EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de León

“Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial”

Tesis

Que presenta:

Jaime Emilio Zavala Barrios

Para obtener el grado de

Maestro en Ciencias de la Computación

Con la Dirección de:

Dr. Víctor Manuel Zamudio Rodríguez

Revisores

Dr. Carlos Lino Ramírez

Dr. David Asael Gutiérrez Hernández

M.C.S. Claudia Leticia Díaz González

León, Guanajuato

Enero 2023

Instituto Tecnológico de León

“Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial”

Tesis

Que presenta:

Jaime Emilio Zavala Barrios

Para obtener el grado de

Maestro en Ciencias de la Computación

Con la Dirección de:

Dr. Víctor Manuel Zamudio Rodríguez

Revisores

Dr. Carlos Lino Ramírez

Dr. David Asael Gutiérrez Hernández

M.C.S. Claudia Leticia Díaz González

León, Guanajuato Enero 2023

León, Guanajuato, 11/enero/2023

OFICIO No. DEPI-002-2023

C. JAIME EMILIO ZAVALA BARRIOS
PRESENTE

De acuerdo al fallo emitido por la Comisión Revisora, integrada por los: **Dr. Víctor Manuel Zamudio Rodríguez, Dr. David Asael Gutiérrez Hernández, Dr. Carlos Lino Ramírez y MCS. Claudia Leticia Díaz González,** y considerando que cubre todos los requisitos establecidos en los Lineamientos Generales para la Operación del Posgrado del Tecnológico Nacional de México, se autoriza la impresión del trabajo de tesis titulado: **"Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de inteligencia artificial".**

Lo que hacemos de su conocimiento para los efectos y fines correspondientes.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Ciencia Tecnología y Libertad

LIC. PATRICIA DONATO JIMÉNEZ
SUBDIRECTORA ACADÉMICA

SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE LEÓN
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

C.c.p. Expediente

JMCV/CLDG

EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

León, Guanajuato., al 12 de enero del 2023

C. ING. LUIS ROBERTO GALLEGOS MUÑOZ
JEFE DE SERVICIOS ESCOLARES
P R E S E N T E

Por este medio hacemos de su conocimiento que la tesis titulada **" Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial."**, ha sido leída y aprobada por los miembros del Comité Tutorial para su evaluación por el jurado del acto de examen de grado al alumno (a) **C. Jaime Emilio Zavala Barrios** con número de control **M20210116** como parte de los requisitos para obtener el grado de Maestro(a) en Ciencias de la Computación (MCCOM-2011-05).

Sin otro particular por el momento, quedamos de Usted.

A T E N T A M E N T E
COMITÉ TUTORIAL

Dr. Víctor Manuel Zamudio Rodríguez
DIRECTOR

Dr. Carlos Lino Ramírez
REVISOR

Dr. David Asael Gutiérrez Hernández
REVISOR

M.C.S Claudia Leticia Díaz González
REVISORA

INSTITUTO TECNOLÓGICO DE LEÓN

DECLARACION DE AUTENTICIDAD Y DE NO PLAGIO

Yo, ***Jaime Emilio Zavala Barrios*** identificado con No. control ***M20210116*** , alumno (a) del programa de la **Maestría en Ciencias de la Computación**, autor (a) de la Tesis titulada: **" Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial. "** DECLARO QUE:

1.- El presente trabajo de investigación, tema de la tesis presentada para la obtención del título de **MAESTRO (A) EN CIENCIAS DE LA COMPUTACIÓN** es original, siendo resultado de mi trabajo personal, el cual no he copiado de otro trabajo de investigación, ni utilizado ideas, fórmulas, ni citas completas "stricto sensu", así como ilustraciones, fotografías u otros materiales audiovisuales, obtenidas de cualquier tesis, obra, artículo, memoria, etc. en su versión digital o impresa.

2.- Declaro que el trabajo de investigación que pongo a consideración para evaluación no ha sido presentado anteriormente para obtener algún grado académico o título, ni ha sido publicado en sitio alguno.

3.- Declaro que las pruebas o experimentos derivados de esta investigación fueron realizados bajo el consentimiento de los involucrados y con fines estrictamente académicos conforme a criterios éticos de confidencialidad.

Soy consciente de que el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades den la tesis, así como de los derechos sobre la obra presentada.

Asimismo, me hago responsable ante el Tecnológico Nacional de México/Instituto Tecnológico de León o terceros, de cualquier irregularidad o daño que pudiera ocasional por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el trabajo de investigación haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas pecuniarias o legales que se deriven de ello sometiéndome a las normas establecidas en los Lineamientos y Disposiciones de la Operación de Estudios de Posgrado en el Tecnológico Nacional de México.

León, Guanajuato a 10 Enero del 2023

Jaime Emilio Zavala Barrios

ACUERDO PARA USO DE OBRA (TESIS DE GRADO)

A QUIEN CORRESPONDA

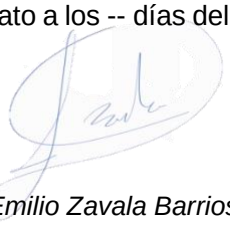
PRESENTE

Por medio del presente escrito, **Jaime Emilio Zavala Barrios** (en lo sucesivo el AUTOR) hace constar que es titular intelectual de la obra denominada**: "Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial"** (en lo sucesivo la OBRA) en virtud de lo cual autoriza al Tecnológico Nacional de México/Instituto Tecnológico de León (en lo sucesivo TECNM/IT León) para que efectué resguardo físico y/o electrónico mediante copia digital o impresa para asegurar su disponibilidad, divulgación, comunicación pública, distribución, transmisión, reproducción, así como digitalización de la misma con fines académicos y sin fines de lucro como parte del Repositorio Institucional del TECNM/ITLeón.

De igual manera, es deseo del AUTOR establecer que esta autorización es voluntaria y gratuita, y que de acuerdo con lo señalado en la Ley Federal del Derecho de Autor y la Ley de Propiedad Industrial el TECNM/IT León cuenta con mi autorización para la utilización de la información antes señalada, estableciendo que se utilizará única y exclusivamente para los fines antes señalados. El AUTOR autoriza al TECNM /IT León a utilizar la obra en los términos y condiciones aquí expresados, sin que ello implique se le conceda licencia o autorización alguna o algún tipo de derecho distinto al mencionada respecto a la "propiedad intelectual" de la misma OBRA; incluyendo todo tipo de derechos patrimoniales sobre obras y creaciones protegidas por derechos de autor y demás formas de propiedad intelectual reconocida o que lleguen a reconocer las leyes correspondientes. Al reutilizar, reproducir, transmitir y/o distribuir la OBRA se deberá reconocer y dar créditos de autoría de la obra intelectual en los términos especificados por el propio autor, y el no hacerlo implica el termino de uso de esta licencia para los fines estipulados. Nada de esta licencia menoscaba o restringe los derechos patrimoniales y morales del AUTOR.

De la misma manera, se hace manifiesto que el contenido académico, literario, la edición y en general de cualquier parte de la OBRA son responsabilidad de AUTOR, por lo que se deslinda al (TECNM/ITLeón) por cualquier violación a los derechos de autor y/o propiedad intelectual, así como cualquier responsabilidad relacionada con la misma frente a terceros. Finalmente, el AUTOR manifiesta que estará depositando la versión final de su documento de Tesis, OBRA, y cuenta con los derechos morales y patrimoniales correspondientes para otorgar la presente autorización de uso.

En la ciudad de León, del estado de Guanajuato a los -- días del mes de Enero del 2023. Atentamente,



Jaime Emilio Zavala Barrios

DEDICATORIA

Dedico este trabajo a mi pareja, padres y hermanos que

me han apoyado en todo este camino con su apoyo,

palabras de aliento y motivación,

gracias por todo .

AGRADECIMIENTOS

Agradezco al CONACYT el apoyo brindado para la realización de estudios de Maestría. Se le agradece al Tecnológico Nacional de México/ IT León y al cuerpo académico de Bioinformática y Tecnología Computacionales (BITEC) las facilidades brindadas para la realización de esta tesis.

De igual manera agradezco al Dr. Carlos Lino y al Dr. David Asael Gutiérrez Hernández quienes durante estos dos años han acompañado a lo largo de estos años desde el plano tanto académico como personal en la realización de esta investigación.

Por último, se le agradece en especial al mi director de tesis el Dr. Víctor Manuel Zamudio Rodríguez por su experiencia, paciencia, asesoramiento y apoyo durante estos años de formación, sin él cual no hubiera sido posible la realización de este trabajo.

TABLA DE CONTENIDOS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

ÍNDICE DE ECUACIONES

ÍNDICE DE ALGORITMOS

ACRÓNIMOS

La Tabla 1 muestra los acrónimos utilizados en este trabajo de investigación.

Acrónimo	Significado	Español
ASR	*Automatic speech recognition*	Sistemas de reconocimiento del habla
EA	*Alzheimer's Disease/Dementia (AD)*	Enfermedad de Alzheimer
EN	*Named entity (NE)*	Entidad nombrada.
GPU	*Graphics Processing Unit*	Unidad de procesamiento gráfico
GRU	*Gated recurrent unit*	Unidad recurrente cerrada
GTTS	*Google Text-to-Speech*	Texto a dicurso - Google API
IDE	*Integrated Development Environment*	Entorno de desarrollo interactivo / Entorno de desarrollo integrado
NER	*Name entity recognition*	Recoconimiento de entidades nombradas.
POS	*Part of Speech*	Partes del discurso
PNL/NLP	*Natural Language Processing*	Procesamiento de lenguaje natura
OMS	*World Health Organization (WHO)*	Organización Mundial de la Salud
RNN	*Recurrent neural network*	Red neuronal recurrente
SDK	*Software Development Kit*	Kit de desarrollo de software
TTS	*Text To Speech*	Texto a discurso
WAV	*Waveform Audio Format*	Formato de audio Waveform

Tabla 1.0.1 *Tabla de Acrónimos.*

GLOSARIO

C

Corpus

Un corpus se puede definir como una colección de documentos de texto. Puede pensarse como un grupo de archivos de texto en un directorio, a menudo junto con muchos otros directorios de archivos de texto., 34

D

Dart

Es un lenguaje de programación de código abierto, desarrollado por Google XE "Google:Es una compañía principal subsidiaria de la estadounidense Alphabet cuya especialización son los productos y servicios relacionados con internet, software, dispositivos electrónicos y otras tecnologías." ., 29

E

EA

Enfermedad de Alzheimer, 27

EN

Entidad nombrada. El Reconocimiento de entidades nombradas es una tarea de extracción de información que busca localizar y clasificar en categorías predefinidas, como personas, organizaciones, lugares, expresiones de tiempo y cantidades, las entidades nombradas encontradas en un texto, 29

F

Firebase

Es una plataforma para el desarrollo de aplicaciones web y aplicaciones móviles lanzada en 2011 y adquirida por Google en 2014., 30

Flutter

Es un SDK de código fuente abierto de desarrollo de aplicaciones móviles creado por Google. Suele usarse para desarrollar interfaces de usuario para aplicaciones en Android, iOS y Web así como método primario para crear aplicaciones para Google Fuchsia. , 29

G

Google

Es una compañía principal subsidiaria de la estadounidense Alphabet cuya especialización son los productos y servicios relacionados con internet, software, dispositivos electrónicos y otras tecnologías., 29

GPU

Es un procesador formado por muchos núcleos más pequeños y especializados. Al trabajar conjuntamente, los núcleos ofrecen un desempeño masivo cuando se puede dividir una tarea de procesamiento y es procesada por muchos núcleos., 88

I

Incidencia

Es la cantidad de casos nuevos de una enfermedad, un síntoma, muerte o lesión que se presenta durante un período de tiempo específico, como un año. La incidencia muestra la probabilidad de que una persona de una cierta población resulte afectada por dicha enfermedad., 27

IOS

Resumen

En México se estima que aproximadamente un millón 300 mil personas padecen la enfermedad de Alzheimer, cuya cifra representa ser entre el 60 y 70 por ciento de los mexicanos diagnosticados con demencia. El Alzheimer afecta con mayor frecuencia a las personas mayores de 65 años. Actualmente no existe una cura para la enfermedad. En esta investigación se plantea la implementación del procesamiento del lenguaje natural para el análisis de recuerdos y la extracción de información con las técnicas de identificación de partes de discurso y reconocimiento de identidades, con la finalidad de preservar los recuerdos en un repositorio digital y enriquecer las historias con la sincronización de material multimedia.

Palabras Clave: aprendizaje máquina, minería de datos, minería de texto, Tokenización, reconocimiento de entidades, reconocimiento de identidades, storytelling …

Abstract

In Mexico it is estimated that approximately one million 300 thousand people suffer from Alzheimer's disease, which figure represents between 60 and 70 percent of Mexicans diagnosed with dementia. Alzheimer's most often affects people over the age of 65. There is currently no cure for the disease. This research proposes the implementation of natural language processing for the analysis of memories and the extraction of information with the techniques of identification of parts of speech and recognition of identities, to preserve memories in a digital repository and enrich the stories with the synchronization of multimedia material.

Keywords: machine learning, data mining, text mining, storytelling, tokenization, named-entity recognition, parts of speech tagging ...

Introducción

A través de las siguientes reflexiones, se invita al lector a conocer sobre el trabajo en progreso de almacenamiento de recuerdos en un repositorio digital, apoyado con estrategias de procesamiento de lenguaje natural y sincronización de contenido multimedia. La investigación en desarrollo tiene como finalidad la preservación de memorias en pacientes con Alzheimer, a través de un repositorio digital. Este repositorio permite la aplicación técnica de procesamiento de lenguaje natural para la clasificación y extracción de la información contenida.

El Alzheimer es un tipo de demencia que afecta directamente a la memoria, el comportamiento y el pensamiento, al ser una enfermedad crónica los síntomas en la última fase avanzan hasta el punto de que logran interferir con las tareas cotidiana.

A inicios del siglo XX, se describió por primera vez por Alois Alzheimer las lesiones neuropatológicas asociadas con un tipo particular de demencia senil. Este hecho fue la pauta para establecer un nuevo concepto etiológico que en la actualidad se le conoce como demencia por enfermedad de Alzheimer [1].

Dicha enfermedad aumenta gradualmente, por lo cual desde su detección se observa como los síntomas de demencia empeoran progresivamente durante varios años. Una de las características en etapas tempranas es la pérdida de memoria sobre detalles pequeños y es considerada como una pérdida de memoria tenue. Sin embargo, las personas que llegan a padecerla en las etapas más avanzadas desarrollan la pérdida de capacidad para mantener una conversación y posteriormente, en la última etapa de la enfermedad, la posibilidad de responder a su entorno [2].

En promedio, una persona con Alzheimer vive de 4 a 8 años después del diagnóstico, pero con una probabilidad de hasta 20 años, dependiendo de múltiples factores como el cuidado, la detección temprana de la enfermedad, el uso de los medicamentos entre otros, pero como factor principal la evolución de la enfermedad [1].

Es entonces que este trabajo gira en torno a la preservación de la memoria de las personas diagnosticas con la enfermedad, mediante la ejecución de estrategias digitales e implementación de algunos algoritmos de inteligencia artificial para el análisis del contenido, extracción de

información, sincronización de contenido multimedia con la narrativa automatizada (videos, fotos, grabaciones, etc.), así como el mapeo de relaciones de datos.

La conservación de la memoria de una persona es un tema muy relevante en el contexto del Alzheimer, pues tal como se abordó con antelación, en las primeras etapas de la enfermedad, la persona comienza a olvidar detalles menores de su día y mientras que la enfermedad avanza, de manera progresiva, aquello que parecía claro y fácil de recordar ahora cuesta completar o recuperar fragmentos y datos de su memoria [1].

La memoria puede sufrir alteraciones por diferentes factores tales como los emocionales [3], entendiéndose estos como aquellos que protegen a la persona de una mala experiencia, por otro lado, el tiempo también permite agregar o remover detalles del suceso original [4].

La memoria puede estimularse y recordar fragmentos olvidados gracias a estímulos sensoriales, o bien con ayuda de material audiovisual, por ejemplo; al observar una fotografía, escuchar una canción o audios, así como ver un algún video pues estos materiales funcionan como detonantes que permiten revivir o recordar experiencias [5].

ESTRUCTURA DE TESIS.

Capítulo 1

Dentro del capítulo 1 se explica la justificación del estudio, se plantea los objetivos buscados durante la investigación reflejada en la presente tesis, el planteamiento del problema, la hipótesis, la viabilidad de a investigación, las limitaciones encontradas durante la investigación. Al final se expone una breve introducción de la metodología seguida que se desglosa más a detalle en el capítulo 4.

Capítulo 2

En este capítulo se expondrán los trabajos de investigación recopilados y analizados que sirvieron como base para el desarrollo del proyecto, algunos antecedentes relevantes dentro del campo de procesamiento de lenguaje de datos, de la narrativa de historias.

Capítulo 3

En este capítulo se muestra el marco teórico de la investigación donde se explicarán los términos y conceptos utilizados durante el desarrollo experimental del proyecto dando formalidad necesaria a los conceptos plasmados dentro del este presente documento.

Se abarca los temas de procesamiento de lenguaje natural, tipos de aplicaciones para dispositivos móviles, se explica las diversas librerías implementadas dentro del desarrollo del estudio.

Capítulo 4

En este capítulo se explica a detalle la metodología implementada con cada una de las 5 etapas en que se dividió la investigación.

Capítulo 5

Se desglosan las pruebas realizadas dentro de la metodología experimental, para la captura de documentos y grabaciones en el repositorio digital.

Se muestras las tareas de procesamiento de lenguaje natural implementadas con la librería SpaCy y la creación de los arreglos de datos para cada palabra dentro del documento seleccionado.

Por último, se pone a prueba el proceso de sincronización de contenido multimedia con los datos incorporados dentro del repositorio digital.

Capítulo 6

En este capítulo se exponen los resultados conseguidos, se muestra la aplicación obtenida y se abre un espacio para discusión sobre los resultados y observaciones que se consiguieron.

Capítulo 7

En este último capítulo se expone las conclusiones a las que se llegó en este trabajo de investigación y se presenta las recomendaciones para dar seguimiento a la tesis y se plantea los trabajos futuros del que pueden partir de la presente investigación.

Anexos

Al presente del documento se localiza las publicaciones realizadas dentro del programa.

Capítulo 1.

Presentación de la investigación

1.1 Justificación de estudio

"La demencia es un nombre colectivo para los síndromes cerebrales degenerativos progresivos que afectan la memoria, el pensamiento, el comportamiento y las emociones" [6]

A partir de los datos oficiales se observa como México se está convirtiendo en un país con más personas mayores que infantes menores de 5 años, se prevé que para el año 2050, la proporción de las juventudes de 15 años en la población total disminuirá de 30.0% a 17.4%. De este modo, el incremento en la población de personas adultas mayores iría en aumento con la utilización de los servicios de salud destinados a la atención de enfermedades propias del envejecimiento como las demencias [7].

La demencia es uno de los factores que más contribuyen a la discapacidad ya la dependencia en las personas adultas mayores [1]. La Enfermedad de Alzheimer (EA), es el tipo de demencia más frecuente y actualmente se conoce que existe una prevalencia del 7.3% y una incidencia de 27.3 (1000 personas/año) de la población adulta mayor mexicana para la EA [8] . En el sitio oficial de la secretaria de salud de México se indica que existe aproximadamente un millón 300 mil personas padecen la enfermedad de Alzheimer, cifra que representa entre 60 y 70 por ciento de los diagnósticos de demencia en el país y que afecta con mayor frecuencia a las personas mayores de 65 años [1] . La enfermedad de Alzheimer se genera debido a cambios en el cerebro por la presencia de la proteína llamada beta amiloide, que se acumula frecuentemente en el lóbulo temporal. Dicha toxina provoca inflamación y muerte progresiva de neuronas [1]. La Organización Mundial de la Salud (OMS) calcula una cifra aproximada de 60 millones de personas a nivel global que viven con Alzheimer, de las cuales 8.1 por ciento son mujeres y 5.4 por ciento, hombres mayores de 65 años [1].

Para una atención adecuada de la persona con enfermedad de Alzheimer, es necesario el diagnóstico temprano y tratamiento integral de alta especialidad que se basa en medicamentos que estimulan y ayudan a prolongar la vida de las neuronas de la memoria para mejorar su calidad de vida. [1].

En el presente trabajo, el procesamiento de recuerdos permitirá a pacientes con Alzheimer poder acceder a recuerdos aun con en el avance de la enfermedad, extraer información de las transcripciones que serán enriquecidas con material multimedia para estimular experiencias y hacerlas más vívidas. Las memorias almacenadas estarán disponibles para ser compartidas con familiares y descendencia, de tal modo que generaciones distintas de una familia puedan conectarse, siempre que la persona dé su consentimiento.

Al preservar las memorias del paciente, a pesar de la perdida de detalles sobre sus recuerdos, podrá revivirlos por medio de la aplicación, como la primera vez que fueron subidos al ser complementados con el contenido multimedia.

El procesamiento de lenguaje natural es una técnica indispensable que permite analizar las transcripciones de los recuerdos, extraer la información y complementarlos con material multimedia relacionado a los datos extraídos. Estos contenidos multimedia pueden ser enriquecidos con archivos subidos de otras experiencias previas, información disponible en la red o con la que contenida en el sistema.

Al almacenar las memorias de la persona dentro de un repositorio digital, se evitará que el paso del tiempo u otros factores como enfermedad, medicamentes, lesiones modifiquen la historia y el recuerdo.

El procesamiento de lenguaje natural se llevará a cabo valiéndose de las transcripciones en formato texto de los audios, controlándose desde la aplicación el formato mp4, y con la captura de recuerdos de manera escrita se aceptarán archivos en diferentes formatos: PDF, Text. y word, para su posterior conversión en formato de texto plano.

La conservación del audio permitirá validar la precisión de las transcripciones escritas con respecto al audio original e identificar palabras que al sistema le cuesta diferenciar por la velocidad del habla de cada persona.

1.2 Objetivos

1.2.1 Objetivo General

Identificar entidades nombradas y partes del discurso por medio del la implementación de técnicas de procesamiento de lenguaje natural (PNL) dentro de archivos con formato de texto plano que contemplan las transcripciones de un repositorio de grabaciones y el corpus integrado por documentos en un repositorio digital con la finalidad de llevar acabo la sincronización de los dentro de los textos con elementos de material multimedia.

1.2.2 Objetivo específicas.

Este triple objetivo general se ha concretado mediante el planteamiento de los objetivos específicos desglosados a continuación:

- Capturar de audios y documentos dentro de un repositorio digital con el soporte de una aplicación móvil.
- Procesamiento de audios para la generación de transcripciones en texto plano.
- Estudiar y seleccionar un corpus de historias, fabulas y cuentos cortos que permitan evaluar dentro de la estructura de la historia los componentes, las cronologías existentes, la representatividad de los momentos, el esquema de aparición de las entidades nombradas.
- Reconocer, clasificar y extraer automáticamente entidades nombradas (EN), partes del discurso y sus relaciones en el corpus seleccionado. Subsiguientemente, realizar el llenado de la plantilla para el proceso de automatización de contenido.
- Construir una base de datos general como sistema de almacenamiento para el contenido multimedia para la recuperación de las entidades nombradas y también construir una base de datos personalizada de acuerdo con el perfil y material privado del usuario.
- Diseñar una aplicación para dispositivos móviles por medio del SDK Flutter y el lenguaje Dart, para ser enlazada con la base de datos Firebase, implementando las funciones de autentificación de usuarios por correo electrónico.

- Utilizar software y lenguajes de código abierto como: Python, Dart, Flutter y Firebase.
- Sincronización y generación de narrativas enriquecidas con material multimedia gracias a la identificación de identidades nombradas.

1.3 Planteamiento del problema/Preguntas de la investigación

Teniendo en cuenta los objetivos desglosados en esta investigación y la problemática expuesta anteriormente dentro de la justificación, se han planteado las siguientes preguntas de investigación:

- ¿Qué dificultades presenta en la captura de historias y recuerdos por medio de una aplicación móvil hibrida y cuáles son los formatos que se adecuan para su almacenamiento?
- ¿Cómo puedo generar transcripciones fiables por medio de las grabaciones o el procesamiento de discurso oral a escrito en tiempo real y por medio de librerías de código abierto son una opción más optima?
- ¿Qué dificultades presenta las funciones de reconocimiento y clasificación de las partes de discurso y entidades nombras dentro de los corpus de prueba y de las historias ingresadas por la aplicación utilizando la librería SpaCy de procesamiento de lenguaje natural?
- ¿Cómo pueden almacenarse la información extraída de las historias preservando las partes del discurso y las entidades nombradas?
- ¿Cómo se puede generar unas relaciones entre una entidad nombrada y contenido multimedia almacenado en la base de datos del perfil del usuario o la base de datos generales de forma que sean accesibles y reutilizables para la tarea de sincronización de material multimedia?
- ¿Las relaciones semánticas establecidas entre la detección de identidades pueden ser consolidadas por la implementación de n-gramas para la extracción de información de las historias para posteriormente realizar la tarea de completar el llenado de la plantilla de datos?

- ¿Los programas, librerías y lenguajes de programación de código abierto pueden ofrecer soluciones al proceso de análisis y extracción de datos del corpus o documento seleccionado?

1.4 Viabilidad de la investigación

El estudio resulta viable, ya que se cuenta con librerías de código abierto para el procesamiento de lenguaje natural permitiendo acortar los tiempos de desarrollo. En segundo plano el manejo de un diseño de aplicación hibrido permite cubrir los dos sistemas operativos para dispositivos móviles con mayor número de usuarios, finalizando que al implementar la base de datos Firebase se reduce los tiempos de desarrollo de infraestructura de bases de datos.

1.5 Hipótesis

1.5.1 Hipótesis de investigación

Es posible generar una plataforma que permita asociar texto o voz con contenido multimedia mediante algoritmos de inteligencia artificial avanzados, priorizando asociaciones que impacten positivamente el bienestar del usuario.

1.6 Limitaciones

En relación con las limitaciones dentro de la investigación no existieron limitaciones del factor económicas al implementar herramientas, bases de imágenes, material audiovisual y software de código abierto. Sin embargo, el haber implementado dichas librerías para acortar los tiempos de programación buscando habilitar funciones como la conversión del discurso hablado al escrito y caso contrario aplicando la conversión del formato escrito a discurso oral el cual tomarían demasiado tiempo de desarrollo; una vez analizado lo anterior se visualizó que se podría tener algunas desventajas en términos de la sincronización para el procesamiento en tiempo real.

1.7 Metodología

Esta tesis es el producto de una investigación de tipo cuantitativo experimental la cual se realiza mediante un estudio de caso con el objetivo de dar con las respuestas a las preguntas de investigación planteadas.

En la primera fase de la investigación se ha dado un enfoque de investigación aplicada para realizar el desarrollo de una aplicación multiplataforma que permitiera la captura de historias y recuerdos en formato oral y escrito de usuarios.

La segunda parte de la investigación se limita al manejo de los datos vía formatos electrónicos para su conversión a texto en formato plano de manera electrónica para la posterior implementación de técnicas de procesamiento de lenguaje natural.

Para la tercera fase se desarrolla la extracción de las partes del discurso y la detección de entidades nombradas implementando en una plantilla que permita organizar los datos del texto seleccionado para ser la base del proceso de sincronización con contenido multimedia.

En la cuarta y quinta fase de la investigación se han realizado de manera experimental o exploratoria mediante la aplicación de una serie de métodos empíricos basados en el análisis del comportamiento e interacción de las partes que componen la estructura del texto y el manejo de las bases de datos del contenido multimedia para el elemento seleccionado del corpus, entendiendo al corpus como el conjunto de archivos en formato texto dentro de la base de datos que contienen el cumulo de historias y recuerdos.

La quinta fase es la encargada de reflejar los resultados de sincronización del material multimedia y la historia dentro de la aplicación de dispositivos móviles.

Dentro de este método se encuentran estructurados cinco etapas que se ejecutan según se representa en la siguiente **Figura 1.1.**

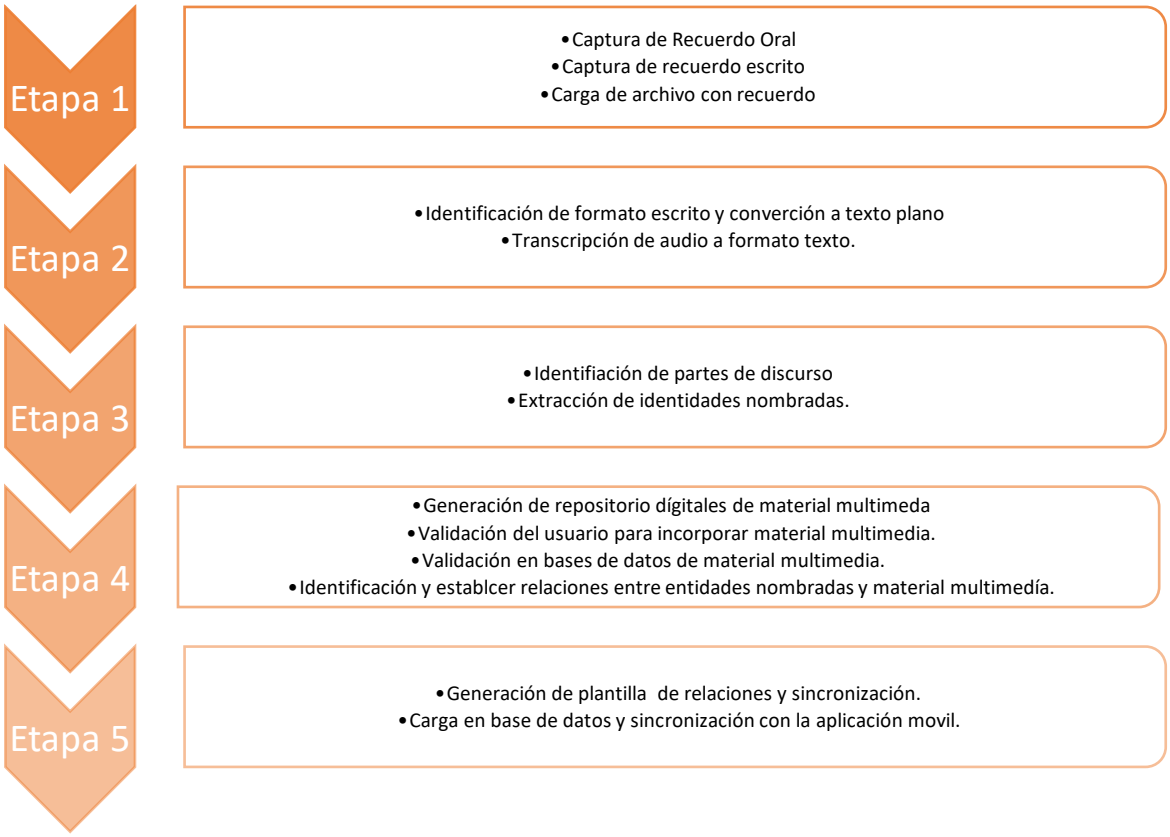


Figura 1.1. *Etapas propuestas para el desarrollo del proyecto.*

Nota. Elaboración propia.

