8. Nadkarni, P.M., Ohno-Machado, L., Chapman, W.W.: Natural language processing: an introduction. *Journal of the American Medical Informatics Association* **18**(5), 544–551 (2011)