A continuación, se exponen de forma breve las metas originales y tareas que se encapsulaban dentro de cada una de las cinco etapas planteadas, las cuales se describirán de forma específica en la sección de implementación y experimentación.

1.7.1 Etapa 1.

Objetivo:

Desarrollo de aplicación hibrida por medio de DART y el SDK Flutter.

Tareas:

- Diseño de la interfaz de la aplicación.
- Evaluación de principales funciones
- Investigar librerías de Flutter para implementación de funciones.
- Selección administradora de base de datos.

1.7.2 Etapa 2.

Objetivo:

Procesamiento de formatos de captura de historias a texto plano.

Tareas:

- Establecer duración y formato para discurso oral por medio de la aplicación.
- Convertir formatos audios a formato WAV.
- Aplicar librerías de Google para conversión de discurso oral a texto en los audios de formato WAV.
- Evaluación de transcripciones y presión con respecto al archivo de origen.
- Conversión de formatos de archivos de texto a texto plano.

1.7.3 Etapa 3.

Objetivo:

Aplicación de técnicas de procesamiento de lenguaje natural para identificación de parte de discurso y entidades nombradas.

Tareas:

- Aplicación librería SpaCy para análisis de procesamiento de lenguaje natural.
- Identificación de partes de discurso.
- Identificación de entidades nombras.
- Aplicación de n-gramas.
- Creación de un corpus de entrenamiento y de evaluación.
- Reconocimiento de entidades nombras en el corpus de entrenamiento de forma automática con la librería SpaCy y el lenguaje Python.
- Reconocimiento de entidades nombras en el corpus de evaluación.
- Clasificación de las entidades nombras dentro del corpus de entrenamiento.
- Evaluación de técnicas y resultados.
- Ajustes de parámetros y configuración del modelo seleccionado de SpaCy.

1.7.4 Etapa 4.

Objetivo:

Elaboración de una base de datos y la generación de un proceso automático de extracción de material multimedia de la red.

Tareas:

- Aplicación librería Selenium y lenguaje Python para proceso de descarga automática de contenido de la red.
- Identificación de imágenes y descarga en repositorio.
- Generación de repositorio de material audiovisual y sonoro.
- Creación de una base de datos relacional que implemente las entidades nombras en los nombres de los archivos.
- Implementación en el sistema de gestión de base de datos con Firebase para el manejo de datos del perfil del usuario.

1.7.5 Etapa 5.

Objetivo:

Generación y evaluación de sincronización de identidades nombradas y bases de datos de material multimedia.

Tareas:

- Aplicación de plantilla para la sincronización de identidades nombras y material multimedia.
- Evaluación de la sincronización con la aplicación.

En la parte experimental se han utilizado lenguajes de programación (Python, DART), el gestor de base de datos Firebase y en la parte de desarrollo de aplicaciones hibridas el SDK Flutter junto con el software Android Studio. La información y los manuales para el uso de la base de datos y software de desarrollo se encuentran disponibles en acceso abierto dentro de las plataformas respectivas [9]. Se han utilizado como fuente de información principal las páginas web oficiales

[9], [10], sus manuales de operación, páginas web elaboradas por las comunidades de desarrolladores y/o grupos de investigación aportando soluciones a consultas realizadas por diferentes usuarios y mantienen informado sobre actualizaciones en las herramientas y cambios realizados en la sintaxis.

Para las tesis de maestría se ha consultado para el estudio del arte el motor de búsqueda es un motor de búsqueda de Google Scholar, el cual está enfocado y especializado en la búsqueda de contenido y bibliografía científico-académica al igual se apoyó en material publicado sobre procesamiento de lenguaje natural y sobre manuales de desarrollo de aplicaciones y gestores de bases de datos.

Capítulo 2

Estado del arte

2.1 Estado del arte

2.1.1 Interacción Chatbot con usuarios y respuestas emocionales.

Podemos encontrar dentro del trabajo de esta investigación "The Design and Implementation of XiaoIce, an Empathetic Social Chatbot" el poder de la generación de un modelo computacional empático con sus usuarios, el cual está dotado con una gran capacidad de respuesta para las interacciones que lleva con personas; en algunos casos creando vínculos emocionales, sin dejar de ser solo un conjunto de algoritmos [11].

Se presenta como un asistente virtual que al poco tiempo de usarlo algunos usuarios logran generar un vínculo emocional con él, mejorando la percepción inicial del asistente. En China con la evolución de las funciones el modelo tiende a usarse como pareja virtual, por las respuestas emocionales que se consiguen de Xiaoice.

En sus orígenes asumió el género femenino y sus programadores lo dataron con rasgos de respuesta de una adolescente de 18 años. Al ser un ecosistema de IA la interacción con el gran volumen de usuarios le permite aprender a responder a las interacciones con los seres humanos. [12]

Xiaoice diseñado por la compañía Microsoft se implementó originalmente en el mercado de China, pero por su popularidad y uso creció en el mercado asiático [11]. Su capacidad de respuesta emocional le permite actualmente ser capaz de estar monitoreando las emociones fuertes con el objetivo de guiar las conversaciones hacia temas que le resulten más positivos para su usuario y evitar que se estanque en situaciones de crisis o estrés emocional. [11]

XiaoIce es una fuente de datos personales para los creadores de la IA, pudiendo recabar datos personales, íntimos de la persona y obteniendo patrones de comportamiento del usuario y sus iteraciones con otras personas. [12]

2.1.2 Aprendizaje por refuerzo.

El artículo publicado bajo el título "*Delayed Rewards in the context of Reinforcement Learning based Recommender Systems*" de Debmalya Biswas nos presenta un enfoque que se basa en el modelo de aprendizaje por refuerzo (RL), esto con la finalidad de implementar sistemas de recomendación. Como el ser humano es un ente en constante cambio los sistemas tienen que ser adaptativos. El proyecto trata de solventar esto implementando 3 constructores: canales de retroalimentación ponderados, recompensas retrasadas y aumento de recompensas. [13]

Las aplicaciones actuales tienen que ser atractivas y que logren capturar los usuarios el mayor tiempo posible. El principio de la aplicación es calcular el sentimiento del usuario en base a la respuesta del usuario al igual que a los aspectos ambientales (tomando en consideración la ubicación por GPS del dispositivo). El autor considera el implementar una serie de elementos que censen las variables a considerar tal es el caso de la temperatura ambiental, iluminación, hora del levantamiento de los datos del dispositivo, además de otros elementos como el día y el clima. Y en el momento que la aplicación esta actividad pueda encontrar usuarios con una relación directa. Esta serie de variables podrá retroalimentar el sistema y permitir adaptarse a cada usuario de manera más personalizada. [13]. Con los modelos planteados y las soluciones aplicadas le permite al sistema aprender, adaptarse a las preferencias y sentimientos del usuario en tiempo real, favoreciendo la respuesta que se le entregan al usuario. [13]

2.1.3 Narrativa asistida por computadora.

En el artículo "*Visual Storytelling*" de Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra et al. Se nos muestra un modelo descriptivo de imágenes que va más allá de mencionar los elementos contenidos dentro de una imagen seleccionada y como el modelo es capaz de interpretar la interacción de los elementos y describir de manera abstracta el contenido de la imagen introducida. [14]

Dentro del conjunto experimental también podemos observar como el modelo al presentársele 5 imágenes de manera secuenciales crea una historia basada en la relación de cada imagen con la que le precede. [14]. El entrenamiento se logró con un conjunto de casi 82 mil imágenes y con un proceso de entrenamiento y retroalimentación de las historias formadas por un

grupo de personas que iban describiendo las imágenes que los investigadores les aportaban, implementando todo dentro de una red neuronal recurrente. [14]

Dentro del artículo *"Guided Neural Language Generation for Automated Storytelling"* publicado por Prithviraj Ammanabrolu, Ethan Tien, Wesley Cheung et al [15] en el 2019, se plantea la utilización de un proceso de evento a evento que pretende brindar una orientación de evento a oración.

En esta investigación se implementa el uso de etiquetas de reconocimiento de identidades enumeradas y el uso de tuplas de 5 elementos para estructurar los eventos los cuales quedan establecidos por la estructura [s, v, p o, m], donde la *v* corresponde al verbo, la *s* es el sujeto del verbo, *o* es el objeto, *p* se usa para las preposiciones y la *m* queda relacionada con los modificadores, objetos indirectos y el objeto preposicional. La estructura planteada no obliga a que todos los elementos estén presentes en cada ocasión. [15]. Dentro de su investigación encontraron que ninguno de los modelos estudiados (*Retrieve-and-Edit, Sentence Templating , Monte-Carlo Beam Search y Finite State Machine Constrained Beams*) por sí solos podían encontrar con éxito un equilibrio entre los objetivos de retener todos los tokens de eventos y por lo cual se llevaba a cabo la generación de interesantes enunciados; indicando que cada uno de los modelos presentaba sus fortalezas y debilidades de acuerdo a las tareas en las que se implementaba [15]. El modelo que planteaba dentro del trabajo buscaba poder por medio de la memoria de las neuronas LTMS completar la tarea de evento a oración implicando que se pudiera llenar la información faltante dentro de la estructura experimental para tener un resultado comprensible de acuerdo con el contexto de la entrada de información [15] . Su trabajo queda abierto a futuras implementaciones de nuevas técnicas para completar la tarea de evento a enunciados, concebida por en el 2018 por Martin et al [15].

Se puede encontrar dentro del campo de generación de *StoryTelling*, el relacionado al campo de procesamiento de imágenes y la generación de una narración o descripción textual de la interpretación de los eventos representada en una imagen o una secuencia de imágenes denominado este proceso como narración visual computacional. En el trabajo titulado *"A Pipeline for Creative Visual Storytelling"* publicado por Stephanie M. Lukin, Reginald Hobbs, Clare R. Voss pertenecientes al Labaratorio de investigación de la armada de Estados Unidos de America (*U.S. Army Research Laboratory*) se explica que estos procesamientos son gracias a los avances y

enfoques transversales en el lenguaje natural procesamiento, generación y visión artificial [16]. Se define dentro del trabajo una narración visual creativa computacional como aquella con la capacidad de alterar la narración de una historia en tres aspectos: describir diferentes ambientes, para poder llevar a cabo la generación de variaciones basadas en objetivos narrativos, y para adaptarla narración a la audiencia. Dentro de la investigación se nos presenta una estructura de pipeline de módulos de tareas las cuales son: identificación de objetos, imagen única Inferencia y narración de imágenes múltiples que son la estructura para la generación de un Strorytelling [16]. Algo importante en este trabajo es que consideran la audiencia la cual será el receptor de la historia obtenida en el procesamiento de las imágenes. Como indicaba el trabajo dependiendo del público debe variar la estructuración de la historia dentro del trabajo se indica que una narración concisa resulta más apropiada cuando se busca compartir noticias o información, mientras que una narración detallada y humorística es adecuada para motivos del entretenimiento.

En la **Figura 2.1** se ejemplifica las tareas que va realizando, siendo la entrada de información una imagen. Se analiza la imagen entrante al sistema se extrae los elementos y se genera una relación entre la interacción de los elementos y la imagen, posteriormente se obtiene una nueva imagen de entrada se obtiene los elementos la relación, pero posteriormente se hace una conjetura entre los eventos de la imagen pasada y la nueva imagen, y el proceso se repetió el número indicado de imágenes que ingresara al sistema.

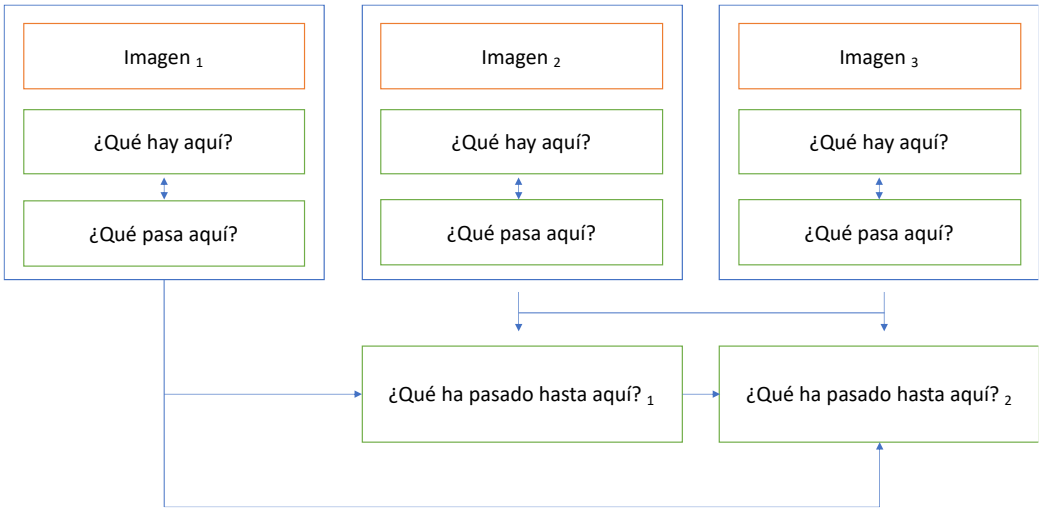


Figura 2.1. *Pipeline de narración visual creativa: T1 (identificación de objetos), T2 (inferencia de imagen única), T3 (narración de imágenes múltiples).*

Nota: *Adaptada del artículo original [16].*

Dentro del campo de la generación de historias (Storytelling) por medio de inteligencia artificial existen trabajos que también buscan dirigir sus investigaciones hacia el campo emocional de los humanos, uno de los trabajos que se pueden mencionar es el publicado bajo el título *"Emotion Reinforced Visual Storytelling"* por Nanxing Li, Bei Liu et al. [17]. La investigación pretende crear historia a un nivel humano al modelar el contenido de las imágenes con emociones y generando los textos con este principio implementando neuronas recurrentes. Proponen un generador de historias visuales reforzado con emociones, que consiste en dos partes: la primera en un codificador de imágenes y la segunda un generador de historias con secuencia de imagen teniendo una emoción como entrada [17]. Dentro de la estructura planteada se cuenta dos discriminadores y un sistema de medición emoción enfocada a emociones afirmativas el cual brindaras recompensas por medir la relevancia de la imagen, el estilo de la historia y la consistencia de la emoción. El codificador de imagen dentro de la investigación es una red neuronal recurrente (*recurrent neural network*, RNN) con unidad recurrente cerrada (*gated recurrent unit,* "GRU") dado que este último genera la salida basada en la entrada actual y las entradas anteriores, por lo cual la salida contiene la información contextual de la imagen actual. Algo importante dentro de su enfoque es cuidar la relación dentro las imágenes para preservar el sentido en la narración [17]. Su diversidad de emociones son felicidad, excitación, tristeza, diversión dentro de tres categorías: correspondencia, razonabilidad y por su relevancia. Estas validaciones se realizan por medio de 10 estudiantes que califican 20 historias seleccionadas al azar [17]. Su conjunto de experimentos mostró que el enfoque propuesto era capaz de generar historias con una mayor diversidad y con emociones más ricas ampliando enormemente la variedad de narraciones visuales [17].

2.1.4 Procesamiento de lenguaje natural y Alzheimer.

La investigación bajo el título "*Procesamiento de lenguaje natura para la búsqueda de patrones en demencia*" de Damián Solís Rosas, Dr. Saúl Tovar Arriaga et al. Plantea un análisis estadístico de métricas lingüísticas que buscan patrones para realizar una clasificación binaria en conversaciones de personas con demencia y personas sin demencia.El estudio experimental se realiza por medio de una base de datos llamada Carolina Conversation Collection (UNCC, 2008), administrada por la Universidad de Carolina del Norte de Charlotte. La base de datos almacena una

serie de transcripciones realizadas entre un entrevistador y un paciente en la cual se plantea que algunos entrevistados pudieran tener algún algún tipo de demencia. Esta base de datos se encuentra en inglés [18]. Dentro de las transcripciones se implementa una cadena de signos para plasmar pausas cortas, las pausas largas y las muestras de aturdimiento como es la presencia del símbolo '[.]' que reflejan una pausa corta, los símbolos '[…]' reflejan una pausa larga y la presencia de los símbolos '[?]' un aturdimiento o confusión. [18].

Interviewer:	How are you?
Mr. X:	I am [.] fine.
Mr. X:	I went to […] my house
Mr. X:	I don´t know [?]

Figura 2.2. *Ejemplo de uso de símbolos en transcripciones.*
Nota: *Adaptada de la tesis original [18].*

La base de datos estaba compuesta de 222 personas de estudio de las cuales 62 pertenecían al grupo que presentaban un grado de demencia y 160 personas formaban el grupo sin demencia [18]. En cada conversación una de las conversaciones con los pacientes se hizo un preprocesamiento de las conversaciones, donde primero se eliminó las interacciones del entrevistador, posteriormente se eliminó el nombre del paciente y se unió todo en un solo párrafo. Esta nueva estructura de las entrevistas permitió poder analizar de una mejor manera el contenido de los pacientes y poder sacar métricas para ser implementadas dentro de una red neuronal de 3 capas y un modelo de máquina de vector soporte [18]. El investigador indica que uno de los factores que puede afectar el resultado de su proceso de clasificación son el grado de demencia, el grado de estudio de la persona el cual puede empobrecer la riqueza léxica de sujeto de estudio y la región donde aprendió el idioma, puesto que esto puede influenciar en su modo de expresarse y comunicarse verbalmente [18]. En el primer modelo una red neuronal se obtuvo una exactitud del 78.01 % y el segundo una máquina de soporte de vectores se obtuvo un 86.42% de exactitud en la clasificación de las conversaciones [18].

En el trabajo publicado bajo el título "Extracción de recuerdos de vídeos de entrevistas con personas con problemas de memoria" del autor Hugo García González se maneja a las historias de

vida como un método de ayuda para las personas que sufren de enfermedades neurodegenerativas como el Alzheimer [19]. La estructura para la extracción de la información se obtiene de un grupo de entrevistas entre el paciente y un investigador, donde ya se cuenta con una plantilla para las preguntas, facilitando la extracción de las respuestas de los pacientes. Su trabajo cuenta con tres líneas para generar transcripciones: la primera es la incorporación de documentos de texto donde se llenó los datos de manera manual, la segunda es la generación de transcripciones por medio de Google Cloud Speech API que permite convertir los videos y audios a texto [19]. El trabajo pretende mapear por medio de grafos las cosas más importantes de la vida de la persona incluyendo eventos, personas y recuerdos, al implementar una estructura de grafos en el proceso de almacenamiento de datos. La finalidad del trabajo es el de ayudar a las personas a recordar y expresar información sobre ellos mismos buscando mejorar su calidad de vida y la capacidad de recibir ayuda [19]. Para el desarrollo de la investigación implementaron Grafeno el cual es una biblioteca de Python *open source* desarrollada en el Departamento de Ingeniería de Software e Inteligencia Artificial de la Universidad Complutense de Madrid [19]. Dicha herramienta está dedicada a la extracción de información semántica en forma de grafo a partir de texto. Grafeno comienza realizando un análisis de dependencias sintácticas al texto de entrada y a partir de los resultados se crea un árbol sintáctico similar a la herramienta SpaCy [19].

Un ejemplo de los resultados obtenidos dentro de su investigación se muestra en **Figura 2.3**. donde se puede observar que el proceso se basa en la generación de una lematización de los verbos como base y de ahí parte la conexión por nodos despreciando ciertas palabras elegidas por la estructura de la herramienta, lo cual puede limitar la información resultante.

Entrevistador: ¿Cuándo naciste?

Mr. X: Nací el 6 de noviembre del 1968

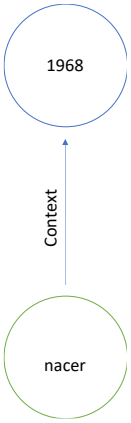


Figura 2.3. *Texto y grafo pregunta 1. [19]*

Nota: *Adaptada de la tesis original.*

2.1.5 Procesamiento de lenguaje natural y manejo de audio

En el trabajo de Maestría generado dentro de la Universidad Autónoma de Madrid bajo el título de "Generación de una base de datos para reconocimiento de voz en español mediante alineamiento de audio y texto" escrito por Martínez Antolín Luis Miguel, se nos muestra la generación de una base de datos en español la cual toma como referencia la construcción de la base de datos Librispeech y partiendo de unas 207 horas de audios de voy y las transcripciones respectivas.

El corpus LibriVox Spanish es una base de datos que contiene cerca de 73 horas de voz en español junto a sus transcripciones proveniente de 154 sujetos divididos en 77 hombres y 77 mujeres hablantes nativos de español [20]. Se compone de audio segmentados de forma manual y con una duración promedio es de entre 3 y 10 segundos de voz.

El proceso de reconocimiento automático del habla es un proceso de decodificación y transcripción de la señal de voz. Los sistemas de ASR reciben como entrada una señal de audio procedente de un hablante y capturada a partir de un micrófono, y se encargan de analizarla a partir de determinados patrones, modelos o algoritmos, dando como resultado de salida la información de la señal de voz en forma de texto. [21]

Por su parte ESPnet es una herramienta de código abierto de procesamiento de voz *end-to-end* la cual está especializada en los sistemas de reconocimiento de voz (ASR) *end-to-end*, para

poder implementar una gran variedad de herramientas de redes neuronales, para poder usar las librerías de Python Chainer y PyTorch [22].

El trabajo de las pautas para poder generar un repositorio propio para la tarea de generación de transcripciones en la tarea de speech- to-text, sin embargo, también es importante recordar que al momento de generar una narrativa automática es necesario la tarea de text- to-speech, que es el proceso de convertir textos a audios y es ahí donde se puede uno cuestionar que tan aceptado son los audios generados de manera digital.

Esta problemática la cubre el trabajo "La inteligencia artificial en la narrativa sonora", el cual es un estudio de caso realizado por Hernán Yaguana Romero y Juan Pablo Arrobo Agila de la Universidad Técnica Particular de Loja. [23]

El trabajo pretende determinar la influencia que la inteligencia artificial, enfocándose concretamente en los discursos generados por la tarea de conversión de texto a discurso oral (Text *To Speech*, TTS), puede ejercer dentro de la narrativa sonora, las sensaciones y los agrados que puede despertar en los oyentes [23]. En el estudio de caso se determinar que el grupo de control de entre los 26 y 35 años son los que dan mayor valor al uso de estas tecnologías y plantea que pronto se construirá una tendencia a corto plazo en la implementación [23].

Capítulo 3

Marco Teórico

3.1 Marco teórico

3.1.1 Procesamiento del lenguaje natural

El lenguaje natural es el lenguaje empleado por los seres humanos para comunicarse de manera oral o escrita. Cada lenguaje posee un conjunto de sintaxis y de una gramática propia, dotados de sus propias características como es la diversidad, la flexibilidad, la ambigüedad de las palabras y la subjetividad de acuerdo con el contexto al cual se hace mención estos atributos dificultan su tratamiento computacional.

Cabe mencionar que se entiende al procesamiento del lenguaje natural como el campo dentro de la informática, dentro de la inteligencia artificial y la lingüística computacional que trata con la interacción entre los ordenadores y los lenguajes humanos conocidos como los lenguajes naturales descritos anteriormente [24].

El principal objetivo dentro del campo de procesamiento del lenguaje natural es lograr el modelado del lenguaje humano para permitir su compresión por medio de herramienta computacionales.

3.1.2 Minería de datos

En la minería de texto, también se busca una especie de representación resumida de una gran cantidad de textos, pero en este caso tal «resumen» no refleja lo que los autores querían expresar en sus textos sino la metainformación sobre estos textos: la información de las tendencias, opiniones prevalecientes, porcentajes de diferentes opiniones, etcétera. Gracias a muchas aplicaciones prácticas en la política, economía, comercio, sociología y otras áreas de la vida social, esta rama del procesamiento del lenguaje natural ha recibido recientemente una gran atención por parte de los investigadores y las empresas.

3.1.3 Parte del discurso / (Part of Speech Tagging - POS)

Parte del discurso se refiere a la categoría de palabras o los términos léxicos en un idioma.

El Español contiene 8 grupo de clases de palabras que hacen referencia a estos términos léxicos: sustantivo, verbo, adjetivo, adverbio, pronombre, preposición, conjunciones y nombres. En la siguiente tabla se muestra un ejemplo de clases de palabras, su abreviación basada en "*The Penn TreeBank Tagset" [3]* y en la tercera columna de la **Tabla 3.1** se logra apreciar un conjunto de palabras de cada categoría.

Término léxico	Etiqueta estándares	Ejemplos
Sustantivo	NN	Algo, nada
Verbo	VB	Comer, correr
Preguntas	WRB	Cómo, Cuándo
Artículos	DT	El, la, los

Tabla 3.1. *Tabla de ejemplo de asignación de etiquetas.*

El *Penn Treebank,* que se muestra en la **Tabla *3.2*** en sus ocho años de funcionamiento (1989–1996), produjo aproximadamente 7 millones de palabras de texto etiquetado de parte del discurso, 3 millones de palabras de texto analizado esqueléticamente, más de 2 millones de palabras de texto analizado para la estructura de argumento predicativo y 1,6 millones de palabras de texto hablado transcrito y anotado por falta de fluidez en el habla [4].

Tag	**Description**
CC	*Coordinating conjunction*
CD	*Cardinal number*
DT	*Determiner*
EX	*Existential there*
FW	*Foreign word*
IN	*Preposition or subordinating conjunction*
JJ	*Adjective*
JJR	*Adjective, comparative*
JJS	*Adjective, superlative*
LS	*List item marker*
MD	*Modal*
NN	*Noun, singular or mass*
NNS	*Noun, plural*
NNP	*Proper noun, singular*
NNPS	*Proper noun, plural*
PDT	*Predeterminer*
POS	*Possessive ending*
PRP	*Personal pronoun*
PRP$	*Possessive pronoun*
RB	*Adverb*
RBR	*Adverb, comparative*

RBS	*Adverb, superlative*
RP	*Particle*
SYM	*Symbol*
TO	*to*
UH	*Interjection*
VB	*Verb, base form*
VBD	*Verb, past tense*
VBG	*Verb, gerund, or present participle*
VBN	*Verb, past participle*
VBP	*Verb, non-3rd person singular present*
VBZ	*Verb, 3rd person singular present*
WDT	*Wh-determiner*
WP	*Wh-pronoun*
WP$	*Possessive wh-pronoun*
WRB	*Wh-adverb*

Tabla 3.2. The Penn TreeBank Tagset. Adaptado de *[27]*

3.1.3.1 Modelos probabilísticos para identificación de partes del discurso.

Se puede inferir que, en el área de procesamiento del lenguaje natural, la detección de partes de discurso es la asignación de etiquetas de acuerdo con el contexto del uso que se le da a cada palabra, en un conjunto ordenado y estructurado que conforman una idea mostrada como una oración o un enunciado. [28]

Algunas de las implementaciones del uso de la técnica de POS son [29]:

- Análisis sintáctico: el etiquetado POS puede mejorar el análisis sintáctico al acelerar sus procesos con la detección de un tipo especifico de etiquetas requerido en las tareas.
- Detección de pronunciación para conversión de text-to-Speech.
- Detección de sentimiento o en ejecución de tareas afectivas: permite detectar los adjetivos de otras etiquetas POS, para su posterior segregación y la implementación en otros procesos.
- Tareas computacionales lingüísticas o analíticas del lenguaje.
- Controlar las etiquetas POS permite poder regresar a una determinada clase de etiqueta para su utilización en tareas de procesamiento de lenguaje natural.

3.1.3.2 Algoritmos estándar para el etiquetado de POS

Los enfoques más familiares en esta tarea son los basados en reglas, redes neuronales artificiales, estocásticos e híbridos. [30]

A continuación, se mencionan algunos de los modelos que pueden ser implementados también. [30]:

- Basado en reglas: utiliza reglas hechas a mano para asignar etiquetas a las palabras en una oración. Las reglas generadas dependen principalmente de las características lingüísticas del idioma, como la información léxica, morfológica y sintáctica.
- Modelos ocultos de Márkov (HMM- Hidden Markov Model).
- Campos Aleatorios Condicionales (CRF- conditional random field).
- Modelos de Márkov de Máxima entropía (MEMM).
- Modelos de secuencias neuronales (RNNs):

1. Una red neuronal recurrente (RNN) se encuentra entre un modelo de red neuronal artificial donde las conexiones entre las unidades de procesamiento forman caminos cíclicos. Es recurrente ya que reciben entradas, actualizan las capas ocultas en función de los cómputos previos y hacen predicciones para todos los elementos de una secuencia. [17]
2. Una memoria a largo plazo a corto plazo (LSTM) es un tipo especial de arquitectura de red RNN, que tiene la capacidad de aprender dependencias a largo plazo. Un LSTM también puede aprender a llenar el vacío en intervalos de tiempo en más de 1000 pasos. [30]
3. Memoria bidireccional largo-corto plazo: el LSTM bidireccional contiene dos capas ocultas separadas para procesar información en ambas direcciones. La primera capa oculta procesa las secuencias de entrada hacia adelante, mientras que la otra capa oculta las procesa hacia atrás; ambos se conectan luego a la misma capa de salida, que brinda acceso al contexto futuro y pasado de cada punto de la secuencia. Por lo tanto, BLSTM supera tanto a los LSTM estándar como a los RNN, y también proporciona significativamente un modelo más rápido y preciso [30] Un ejemplo son los modelos como BERT de la compañía Google.

3.1.3.3 Cadena de Márkov

Las cadenas de Márkov resultan útiles cuando se necesita calcular la probabilidad de una secuencia de eventos observables. Es una herramienta de la teoría de procesos aleatorios, que consta de una secuencia de n números de estados. En este caso, las conexiones entre los nodos (valores) de la cadena se crean solo si los estados están estrictamente uno al lado del otro, se puede calcular la probabilidad de una secuencia de eventos simplemente multiplicando las probabilidades de transición entre los estados que nos interesan.

Las cadenas de Márkov son extensiones del autómata finito determinista, en concreto, son un tipo de autómata en el que las transiciones representan las probabilidades de ir de un estado a otro y los estados que se visitan dependen únicamente de la cadena de entrada [31]. Formalmente, podemos definir una cadena de Márkov con los siguientes elementos en la **Tabla 3.3**:

$Q = q_1, q_2 \ldots . q_n$	Conjunto de N estados.
$A = a_{11}, a_{12} \ldots . a_{n1} \ldots a_{nn}$	Matriz de probabilidades de transición, donde cada casilla a_{ij} representa la probabilidad de ir del estado i al estado j. La suma de todos los arcos que salen de un estado i debe ser igual a 1. $$\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$$
$\pi = \pi_1, \pi_2 \ldots . \pi_n$	Estados artificiales que se añaden al inicio y final de la cadena de entrada, respectivamente.

***Tabla 3.3**. Tabla de elementos de cadena de Márkov.*

Nota. Libro "Speech and Language Processing. Daniel Jurafsky & James H. Martin. Copyright © 2021." [32]

Estados

$$Q = \{Q_1, Q_2, Q_3 \ldots, Q_n\}$$

Matriz de transición

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{N+1,1} & \cdots & a_{n+1,n} \end{bmatrix}$$

Formalmente, considere una secuencia de variables de estado $q_{1,}q_2 \ldots q_n$. Un modelo de Márkov toma la suposición sobre las probabilidades de la secuencia estudiada planteando que, al predecir el futuro, no importa el pasado, solo el presente. [31]

$$P(q_i = a \mid q_1 \ldots q_{i-1} = \ P(q_i = a \mid q_{i-1} \ )$$

Ecuación 3.1 Supuesto de Márkov. *[32]*

Un modelo de secuencia o clasificador de secuencia es un modelo cuyo trabajo es asignar una etiqueta o clase a cada unidad en una secuencia. mapeando así una secuencia de observaciones a una secuencia de etiquetas. [31]

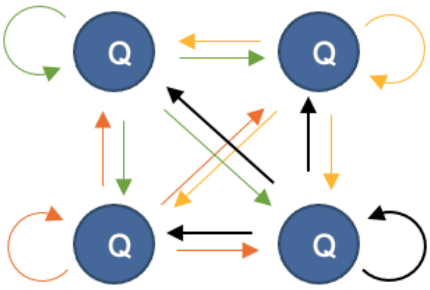
Se puede usar las cadenas de Márkov en procesamiento de lenguaje natural para identificar la probabilidad del tipo de clase al que pertenece la palabra que precede a la actual. En el siguiente **Figura 3.1** se muestra un ejemplo donde se observa la aplicación de las etiquetas POS a dos palabras de la oración.

El perro corre hacia el jardín

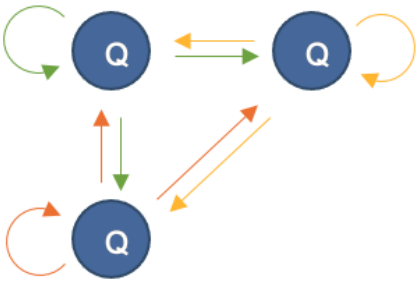
Sust verbo

Figura 3.1 *. Ejemplo de conversión de palabras a etiqueta de discurso (POS).*

En las **Figura 3.2** se representan dos modelos de cadenas de Márkov de 3 y 4 estados, recordando que este modelo es esencialmente el modelo de Márkov más simple, donde debemos determinar el conjunto de las probabilidades iniciales con respecto a los estados que lo conforman y, determinar las probabilidades de transiciones de un estado a otro o de la permanencia en el mismo estado después de una transición.



$Q = \{Q_1, Q_2, Q_3, Q_4\}$

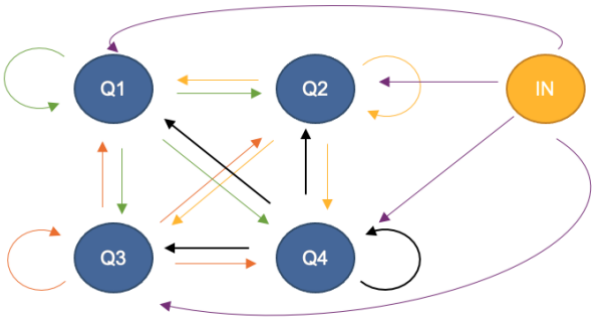


$Q = \{Q_1, Q_2, Q_3\}$

Figura 3.2 *. Diagrama de estados de cadenas de Márkov.*

En la ilustración 5 los círculos representan los estados del modelo de la cadena de Márkov planteado. Un estado hace referencia a una determinada condición especifica que se busca cubrir, dentro de la implementación del modelo de Márkov en la tarea de POS se puede considerar los estados como el total de las etiquetas implementadas en la clasificación.

En la **Figura 3.3** podemos observar la matriz de transición que debe completarse de acuerdo con el análisis de comportamiento entre los estados o al conjunto de valores propuestos dentro del experimento o el proceso de clasificación.



$Q = \{Q_1, Q_2, Q_3, Q_4\}$

Figura 3.3 *. Diagrama de estados de cadenas de Márkov.*

Estados	Q1	Q2	Q3
Q1			
Q2			
Q3			

Tabla 3.4. *Tabla de transiciones de estados en cadena de Márkov, donde se obtiene probabilidades de cambiar de un estado al otro de acuerdo a la interacción que tienen dentro del corpus estudiado.*

En la matriz de transición como en la **Tabla *3.4***, los valores en la intersección de columnas y filas reflejan las probabilidades de transiciones entre estados, la tabla se va llenando al ir analizando los corpus estudiados, obteniendo las probabilidades de que un estado cambie a otro o se mantenga en sí mismo dado el objeto observado en una secuencia finita de elementos, el cual corresponde a la palabra estudiada contra la que le sigue en un enunciado, este análisis se hace por todo los documentos dentro del corpus.

En el diagrama anterior, los círculos azules corresponden a las etiquetas de parte del discurso, y las flechas corresponden a las probabilidades de transición de un tipo de clase a otro. El circulo amarillo corresponde al conjunto de probabilidades de iniciar con un estado en particular sin estar sujeto a una condición de comienzo. En la tarea de procesamiento de lenguaje natural, el estado inicial corresponde a la primera palabra que aparece en cada enunciado, la cual se relaciona directamente con la frecuencia de aparición de las etiquetas que determinada a la clase dentro de los enunciados del corpus de entrenamiento. Sí en el corpus de entrenamiento nunca se mostró al inicio de ningún enunciado una palabra de una determinada clase la probabilidad sobre esa clase estaría establecida como un 0.

3.1.3.4 Modelo oculto de Márkov/Hidden Márkov models

En los modelos ocultos de Márkov, se utiliza probabilidades de emisión. Considerándola como la probabilidad de usar de una determinada etiqueta sobre (etiqueta POS) la palabra observada.

Por lo cual se incorpora al modelo de cadenas de Márkov la siguiente matriz conocidas como matriz de emisión. En la matriz de emisión las filas corresponden a las etiquetas usadas en el modelo y la columna al conjunto de palabras sin repetición sobre el corpus de entrenamiento. [32]

Al sumar todos los valores sobre una misma fila esta debería ser igual o muy acercada a 1. [32]

$$ME = \begin{bmatrix} b_{1,1} & \cdots & b_{1,v} \\ \vdots & \ddots & \vdots \\ b_{V+1,1} & \cdots & b_{v+1,v} \end{bmatrix}$$

$$\sum_{j=1}^{v} bij = 1$$

Ecuación 3.2 *Ecuación de cadena oculta de Márkov. [32]*

En los modelos ocultos existen ahora dos probabilidades que se pueden ver en los estados.

- Uno son las probabilidades de emisión, que representan las probabilidades de hacer ciertas observaciones dado un estado particular.
- Los otros son las probabilidades de transición, que representan la probabilidad de pasar a otro estado dado un estado particular.

Para completar la matriz de emisión, se puede tener un conjunto de datos ya etiquetado y por medio de los textos de un corpus observar la interacción de las etiquetas para calcular las probabilidades de asignar una determinada POS a la palabra observada. Anteriormente se comentó que existen dos categorías entre las 8 clases bases usadas en POS, las cuales son las cerradas y abiertas. Es importante al usar bases de datos, analizar si estas consideran las ambigüedades dentro de las clases abiertas, ya que pueden afectar el proceso de entrenamiento y generar ciertos sesgos para palabras específicas.

Si se realizara un entrenamiento desde cero de un corpus especifico, se debe analizar la estructura de todos los documentos que lo conforman, extraer la bolsa de palabra de cada uno de los textos individuales, eliminando las palabras repetidas entre ellos y dentro de los textos. Este vocabulario se clasifica de manera manual o se le pueden asignar las etiquetas para relacionar el contenido de la bolsa de palabras a una base de datos prediseñada. Se recomienda hacer una revisión de las palabras polisémicas y homónimas y hacer un ajuste en el set de entrenamiento con respecto a la POS, si se presentan errores.

Para la extracción de la matriz de trasmisión, se contará el número de veces que la etiqueta seleccionada fue seguida de otra y se realizará un incremento dentro de la columna que contiene la segunda etiqueta, sobre la fila de la etiqueta actual, repitiendo el proceso para el total del texto.

En el proceso de llenar las matrices de emisión para la aplicación de procesamiento de lenguaje natural, se debe analizar las etiquetas que están relacionadas a la palabra observada en cada enunciado durante el set de entrenamiento, e ir incrementando en uno de acuerdo con dicha etiqueta dentro de la matriz de entrenamiento, al finalizarlo se hace un ajuste a la matriz, al contabilizar las palabras totales por clase y dividir cada una de estas relacionadas a esa clase sobre el total, repitiéndose la misma tarea para el total de las clases, de esta manera se puede obtener la matriz de emisión.

En la siguiente ecuación se muestra un modo de contabilizar la transición de una etiqueta con respecto a observar la etiqueta previa. La letra "C" hace referencia al conteo de las veces que t_{i-1} (etiqueta previa) se mostró antes que la etiqueta observable t_i.

$$C(t_{i-1}, t_i)$$

Ecuación 3.3 *Ecuación de conteo de apariciones de etiqueta.*

Se puede llevar a cabo la obtención de la relación de probabilidad de una etiqueta con respecto a la que la antecede al realizar el conteo de veces que se presenta la transición rentre ambas ($C(t_{i-1}, t_i)$) dentro del corpus de entrenamiento. Dicho resultado se dividirá sobre la sumatoria de todas las interacciones que tuvo la etiqueta previa con el resto de las etiquetas implementadas en la clasificación definidos por la formula $\sum_{j=1}^{N} C(t_{i-1}, t_i)$.

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{\sum_{j=1}^{N} C(t_{i-1}, t_i)}$$

Ecuación 3.4 *Ecuación de conteo de calcula de probabilidad usando conteo de etiquetas.*

Los componentes del modelo oculto de Márkov se definen en la siguiente **Tabla 3.5** [31]:

$Q = q_1, q_2 \ldots . q_n$	Conjunto de N estados.
$A = a_{11}, a_{12} \ldots . a_{n1} \ldots a_{nn}$	Matriz de probabilidades de transición, donde cada casilla a_{ij} representa la probabilidad de ir del estado i al estado j. La suma de todos los arcos que salen de un estado i debe ser igual a 1. $\sum_{j=1}^{n} a_{ij} = 1 \; \forall i$
O= $o_1, o_2 \ldots . o_T$	una secuencia de T observaciones, cada una extraída de un vocabulario V= $v_1, v_2 \ldots . v_n$
$B = b_i(o_i)$	Es una secuencia de probabilidades de observación, también llamadas probabilidades de emisión, cada una de las cuales expresa la probabilidad de que se genere una observación o_i de un estado i.
$\pi = \pi_1, \pi_2 \ldots . \pi_n$	Estados artificiales que se añaden al inicio y final de la cadena de entrada, respectivamente.

Tabla 3.5. Tabla de elementos de cadena de Márkov.

3.1.3.5 Algoritmo de Viterbi

El algoritmo de Viterbi hace uso de las probabilidades de transición y las probabilidades de emisión, siendo un algoritmo de programación dinámica cuyo objetivo es encontrar la ruta óptima. Este algoritmo encuentra cual es el camino más probable para llegar a un estado determinado i, tomando en cuenta los estados anteriores q y las observaciones O.

El algoritmo de Viterbi se define como $V(q, t)$, como la probabilidad máxima de que el instante *t* se alcance en el estado q emitiendo el prefijo $x_1 \ldots x_t$. [33]

Ecuación 3.5 *Ecuación de algoritmo de Viterbi.*

$$v(q,t) = \underset{q_0 q_1, \ldots, q_{t-1}}{máx} P(\ \ q_0 q_1, \ldots, q_{t-1},\ \ o_1, o_2 \ldots o_t,\ \ q_t = t)$$

Para ir del punto inicial a la etiqueta O(*otros) necesita multiplicar la probabilidad de transición correspondiente y la probabilidad de emisión correspondiente, se sigue haciendo el mismo proceso para todas las palabras, hasta que se obtiene la probabilidad de una secuencia completa.[17]

Se trata también de un algoritmo que puede ser recursivo ya que para el paso t necesitaremos el camino más probable en t − 1. Si escribimos la fórmula de forma recursiva obtenemos:

$$v_t(j) = máx_{i=1}^{N}\ v_{t-1}(i)\ a_{ij}\ \ b_j(O_t)$$

Ecuación 3.6. *Ecuación recursiva del algoritmo de Viterbi.*

$v_{t-1}(i)$	Probabilidad del camino más probable para el paso anterior.
a_{ij}	Probabilidad de transición del estado anterior q_i al estado actual q_j.
$b_j(O_t)$	Probabilidad de emisión para el evento observable O_t dado el estado q_j.

Tabla 3.*6. Tabla de elementos de ecuación del algoritmo de Viterbi recursivo.*

El algoritmo de Viterbi se apoya de un grafo denominado *trellis,* el cual permite el cálculo iterativo eficiente por programación dinámica. Se puede observar en la **Tabla *3.4*** el trellis para el enunciado *"Janet will back the bill"* que se encuentra en inglés. Las etiquetas en color gris para este modelo son aquellas cuya posibilidad de ocurrencia quedó en cero, esto se pudo deber a diferentes razones, entre las cuales podemos encontrar: el set de entrenamiento no tenía ejemplo de frases que comenzaran con dichas etiquetas por lo cual, la probabilidad de iniciar con ellas era cero, la probabilidad de emisión para esa etiqueta dada la palabra observada nos daba cero, entre otras.

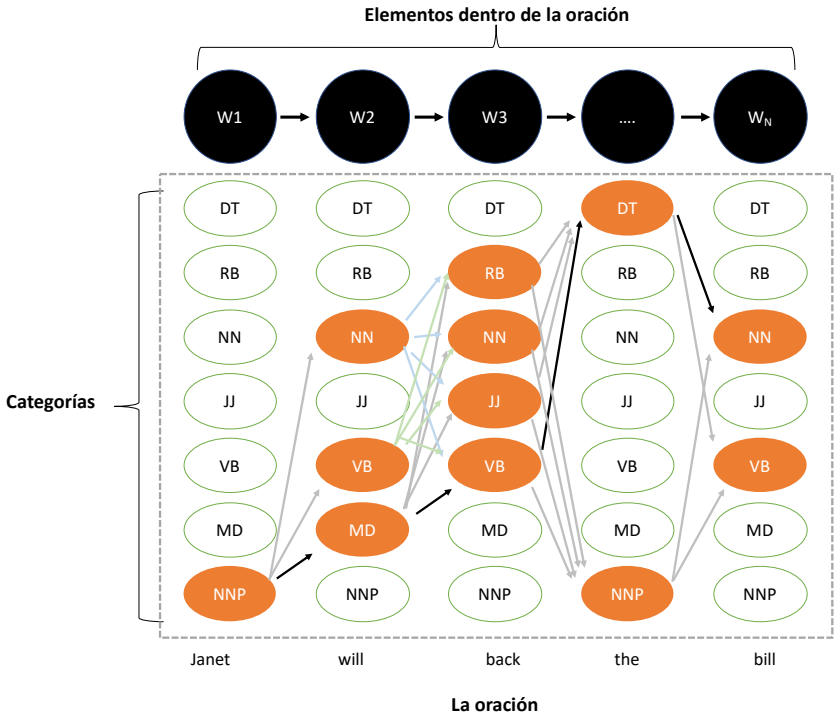


Figura 3.4. Ejemplo de implementación de algoritmo de Viterbi en la asignación de POS. [1]

Los trellis se diseñan de acuerdo con la longitud de objetos observables (palabras) que determinan la cantidad de columnas y número de filas del grafo y están relacionadas a la cantidad de etiquetas empleadas en el modelo.

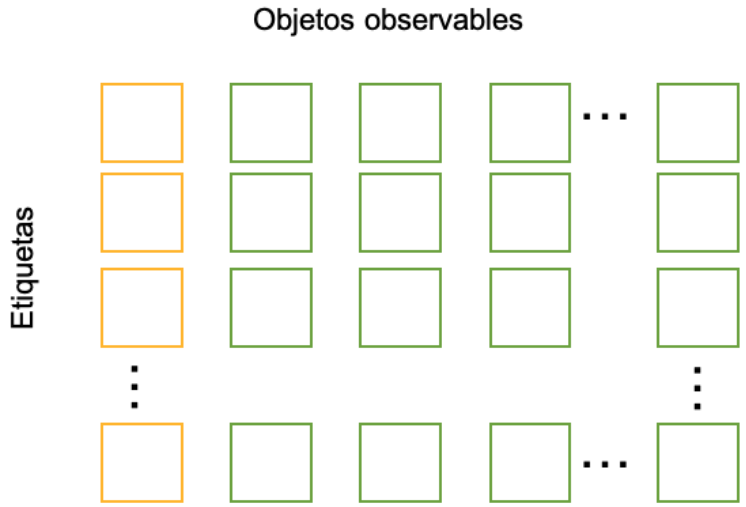


Figura 3.5. Diseño de trellis.

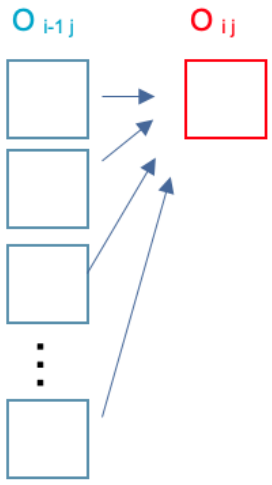


Figura 3.6. Paso 2 Viterbi

El algoritmo de Viterbi consta de tres pasos principales: inicialización, forward y recorrido hacia atrás.

El paso de inicialización se consigue en la columna de amarillo, donde se calcula para cada una de las posiciones la probabilidad de ocurrencia, la cual se consigue al multiplicar la probabilidad de comenzar con la etiqueta de esa fila por la probabilidad de emisión de la primera palabra, por el tipo de etiqueta asignado a esa posición.

Una vez obtenidos todos los valores de la primera columna, se prosigue al segundo proceso del algoritmo, él cual es el recorrido hacia adelante desde la segunda palabra o_2 hasta llegar al último observable o_n. Este se obtiene de la selección de la probabilidad máxima de los posibles caminos previos. El proceso es simple, se multiplica la probabilidad de emisión del observable o_i por el tipo de etiqueta asignada a dicha posición, luego se multiplica por la probabilidad de transición de la etiqueta, de donde parte el camino a la nueva etiqueta y por último, la probabilidad de ese trayecto. El proceso se repite para cada uno de los caminos previos, que corresponde al número de etiquetas, se va almacenando el valor obtenido para cada uno y al finalizar, se las etiquetas se seleccionan y guarda el valor más alto y la etiqueta que lo generó. En la *ilustración 8* se puede observar la trayectoria previa o_{i-1}.

En el tercer paso ocurre al finalizar todos los observables, se selecciona de la última columna el valor más grande y se empieza hacer un recorrido hacia atrás seleccionando el camino que lo generó, de esta manera se puede obtener el conjunto de etiquetas POS con mayor probabilidad para el enunciado.

3.1.4 Reconocimiento de entidad nombrada.

La tarea de reconocimiento de entidad nombrada *("named entity recognition"* NER por sus siglas en inglés), identifica entidades en las oraciones y las clasifica en categorías (etiquetas o clases) como son: nombres de personas, organizaciones, lugares, fechas, tiempos, etc. Está tarea aún no se encuentra totalmente definida debido a que depende del tipo de lenguaje y dominio donde se aplica. Actualmente existen trabajos sobre los dominios en noticias periodísticas, medicina y artículos científicos.

Con la tarea de identificación de identidades, se pueden reconocer patrones numéricos para ser categorizados como entidades de tiempo, formulas químicas, entidades monetarias, expresiones de porcentajes y patrones de fechas.

Anterior a los modelos de aprendizaje profundo se utilizaba modelos probabilísticos tales como los Modelos Ocultos de Markov [5] y Campos Aleatorios Condicionales. [35]

A partir de la implementación del Aprendizaje profundo o Deep Learning se han ido desarrollando diferentes enfoques para poder resolver la tarea del reconocimiento de entidades nombradas, algunos son implementando modelos con incrustaciones de vectores Word2Vec, Glove, Senna y FastText [6].

3.1.5 Librería SpaCy

SpaCy es una librería desarrollada por Matt Honnibal y programado en lenguaje Python con la cual se pude construir aplicaciones de procesamiento de lenguaje natural (NLP). Siendo de Software libre con Licencia MIT [37].

SpaCy proporciona modelos pre entrenados en 72 lenguajes actualmente, lo cual junto a una sintaxis clara y su diseñado de implementación logra ayudar para crear productos reales o recopilar información real [38].

La librería SpaCy estructura en módulos o funciones que realizan el proceso de análisis de texto a diferentes niveles, como el análisis morfológico, el léxico, el sintáctico, semántico y de discurso. Sobresaliendo en tareas de extracción de información a gran escala. Está escrito desde cero en Cython cuidadosamente administrado por memoria [37].

Las principales características de esta librería son [37]:

- Soporte para más de 72 idiomas.
- 80 canalizaciones entrenadas para 24 idiomas.
- Aprendizaje multitarea con transformadores pre entrenados como BERT.
- Vectores de palabras pre entrenadas.
- Velocidad de última generación.

- Sistema de formación listo para la producción.
- Tokenización motivada lingüísticamente.
- Componentes para el reconocimiento de entidades nombradas, etiquetado de partes del discurso, análisis de dependencia, segmentación de oraciones, clasificación de texto, lematización, análisis morfológico, vinculación de entidades y más.
- Fácilmente extensible con componentes y atributos personalizados.
- Compatibilidad con modelos personalizados en PyTorch, TensorFlow y otros marcos.
- Visualizadores integrados para sintaxis y NER.
- Fácil empaquetado de modelos, implementación y gestión del flujo de trabajo.
- Precisión robusta y rigurosamente evaluada.

Su estructura está basada en una tubería de procesamiento mostrada en la **Figura 3.7**, compuesta por una serie de elementos definidos a continuación:

- Tokenización: módulo destinado a tokenizar textos, que es la división del contenido en elementos únicos.
- Etiquetador de partes de discurso: encargado de etiquetar el tipo al que corresponde cada palabra de acuerdo con el contexto en que se emplea dentro del texto, alguna de las categorías es: los verbos, los sustantivo, los adjetivos, las preposiciones, los adverbios, etc.
- Parser (Parsing en inglés): es el término con el que se denomina el proceso de análisis sintáctico realizado en PLN.
- Etiquetador de entidades (NER): identifica las diferentes entidades nombradas dentro del texto evaluado.

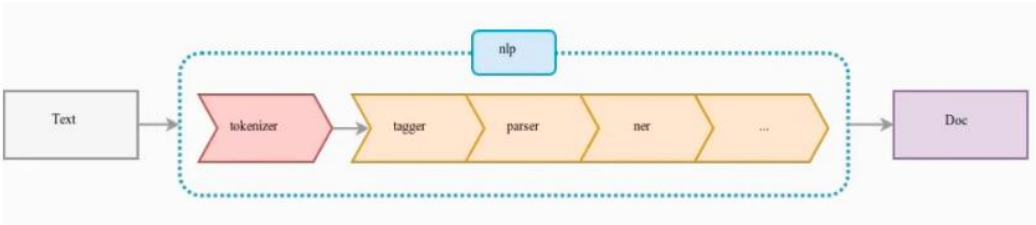


Figura 3.7. Diagrama de tubería en tareas de procesamiento de lenguaje Natural.

Nota. Imágenes obtenidas de https://spacy.io/api/doc

El proceso de descarga se puede realizar por medio de la página https://spacy.io/usage , donde se ingresa los datos del equipo en el cual se requiere instalar como se muestra en la **Figura 3.8**, el cual es un proceso sencillo.

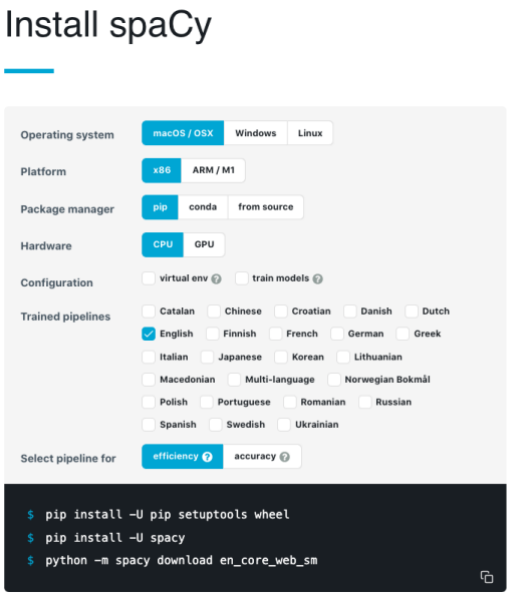


Figura 3.8. Diagrama de tubería en tareas de procesamiento de lenguaje Natural.

Nota. Imágenes obtenidas de https://spacy.io/usage [37]

3.1.6 Sistemas operativos.

El sistema operativo de Apple iOS, y diseñado exclusivamente para el hardware producido por dicha compañía fue el primero en implementar el interfaz usuario multitouch y, en buena medida, se puede ver como el responsable de la explosión y universalización en el uso de dispositivos móviles. Al igual que el sistema operativo que emplean para sus equipos de escritorio, MacOS X, iOS está basado en el núcleo Darwin, derivado de FreeBSD, un sistema libre tipo Unix [39].

El sistema operativo Android Diseñado por la compañía Google, basa la mayor parte de su operación en software libre (un núcleo Linux, máquina virtual Java, y muchas de las bibliotecas de sistema comunes en sistemas Linux), agregando una capa de servicios propietarios. La estrategia de Google ha sido inversa a la de Apple: en vez de fabricar sus propios dispositivos, otorga licencias

para el uso de este sistema operativo a prácticamente todos los fabricantes de hardware, con lo que la amplia mayoría de los modelos de teléfonos inteligentes y tabletas corren sobre Android. [39]

3.1.7 Tipo de aplicaciones

3.1.7.1 Aplicaciones nativas

Las aplicaciones nativas son aquellas que su desarrollo está basado en el lenguaje nativo del terminal. Se desarrollan en un lenguaje específico según la plataforma para la que se vayan a crear las aplicaciones. Por ejemplo, para desarrollar en Android se utilizará Java o Kotlin.

Por lo cual se obtiene aplicaciones desarrolladas y optimizadas específicamente para cada sistema operativo. [40]

3.1.7.2 Aplicaciones híbridas

Son aplicaciones móviles diseñadas en un lenguaje de programación web (CSS, JavaScript, HTML5) junto a un framework que hace posible la adaptación de la vista web a cualquier vista de un dispositivo móvil. Dicho de otra forma, son aplicaciones creadas para ser usadas en distintos sistemas operativos móviles, como Android o iOS. Estas apps aprovechan las ventajas del desarrollo web y se adaptan perfectamente al dispositivo como si fuesen aplicaciones nativas. . [40]

Diferencias entre aplicaciones nativas y las aplicaciones híbridas. [40]

- *Construcción:* las aplicaciones nativas son construidas sobre un lenguaje nativo del propio terminal mientras la aplicación híbrida se desarrolla sobre un lenguaje de programación web (JavaScript, CSS, HTML5, DART).
- Rendimiento: las aplicaciones nativas logran tener un mayor rendimiento las aplicaciones híbridas, independiente que ambas sean iguales en su usabilidad. Puesto que las nativas logran aprovechar de una mejor manera los recursos de hardware del dispositivo móvil o de la tableta. El rendimiento puede llegar a ser rendimiento similar siendo complicado encontrar indicios de que tipo de desarrollo se llevó dentro de la aplicación.

- *Coste del desarrollo:* En las aplicaciones nativas se requiere desarrollar la aplicación para cada sistema operativo donde se quiere lanzar su implementación, por lo que el tiempo de desarrollo se incrementa aumentando los costos, por el contrario, en las aplicaciones hibridas se pude partir de un solo código para las plataformas e implementar código especializado para casos particulares de los dispositivos de ser necesario.
- *Diseño visual:* Si se observamos una estructura igual para la misma aplicación dentro de dos plataformas distintas, seguramente se trata de una aplicación híbrida, por el contrario, si los elementos visuales cambian en forma y distribución, lo más seguro es que se trate de una aplicación nativa. Aunque existen casos en que el mismo diseño puede ser implementado para las aplicaciones nativas
- *Curva de aprendizaje:* Las aplicaciones nativas requieren el conocimiento de desarrollo para cada una de las plataformas que se vayan a implementar, desde el manejo de los hardware la implementación de APIs, por lo que el aprendizaje es más lento para dominar dichas herramientas. En el caso de las híbridas existen SDK y herramientas que permite desarrollar una aplicación con un mismo lenguaje caso de Flutter que implementa DART en la construcción de las aplicaciones.

3.1.7.3 Aplicaciones multiplataforma

Los equipos pueden codificar en la aplicación en los lenguajes y marcos de su elección y luego compilar el código para cada sistema operativo en el que la aplicación está destinada a ejecutarse utilizando plataformas como Xamarin. Las aplicaciones multiplataforma, como se denominan, permiten a los equipos ejecutarse en muchos dispositivos diferentes y reutilizar una gran parte del código, aunque el puente del código requiere más trabajo que el desarrollo para una sola plataforma. [40]

3.1.7.4 Aplicaciones web progresivas

Al igual que las aplicaciones web, las aplicaciones web progresivas se ejecutan en los navegadores móviles y suelen estar creadas con JavaScript, CSS y HTML5. Lo que hace que las aplicaciones web sean progresivas es su capacidad para proporcionar notificaciones push, mejores

gestos táctiles e interacción con el hardware utilizando las capacidades avanzadas del navegador. No hay SDK que ayuden a construir aplicaciones web progresivas, pero pueden implementarse sin pasar por un Marketplace. [40]

3.1.8 Extracción de datos de la web

El raspado web es la práctica de recopilar datos a través de cualquier otro medio que no sea un programa que interactúa con una API también considerado a través de un humano que usa un navegador web. Comúnmente se logra este proceso al escribir un programa automatizado que consulta un servidor web determinado, solicita los datos ayudándose de identificadores de lenguaje HTML y de otros archivos que compongan la página o páginas web seleccionadas y luego analiza los datos para extraer la información necesaria. [41]

El web scraping abarca una amplia variedad de técnicas y tecnologías de programación, como el análisis de datos y la seguridad de la información [41].

3.1.8.1 Librería Selenium

Selenium es un proyecto de código abierto que permite la automatización de tareas en navegadores. Su principal enfoque es el testeo de aplicaciones web, pero también logra realizar desarrollar potentes flujos de trabajo como es el caso de las técnicas de extracción de información de la red (scraping en inglés). [42]

Selenium requiere de un navegador web que se encuentre instalado dentro del sistema para poder funcionar. Algunas de las opciones disponibles para trabajar con Selenium se encuentra Chrome, Firefox, Edge, Internet Explorer y Safari. [42]

DRIVER

Además de esto, también es necesario disponer un «webdriver» que permita manejar el navegador (a modo de marioneta). Cada navegador tiene asociado un tipo de «driver». [42]

Entre las principales tareas que se deben de llevar a cabo al momento de extraer datos de la red son:

- Ingreso a sitio web: n La forma de acceder a una url es utilizar el método. get():

  >>> driver.get('url)

- Localización de elementos: El objeto ***driver*** ofrece las funciones desglosadas en

Acceso	**Función**	**Localizador**
Clase	find_elements_by_class_name()	By.CLASS_NAME
Selector CSS	find_elements_by_css_selector()	By.CSS_SELECTOR
Atributo ID	find_elements_by_id()	By.ID
Texto en enlace	find_elements_by_link_text()	By.LINK_TEXT
Texto en enlace (parcial)	find_elements_by_partial_link_text()	By.PARTIAL_LINK_TEXT
Atributo NAME	find_elements_by_name()	By.NAME
Nombre de etiqueta	find_elements_by_tag_name()	By.TAG_NAME
XPath	find_elements_by_xpath()	By.XPATH

Tabla 3.7. Tabla de funciones driver librería Selenium.

Nota. Tabla extraída de https://www.selenium.dev/documentation/ . [42]

Todas las funciones de búsqueda tienen su correspondiente versión para devolver un único elemento que cumpla con el filtro especificado. Cuando existe el caso de que haya varios elementos con el mismo identificador dentro del url, se devolverá el primer elemento encontrado entre todos ellos. [42]

- Click en botones: Se implementa la función homónima.

```
>>>  driver.get('url)
>>>  accept_click = driver.find_element_by_id('id')
>>>  accept_click.click()
```

- Borrar contenido: El objeto ***driver*** ofrece el método ".clear()"

```
>>>  driver.get('url)
>>>  accept_click = driver.find_element_by_id('id')
```

- Envío de texto: El objeto ***driver*** ofrece las funciones de envío de texto y envío de teclas especiales. En código mostrado a continuación se ejemplifica el uso para el ingreso de un usuario.

```
>>>  driver = webdriver.Firefox()
>>>  driver.get('url/login')
>>>  username = driver.find_element_by_id('username')
```

```
>>>  password = driver.find_element_by_id('password')
>>>  username.send_keys('abcdefg')
>>>  password.send_keys('12345678')
>>>  login_btn = = driver.find_element_by_id('buttonname')
>>>  login_btn.click()
```

Para el envoi de caracteres especiales se importa “.common.keys”, la implementación se muestra a continuación.

```
>>>  from selenium.webdriver.common.keys import Keys
>>>  element.send_keys(Keys.RIGHT)
```

- Extracción de texto de elemento: es posible acceder a esta información cuando un elemento incluye texto en su contenido, ya sea de manera directa o mediante elementos anidados.

```
>>>  driver.get('url)
>>>  accept_click = driver.find_element_by_id('id’).txt
```

Capítulo 4

Metodología

4.1 Metodología experimental propuesta

El proceso de desarrollo de la investigación contemplaba 5 etapas que se encuentra desglosada en la **Figura 1.1** del capítulo 1.

La primera etapa de la investigación contempla el desarrollo de una aplicación móvil para los sistemas operativos IOS y Android, por lo cual se optó por el SDK Flutter, el cual es de código fuente abierto creado por la compañía Google [7].

El SDK mencionado anteriormente se emplea para el desarrollo de aplicaciones híbridas usando el lenguaje de DART. El desarrollo de la aplicación se llevó por medio del software Android Studio.

El desarrollo de una aplicación híbrida fue la solución encontrada al buscar ahorrar tiempos de desarrollo para ambos sistemas operativos de Android y IOS.

Una de las funciones principales de la aplicación es la de conservar historias con tres opciones:

- El uso de formato de narración oral el cual queda guardado en un archivo mp4 con una duración máxima de 30 min.
- Subir un documento escrito.
- Escribir dentro de la aplicación y editarlo en cualquier momento.

En los tres modelos de carga de historia se busca poder extraer la información dentro de ellas para enriquecerla al sincronizar los elementos detectados con contenido multimedia.

La aplicación tenía considerado implementar la conservación de recuerdos en pacientes con Alzheimer por medio del discurso oral y escrito del usuario, o con el apoyo de un tutor o familiar.

En la **Figura 4.1** se plasma el diagrama propuesto que incorpora la división de pantallas para limitar las áreas de funciones dentro de la aplicación, dejando a la pantalla principal con un diseño que incorpora dentro de sus estructuras plantillas divididas en 4 categorías: total de historias, personas dentro de la historia, favoritos y rango de edad.

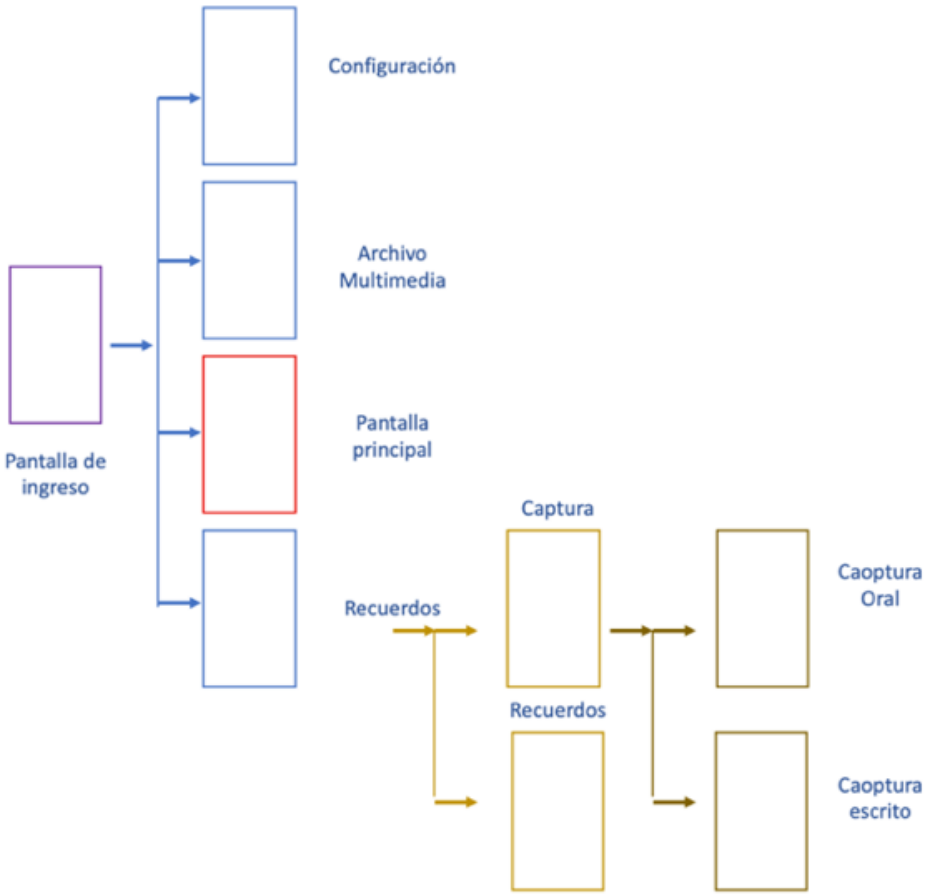


Figura 4.1. Diagrama de estructura de aplicación.

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

Por las implementaciones en personas mayores el desarrollo de la interfaz considero las brechas [26] generacionales para simplificar el llamado a las funciones y dejar de manera intuitiva el acceso a las herramientas.

Enfocándose en la parte de captura de recuerdo se aprecia en la **Figura *4.2*** el diseño planteado para el manejo de las grabaciones, donde se relación el grabado de nuevos archivos, la búsqueda por nombre de las grabaciones y el poder escuchar la grabación original dentro de la aplicación.

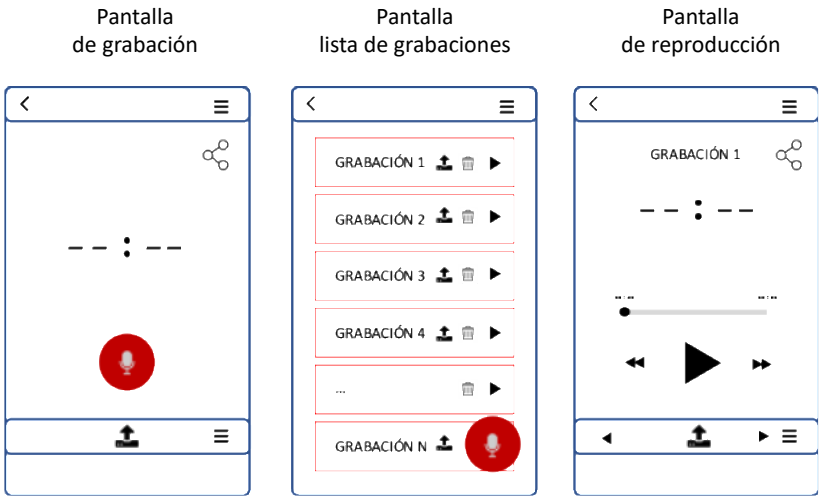


Figura 4.2. *Diseño de estructura de aplicación para grabación de audios.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

En la **Figura *4.3*** se realiza la explicación a detalle de los componentes dentro de cada pantalla de las tareas de grabaciones entre los cuales se destaca el botón de grabar, el modelo en formato de lista donde se almacenan las grabaciones, y los controles reproducción de la parte de captura.

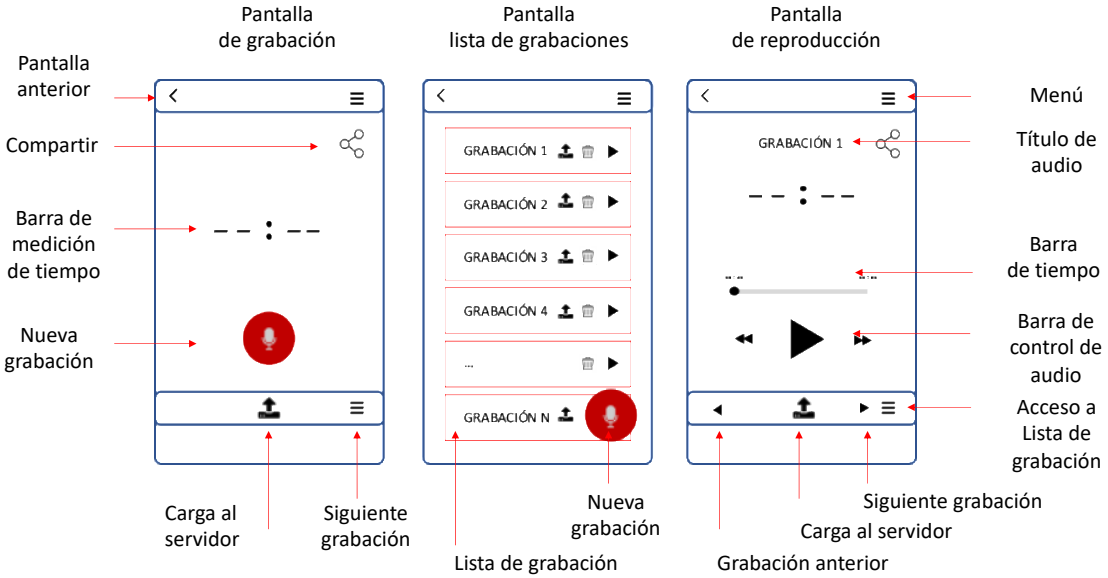


Figura 4.3. *Componentes de estructura de aplicación para grabación de audios.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

Con relación al diseño de las tareas de la carga de archivos de texto y escritura de nuevos archivos se muestra el diseño seleccionado en la **Figura *4.4***, donde se aprecia las funciones relacionadas a estas tareas, como es el subir elementos relacionados al documento como fotografías, la lista de documentos subidos al sistema y un editor de notas para la escritura manual de las historias.

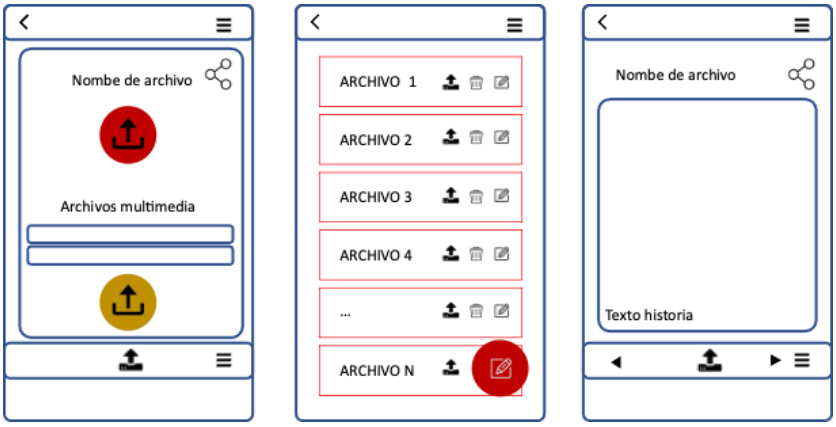


Figura 4.4. *Diagrama de estructura de captura de documentos y escrituras dentro de la aplicación.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

En la **Figura *4.5*** se realiza la explicación a detalle de los componentes dentro de cada pantalla de las tareas de carga, modificación de archivos adjuntos y escritura de archivos.

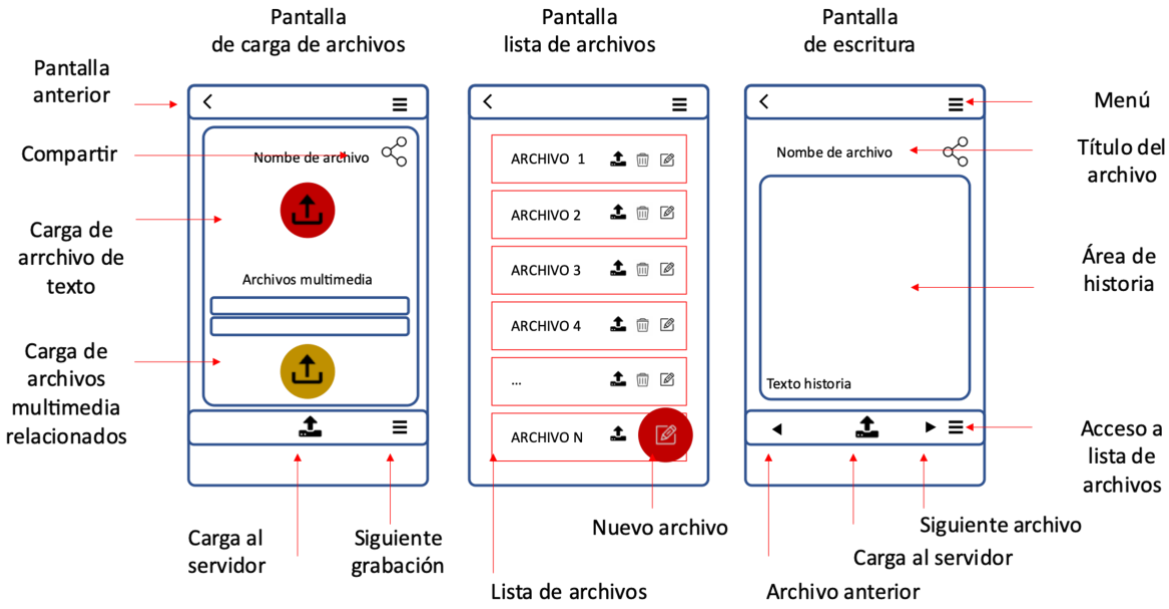


Figura 4.5. *Componentes de estructura de captura de documentos y escrituras dentro de la aplicación.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

El diseño de la pantalla de inicio se estructura para una visualización en formato de tarjetas que el usuario pueda seleccionar por 4 categorías: todos los archivos, favoritos, personas y época de vida, esta estructura se puede apreciar en la **Figura *4.6***.

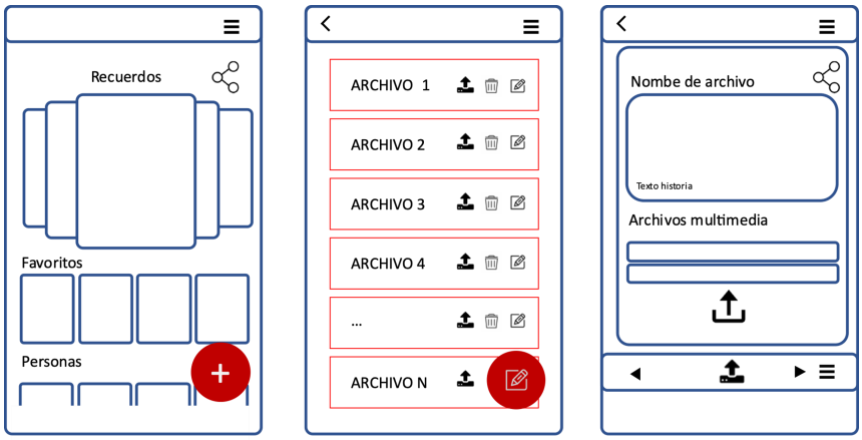


Figura 4.6 . *Diagrama de estructura de pantalla de inicio y visualización de historias.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

En la **Figura *4.7*** se realiza la explicación a detalle de los componentes dentro de la pantalla de inicio como son la división por categorías o elementos dentro de la aplicación y la administración para la visualización de historias y los archivos relacionados a ellas.

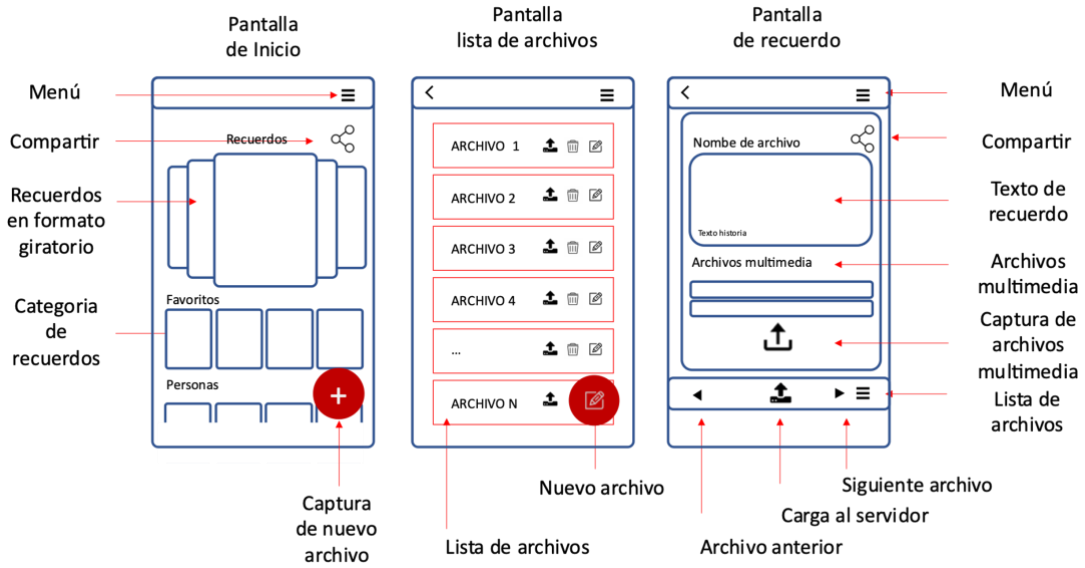


Figura 4.7 . *Componentes de estructura de captura de documentos y escrituras dentro de la aplicación.*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

En la **Figura *4.8*** se muestran las imágenes del desarrollo de la aplicación por medio del software Android Studio y el SDK Flutter.

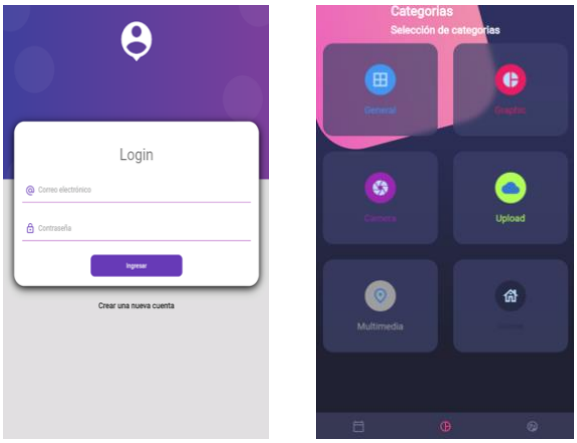


Figura 4.8 .*Captura de pantallas de la aplicación*

Nota. Elaboración propia a partir de notas Android Studio y SDK Flutter.

La aplicación móvil desarrollada en la investigación buscaba ser la herramienta que funcione como puente entre la parte de la captura de los recuerdos en su versión oral o escrita, con el repositorio digital y el sistema de narrativa inteligente, implementando este último en sus funciones con el diseño que se muestra en la **Figura 4.9.**

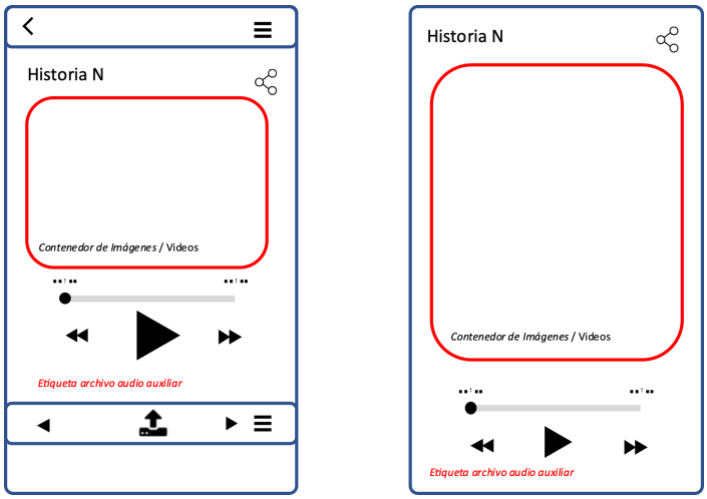


Figura 4.9 . *Diagrama de estructura de pantalla de narrativa automatizada dentro de la aplicación*

Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

En la **Figura 4.10** se realiza la explicación a detalle de los componentes dentro de la pantalla de inicio y administración de visualización de historias con la narrativa automática, esta pantalla busca poder ampliar las imágenes mostradas para mayor visibilidad del espectador, y aun permitir tener accesible los controles de reproducción del contenido.

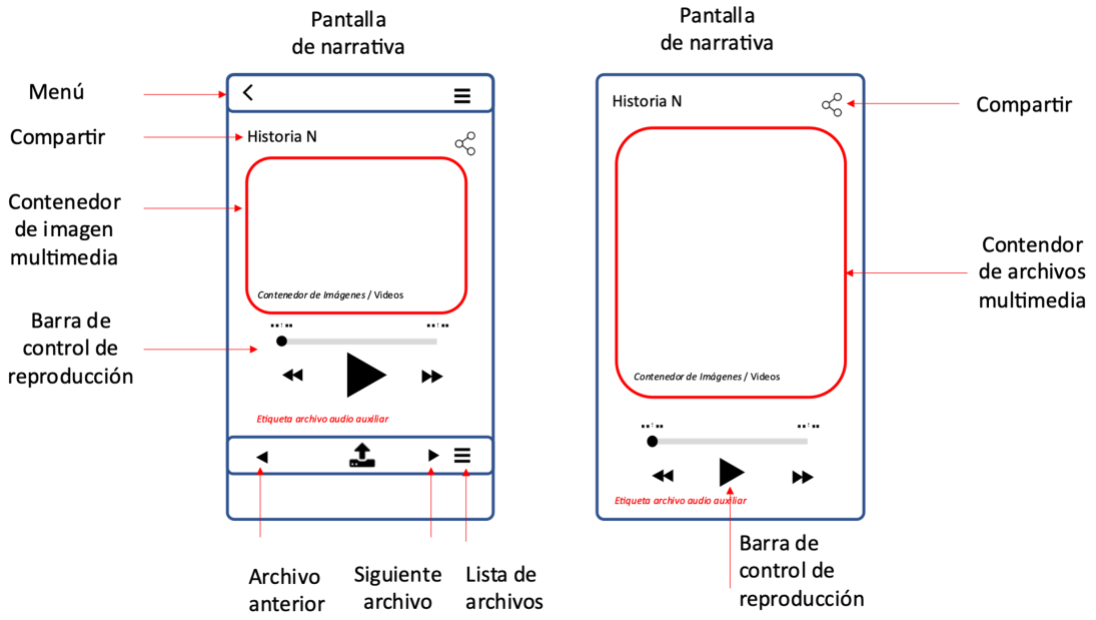


Figura 4.10. *Componentes de estructura de captura de documentos y escrituras dentro de la aplicación.*
Nota. Elaboración propia a partir de notas obtenidas de https://pub.dev/packages.

Entre las características que se implementaron para la aplicación que se encuentran:

- La aplicación es responsiva (se adapta a las diferentes dimensiones de equipos).
- Cuenta con características de usos intuitivo para el usuario.
- Tener funciones integradas para poder actualizarse de manera remota.
- La herramienta debe estar diseñada para ser compatible con el mayor número de dispositivos Android y permitir el uso de código para el sistema IOS.

El primer paso del proyecto se enfoca en la captura de los recuerdos e historias como se muestra en la **Figura 4.11.**, para la posterior extracción y generación de transcripciones en texto

plano “.txt” con la finalidad de extraer la identidades nombradas y partes del discurso de los textos generados por medio de técnicas de procesamiento de lenguaje natural.

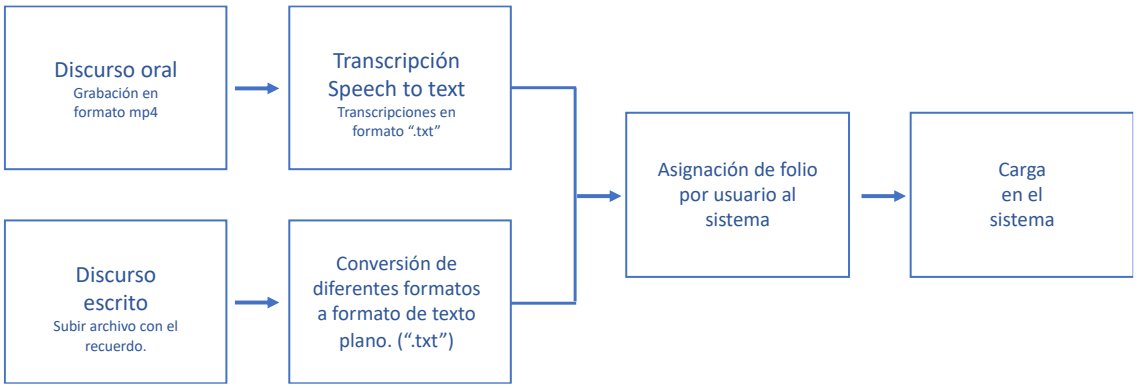


Figura 4.11. *Componentes de diagrama de captura de recuerdo.*

Nota. Elaboración propia.

En la **.,** se desglosa el esquema de captura. Los archivos en formato audio se almacenarán dentro del perfil del usuario aun cuando se haya realizado la tarea de generar su de transcripción como una medida de validación para medir el nivel de la precisión realizada por las librerías de Google que se implementaron dentro de la investigación dentro del proceso de conversión del discurso oral al escrito.

Entre las pruebas experimentales realizadas con los formatos con extensión “.pdf”, “.docx”, “.doc” y “.txt” relacionado a documentos de almacenamiento de texto, se identificó que el formato de texto plano resultaba el que menos recibía cambios en su estructura dentro de las tareas de procesamiento.

Para las tareas de procesamiento de lenguaje natural implementadas en la etapa 3 se emplea el uso de la librería SpaCy de código abierto para el lenguaje de programación Python.

Como se mencionó en el **Capítulo 3** relativo al de marco teórico, SpaCy es una biblioteca gratuita de código abierto para el procesamiento avanzado del lenguaje natural (NLP) en Python, diseñada para ayudar a crear aplicaciones procesamiento avanzado del lenguaje natural, cuyo uso no este restringido por cuota de palabras o eventos, haciendo que no sea un servicio consumible y siendo una opción gratuita [9] implementando diversos 4 módulos por idioma entre los cuales está disponible el español. Los cuales se pueden encontrar en la página https://spacy.io/models/es.

Los datos técnicos del módulo implementados (es_core_news_sm) se observan en la **Tabla *4.1*** el cual es el segundo modulo con tamaño media y entrenado con textos de noticias y medios.

Los datos técnicos de los 4 módulos en español se pueden encontrar se pueden encontrar en el sitio oficial de SpaCy o en el enlace https://spacy.io/models/es#es_core_news_sm.

MÓDULO	es_core_news_md
IDIOMA	Español
TIPO	COREVocabulary, syntax, entities, vectors
GENRE	NEWSwritten text (news, media)
TAMAÑO	MD - 40 MB
COMPONENTES	tok2vec, morphologizer, parser, senter, attribute_ruler, lemmatizer, ner
PIPELINE	tok2vec, morphologizer, parser, attribute_ruler, lemmatizer, ner
VECTORES	500k keys, 20k unique vectors (300 dimensions)
ENLACE DESCARGA	es_core_news_md-3.4.0-py3-none-any.whl
FUENTES	UD Spanish AnCora v2.8 (Martínez Alonso, Héctor; Zeman, Daniel)
	WikiNER (Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, James R Curran)
	spaCy lookups data (Explosion)
	Explosion fastText Vectors (cbow, OSCAR Common Crawl + Wikipedia) (Explosion)
AUTOR	Explosion
LICENCIA	GNU GPL 3.0

Tabla 4.1. *Datos técnicos del módulo implementados (es_core_news_sm).*

Nota: Datos adaptados del sitio oficial de SpaCy [38].

Se requiere entender a la tarea de identificación de discurso como la generación de un etiquetado para cada palabra que componen el texto. Generalmente se modela como una tarea estructurada donde la etiqueta de la primera palabra puede depender de la etiqueta de hasta la tercera palabra dependiendo de cada idioma, pero se puede llegar a aproximar la clasificación de cada palabra de forma aislada en una etiqueta (“part-of-speech” en inglés, siglas POS) dependiendo del modelo implementado, Cuando está basado en una ventana de dos palabras a cada lado de la

palabra este logra buenos resultados en el proceso de clasificación, cuyas clases están predefinidas por e tipo de palabras que existen dentro de cada idioma. Si se etiquetan las palabras en un orden fijo, por ejemplo, de izquierda a derecha, también se puede condicionar cada predicción de etiquetado a las predicciones de etiquetas realizadas en etiquetas anteriores [8].

En la tarea de reconocimiento de identidades nombradas se proporciona un documento en el cual se necesita encontrar las entidades nombradas y desestimar las demás palabras; dentro de esta tara también es necesario realizar el proceso de categorizar las entidades en un conjunto predefinido de categorías como pudiesen ser la locación, si la etiqueta hace referencia a una organización, si es una persona, entre otras categorías que se definen desde el entrenamiento [8]. Dentro del corpus algunas palabras pueden llegar a recibir una categoría diferente de acuerdo con el contexto en el que se emplean dicha palabra dentro del texto.

Estas dos tareas que se realizan de manera independientes y permiten en la etapa 4 de la investigación poder extraer información relevante para la generación y rellenado del arreglo que contendrá la información de cada palabra, generando una matriz de **n x m** longitud, donde **n** es el número de datos que contienen la palabra, y **m** es la cantidad de palabras contenida dentro del texto.

En el arreglo que se opta implementar se almacena la información que se considera relevante de cada palabra cuya implementación está relacionada a la sincronización de material multimedia y a los cálculos en el proceso sincronización como se aprecia en la **Figura 4.12.**

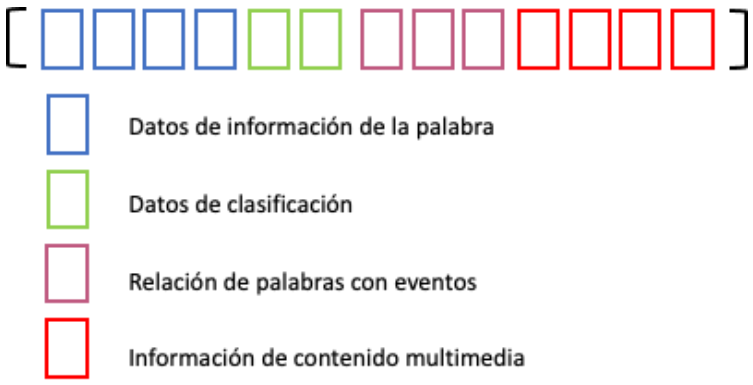


Figura 4.12. *Diagrama de estructura de cada elemento del texto, excluyendo signos de puntación.*

Nota. Elaboración propia.

En la

Figura *4.13* se desglosa más a detalle las funciones dentro de cada una de las categorías delimitándola por colores y se describe la información que contiene cada posición del arreglo, cuya longitud n se limitó en 12 datos.

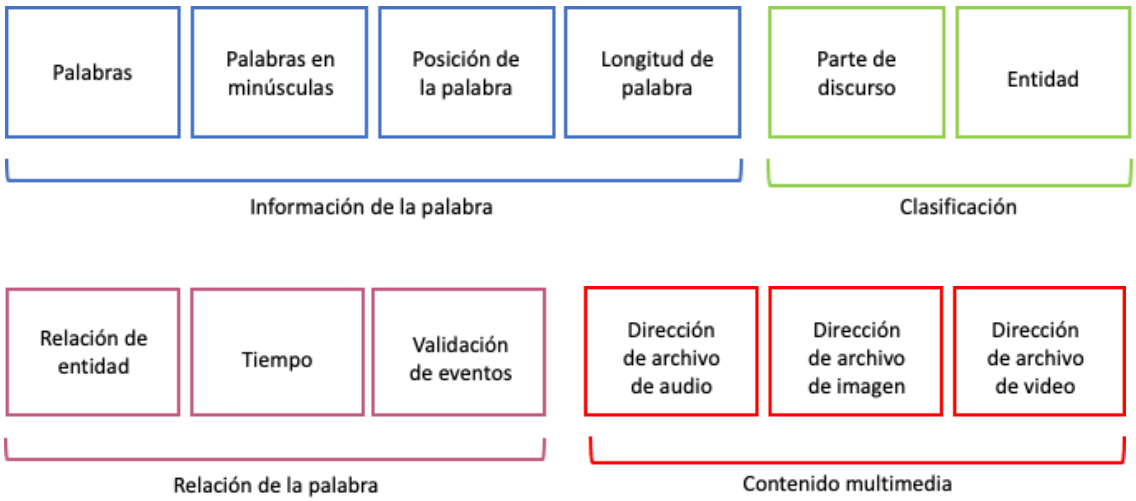


Figura 4.13. *Desglosado de procesamiento de cada palabra dentro del recuerdo.*

Nota. Elaboración propia.

Por medio de la librería referida, se pueden construir sistemas de extracción de información o comprensión del lenguaje natural, procesar y entender grandes volúmenes de texto contenidas dentro del corpus delimitados en el perfil del usuario.

En lo que corresponde a la parte experimental, se utilizó el modelo en español de nivel medio, sin desestimar las palabras vacías (“stop words” en inglés) las cuales son palabras muy frecuentes en el idioma, pero cuyo uso no genera un significado importante al contenido.

Se asume que el conjunto de textos fue revisado y ajustado para que su composición tenga sentido y cumpla con las reglas gramaticales del idioma implementado. El proceso anterior hace referencia a la corrección de errores en la escritura o transcripción de determinadas palabras para proceder con el procesamiento y extracción de información de los documentos.

Dentro de las tareas de identificación de parte de discurso e identidades nombradas donde el conjunto de palabra es separado en elementos individuales denominados tokens mismos que se obtienen al separar cada frase en sus elementos individuales sin considerar los signos de puntación.

Este proceso implementando el lenguaje de Python se puede llevar a cabo por medio de un ciclo que separe todo el texto en frases y otro que recorra cada frase donde pueda encontrar una serie de condiciones para realizar el proceso de separación de las palabras, generando un recorrido de izquierda a derecha para el español.

Explicando más a detalle el proceso de la etapa 2 para se debe desglosar lo que sucede dentro del proceso de separar la oración, que se convertirá de discurso escrito a oral en palabras individuales y considerarlas como estados independientes de otros dentro de un arreglo secuencial. Al visitar cada uno de estos estados individualmente, y de manera ordenada se puede llamar a los eventos que contienen el estado de manera paralela a otros eventos generados por otros estados o pausar un evento que inicio en un estado q_x. de la lista de palabras que conforman una frase como en la **Figura *4.14***.

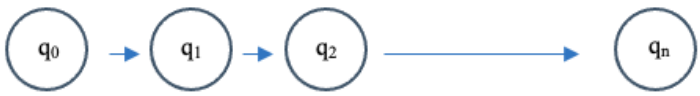


Figura 4.14 . *Diagrama secuencial de una frase.*

Nota. Elaboración propia.

En cada uno de los estados "q" se hace referencia a una única palabra dentro de una frase, por lo cual tienen un número finito de elementos dentro de su estructura, siendo el estado q_n la palabra que se encuentra al final de la oración delimitada en el español por signo especial "." que indica el cierre de una oración.

En **Figura *4.15*** se ejemplifica por medio de un diagrama el recorrido requerido para la identificación de cada token y el proceso de encapsularlo de frases o la delimitación de párrafos

según la implementación de la función escrita en Python. Es necesario recalcar que se debe de ajustar el proceso para considerar las reglas gramaticales del idioma trabajado.

Dentro de la investigación se delimito las reglas del idioma español por lo cual como anteriormente se mencionó el recorrido de las palabras debe realizarse de izquierda a derecha, donde cada espacio encontrado delimita a una palabra, habiendo casos especiales como la presencia de signos de puntuación tales como: los puntos, las comas, entre otros que identificaran el fin de la palabra. Un caso particular de los textos planos es la presencia de la combinación de caracteres "/n" el cual hace referencia a un salto de línea, estos saltos se dejan a discreción del programador el removerlos dentro de su proceso o implementarlos para recuperar la estructura original del texto. Para el caso de presente se remueven en el proceso de generación de frases, pero se consideran en el proceso de generación de párrafos.

Para la etapa 5 del proceso de sincronización se usa el resultado de la etapa 4, donde se obtenida la información de cada palabra generando una matriz de datos. En las siguientes líneas de seudocódigo se observa el desglose de los pasos para la selección y sincronización del material de las transcripciones mostrados en el **Figura 4.15.**.

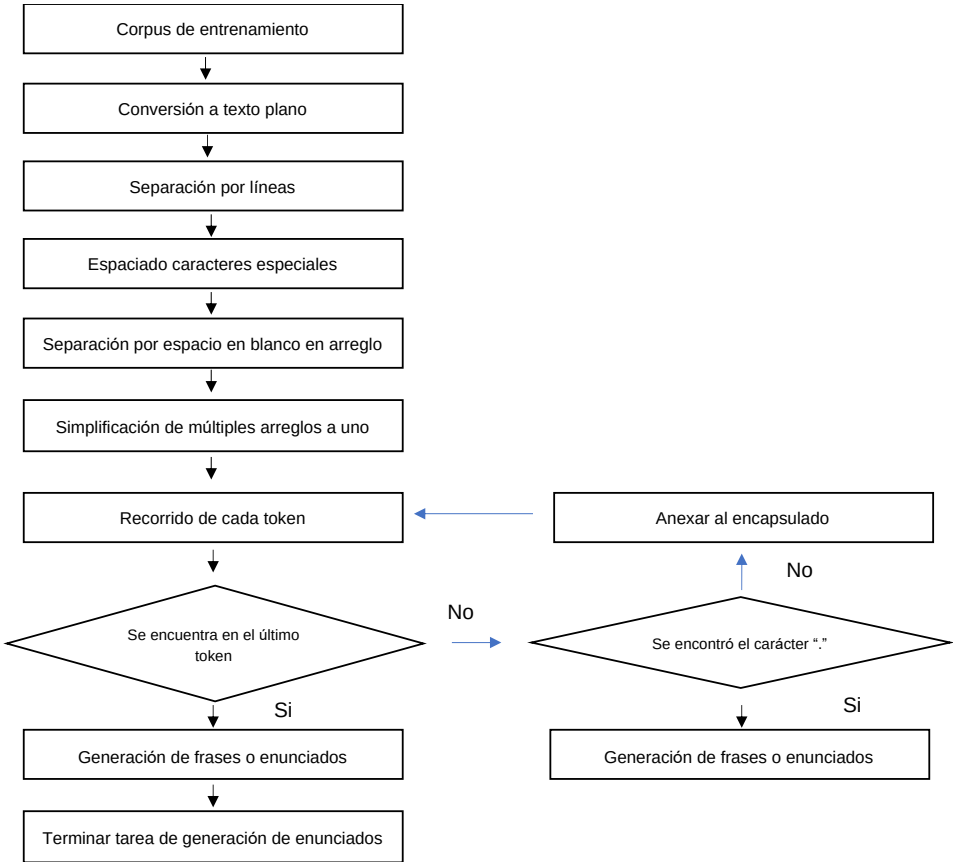


Figura 4.15 . *Diagrama de procesamiento de texto.*

Nota. Elaboración propia.

Es importante mencionar que no todas las categorías de partes de discurso (POS) aparecen dentro de una oración o requieren un proceso de sincronización con el material multimedia, como son los casos de la presencia de artículos, verbos, conjunciones entre otras categorías, pero la presencia posterior a una etiqueta que si lo requiera puede permitir el aumentar el tiempo del contenido a mostrado con referencia a una entidad nombrada en la oración o frase que si requiere una sincronización y llamado a este material. Ello se refleja en el código de la **Algoritmo 4.1**.

Algoritmo 4.1 *Diagrama de procesamiento de recuerdos y sincronización.*

1: Selección de texto (recuerdo).

2: Extracción de frases.

3: Duplicado de frase

4: Eliminación de signos ortográficos dentro de duplicado.

5: Generación de bolsa de palabras.

6: Incorporación de palabras nuevas a la bolsa de palabras del perfil del usuario.

7: Recorrido de todas las frases dentro del texto.

Extracción de Tokens por frase.

Clasificación de partes de discurso

Clasificación de entidades.

Recorrer todos los tokens dentro de una frase.

Rellenado de arreglo de datos por token.

Si se requiere material por tipo de POS.

Revisión de material multimedia.

Si existe material dentro del perfil del usuario.

Selección de archivos multimedia.

Si no se consulta la base de datos de la aplicación.

Si existe material.

Selección de archivos multimedia.

Si no existe material

Descarga de la red los archivos multimedia.

Guardar el archivo con descripción del Token.

Selección de archivos multimedia.

Si no se requiere material por tipo de POS.

Continuar con siguiente Token.

8: Subir nuevo archivo al servidor.

9: Ligar recuerdo del usuario a recuerdo dentro de la aplicación.

Nota. Elaboración propia.

Los recorridos por cada frase permiten identificar las expresiones regulares por medio de n-gramas, controlando la cantidad de palabras que se agrupan.

Un n-grama es un conjunto de n elementos consecutivos en un documento de texto, que puede incluir palabras, números, símbolos y puntuación. Los modelos de n-gramas son útiles en muchas aplicaciones de análisis de texto en que la secuencia de palabras es pertinente, tales como análisis de sentimiento, clasificación de texto y generación de texto. El modelado de n-gramas es una de las técnicas utilizadas para convertir texto de un formato no estructurado a un formato estructurado [10]. Para dicha implementación en la investigación se deja solo el uso de n-gramas de palabras, excluyendo signos de puntación o exclamación.

Una de las ventajas de la identificación de las expresiones regulares, es que se puede poseer una mayor precisión sobre el dominio de patrones con los que se está trabajando con cada usuario. Entre más recuerdos se van ingresando al perfil del usuario, se ira mejorando el proceso de personalizar el sistema. También esto permitirá determinar la frecuencia del uso de palabras, la extensión de su vocabulario, los patrones del habla, entre algunas de las opciones lo cual permitirá facilitar la extracción de información y clasificación de los textos.

El proceso de n-gramas implementado permite identificar una entidad que va agrupada en una expresión regular. Es importante mencionar que el uso de las expresiones regulares se considera para patrones individuales de una única persona ya que es susceptible a patrones definidos del habla o de la expresión escrita y no llegan a ser reglas gramaticales que puedan repetirse entre un grupo de individuos o en cada caso.

Es importante destacar que los n gramas son susceptibles a palabras con: caracteres extras, repetición de letras, errores ortográficos entre otros, por eso se hace énfasis en la revisión y correcciones como preprocesamiento de los textos para posteriormente hacer su análisis y extracción de información en la etapa 3 donde se lleva a cabo el proceso de generación de transcripciones y creación de los textos en formato de texto plano.

La obtención de bolsa de palabras por el procesamiento de cada memoria permitirá realizar una actualización recurrente dentro del perfil del usuario. Y al segregar a usuarios de la misma zona geográficas se podrá detectar regionalismos dentro de un mismo idioma e incorporar estas nuevas palabras al sistema para mejorar los niveles de precisión.

Para todas aquellas palabras desconocidas se optó por implementar un modelo de extracción de datos de la web que ingresa a la página oficial de la Real Academia de la Lengua Española (RAE), buscando la palabra deseada y extrayendo la implementación de la palabra en el español. El código mostrado en el **Algoritmo 4.2** ejemplifica la secuencia de pasos para realizar esta tarea.

Se realiza la búsqueda en la página oficial de la Real Academia de la Lengua Española, ya que es una institución cultural dedicada a la regularización lingüística entre el mundo hispanohablante [11].

Algoritmo 4.2 *Pseudocódigo de búsqueda de categorías y entidades por medio de Python.*

Entrada: Selección de la palabra o lista de palabras

10: Abrir navegador predefinido.

11: Ingreso a sitio de la Real Academia Española (RAE)

12: Si es solo una palabra

Búsqueda por palabra

Extracción de categorías del discurso de las palabras

Extracción de la entidad.

13: Si no es solo una palabra

14: Recorrido de toda la lista de palabras.

Búsqueda por palabra

Extracción de categorías del discurso de las palabras

Extracción de la entidad.

15: Generación de bolsa de palabras.

16: Incorporación de palabras nuevas a la bolsa de palabras del perfil del usuario.

17: Recorrido de todas las frases dentro del texto.

Extracción de Tokens por frase.

Clasificación de partes de discurso

Clasificación de entidades.

18: Cerrar navegador.

Nota. Elaboración propia.

La librería de SpaCy implementada en la etapa 3 para la tarea de identificación de partes de discurso y entidades nombradas permite reentrenar el modelo, con la finalidad de permitir incorporar nuevos elementos para ir mejorando los niveles de precisión en la tarea de clasificación.

En la etapa 4 donde se requiere realizar la tarea de sincronización cuando previamente se identifica cada uno de los elementos del texto, se genera las etiquetas para cada elemento y se realiza el proceso de captura de datos dentro de la matriz de n x m. Se procede a buscar elementos multimedia dentro de las bases de datos para completar el espacio designado en el arreglo de cada palabra, es importante destacará que, por las ambigüedades del idioma español, aunque se repita una palabra dentro del texto puede tener implementaciones diferentes por lo cual no necesariamente, tendrá su arreglo los mismos datos.

Al terminar estos procesos se podrá realizar la tarea de sincronizar el contenido del texto con el contenido multimedia necesario para la narrativa como se muestra en la **Figura *4.16***.

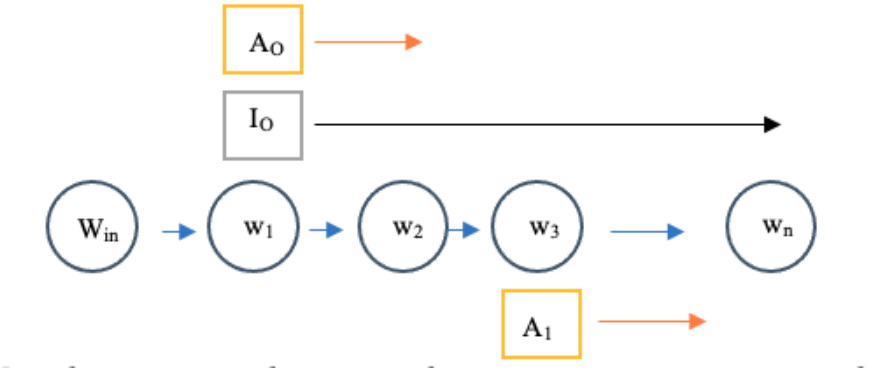


Figura 4.16. *Sincronización de elementos con narrativa.*

Nota. Elaboración propia.

La parte de la identificación de entidades y partes del discurso en la etapa 3 nos permite identificar contenido multimedia que anteriormente fue subido por la persona a su perfil y poder llamarlo a la historia seleccionada que se ingresa en el sistema de sincronización implementado en la etapa 5.

En la etapa 4 donde se realiza la tarea de rellenado del arreglo de datos, se buscó la flexibilidad de acceder primero a los archivos del usuario para posteriormente ingresar a la base de datos de archivos de uso general y poder generar la relación con el elemento que se buscado.

El algoritmo propuesto debe ser capaz de generar un preprocesamiento de la historia subida al sistema para llevar a cabo el análisis del contenido dentro de un contenido antes de permitir comience la narrativa enriquecida con contenido multimedia.

Por las limitantes del uso de librerías de terceros se deben estimar los tiempos de duración de los elementos a aparecer dentro del recuerdo y de manera automática los llamados. La librería que se implementa en la conversación de texto a discurso hablado genera un audio en formato MP3 el cual no se puede separar por elemento si no que incorpora todos los elementos dentro del texto ingresado.

El llamado de los elementos multimedia debe ser un proceso paralelo asíncrono ejemplificado en la **Figura *4.16***. De tal manera que no dependan el llamado de un elemento de la aparición de otro si no del contexto y el momento en el que se encuentra la historia así pudiendo complementar los contenidos de imágenes, audio o video sin requerir que estén todos u otro elemento, como se muestra en la **Figura *4.17*.**

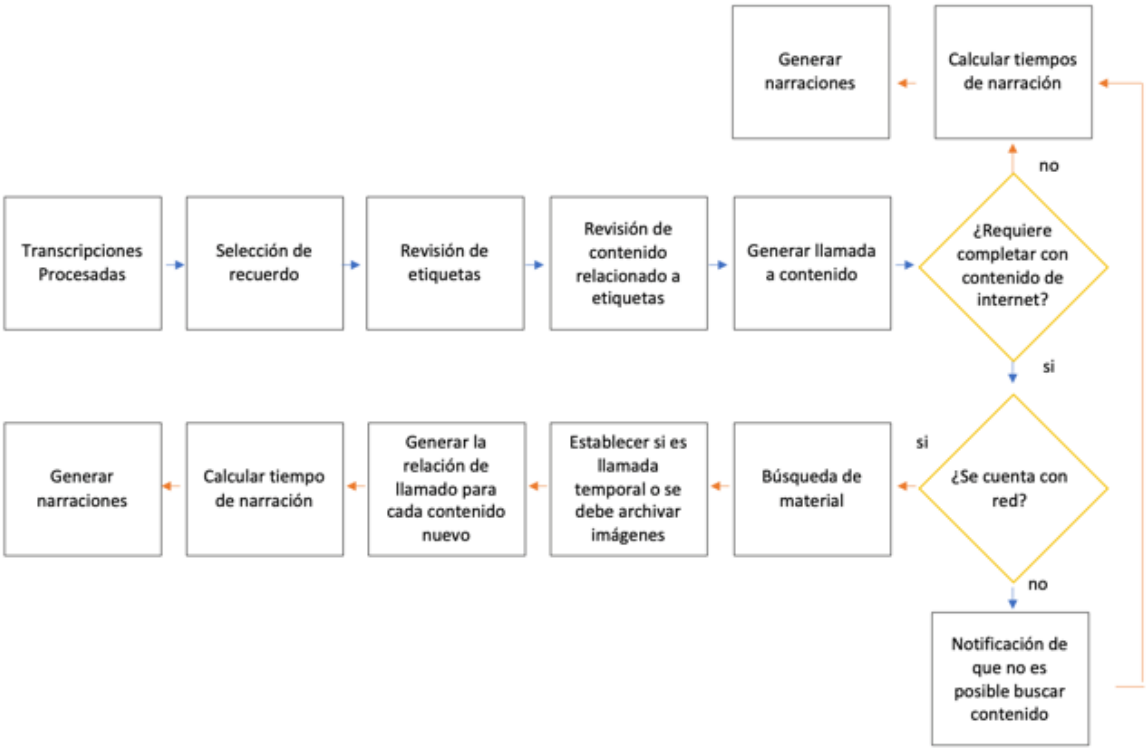


Figura 4.17. *Diagrama de flujo de procesamiento de recuerdo*

Nota. Elaboración propia.

A partir de este punto al generar la narrativa de la historia, una imagen puede ser mostrada en la pantalla del dispositivo sin pausar la narrativa y a su vez, otro evento puede solicitar un efecto sonoro o arrancar de una pista audiovisual sin opacar la voz principal o parar los otros dos eventos, siendo esto calculado por el sistema.

Los eventos se pueden poner a dormir o concluir según indique el arreglo la configuración predefinida para cada palabra de acuerdo con el texto. Por lo cual, la manera que se comparta un elemento relacionado a una palabra en texto va a variar al contexto de la historia y a los otros elementos que se encuentran dentro de ella.

La opción de complementar las historias con elementos fuera de los archivos, el usuario o de los archivos generales cargados al momento queda a discreción de usuario. De igual modo queda a discreción pues no se requiere la función del llamado a elementos multimedia para la narración por defecto.

Al activarse la función de llamada a elementos multimedia, el algoritmo será capaz revisar con el primer preprocesamiento los elementos con los que cuenta en su base de datos interna del dispositivo que tengan una relacionados al recuerdo, complementarlo con elementos del repositorio general o permitir una descarga de elementos de la red tal como imágenes, videos, elementos de audio, etc., para complementar los vacíos de elementos.

Si el dispositivo cuenta con una conexión a la red y estos vacíos son de carácter universal como ciudades, países, lugares públicos, por medio de algoritmos de raspado de web ("scraping" en inglés) se podrá descargar los elementos que complementen la historia y añadirlos a una base de datos temporal de donde serán llamados. Si, por el contrario, son de carácter privado no se llamará a ningún archivo o elemento haciendo referencia a fotografía personales con las que el sistema no cuente.

Capítulo 5

Evaluación Experimental

5.1 Evaluación experimental

La parte de captura se realiza con dos textos populares, el primero tiene por nombre “La leyenda del Conejo en la Luna” y el segundo “La liebre y la tortuga.”. Ambos textos se implementan para realizar la captura de audios y observar el proceso de la generación de transcripciones automatizadas.

Por medio de los documentos seleccionados se busca medir la relación de 150 palabras por minuto para establecer el estándar de tiempo implementado para la narrativa automática, al comparar el total de las palabras contenidas dentro del texto con el tiempo de duración de la grabación dejando un porcentaje de tolerancia de +- 5%.

Historia	Palabras	Tiempo relación 150 palabras/min
Leyenda - “La leyenda del Conejo en la Luna”	188	1:16
Fábula - “La liebre y la tortuga.”.	154	2:22

Tabla 5.1. *Tabla de datos sobre textos de evaluación.*

La relación de las palabras por minutos se obtiene del valor mínimo entregado por el gobierno de México [46] para personas que tienen un nivel mínimo de estudio de 3 ª de secundaria. En la siguiente tabla se obtiene los valores promedios por nivel de grado comenzando desde primero de primaria.

Historia	Palabras	Tiempo relación 150 palabras/min
Primaria	1º	35 a 59
	2º	60 a 84
	3º	85 a 99
	4º	100 a 114
	5º	115 a 124
	6º	125 a 134
Secundaria	1º	135 a 144
	2º	145 a 154
	3º	155 a 160

Tabla 5.2. *Relación de velocidad de lectura en relación con nivel académico.*

La captura de los documentos se realiza en 4 sujetos para medir los tiempos de grabaciones entre diferentes edades y sexos.

Sujeto	Genero	Tiempo Historia 1 en min	Tiempo Historia 2 en min
Sujeto A	Mujer	1:13	2:16
Sujeto B	Mujer	1:20	2:10
Sujeto C	Hombre	0:55	1:44
Sujeto D	Hombre	1:10	2:11

Tabla 5.3. *Tabla de datos de captura de grabaciones.*

Figura 5.1. *Archivos de captura de textos seleccionados.*

El desarrollo experimental se realiza por medio de Pycharm un IDE para Python y de Colaboratory de la compañía Google el cual permite programar y ejecutar Python en los navegador, sin configuraciones, con acceso a GPUs sin costo y permite compartir contenido fácilmente.

Figura 5.2 . *Pycharm, IDE para Python.*

5.1.1 Texto de control

La leyenda del Conejo en la Luna

Durante la búsqueda de su alimento, *Quetzalcóatl* se encontró con un *conejo* que estaba cenando.

Su estómago se retorcía de hambre, por lo que se acercó al *animal*:

- ¿Qué comes?, - le preguntó *Quetzalcóatl* al *conejo*

- *Zacate*, ¿quieres?, - Preguntó el animal

- No, yo no puedo comer eso, si lo hago podría morir

- Soy pequeño y tal vez conmigo no te llenes, pero puedes comerme para poder seguir tu *camino*

A pesar de que el *conejo* moría de miedo, humildemente se ofreció para poder ayudar al dios *Quetzalcóatl*.

La acción de valentía y humildad del conejo sorprendió al *dios azteca*, por lo que no dudó en acariciarlo y decirle:

- Podrás ser un *animal* muy pequeño, pero a partir de ahora, mereces que los hombres te admiren por la grandeza de tu *corazón*.

Seguidamente, *Quetzalcóatl* procedió a levantar al *conejo* muy alto, apuntando hacia la *luna*, haciendo que su imagen se quedara plasmada por siempre en este astro.

Desde entonces, cada noche de *luna llena*, puede apreciarse con claridad la forma imperdible del *conejo* en la *luna*.

Elementos destacables: Quetzalcóatl, conejo, estómago, animal, zacate, dios Azteca, corazón, luna, astro.

La liebre y la tortuga.

En el mundo de los *animales* vivía una *liebre* muy orgullosa y vanidosa, que no cesaba de pregonar que ella era el animal más veloz del bosque, y que se pasaba el día burlándose de la lentitud de la *tortuga*.

- ¡Eh, *tortuga*, no corras tanto! Decía la liebre riéndose de la *tortuga*.

Un día, a la *tortuga* se le ocurrió hacerle una inusual apuesta a la *liebre*:

- *Liebre*, ¿vamos a hacer una carrera? Estoy segura de poder ganarte.

- ¿A mí? Preguntó asombrada la *liebre*.

- Sí, sí, a ti, dijo la *tortuga*. Pongamos nuestras apuestas y veamos quién gana la carrera.

La *liebre*, muy engreída, aceptó la apuesta prontamente.

Así que todos los *animales* se reunieron para presenciar la carrera. El *búho* ha sido el responsable de señalizar los puntos de partida y de llegada. Y así empezó la carrera:

Astuta y muy confiada en sí misma, la *liebre* salió corriendo, y la *tortuga* se quedó atrás, tosiendo y envuelta en una nube de polvo. Cuando empezó a andar, la *liebre* ya se había perdido de vista. Sin importarle la ventaja que tenía la *liebre* sobre ella, la tortuga seguía su ritmo, sin parar.

La *liebre*, mientras tanto, confiando en que la *tortuga* tardaría mucho en alcanzarla, se detuvo a la mitad del camino ante un frondoso y verde *árbol*, y se puso a descansar antes de terminar la carrera. Allí se quedó dormida, mientras la *tortuga* seguía caminando, paso tras paso, lentamente, pero sin detenerse.

No se sabe cuánto tiempo la *liebre* se quedó dormida, pero cuando ella se despertó, vio con pavor que la *tortuga* se encontraba a tan solo tres pasos de la *meta*. En un sobresalto, salió corriendo con todas sus fuerzas, pero ya era muy tarde: ¡la *tortuga* había alcanzado la meta y ganado la carrera!

Ese día la *liebre* aprendió, en medio de una gran humillación, que no hay que burlarse jamás de los demás. También aprendió que el exceso de confianza y de vanidad, es un obstáculo para alcanzar nuestros objetivos. Y que nadie, absolutamente nadie, es mejor que nadie.

Elementos destacables: liebre, árbol, tortuga, meta, animales y búho.

5.1.2 Generación de transcripciones y procesamiento de audios.

Para la implementación de la generación de transcripciones por medio de audios se implementó la librería *Pydub* que permite convertir los formatos por medio del lenguaje Python de MP4 o del formato MA4 a formato de WAV sin modificar la duración o el contenido del audio.

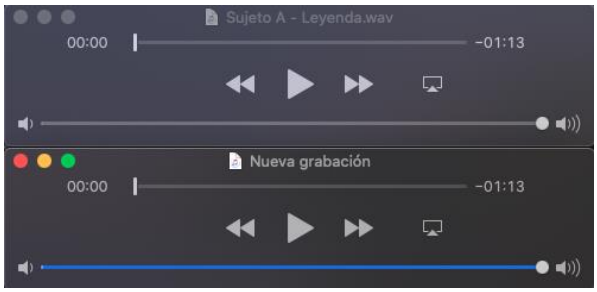


Figura 5.3. *Comparación de tiempos entre formato origen y resultante*

Siguiendo el siguiente Seudocódigo en el **Algoritmo 5.1**, podemos obtener la conversión de los formatos y la generación de las transcripciones.

Algortimo 5.1 *Pseudocódigo de conversión de formatos de audio y generación de transcripciones en texto.*

Entrada: Selección del audio

1: Establecer formato de origen
2: Seleccionar formato de conversión.
3: Establecer nombre de archivo resultante.
4: Implementar de la librería Pydub la función *AudioSegment*.
5: Guardar nuevo formato.
6: Configurar librería *SpeechToText_short* con idioma y formato de salida.
7: Generar transcripción.
8: Guardar archivo resultante en texto plano.

Los formatos obtenidos en el tratamiento de los 8 audios se dividirán en dos grupos los cuales están relacionados a su audio de origen, los cuales fueron clasificados manualmente para la detección de las identidades y de las partes importantes del discurso como se muestra en los

siguientes textos, donde las identidades nombradas están identificadas en color verde y los verbos en naranja.

5.1.2.1 Generación de transcripciones.

En las siguientes transcripciones se observa la variación generada a pesar de trabajar con el mismo texto, donde la pronunciación, la velocidad del habla crean divergencias en las transcripciones. En la parte final se agrupa cada 10 palabras de los textos generados para observar las diferencias. Los textos resultantes quedan almacenados en un archivo de texto plano.

En los siguientes documentos se muestra las transcripciones generadas por las conversiones de speech to text de la librería, para los primeros sujetos de evaluación, los textos mostrados son tal y como se generan no reciben modificaciones, como se puede observar que en algunos casos se visualiza "quetzalcoatl" y en otros "Quetzalcóatl".

Sujeto A– Leyenda

la leyenda del conejo en la luna durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando su estómago se retorcía de hambre por lo que se acercó al animal que comes le pregunto quetzalcoatl al conejo zacate quieres pregunto el animal No yo no puedo comer eso sí lo hago podría morir soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna así Endo que su imagen se quedará plasmada por siempre en este Astro desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del conejo en la luna

Sujeto B – Leyenda

la leyenda del cole del conejo en la luna durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando su estómago se retorcía de hambre que comes le pregunto Quetzalcóatl conejo Jaja Te quieres pregunto el animal No yo no podía comer eso sí lo hago podría morir soy pequeño y tal vez conmigo no te llenes pero pues cómeme comerme para poder seguir tu camino a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl la acción de valentía y humildad del conejo y sorprender al dios Azteca por lo que no dudo en acariciarlo y decirle podrá ser un animal muy pequeño pero a partir de ahora me dices que los hombres terminé por la grandeza de tu corazón seguidamente quetzalcoátl procedió a levantar al conejo muy alto apuntándose la luna

haciendo que su imagen se quedará plasmada por siempre en este Astro desde entonces cada noche de luna llena puede se puede apreciarse con Claridad la forma imperdible del conejo en la luna

Sujeto C – Leyenda

la leyenda del conejo en la luna Durante los alimentos que se encontró con un conejo que estaba cenando su estómago se retorcía de hambre por lo que se le acerco el animal que come el conejo zacate quieres yo no puedo comer eso

Sujeto D – Leyenda

leyenda del conejo en la luna durante la búsqueda cemento que se encontró con un conejo que estaba cenando su estómago se retorcía de hambre por lo que se acercó al animal que come le pregunto qué es el conejo zacate quieres preguntar animal No yo no puedo comer eso sí lo hago podría morir soy pequeño y tal vez conmigo no te llenes Pero puedes comer para poder seguir tu camino a pesar de que conejo murió de miedo humildemente su precio para poder ayudar al dios quezalcoatl la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudan acariciarlo decirte podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiran por la grandeza de tu corazón seguidamente cosaco accesible matar el conejo muy alto apuntando hacia la luna haciendo que su imagen que se quedará plasmado por siempre de teatro desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del conejo en la luna

5.1.2.2 Extracción de identidades nombradas y partes del discurso.

En esta sección experimenta se analiza con respecto a la versión original las traducciones de los 4 sujetos de evaluación donde se reorganiza para mantener la estructura de las líneas del texto original con respecto a la leyenda del conejo y la luna, para posteriormente hacer un ajuste en los textos de los sujetos de estudios al completar palabras faltantes y marcar palabras con error.

Acomodo de transcripciones.

Original: La leyenda del Conejo en la Luna
Sujeto A: la leyenda del conejo en la luna
Sujeto B: la leyenda del conejo en la luna
Sujeto C: la leyenda del conejo en la luna
Sujeto D: leyenda del conejo en la luna

Original: Durante la búsqueda de su alimento, *Quetzalcóatl* se encontró con un *conejo* que estaba cenando.
Sujeto A: durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando
Sujeto B: durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando
Sujeto C: durante los alimentos que se encontró con un conejo que estaba cenando
Sujeto D: durante la búsqueda cemento que se encontró con un conejo que estaba cenando

Original: Su estómago se retorcía de hambre, por lo que se acercó al *animal*:
Sujeto A: su estómago se retorcía de hambre por lo que se acercó al animal
Sujeto B: su estómago se retorcía de hambre por lo que se acercó al animal al conejo
Sujeto C: su estómago se retorcía de hambre por lo que se le acerco el animal
Sujeto D: su estómago se retorcía de hambre por lo que se acercó al animal

Original: - ¿Qué comes?, - le preguntó *Quetzalcóatl* al *conejo*
Sujeto A: que comes le pregunto quetzalcoatl al conejo
Sujeto B: que comes le pregunto quetzalcoatl
Sujeto C: que come el conejo
Sujeto D: que come le pregunto qué es el conejo

Original: - *Zacate*, ¿quieres?, - Preguntó el animal
Sujeto A: zacate quieres pregunto el animal
Sujeto B: zacate quieres pregunto el animal
Sujeto C: zacate quieres

Sujeto D: zacate quieres preguntar animal

Original: - No, yo no puedo comer eso, si lo hago podría morir

Sujeto A: No yo no puedo comer eso sí lo hago podría morir
Sujeto B: No yo no puedo comer eso sí lo hago podría morir
Sujeto C: yo no puedo comer eso
Sujeto D: No yo no puedo comer eso sí lo hago podría morir

Original: - Soy pequeño y tal vez conmigo no te llenes, pero puedes comerme para poder seguir tu *camino*

Sujeto A: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino
Sujeto B: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino
Sujeto C:
Sujeto D: soy pequeño y tal vez conmigo no te llenes Pero puedes comer para poder seguir tu camino

Original: A pesar de que el *conejo* moría de miedo, humildemente se ofreció para poder ayudar al dios *Quetzalcóatl*.

Sujeto A: a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl
Sujeto B: a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl
Sujeto C:
Sujeto D: a pesar de que conejo murió de miedo humildemente su precio para poder ayudar al dios quezalcoatl

Original: La acción de valentía y humildad del conejo sorprendió al *dios azteca*, por lo que no dudó en acariciarlo y decirle:

Sujeto A: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle
Sujeto B: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle
Sujeto C:
Sujeto D: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudan acariciarlo decirte

Original: - Podrás ser un *animal* muy pequeño, pero a partir de ahora, mereces que los hombres te admiren por la grandeza de tu *corazón*.

Sujeto A: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón
Sujeto B: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón
Sujeto C:
Sujeto D: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiran por la grandeza de tu corazón

Original: Seguidamente, *Quetzalcóatl* procedió a levantar al *conejo* muy alto, apuntando hacia la *luna*,

Sujeto A: seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna

Sujeto B: seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna

Sujeto C:

Sujeto D: seguidamente cosaco accesible matar el conejo muy alto apuntando hacia la luna

Original: haciendo que su imagen se quedara plasmada por siempre en este astro.

Sujeto A: así Endo que su imagen se quedará plasmada por siempre en este Astro

Sujeto B: así Endo que su imagen se quedará plasmada por siempre en este Astro

Sujeto C:

Sujeto D: haciendo que su imagen que se quedará plasmado por siempre de teatro

Original: Desde entonces, cada noche de *luna llena*, puede apreciarse con claridad la forma imperdible del

Sujeto A: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Sujeto B: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Sujeto C:

Sujeto D: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Original: *conejo* en la *luna*.

Sujeto A: conejo en la luna

Sujeto B: conejo en la luna

Sujeto C:

Sujeto D: conejo en la luna

Correcion de transcripciones.

Dentro de este apartado se hacen los ajustes de cada transcripcion de los sujestos al completar con respecto a la linea original y marcar los errores de transcrpción-. Con el analisi se pretende calificar al sujetos con el mayor puntaje y evaluar si varia las transcripciones del mismo audio en 4 experimentos de generacción de converción de audio a texto.

Se usa código de colores para marcar las variaciones y palabras faltantes en cada linea a evaluar usando el siguiente formato:

- Rojo: palabras faltantes
- Naranja paralabras con errores.
- Negro: Palabras bien escritas

Original: La leyenda del Conejo en la Luna

Sujeto A: la leyenda del conejo en la luna

Sujeto B: la leyenda del conejo en la luna

Sujeto C: la leyenda del conejo en la luna

Sujeto D: la leyenda del conejo en la luna

Original: Durante la búsqueda de su alimento, *Quetzalcóatl* se encontró con un *conejo* que estaba cenando.

Sujeto A: durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando

Sujeto B: durante la búsqueda de su alimento quetzalcoatl se encontró con un conejo que estaba cenando

Sujeto C: durante (los) la búsqueda de su alimentos quetzalcoatl (que) se encontró con un conejo que estaba cenando

Sujeto D: durante la búsqueda (cemento) de su alimentos quetzalcoatl (que) se encontró con un conejo que estaba cenando

Original: Su estómago se retorcía de hambre, por lo que se acercó al *animal*:

Sujeto A: su estómago se retorcía de hambre por lo que se acercó al animal

Sujeto B: su estómago se retorcía de hambre por lo que se acercó al animal

Sujeto C: su estómago se retorcía de hambre por lo que se (le) acerco (el) al animal

Sujeto D: su estómago se retorcía de hambre por lo que se acercó al animal

Original: - ¿Qué comes?, - le preguntó *Quetzalcóatl* al *conejo*

Sujeto A: que comes le pregunto quetzalcoatl al conejo

Sujeto B: que comes le pregunto quetzalcoatl al conejo

Sujeto C: que come le pregunto quetzalcoatl al (el) conejo

Sujeto D: que come le pregunto quetzalcoatl (qué es el) al conejo

Original: - *Zacate*, ¿quieres?, - Preguntó el animal

Sujeto A: zacate quieres pregunto el animal

Sujeto B: zacate quieres pregunto el animal

Sujeto C: zacate quieres pregunto el animal

Sujeto D: zacate quieres preguntar el animal

Original: - No, yo no puedo comer eso, si lo hago podría morir

Sujeto A: No yo no puedo comer eso sí lo hago podría morir

Sujeto B: No yo no puedo comer eso sí lo hago podría morir

Sujeto C: No yo no puedo comer eso sí lo hago podría morir

Sujeto D: No yo no puedo comer eso sí lo hago podría morir

Original: - Soy pequeño y tal vez conmigo no te llenes, pero puedes comerme para poder seguir tu *camino*

Sujeto A: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino

Sujeto B: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino

Sujeto C: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino

Sujeto D: soy pequeño y tal vez conmigo no te llenes Pero puedes comerme para poder seguir tu camino

Original: A pesar de que el *conejo* moría de miedo, humildemente se ofreció para poder ayudar al dios *Quetzalcóatl*.

Sujeto A: a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl

Sujeto B: a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl

Sujeto C: a pesar de que el conejo moría de miedo humildemente se ofreció para poder ayudar al dios Quetzalcóatl

Sujeto D: a pesar de que el conejo murió de miedo humildemente (su precio) se ofrecio para poder ayudar al dios quezalcoatl

Original: La acción de valentía y humildad del conejo sorprendió al *dios azteca*, por lo que no dudó en acariciarlo y decirle:

Sujeto A: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle

Sujeto B: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle

Sujeto C: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo en acariciarlo y decirle

Sujeto D: la acción de valentía y humildad del conejo sorprendió al dios Azteca por lo que no dudo (dudan) en acariciarlo y decirle (decirte)

Original: - Podrás ser un *animal* muy pequeño, pero a partir de ahora, mereces que los hombres te admiren por la grandeza de tu *corazón*.

Sujeto A: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón

Sujeto B: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón

Sujeto C: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiren por la grandeza de tu corazón

Sujeto D: podrá ser un animal muy pequeño pero a partir de ahora mereces que los hombres te admiran por la grandeza de tu corazón

Original: Seguidamente, *Quetzalcóatl* procedió a levantar al *conejo* muy alto, apuntando hacia la *luna*,

Sujeto A: seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna

Sujeto B: seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna

Sujeto C: seguidamente quetzalcoatl procedió a levantar al conejo muy alto apuntando hacia la luna

Sujeto D: seguidamente quetzalcoatl (cosaco accesible matar el) procedió a levantar al conejo muy alto apuntando hacia la luna

Original: haciendo que su imagen se quedara plasmada por siempre en este astro.

Sujeto A: haciendo (así Endo) que su imagen se quedará plasmada por siempre en este Astro

Sujeto B: haciendo (así Endo) que su imagen se quedará plasmada por siempre en este Astro

Sujeto C: a haciendo que su imagen se quedará plasmada por siempre en este Astro

Sujeto D: haciendo que su imagen que se quedará plasmado por siempre de (teatro) en este Astro

Original: Desde entonces, cada noche de *luna llena*, puede apreciarse con claridad la forma imperdible del

Sujeto A: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Sujeto B: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Sujeto C: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Sujeto D: desde entonces cada noche de luna llena puede apreciarse con Claridad la forma imperdible del

Original: *conejo* en la *luna*.

Sujeto A: conejo en la luna

Sujeto B: conejo en la luna

Sujeto C: conejo en la luna

Sujeto D: conejo en la luna

La comparación entre las transcripciones y los audios de los cuatro sujetos se pueden encontrar resumidas en la **Tabla *5.4***, pero un desgloce más a detalle se encuentra en la **Tabla *5.5***, donde se observa cada una de las comparaciones de las lineas del texto original.

Sujeto	Total Palabras	Total Palabras Correctas	Total Palabras Faltantes	Total Palabras con errores
Sujeto A	181	180	1	2
Sujeto B	181	180	1	2
Sujeto C	181	38	143	5
Sujeto D	181	157	24	14

Tabla 5.4. *Tabla de validación de diferencias entre sujetos y texto original.*

Línea del Texto original	Sujeto	Total Palabras	Total Palabras Correctas	Total Palabras Faltantes	Total Palabras con errores
1	A	7	7	0	0
	B	7	7	0	0
	C	7	7	0	0
	D	7	6	1	0
2	A	15	15	0	0
	B	15	15	0	0
	C	15	10	5	2
	D	15	11	4	1
3	A	13	13	0	0
	B	13	13	0	0
	C	13	12	1	2
	D	13	13	0	0
4	A	7	7	0	0
	B	7	7	0	0
	C	7	3	4	1
	D	7	5	2	3
5	A	5	5	0	0
	B	5	5	0	0
	C	5	2	3	0
	D	5	4	1	0
6	A	11	11	0	0
	B	11	11	0	0
	C	11	4	7	0
	D	11	11	0	0
7	A	17	17	0	0
	B	17	17	0	0
	C	17	0	17	0
	D	17	16	1	1
8	A	18	18	0	0
	B	18	18	0	0
	C	18	0	18	0
	D	18	15	3	2
9	A	21	21	0	0
	B	21	21	0	0
	C	21	0	21	0
	D	21	17	4	2
10	A	23	23	0	0
	B	23	23	0	0
	C	23	0	23	0
	D	23	23	0	0
11	A	13	13	0	0
	B	13	13	0	0
	C	13	0	13	0
	D	13	8	5	4
12	A	12	11	1	2
	B	12	11	1	2
	C	12	0	12	0
	D	12	9	3	1
13	A	15	15	0	0
	B	15	15	0	0
	C	15	0	15	0
	D	15	15	0	0
14	A	4	4	0	0
	B	4	4	0	0
	C	4	0	4	0
	D	4	4	0	0

Tabla 5.5. *Tabla desglosada de la validación de diferencias entre sujetos y texto original.*

Validación de transcripciones consecutivas sobre el mismo audio.

Se lleva a cabo una revisión para confirmar que el mismo audio genere la misma cantidad de palabras y con la misma secuencia dentro de las transcripciones generadas del sujeto que obtuvo un mayor número de palabras de la historia original y el menor número de palabras con errores y palabras faltantes.

Esta validación busca desestimar diferencias entre el mismo audio y el proceso de generación de transcripciones en diferentes momentos.

Estos archivos culla estructura gramatical ha sido eliminada casi en su totalidad como se puede apreciar en las transcripciones mostradas previamente. Al pasar estos textos sin formato se genera muchos errores en el proceso de clasificación tato para las entidades nombradas como para las POS al implementar la librería de SpaCy, por lo cual se decidido usar un modelo Oculto de Markov combinado con el algoritmo de Viterbi para el proceso de clasificación de partes del discurso, el cual determina la clase al observar una palabra actual y la interacción con la siguiente y sus posibles categorías como se muestra en **Figura *5.4***.

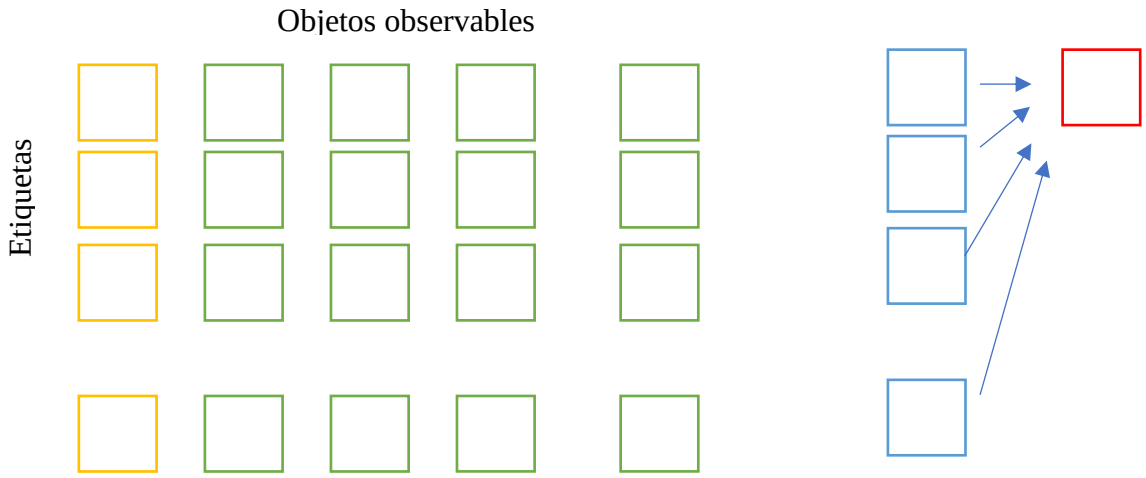


Figura 5.4. *Diseño de Trellis y pasos de Viterbi*

La implementación del algoritmo de Markov y de Viterbi conlleva un reto en el proceso de etiquetado de las palabras. Y como todo proceso de clasificación debe estar validado esta generación de etiquetas por una fuente especializada y confiable. Es donde aquí donde entra un proceso automatizado de descarga de datos de la red de la página oficial de la Real Academia

Española (RAE). Este proceso está limitado al universo de palabras de dos libros del autor Gabriel Gracia Márquez los cuales son “Amor en tiempos de Colera” y “Cien años de soledad”.

Para entrenar esta clase de modelos se debe separar el corpus en enunciados y asignar a cada una de las palabras la etiqueta que le corresponde, lo cual resulta en un proceso muy manual. Por ello se decidió usar el Software Excel y el lenguaje VBA para establecer un código que permitiera visitar cada oración y poder clasificarlo como se muestra en la figura.**Figura *5.5***.

Figura 5.5. *Diseño de Trellis y pasos de Viterbi*

En la siguiente ilustración **Figura *5.6*** se muestra la asignación de la fórmula para la obtención de la POS del diccionario de la bolsa de palabras del corpus de entrenamiento.

```
Sub Formula()

    Sheets("POS_separacion").Select

    Cells(3, 1).Select
    ActiveCell.FormulaR1C1 = "=IF(R[-1]C<>"""",VLOOKUP(R[-1]C,POS!C1:C2,2,0),"""")"
    Range("A3").Select
    Selection.AutoFill Destination:=Range("A3:EL3"), Type:=xlFillDefault
    'Range("B3:P3").Select

End Sub
```

Figura 5.6. *Hoja de cálculo adaptada para manejo de macros y revisión de POS manual.*

Es importante destacar que las cadenas de Markov no toman a bien palabras nueva y al aparecer en textos llegan a fallar los procesos de clasificación, en este tipo de casos solo será necesario validar que todas las palabras estén presentes dentro de las matrices del modelo y de no ser el caso mandar a llamar la función para obtención de datos de la página oficial de la RAE.

Algoritmo 5.2 *Pseudocódigo de búsqueda de categorías y entidades por medio de Python.*

Entrada: Selección de la palabra o lista de palabras

```
1:  Abrir navegador predefinido.
2:  Ingreso a sitio de la Real Academia Española (RAE)
3:  Si es solo una palabra
        Búsqueda por palabra
        Extracción de categorías del discurso de las palabras
        Extracción de la entidad.
4:  Si no es solo una palabra
5:  Recorrido de toda la lista de palabras.
        Búsqueda por palabra
        Extracción de categorías del discurso de las palabras
        Extracción de la entidad.
6:  Generación de bolsa de palabras.
7:  Incorporación de palabras nuevas a la bolsa de palabras del perfil del usuario.
8:  Recorrido de todas las frases dentro del texto.
    Extracción de Tokens por frase.
    Clasificación de partes de discurso
    Clasificación de entidades.
9:  Cerrar navegador.
```

Nota. Elaboración propia.

Descarga WEB de descriptores de palabras automatizado.

Para la parte de las técnicas de extracción de datos de la web se usó el software PyCharm en su versión gratuita y la librería Selenium [47].

El paquete Selenium se utiliza para automatizar la interacción del navegador web desde Python. Selenium admiten varios navegadores/controladores (Firefox, Chrome, Internet Explorer), así como el protocolo Remoto para su manejo. [48]

```
from selenium import webdriver
from selenium.webdriver.common.by import By
import pandas as pd
import xlrd
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
```

Figura 5.7. *Librerías para extracción de datos de la web.*

Se hace un análisis previo de la estructura de la página web de la RAE para la identificación de los elementos y clases que contienen la información necesaria de los atributos de cada palabra para poder realizar la selección de las partes del discurso.

```
def extraccion_palabras() :
    variacion_palabra = []
    doc = ('/Users/emilio/Desktop/Palabras.xlsx')
    book = xlrd.open_workbook(doc)
    sh = book.sheet_by_index(0)
    spreadsheet = book
    driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
    url = "https://dle.rae.es/"
    for rx in range(1, sh.nrows):
        palabra = str(sh.cell_value(rx, 0))
        url_total = url + palabra
        try:
            driver.get(url_total)
            tag_clase = driver.find_element(By.CLASS_NAME, 'd')
            tag_titulo = driver.find_element(By.CLASS_NAME, 'd').get_attribute("title")
            variacion_palabra.append([palabra, tag_clase.text,tag_titulo])
            variacion_palabra: list[list[str]] = [] ;
        exce...
            variacion_palabra.append([palabra, 'error', 'error'])
            print(' error ' + str(rx) + ' ' + str(palabra))
            next

        if (rx % 200 == 0):
            print('Palabras : ', rx)
    print(variacion_palabra)
    return variacion_palabra
```

Figura 5.8. *Función de búsqueda y extracción de descriptores de palabras en la web.*

5.1.2.3 Algoritmo de Viterbi Python

El código está dividido en inicialización, forward y obtención de mejor camino.

```
369] def viterbi(Enunciado, Matriz_Transicion, Matriz_Emision, Matriz_inicializacion, Tags ):

      Len_tag         = len(Tags)
      Len_Enunciado   =len(Enunciado)

      Matriz_Trans = Matriz_Transicion
      Matriz_Emis = Matriz_Emision
      #Matrices de que almacenan trancisiones y probabilidades

      Start       = Matriz_inicializacion.copy()
      Observables = eliminación_simbolos(Enunciado)

      #Matriz de indice del estado de la mejor posicion.
      M_T = np.zeros((Len_tag, len(Observables)))
      #Matriz de mejor probabilidad
      M_P = np.zeros((Len_tag, len(Observables)))

      #Preporocesamiento de tomar el enunciado y tokenizarlo.

      for l in range(len(Observables)):
          Observables[l]= Observables[l].lower()


      Estados = Tags
```

Figura 5.9. *Proceso de inicialización de variables.*

Se aprecia en la **Figura *5.9*** el proceso de inicialización de matrices que se adapta a la longitud de las oraciones; con la finalidad de obtener las probabilidades para la palabra que ocupa la primera posición se debe comenzar el proceso **Figura *5.30*** con cada una de las etiquetas multiplicando el valor obtenido por la probabilidad de emisión de la primera palabra observada con respecto a dicha etiqueta.

```
# ------------------------------------------------------------------

# Primera estapa de algoritmo Viterbi calculo de probabilidades
# de iniciar con determinada etiqueta

# ------------------------------------------------------------------

#Obtiene la primera posibilidad de ser cada uno de los estados
#pero relacidonado al observable.

for Es in range(len(Estados)):

  # Busca en el diccionario el indice de la palabra en la tabla de emisiones
  # y regresa la probabilidad para cada etiqueta con respecto a la primera palabra
  Obs_ = Diccionario_emision[Observables[0]]

  Prob_inici = Matriz_inicializacion[Es]
  Prob_obse =  Matriz_Emis[Es][Obs_]

  M_P[Es][0] = (Prob_inici) * (Prob_obse)     # Se alamacena la probabilidad para cada camino
  M_T[Es][0] = 0                               # Matriz de trayectoria se iguala a 0
```

Figura 5.10. *Parte de inicialización de variables.*

Como se mencionó dentro de la sección de marco teórico el algoritmo de Viterbi para el proceso de forward parte de la segunda palabra hasta completar todos los elementos del enunciado y llegar a la última palabra observable.

Para cada nueva palabra observable considera todos los estados que puede adquirir la palabra y genera una trayectoria con respecto a todos los posibles caminos que la pudieron llevar a ella.

Para el algoritmo de Viterbi se infiere que el número de caminos posibles se define por:

N caminos = (Número de elementos observables) * (Numero de etiquetas ^ 2)

Dentro del modelo se usan las 8 etiquetas básicas de idioma español y no se amplía al mismo número de etiquetas de la librería SpaCy la cual puede ser consultada dentro del Anexo 2 en la parte final de esta tesis.

Para concluir el proceso se debe observar el camino que genero se considera la mayor probabilidad de ocurrencia y a partir de esa etiqueta se implementa como índice y se comienza un proceso en reversa a través de la matriz de memoria donde se almacenaba la transición de los estados que se van obteniendo, el algoritmo implementado se encuentra en la **Figura *5.30***.

```
# ------------------------------------------------------------------

#           Segundo proceso de ir calculando hacia adelante

# ------------------------------------------------------------------

# Evita el primer punto por que se considero ya de partida.
# Recorre las etiquetas
# Probabilidad acumulada *
# [Probabilidad de transicion al estado nuevo * Proobabilidad emision del obsevable por el estado * ]

for ob in range(1, len(Observables)):
  Obs_re = Diccionario_emision[Observables[ob]]  #Devuelve el indice de esa palabra para la matriz de emisiones

  #Recupera todas las probabilidades de la columna anterior

  # Cada nodo corresponde a un espacio en los grafos Trellis
  # Se genera un recorrido de cada nodo de una columna.
  Matriz_Transicion= np.reshape(Matriz_Trans,(Len_tag,Len_tag))

  for nodo in  range(len(Estados)):
      pro_nod=[]
      for Es in  range(len(Estados)):
         Prob_camino      = M_P[Es][ob-1]
         Prob_emisi       = Matriz_Emis[nodo][Obs_re]
         Prob_transicion  = Matriz_Trans[Es][nodo]

         virtebi_nodo     = Prob_camino * Prob_transicion * Prob_emisi
         pro_nod.append(virtebi_nodo)
      Max_es = np.argmax(pro_nod)
      Max_pr = Decimal(np.max(pro_nod))

      M_P[nodo][ob] = (Max_pr)        # Guarda la mejor posibilidad para dicho nodo.
      M_T[nodo][ob] = (Max_es)           # Guarda el mejor camino de llegada con respecto
                                            # al que produjo la mejor probabilidad en el nodo
```

Figura 5.11 *Recorrido en Forward del algoritmo de Viterbi.*

```
  # ------------------------------------------------------------------

  #            Tercer proceso seleccionar el ultimo mejor valor
  #         Y hacet recorrido en reversa para seleccionar camino

  # ------------------------------------------------------------------
#Devuelve el indice del valor más alto que corresponde al último estado
#Recupera de la matriz de memoria de las trayectorias.
  Mejor_camino=[]
  Contador = 1
  for col in range(len(Observables)-1,-1,-1):

     if(Contador ==1):
        prob_nodo=[]
        for nodo in  range(len(Estados)):
            prob_nodo.append(M_P[nodo][col])

        Max_pr = int(np.argmax(prob_nodo))
        Mejor_camino.append(Max_pr)
        Contador = Contador + 1
     elif(Contador > 1):
        Max_pr = int(M_T[Max_pr][col+1] )
        Mejor_camino.append(Max_pr)

  print("La mejor clasifiacación desde atras es  : ", Mejor_camino)

  #--------------  Asigna los valores de las Etiquetas ---------------
  Mejor_camino_tag=[]
  for col2 in range(len(Mejor_camino)-1,-1,-1):
      Tag_tray = diccionario_num_tag[Mejor_camino[col2]]
      Mejor_camino_tag.append(Tag_tray)

  print("La mejor clasifiacación es               : ", Mejor_camino_tag)
  # ------------------------------------------------------------------
  M_P= np.reshape(M_P,(Len_tag,len(Observables)))
  print("\n Matriz resultante Probabilidad")
  print("\n",M_P,"\n")
  print("Matriz resultante de trayectora")
  print("\n",M_T,"\n")
  return M_P
```

Figura 5.12 *Recorrido en reversa del algoritmo de Viterbi, para determinar mejor camino.*

5.1.2.4 Implementación SpaCy para procesamiento de textos.

El modelo anterior debe ser implementado para los textos sin formato ya que nos dan una buena proximidad en las clasificaciones de partes del discurso al ir observando cada uno de los elementos de la secuencia del texto. Se ha de recordar que al generar una traducción de audio a texto el resultado es un documento donde todo el contenido queda como una línea continua de información la cual debería ser separada manualmente.

Para los casos de ingresar documentos escritos al ya contar con una estructura de ideas separadas en enunciados. Se puede usar la librería de SpaCy que se apoya de estas estructuras para el proceso de detección de identidades y partes del discurso.

SpaCy es una librería de software para procesamiento de lenguajes naturales desarrollado por Matt Honnibal y programado en lenguaje Python. Fue lanzado en febrero de 2015 estando su desarrollo activo y siendo utilizado en distintos entornos. [37] [38]

En el Anexo 2 de la parte final de este trabajo se puede consultar a el significado de las abreviaciones usadas para los procesos de detección de partes del discurso y detección de identidades.

Lista de partes del discurso.

['ADJ','ADP','ADV','AUX','CONJ','CCONJ','DET','INTJ','NOUN','NUM',
'PART','PRON','PROPN','PUNCT','SCONJ','SYM','VERB','X','SPACE']

Lista de clases de identidades nombradas.

['PERSON','GPE','ORG','LOC','LANGUAGE','CARDINAL','DATE','EVET',
'FAC','LAW','MONEY','NORP','ORDINAL','PERCENT','PRODUCT',
'QUANTITY','TIME','WORK_OF_ART']

En el Anexo 2 de la parte final de este trabajo se puede consultar a el significado de las abreviaciones usadas para los procesos de detección de partes del discurso y detección de identidades.

SpaCy es una librería con diversas funciones para el procesamiento de lenguaje natural, sin embargo, para el caso de estudio lo que se requiere es la detección de identidades nombradas y las partes del discurso.

Algoritmo 5.3 *Pseudocódigo de implementación de SpaCy*

Entrada: Selección de texto

1: Carga de Librería SpaCy
2: Carga de paquetería según el modelo e idioma a implementar.
3: Establecer variable de objetivo para el modelo
4: Carga de Texto dentro de variable de modelo.
5: Tokenización.
6: Detección de partes del discurso (POS)
7: Detección de entidades nombradas (NER)
8: Guardar datos en arreglo resultante.

En la **Figura *5.13*** se ven la implementación en el lenguaje Python de la primera etapa del **Algoritmo 5.2** y en la **Figura *5.26*** la selección entre los múltiples módulos para el idioma español. Cabe destacar que mientras se seleccione un módulo de mayor tamaño los niveles de precisión en las tareas mejoran, debido al tipo y número de corpus implementado en el entrenamiento de dichos módulos.

Figura 5.13 *Paso 1 del Pseudocódigo de implementación de SpaCy*

Figura 5.14 *Paso 2 del Pseudocódigo de implementación de SpaCy, selección de módulos en español*

Ejemplo 1:

En este ejemplo de identificación de POS y NER se usa el texto resultante de la leyenda Mexicana del “El conejo en la luna”, la cual se usó como documento de control para la tarea de generación de transcripciones.

Al usar la detección de identidades a simple vista en la **Figura *5.15*** se puede observar que algunos elementos no fueron bien clasificados, esto debido a que dichas transcripciones se generaron en un texto sin estructura gramatical lo cual le dificulta al módulo el proceso.

```
for named_entity in document.ents:
    print(named_entity, named_entity.label_)
```

```
la luna LOC
No yo no puedo MISC
Quetzalcóatl PER
Azteca LOC
Endo PER
Astro MISC
Claridad LOC
```

Figura 5.15 *Extracción de identidades en texto generado por transcripciones de audio.*

Al intentar la extracción de datos de un segundo archivo generado de transcripciones como se muestra en la **Figura *5.16*** los resultados aun no son muy precisos, esta librería puede reentrenarse para mejorar los niveles de precisión. Y aunque esta estrategia es muy útil cuando se trabaja con archivos que contienen una estructura o una bolsa de palabras similar, para este caso de estudio no se aplica al manejar corps con múltiple categorías y temas.

```
document.ents
```

```
(Conejo en la Luna,
 Quetzalcóatl,
 ¿Qué comes?,
 Quetzalcóatl,
 No, yo no puedo,
 Soy pequeño,
 Quetzalcóatl,
 La acción de valentía,
 decirle.:- Podrás,
 Quetzalcóatl)
```

Figura 5.16 *Extracción de identidades en texto generado por transcripciones de audio*

Se puede observar mejor en la **Figura *5.17*** los resultados cuando se imprime cada uno de los elementos de manera vertical. Al encontrar “Entity: ” vacío indica que esa palabra no es una identidad y en algunos casos es algo lógico, en el primer artículo, pero posteriormente se puede observar que en la línea 6 si lo considera una “Entity : Misc” y este caso se da por que considera al artículo dentro de un todo de la identidad “conejo en la luna”.

```
[ ]  for tokens in document[:]:
       _Tok = str(tokens)
       _Ner = str(tokens.ent_type_)
       print("Token:  {0}, Entity : {1}".format(_Tok,_Ner))
```

```
Token:  La, Entity :
Token:  leyenda, Entity :
Token:  del, Entity :
Token:  Conejo, Entity : MISC
Token:  en, Entity : MISC
Token:  la, Entity : MISC
Token:  Luna, Entity : MISC
Token:  ., Entity :
Token:  Durante, Entity :
Token:  la, Entity :
Token:  búsqueda, Entity :
Token:  de, Entity :
Token:  su, Entity :
Token:  alimento, Entity :
Token:  ,, Entity :
Token:  Quetzalcóatl, Entity : PER
Token:  se, Entity :
Token:  encontró, Entity :
Token:  con, Entity :
```

Figura 5.17 *Impresión vertical de identidades nombradas.*

Ejemplo 2:

Para el siguiente ejemplo **Figura *5.18*** se selección al azar de una lista de 33 enunciados analizados, el cual se muestra la detección correcta de elementos dentro de un enunciado.

```
[ ]  print(enun_1)
```

```
Juan Pablo esta en Burgerking comiendo un hamburguesa y tomando un refresco con su hermano Roberto
```

Figura 5.18 *Enunciado para extracción de POS y NER por SpaCy.*

En la tarea de extracción de identidades mostrada en la **Figura *5.19***, se ve que los tres elementos en la historia son detectados con precisión.

```
print(enun_1.ents)
```

```
(Juan Pablo, Burgerking, Roberto)
```

Figura 5.19 *Detección de NER por SpaCy.*

Para el análisis de cada uno de los elementos se genera una función que detecte por separado la POS y si existe o forma parte de un NER la palabra como se muestra en la **Figura *5.20***, la última

parte de cada uno de los arreglos “Xxxx” nos muestra la estructura de la palabra, esta estructura al incluir una X mayúscula indica que es una palabra que inicia con mayúscula y según las reglas ortográficas del idioma en cuestión ayuda en los proceso de clasificación y extracción de información.

```
for token in enun_1[:]:
    _data = [token.text, token.ent_type_ ,  token.pos_, token.tag_,
             token.shape_]
    print(_data)
```

```
['Juan', 'PER', 'PROPN', 'PROPN', 'nsubj', 'Xxxx']
['Pablo', 'PER', 'PROPN', 'PROPN', 'flat', 'Xxxxx']
['esta', '', 'DET', 'DET', 'flat', 'xxxx']
['en', '', 'ADP', 'ADP', 'case', 'xx']
['Burgerking', 'LOC', 'PROPN', 'PROPN', 'nmod', 'Xxxxx']
['comiendo', '', 'VERB', 'VERB', 'ROOT', 'xxxx']
['un', '', 'DET', 'DET', 'det', 'xx']
['hamburguesa', '', 'NOUN', 'NOUN', 'obj', 'xxxx']
['y', '', 'CCONJ', 'CCONJ', 'cc', 'x']
['tomando', '', 'VERB', 'VERB', 'conj', 'xxxx']
['un', '', 'DET', 'DET', 'det', 'xx']
['refresco', '', 'NOUN', 'NOUN', 'obj', 'xxxx']
['con', '', 'ADP', 'ADP', 'case', 'xxx']
['su', '', 'DET', 'DET', 'det', 'xx']
['hermano', '', 'NOUN', 'NOUN', 'obl', 'xxxx']
['Roberto', 'PER', 'PROPN', 'PROPN', 'appos', 'Xxxxx']
```

Figura 5.20 *Extracción de POS y NER por SpaCy.*

La tarea de reconocimiento de identidades no siempre resulta tan sencilla como se mostró en el ejemplo anterior. Depende del modo en que se estructuran las ideas, la complejidad de los textos y el uso de connotaciones especiales, pero para un caso de demostrativo resulta ideal.

Una vez identificado los elementos de POS y NER se prosigue a validar la existencia de material multimedia para la sincronización, calcular los tiempos de aparición de cada uno de los elementos y a generar un arreglo matricial para mostrar la historia por medio de la librería Pygame.

Las librerías a usar fueron:

- Pygame : Sincronización de elementos multimedia.
- Panda : Creación de archivo para guardar tiempos de pronunciación.
- Time : Procesos de espera.
- Os : Procesos de búsqueda de archivos.
- Gtts : Generación de tarea de Text to Speech.
- Librosa : Calculo de duración de archivo de audio.

5.1.2.5 Búsqueda de archivos multimedia.

La tarea de búsqueda de archivos multimedia nos permite identificar la existencia de los archivos según una palabra clave o por un formato especifico, la cual se definió en 8 funciones, que se pueden agrupar en 4 tareas: 2 para detección de archivo de historias, 2 para detección de archivos para efectos sonoros. 2 para identificación de formatos de imagen y 2 para detección de archivo de audio de palabras individuales para el proceso del cálculo de sincronización.

La tarea de búsqueda se puede establecer con ayuda del algoritmo siguiente: y mostrado su implementación en Python en la **Figura *5.21***.

Algoritmo 5.4 *Pseudocódigo de búsqueda en repositorios.*

Entrada: Selección de tipo de búsqueda

1: Establecer directorio de búsqueda.

2: Extracción de directorios y archivos dentro de la dirección.

3: Clasificación por directorio y clase archivos.

4: Guardar archivos con extensión permitida.

5: Devolver arreglo con archivos que cumplen parámetros.

Excepción:

1: En caso de no encontrar archivos o directorio indicar carpeta vacía

---

```
# --------- Audio Records  ----------
def search_audio_memories():
    contenido = os.listdir('Assets/memories/')
    memories = []
    for fichero in contenido:
        if fichero.endswith('.mp3') or fichero.endswith('.wav') or fichero.endswith('.ma4'):
            memories.append(fichero)
    return (memories)
```

Figura 5.21 *Tarea de búsqueda de archivos dentro de repositorio de archivo de audios de las historias.*

La tarea de búsqueda especializada por palabra se puede establecer con ayuda del algoritmo siguiente, el cual tiene su implementación en Python en la **Figura *5.22***.

Algoritmo 5.5 *Pseudocódigo de búsqueda en repositorios.*

Entrada: Selección de palabra de búsqueda

1: Establecer directorio de búsqueda.
2: Extracción de directorios y archivos dentro de la dirección.
3: Clasificación por directorio y clase archivos.
4: Guardar archivos con extensión permitida.
5: Clasificación por palabra buscada.
6: Guardar archivos con palabra buscada.
7: Devolver arreglo con archivos que cumplen parámetros.

Excepción:

1: En caso de no encontrar archivos o directorio hay que indicar que no se cuenta con archivo con ese criterio.

```
def search_audio_memories_palabra(_palabra):

    try:
        contenido = os.listdir('Assets/memories/')
        memories = []
        for fichero in contenido:
            print(fichero)
            if fichero.endswith('.mp3') or fichero.endswith('.wav') or fichero.endswith('.ma4'):
                print("Formato Correcto")
                fichero_low = fichero.lower()
                if fichero_low.find(_palabra.lower()) != -1:
                    memories.append(fichero)
        return memories
    except:
        print("Not File related")
```

Figura 5.22 *Tarea de búsqueda por palabra clave de archivos dentro de repositorio de archivo de audios de las historias.*

En la **Figura *5.23*** se ve un caso práctico en el cual se buscó la palabra “Sujeto” y se devuelve el único archivo dentro de la ubicación seleccionada que cumple dicho criterio.

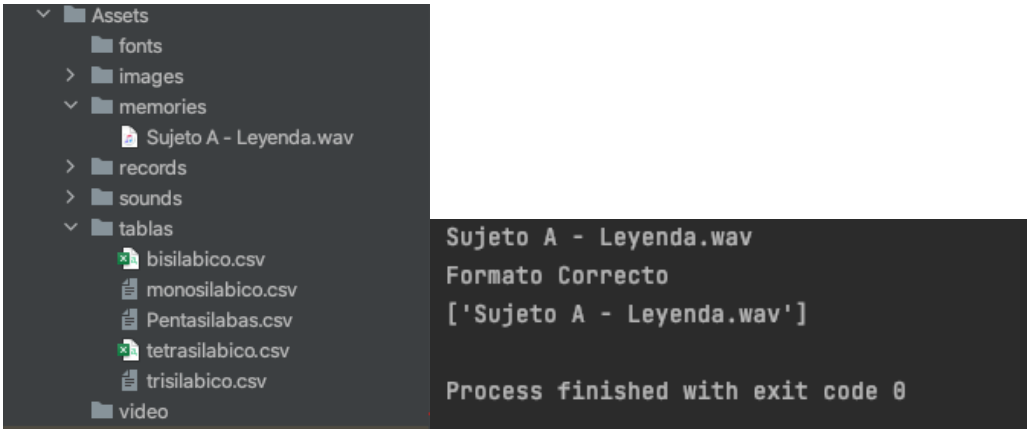


Figura 5.23 *Tarea de búsqueda por palabra clave.*

5.1.2.6 Generación de audio a partir de archivos de texto.

La tarea de generación de archivos de audio partir de textos o de Strings se realiza por medio de la librería gtts (Google Text-to-Speech) la cual cuenta con una API de Google, nos permite convertir a archivos de audio en formato MP3.

Cuenta con una única función reducir la velocidad del audio para aumentando la duración del archivo, se hizo un análisis de un conjunto de palabras: monosilábicas, bisilábicas, trisilábicas, tetrasílabas y pentasílabas. Los resultados se pueden encontrar para su consulta en el **Anexo 1** en la parte final de este trabajo.

En el **Algoritmo 5.6** se muestra el algoritmo para la conversión de textos a audios y se observa su implementación en la **Figura *5.24***.

Algoritmo 5.6 *Pseudocódigo de generación de audio individual.*

Entrada: Selección de palabra

1: Llamado a librería gtts
2: Establecer directorio de guardado.
3: Establecer nombre de nuevo archivo.
4: Generar audio.
5: Guardar archivo resultante.

```python
def text_to_speech_narrative(File_text, NewFileName_, _slow= False):

    Name_dir = 'Assets/records/'+ NewFileName_

    if _slow == False:
        tts = gTTS(text=File_text
                   , lang='es', slow=_slow)
        tts.save(Name_dir+"_n.mp3")
        time.sleep(1)
        return str((NewFileName_ + "_n.mp3"))
    else:
        tts = gTTS(text=File_text
                   , lang='es', slow=_slow)
        tts.save(Name_dir + "_s.mp3")
        time.sleep(1)
        return str((NewFileName_ + "_s.mp3"))
```

Figura 5.24 *Tarea de búsqueda por palabra clave.*

En el **Algoritmo 5.7** se muestra el algoritmo para la obtención de los tiempos de audios de archivo, que se puede implementar para la obtención de la duración de una palabra como de un texto extenso convertido a audio. En la **Figura *5.24*** se ve la implementación en el lenguaje Python, los resultados de la implementación de estas líneas de código se pueden observar en el conjunto de tablas del **Anexo 1.**

Algoritmo 5.7 *Pseudocódigo de cálculo de tiempo sobre archivo de audio*

Entrada: Selección de archivo de audio

1: Llamado a librería librosa

2: Extracción de tiempo de duración.

3: Regresar tiempo de duración de archivo de audio.

```python
def time_audio(audioFile):
    Name_dir = 'Assets/records/' + audioFile
    return librosa.get_duration(filename=Name_dir)
```

Figura 5.25 *Tarea de búsqueda por palabra clave.*

5.1.2.7 Generación de arreglos de datos para sincronización de narrativa automatizada.

El algoritmo propuesto para la generación de arreglos que nos permita mostrar las narrativas automatizadas se puede encontrar en el **Algoritmo 5.8** donde cada elemento es contemplado y se asocia de con el material multimedia disponible en los repositorios.

Se pretende que para futuros trabajos el material no encontrado en los repositorios digitales se pueda descargar de manera automática de la web y pasar por un proceso de análisis de imágenes para validar que la imagen coincida con la etiqueta de la entidad o palabra buscada.

Algoritmo 5.8 *Pseudocódigo de generación de arreglos de datos para sincronización de elementos multimedia*

Entrada: Ingreso de arreglo SpaCy.

1: Agrupamiento por identidades
2: Extracción de bolsa de palabras
3: Arreglo con palabras únicas.
4: Ciclo For para visitar cada elemento dentro de la matriz o arreglo.
    a. Extracción de datos del arreglo original POS y NER
    b. Búsqueda de archivo de audio de palabra.
        I. Si se tiene archivo de audio de la palabra extraer tiempo de archivo y agregar al arreglo.
        II. Si no se cuenta con archivo
            i. Generar archivo único de palabra.
            ii. Extraer tiempo de duración de la palabra.
            iii. Agregar tiempo al arreglo.
    c. Si es una identidad extracción datos de archivos multimedia.
        I. Búsqueda de archivo de audio.
            i. Si se cuenta con audio agregar ubicación al arreglo.
            ii. Si no se cuenta con audio agregar string vacío al arreglo (“ ”).
        II. Búsqueda de archivo de imagen.
            i. Si se cuenta con imagen agregar ubicación al arreglo.
            ii. Si no se cuenta con imagen agregar string vacío al arreglo (“ ”).
        III. Búsqueda de archivo de video.
            i. Si se cuenta con video agregar ubicación al arreglo.
            ii. Si no se cuenta con video agregar string vacío al arreglo (“ ”).
5: Establecer directorio de guardado.
6: Guardar archivo resultante.

5.1.2.8 Selección y edición de imágenes relacionada a partes del discurso e identidades nombradas.

Dentro de la asignación de imágenes los elementos dentro de los documentos que están contemplados para el proceso de sincronización son los sustantivos y en algunos casos dentro de la historia los pronombres personales en los casos en que se hable de primera persona.

El proceso de asignar un texto a una imagen para desplegarla en la sincronización se observa en el **Algoritmo 5.2,** donde se ve a detalle los ajustes necesarios para poder tener una relación entre el tamaño de la imagen y el texto desplegado, implementando la librería de Python "PIL" y las funciones Image, ImageDraw y ImageFont.

Algoritmo 5.9 *Pseudocódigo de colocación de texto sobre imagen.*

Entrada: Selección de palabra

1: Búsqueda de imagen disponible en repositorio.
2: Generación de listado de imagen o imágenes disponible sobre palabra de búsqueda.
3: Establecer imagen a usar.
4: Obtener atributos de ancho y altura de imagen.
5: Obtener relación del centro de la imagen.
6: Generar relación del tamaño de la letra con respecto a los atributos de altura y anchura.
7: Asignar capa de texto sobre la imagen.
8: Guardar archivo resultante.
9: Mostrar imagen nueva.

Dentro de la implementación experimental se implementó la fuente "Roboto-Italic.ttf" descargada de https://fonts.google.com/knowledge, la cual puede ser intercambiada al seleccionar otro archivo con extensión ".ttf", como se observa en **Figura 5.26.** Comparación de tiempos entre formato origen y resultante , donde al intercambiar los archivos en esas líneas de código se tiene acceso a nuevas fuentes.

```
#Tipo de letras
# fetch the name of font that was installed
pathFont = 'drive/My Drive/Memorias/Font/Roboto-Italic.ttf'
```

Figura 5.26. *Comparación de tiempos entre formato origen y resultante*

La extracción de los atributos de la imagen es importante ya que se puede obtener una relación del tamaño de fuente necesario para que el texto se observe y poder generar un efecto de sombra que facilite la visibilidad de los textos, mostrados en el código de la **Figura *5.27*.** y cuyos resultados se pueden ver en la **Figura *5.28***.

```
aj_h_ima= len(mensaje)*(font_size/2)

pos_w_ima_text = (w_image-aj_h_ima)/2
pos_h_ima_text = (h_image-font_size)/2

pos_w_ima_text_somb = pos_w_ima_text+5
pos_h_ima_text_somb = pos_h_ima_text-5

posicion  = (pos_w_ima_text,pos_h_ima_text)
posicion_somb  = (pos_w_ima_text_somb,pos_h_ima_text_somb)
```

Figura 5.27. *Obtención de atributos de altura y anchura de una imagen.*

```
print(w_image)
print(h_image)

1800
1013

print(font_size)

128
```

Figura 5.28. *Relación de altura y anchura con tamaño de letra.*

El proceso de asignación de los colores se diseña con el uso de un diccionario con los colores en nombres en inglés y relacionados a los valores RGB de cada color como se observa en la **Figura *5.30***.

Diccionario

```
[ ] colores_texto={
'black':(0,0,0),'white':(255,255,255),'red':(255,0,0),'lime':(0,255,0),'blue':(0,0,255),'yellow':(255,255,0),'cyan_':(0,255,255),
'magenta_':(255,0,255),'silver':(192,192,192),'gray':(128,128,128),'maroon':(128,0,0),'olive':(128,128,0),'green':(0,128,0),
'purple':(128,0,128),'teal':(0,128,128),'navy':(0,0,128),'maroon':(128,0,0),'dark_red':(139,0,0),'brown':(165,42,42),'firebrick':(178,34,34),
'crimson':(220,20,60),'red':(255,0,0),'tomato':(255,99,71),'coral':(255,127,80),'indian_red':(205,92,92),'light_coral':(240,128,128),
'dark_salmon':(233,150,122),'salmon':(250,128,114),'light_salmon':(255,160,122),'orange_red':(255,69,0),'dark_orange':(255,140,0),
'orange':(255,165,0),'gold':(255,215,0),'dark_golden_rod':(184,134,11),'golden_rod':(218,165,32),'pale_golden_rod':(238,232,170),
'dark_khaki':(189,183,107),'khaki':(240,230,140),'olive':(128,128,0),'yellow':(255,255,0),'yellow_green':(154,205,50),
'dark_olive_green':(85,107,47),'olive_drab':(107,142,35),'lawn_green':(124,252,0),'chartreuse':(127,255,0),'green_yellow':(173,255,47),
'dark_green':(0,100,0),'green':(0,128,0),'forest_green':(34,139,34),'lime':(0,255,0),'lime_green':(50,205,50),'light_green':(144,238,144),
'pale_green':(152,251,152),'dark_sea_green':(143,188,143),'medium_spring_green':(0,250,154),'spring_green':(0,255,127),'sea_green':(46,139,87),
'medium_aqua_marine':(102,205,170),'medium_sea_green':(60,179,113),'light_sea_green':(32,178,170),'dark_slate_gray':(47,79,79),'teal':(0,128,128),
'dark_cyan':(0,139,139),'aqua':(0,255,255),'cyan':(0,255,255),'light_cyan':(224,255,255),'dark_turquoise':(0,206,209),'turquoise':(64,224,208),
'medium_turquoise':(72,209,204),'pale_turquoise':(175,238,238),'aqua_marine':(127,255,212),'powder_blue':(176,224,230),'cadet_blue':(95,158,160),
'steel_blue':(70,130,180),'corn_flower_blue':(100,149,237),'deep_sky_blue':(0,191,255),'dodger_blue':(30,144,255),'light_blue':(173,216,230),
'sky_blue':(135,206,235),'light_sky_blue':(135,206,250),'midnight_blue':(25,25,112),'navy':(0,0,128),'dark_blue':(0,0,139),'medium_blue':(0,0,205),
'blue':(0,0,255),'royal_blue':(65,105,225),'blue_violet':(138,43,226),'indigo':(75,0,130),'dark_slate_blue':(72,61,139),'slate_blue':(106,90,205),
'medium_slate_blue':(123,104,238),'medium_purple':(147,112,219),'dark_magenta':(139,0,139),'dark_violet':(148,0,211),'dark_orchid':(153,50,204),
'medium_orchid':(186,85,211),'purple':(128,0,128),'thistle':(216,191,216),'plum':(221,160,221),'violet':(238,130,238),'magenta':(255,0,255),
'orchid':(218,112,214),'medium_violet_red':(199,21,133),'pale_violet_red':(219,112,147),'deep_pink':(255,20,147),'hot_pink':(255,105,180),'light_pink':(255,182,193),
'pink':(255,192,203),'antique_white':(250,235,215),'beige':(245,245,220),'bisque':(255,228,196),'blanched_almond':(255,235,205),'wheat':(245,222,179),
'corn_silk':(255,248,220),'lemon_chiffon':(255,250,205),'light_golden_rod_yellow':(250,250,210),'light_yellow':(255,255,224),'saddle_brown':(139,69,19),
'sienna':(160,82,45),'chocolate':(210,105,30),'peru':(205,133,63),'sandy_brown':(244,164,96),'burly_wood':(222,184,135),'tan':(210,180,140),
'rosy_brown':(188,143,143),'moccasin':(255,228,181),'navajo_white':(255,222,173),'peach_puff':(255,218,185),'misty_rose':(255,228,225),
'lavender_blush':(255,240,245),'linen':(250,240,230),'old_lace':(253,245,230),'papaya_whip':(255,239,213),'sea_shell':(255,245,238),
'mint_cream':(245,255,250),'slate_gray':(112,128,144),'light_slate_gray':(119,136,153),'light_steel_blue':(176,196,222),'lavender':(230,230,250),
'floral_white':(255,250,240),'alice_blue':(240,248,255),'ghost_white':(248,248,255),'honeydew':(240,255,240),'ivory':(255,255,240),'azure':(240,255,255),
'snow':(255,250,250),'black':(0,0,0),'dim_gray_':(105,105,105),'gray_':(128,128,128),'dark_gray_':(169,169,169),'silver':(192,192,192),'light_gray_':(211,211,211),
'gainsboro':(220,220,220),'white_smoke':(245,245,245),'white':(255,255,255),

}
```

Figura 5.29. *Relación de nombre de color y valor RGB dentro de diccionario en Python.*

El resultado final del proceso se observa en la **Figura *5.30*** donde el texto de la imagen se centra con respecto a la anchura y altura de la imagen.

```
draw = ImageDraw.Draw(image)
```

```
draw.text(xy= posicion_somb, text=mensaje_somb, fill=colorText_somb, font=font_type)
draw.text(xy= posicion, text=mensaje, fill=colorText, font=font_type)
```

Figura 5.30. *Relación de altura y anchura con tamaño de letra.*

Librería PyGame

Pygame es una librería del lenguaje de programación Python implementada para el desarrollo de videojuegos. [49] Pygame está basada en SDL, que es una librería que nos provee acceso de bajo nivel al audio, teclado, ratón y al hardware gráfico de nuestro ordenador.

Pygame nos permite manejar de manera paralela los efectos sonoros y visuales dentro de su entorno al implementar la estrategia de inmersión que manejan los videojuegos. Nos da control sobre las dimensiones de nuestro espacio y permite sincronizar eventos en el proceso de la narrativa.

En el siguiente **Algoritmo** se muestra el proceso de sincronización de los eventos con el arreglo de datos previamente preparado, donde este arreglo incluye los tiempos que dura los eventos y los archivos que serán llamado para dichos eventos.

Algoritmo 5.10 *Pseudocódigo de colocación de texto sobre imagen.*

Entrada: Selección de historia

1: Ingreso de arreglo de datos
2: Ingreso de archivo audio de historia.
3: Selección de fondo de la historia o dejar valor predeterminado.
4: Inicio del audio de la narrativa.
5: Inicialización del espacio visual de la aplicación.
6: Ciclo For que visita cada elemento en paralelo al sonido de historia.
    a. Si es requerido llamar imagen de acuerdo con el tiempo establecido en el arreglo.
    b. Si es requerido llamar archivo de audio de acuerdo con el tiempo establecido en el arreglo.
    c. Si es requerido llamar archivo de video de acuerdo con el tiempo establecido en el arreglo.
    d. Actualización de estados y del entorno.
7: Finalizar narrativa.

El resultado visual es similar al mostrado en la **Figura *5.30***, pero permitiendo más flexibilidad y control sobre los eventos que se realizan en paralelo como se muestra en el flujo de la **Figura *5.31***, donde se puede o no tener relación entre cada uno de los materiales audiovisuales, haciendo referencia que la imagen mostrada puede durar solo el tiempo que dura la palabra o hasta que otro elemento sea llamado.

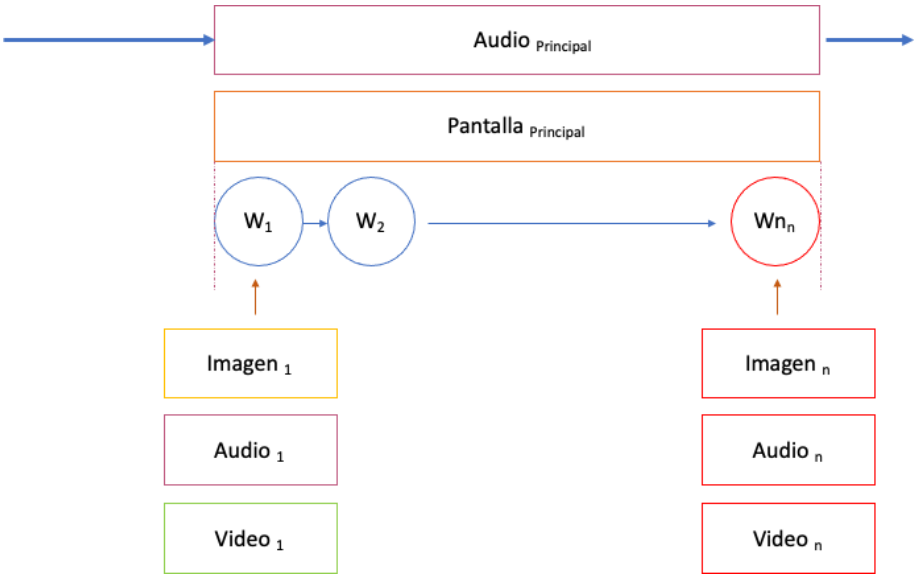


Figura 5.31. *Flujo de animación.*

En la **Figura *5.32*** se muestra una breve parte del código de la función encargada de estar sincronizando y manteniendo cada uno de los estados dentro del proceso de animación de la narrativa automatizada.

```
# Game Loop
running = True
while running:
    screen.fill((255, 0, 78))
    screen.blit(background, (0, 0))
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Enemy Movements
    for index in range(0,len(arrayData)):
        screen.blit(background, (0, 0))
        _palabra = str(arrayData[index])
        print(_palabra)
        _font, positionX, _positionY = mensajeScreen(_height, _width, _palabra)
        font_word = pygame.font.Font("freesansbold.ttf", _font)
        word_update = font_word.render(str(_palabra), True, (255, 255, 255))
        _sound = mixer.Sound('Assets/sounds/'+ arrayData[index][s])
        screen.blit(word_update, (positionX, _positionY))
        time.sleep(2)
        pygame.display.update()
```

Figura 5.32. *Trozo de código implementando librería Pygame para pruebas de animación.*

Capítulo 6

Análisis de resultados

6.1 Análisis de Resultados / Análisis computacional

El enfoque dado originalmente era la generación de una aplicación hibrida implementando el SDK de Flutter. Esta aplicación iba a fungir como el receptor y capturista de las historias al igual que el proceso de narrativa automatizada. Por limitantes de las funciones de Flutter se implementó la librería Pygame para la proyección de las narrativas.

La aplicación queda habilitada para la parte de captura de historias en formato de narrativas orales o escritas, el ingreso de textos e historias en formato de PDF y DOC, al igual que queda habilitado dentro de la aplicación el reconocimiento y extracción de caracteres para poder extraer textos de imágenes o archivos impresos con letras impresas y legibles.

Los primeros desarrollos de la aplicación se enfocaban en realizar el proceso de transcripciones en tiempo real de los recuerdos, pero fue donde la configuración del uso de librerías de terceros generó limitantes. La primera es que la aplicación al detectar una pausa en la historia se detenía por completo y la transcripción quedaba cortada esto era un impedimento al momento de intentar obtener traducciones continuas y no se contaba con un archivo o grabación para comparar la precisión de las transcripciones, por lo cual se optó por la captura de un audio en formato MP4 p M4A con la aplicación y su conversión al formato WAV para aplicar la librería de Google de conversión de discurso hablado a escrito por medio del lenguaje Python.

La preservación del formato de audio nos permitió medir la precisión en el proceso de conversión del discurso oral entre sujetos de evaluación y revisar la correlación entre el sentido de las ideas que se transcribieron y el texto original.

Uno del problema encontrado en la generación de las transcripciones fue la perdida de estructuras de ideas y la falta de manejo de signos de interrogación. La transcripción se genera como si fuera una única frase donde se contienen todas las palabras haciendo difícil la detección de identidades al implementarse la librería de SpaCy por lo cual se optó por usar un modelo de cadena ocultas de Markov junto al algoritmo de Viterbi.

Uno de los inconvenientes al implementar este modelo fue que cuando se usaba en palabras desconocidas generaba un error y la probabilidad se iba a cero. Se resolvió dicho inconveniente

con la extracción de la bolsa de palabras y todas aquellas palabras fueran nuevas se ingresaban a una función escrita en Python que obtenía la clase de la página oficial de la RAE.

Dicho modelo se entrenó usado dos libros del autor colombiano Gabriel García Márquez, el primer libro es “El amor en los tiempos del Cólera” y “Cien años de soledad” los cuales se implementaron en el set de entrenamiento de las matrices de transición y emisión para el algoritmo de cadenas ocultas de Márkov y el algoritmo de Viterbi.

	Amor en tiempo de cólera	**Cien años de Soledad**
Enunciados	4 879	5415
Palabras	144 465	137 984
Palabras sin repetir	14 866	15830
Palabras compartidas	6326	6326
Palabras diferentes	9504	9504

Tabla 6.1. *Tabla de comparación entre texto de entrenamiento y pruebas.*

Comprobación de clasificación del algoritmo de Viterbi.

El enunciado original:

La tercera cubeta , la del líquido fijador , era la que estaba junto al cadáver .

La mejor clasificación es

['ART', 'ADJ', 'SUS', 'ART', 'ART', 'SUS', 'ADJ', 'VER', 'ART', 'PRO', 'VER', 'ADJ', 'ART', 'SUS']

La conversión de diferentes formatos de textos de las transcripciones y los textos ingresados por medio de la aplicación se lograron con la librería de Python Tika, que detecta el formato en el que se encuentra y lo convierte a texto plano. Uno de los inconvenientes que se detectó con esto fue que requiere el uso de internet y genera una conexión con el servidor que aloja Tika, por lo cual podría ser un problema de seguridad para el manejo de datos personales en futuros trabajos.

El manejo de la librería SpaCy en su modelo mediano (es_core_news_md) da un buen proceso en la parte de reconocimiento de identidades y partes del discurso. Como se aprecia en la **Figura *6.1*** los arreglos que se generan se realizan a partir del proceso de Tokenización por lo cual en el proceso de agrupamiento se usa la columna 2 que contiene una identidad el texto.

```
for token in enun_1[:]:
    _data = [token.text, token.ent_type_ ,  token.pos_, token.tag_,
             token.shape_]
    print(_data)
```

```
['Juan', 'PER', 'PROPN', 'PROPN', 'nsubj', 'Xxxx']
['Pablo', 'PER', 'PROPN', 'PROPN', 'flat', 'Xxxxx']
['esta', '', 'DET', 'DET', 'flat', 'xxxx']
['en', '', 'ADP', 'ADP', 'case', 'xx']
['Burgerking', 'LOC', 'PROPN', 'PROPN', 'nmod', 'Xxxxx']
['comiendo', '', 'VERB', 'VERB', 'ROOT', 'xxxx']
['un', '', 'DET', 'DET', 'det', 'xx']
['hamburguesa', '', 'NOUN', 'NOUN', 'obj', 'xxxx']
['y', '', 'CCONJ', 'CCONJ', 'cc', 'x']
['tomando', '', 'VERB', 'VERB', 'conj', 'xxxx']
['un', '', 'DET', 'DET', 'det', 'xx']
['refresco', '', 'NOUN', 'NOUN', 'obj', 'xxxx']
['con', '', 'ADP', 'ADP', 'case', 'xxx']
['su', '', 'DET', 'DET', 'det', 'xx']
['hermano', '', 'NOUN', 'NOUN', 'obl', 'xxxx']
['Roberto', 'PER', 'PROPN', 'PROPN', 'appos', 'Xxxxx']
```

Figura 6.1. *Arreglo generado con librería de SpaCy para POS y NER.*

El proceso de generar un arreglo que contenga los archivos multimedia tiene diversos pasos entre los cuales destacaron: agrupamiento de entidades, búsqueda de archivos multimedia, asignación de direcciones de archivos al arreglo, cálculo de aparición de la palabra quedando restringidos al tiempo en que se tarda en decir la pronunciación de la palabra pero no limitado a esto ya que existen casos concretos en el cual la entidad se agrupa de un conjunto de palabras, por lo cual debe tomarse en cuenta el cumulo de la suma de los tiempos de ellas o pudiendo dejarse hasta la aparición de un nuevo elemento del mismo tipo(imagen, video y audio).

En la parte de sincronización al utilizar un estándar de 150 palabras por minutos y una relación de 7.5 ms por silaba, con un espacio entre las palabras de 8 ms de espera, se logró mostrar el material en forma línea, pero en algunos momentos existía un retraso en lo que sé que se apreciaba con la grabación que contenía la conversión del texto plano a discurso oral. Por ello se desarrolló un a función que revisa si se tienen previamente la palabra dentro de un repositorio de audio y de no ser el caso separa la palabra y genera dos archivos de audios con la librería gtts para futuras consultas, apoyándose de la librería labrosa se obtiene el tiempo que dura cada audio, el cual es la relación de la pronunciación de una palabra y su aparición dentro del entorno de Pygame.

El manejo de datos personales, gracias a Firebase quedan protegidos por los protocolos de seguridad que maneja, el montar una aplicación en estos servicios de Google nos permite no tener

que aplicar por el momento un proceso de encriptado a los documentos que se están generando. Pero aún queda pendiente el manejo sensible de datos personales, dentro de los documentos.

El manejo de la librería Pygame nos da un buen control en paralelo de las diferentes partes y da la sensación de inmersión que manejan los videojuegos entre los diferentes contenidos multimeda, como se manejo en el capítulo 5 de la parte experimental. Aun queda limitado la aparición de videos dentro de la historia y el entorno.

Capítulo 7

Conclusiones

7.1 Conclusiones

Las transcripciones del discurso hablado al escrito generadas por la librería de Google son susceptibles a la velocidad y modo de pronunciación del sujeto que genera el audio, como se vio en los casos de estudio, donde el género, y la velocidad del habla obtenían resultandos distintos en textos controlados.

La identificación y extracción con técnicas de procesamiento natural son susceptible a la estructura contenida dentro de las transcripciones en formato de texto de plano, como es el caso de la falta de signos de puntación en la generación de transcripciones. Otro de los casos que se ve sensible es el uso de las mayúsculas en el texto, ya que al encontrarlas sin signos de interrogación las da por entidades, aunque en una de las transcripciones estaba relacionado a un adjetivo.

El cálculo en los tiempos de aparición está basado en la relación de 150 palabras por minuto que es el promedio de una persona al leer, este estándar se mostró en los casos de estudio, y se refuerza con lo mostrado Enel sitio oficial de la secretaria de Educación de México. Por ello se maneja una relación de 7.6 ms por silaba permitió sincronizar de los tokens que con el contenido multimedia en los casos necesarios.

Es importante mencionar que al alterar la velocidad del audio se debe recalcular los tiempos del material multimedia, y una posible solución propuesta es la planteada de manejar un repositorio dentro de la aplicación que contenga la bolsa de palabras ingresadas hasta el momento y se actualice con cada nueva palabra.

Gracias a la realización de este trabajo se pudieran implementar y obtener resultados en análisis de textos, extracción de información, detección de identidades nombradas y relación con partes de la historia, para posteriormente complementarse con la sincronización de material multimedia, el cual está contenido dentro de sus propios repositorios, uno para audio, uno para imágenes y otro de videos del usuario.

7.2 Recomendaciones y trabajos Futuros.

En la parte de trabajos a futuro se planea el desarrollo de un proceso automático de descarga del contenido multimedia y la validación de los elementos implementando técnicas de Machine Learning como es el caso de poder identificar elementos de la fotografía y validar que esté relacionada a la identidad o elemento buscado. Otro enfoque que pudiese implementarse es el uso de librerías abiertas de generación de imágenes a partir del ingreso de texto.

El uso de derechos de autor es una limitación para implementar música de la cual no se cuenta con licencia de uso, con lo cual se buscará desarrollar herramientas especializadas que puedan sincronizare en tiempo real con aplicaciones o sitios web como YouTube donde el material puede encontrase y accederse. Para generar una modelo que evite cuellos de botella en la transmisión de datos, para funcionar de manera paralela a la aplicación, pero pueda importarse por su parte el material multimedia.

La librería SpaCy permite el perfilamiento emocional de los textos apoyando a generar un análisis del contenido de la historia, junto con la retroalimentación de los usuarios se puede plantear el desarrollo de funciones que permitan poder revertir una emoción negativa dentro de un texto o poder enriquecer una emoción positiva.

Y se plantea el desarrollo de entornos más inmersivos con tecnologías de realidad aumentada y narrativa automatizada, para revivir historias de manera más real. Otro enfoque propuesto es la implementación del modelo para la educación donde se pueda usar las mismas herramientas ya desarrolladas en modelos de reforzamiento de temas.

Bibliografía

[1] S. d. S. d. México, «https://www.gob.mx/,» Secretaría de Salud , 05 Octubre 21. [En línea]. Available: https://www.gob.mx/salud/es/articulos/enfermedad-de-alzheimer-demencia-mas-comun-que-afecta-a-personas-adultas-mayores?idiom=es. [Último acceso: 01 Octubre 22].

[2] F. A. ESPAÑA, «tp://www.alzfae.org/,» 13 04 2015. [En línea]. Available: http://www.alzfae.org/actualidad/mundo-alzheimer/1158/30-congreso-annual-de-alzheimrer-europe-la-demencia-en-un-mundo-cambiante-congreso-virtual-celebrado-entre-20-22-de-octubre-de-2020. [Último acceso: 02 10 2022].

[3] N. Justel, M. Psyrdellis y E. Ruetti, «Modulación de la memoria emocional: una revisión de los principales factores que afectan los recuerdos,» *Suma Psicológica,* vol. 20, nº 2, 2013.

[4] E. Echeburúa y P. J. Amor, «Memoria traumática: estrategias de afrontamiento adaptativas e inadaptativas,» *Terapia psicológica,* vol. 37, nº 1, p. 2, 2019.

[5] D. A. Muñoz González, «La estimulación cognitiva como estrategia para la atención psicogerontológica a los adultos mayores con demencia,» *Revista Cubana de Salud Pública,* vol. 44, nº 3, 2018.

[6] C. d. I. p. D. e. S. Pública, «Instituto Nacional de Salud Pública,» Instituto Nacional de Salud Pública, 21 Septiembre 2020. [En línea]. Available: https://www.insp.mx/avisos/hablemos-de-demencia#sup3. [Último acceso: 04 Octubre 2022].

[7] T. Álvarez Cisneros, S. Torres Castro, B. Mena Montes y . N. M. Torres Carrillo, «Factores de protección para la demencia en adultos mayores en México:

¿dónde estamos? En el libro: Gerontología para la vida activa.,» *Género y salud en cifras,* vol. 15, nº 3, p. 13, 2016.

[8] M. d. L. Reyna Carrizales, M. V. González Rubio, F. J. López Esqueda y G. R. González González , «Demencias, Una visión panorámica,» de *V Simposio de Medicina Geriátrica*, San Luis Potosí, 2014.

[9] Google, «flutter.dev,» Google, 01 Octubre 2022. [En línea]. Available: https://flutter.dev/. [Último acceso: 01 octubre 2022].

[10] G. Developers, «Google Developers,» Google, 2022. [En línea]. Available: https://developer.android.com/studio. [Último acceso: 29 Octubre 2022].

[11] L. Zhou, J. Gao, D. Li y H.-Y. Shum, «The Design and Implementation of XiaoIce, an Empathetic Social Chatbot,» *Computational Linguistics,* nº 46(1), p. 53–93., 2020.

[12] M. IA, «Virtual Beings,» Virtual Beings, 22 08 2019. [En línea]. Available: https://www.youtube.com/watch?v=LLFVz6buGXo. [Último acceso: 10 2022].

[13] D. Biswas, «Delayed Rewards in the context of Reinforcement Learning based Recommender Systems,» *AAI4H@ ECAI,* pp. 49-53, 2020.

[14] T.-H. (. Huang, F. Ferraro, N. Mostafazadeh y . I. Misra, «Visual Storytelling,» *Proceedings of the 2016 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies",* Vols. %1 de %2N16-1147v2, p. 1233–1239, 2016.

[15] P. Ammanabrolu, E. Tien, W. Cheung, Z. Luo, W. Ma, L. Martin y M. Riedl, «Guided Neural Language Generation for Automated Storytelling,» *Association for Computational Linguistics,* vol. Proceedings of the Second Workshop on Storytelling, nº W19-3405, p. 46–55, 2019.

[16] S. M. Lukin, R. Hobbs y C. R. Voss, «A Pipeline for Creative Visual Storytelling,» *Proceedings of the First Workshop on Storytelling (StoryNLP)*, vol. arXiv:1807.08077 , 2018.

[17] N. Li, B. Liu, Z. Han, Y.-S. Liu y J. Fu, «Emotion Reinforced Visual Storytelling,» *ICMR '19: Proceedings of the 2019 on International Conference on Multimedia Retrieval*, vol. ICMR '19, p. 297–305, 2019.

[18] D. S. Rosas, *Procesamiento de lenguaje natura para la búsqueda de patrones en demencia*, Querétaro, Qro. : Universidad Autónoma de Querétaro, Facultad de Ingeniería., 2020.

[19] H. arcía González, *Extracción de recuerdos de vídeos de entrevistas con personas con problemas de memoria*, Madrid: Facultad de Informática Universidad Complutense de Madrid, 2022.

[20] J. Kearns, *Librivox: Free public domain audiobooks.*, 2014.

[21] J. Levis y . R. Suvorov, Automatic Speech Recognition., 2012.

[22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, Y. Unno, J. Nishitoba, Y. Soplin, J. Heymann, N. Enrique , N. Chen y M. Wiesner, «Espnet: End-to-end speech processing toolkit.,» *arXiv preprint arXiv:1804.00015*, 2018.

[23] H. Yaguana Romero y J. P. Arrobo Agila, «23La inteligencia artificial en la narrativa sonora. Estudio de caso,» *Anàlisi: Quaderns de Comunicació i Cultura*, nº 66, pp. 9-23, 2021.

[24] J. C. Alfonso, «Desarrollo de tecnologías de procesamiento del lenguaje de alto rendimiento para transcripción y traducción audiovisual,» Universitat Politècnica de València , 2020. [En línea]. Available: https://aplicat.upv.es/exploraupv/ficha-servicio/capacidad/12512. [Último acceso: 04 Octubre 2022].

[25] B. Santorini, «Part-of-Speech T t-of-Speech Tagging Guidelines for the P agging Guidelines for the Penn Treebank Pr eebank Project (3rd Revision),» *Computer Sciences Commons, University of Pennsylvania,* 1990.

[26] A. M. M. S. B. Taylor, «The Penn Treebank: An Overview.,» *Springer, Dordrecht.,* Vols. %1 de %2In: Abeillé, A. (eds) Treebanks. Text, Speech and Language Technology, vol 20. , nº 20, p. 18, 2003.

[27] A. Taylor, B. Santorini y M. Marcus, «The Penn Treebank: An overview,» Springer, Dordrecht, 2003.

[28] A. Gelbukh, «Alexander,» de *Organización del conocimiento: bibliotecología y terminología.* , Ciudad de México, Instituto Politécnico Nacional, p. 343.

[29] C. D. Manning, «Stanford University,» [En línea]. Available: https://nlp.stanford.edu/pubs/CICLing2011-manning-tagging.pdf. [Último acceso: 2 Junio 2022].

[30] A. Chiche y B. Yitagesu , «Part of speech tagging: a systematic review of deep learning and machine learning approaches,» *SpringerOpen*, Junio 2022.

[31] J. H. Martin. y D. Jurafsky, «Speech and Language Processing,» Diciembre 2021. [En línea]. Available: https://web.stanford.edu/~jurafsky/slp3/A.pdf. [Último acceso: 3 Junio 2022].

[32] D. Jurafsky y . J. H. Martin, Hidden Markov Models, Palo Alto: Stanford, 2021.

[33] *Modelos ocultos de Markov: algoritmo de Viterbi. | | UPV*. [Película]. España: Universitat Politècnica de València - UPV , 2017.

[34] S. a. L. Processing, «Hidden Markov Models,» de *Speech and Language Processing*, Palo Alto, Stanford, 2021, p. Apendix A.

[35] R. Cabeza Ruiz, «Aplicando los Campos Aleatorios Condicionales en la Segmentación de Textos por Idiomas,» de *Conditional Random Fields in Text Segmentation by Language*, Holguin, 2019.

[36] S. K. Siencnik, «Adapting word2vec to named entity recognition,» *Proceedings of the 20th nordic conference of computational linguistics*, vol. 20, p. 239–243., 2015.

[37] Spacy, «Spacy.io,» 01 Octubre 2022. [En línea]. Available: https://spacy.io/usage/spacy-101. [Último acceso: 02 Octubre 2022].

[38] Spacy, «https://spacy.io,» SpaCy, 02 Octubre 2016-2022. [En línea]. Available: https://spacy.io/models/es#es_core_news_sm. [Último acceso: 02 Octubre 2022].

[39] G. Wolf, E. Ruiz, F. Bergero y E. Meza, FUNDAMENTOS DE SISTEMAS OPERATIVOS, Ciudad de México: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2015.

[40] Azure, «¿Qué es el desarrollo de aplicaciones móviles?,» microsoft, [En línea]. Available: https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-mobile-app-development/#definition. [Último acceso: 23 Octubre 2022].

[41] R. Mitchell, Web Scraping with Python Collecting. Data from the Modern Web, Sebastopol: 2015, Reilly Media.

[42] Selenium, «Selenium,» Selenium, Octubre 2022. [En línea]. Available: https://www.selenium.dev/. [Último acceso: 04 Octubre 2022].

[43] Y. Goldberg, Neural Network Methods for Natural Language Processing, Toronto: Graeme Hirst, University of Toronto, 2017.

[44] mathworks, «mathworks,» 01 Octubre 2022. [En línea]. Available: https://la.mathworks.com/discovery/ngram.html. [Último acceso: 01 Octubre 2022].

[45] R. A. Española, « Real Academia Española,» 01 Octubre 2022. [En línea]. Available: https://www.rae.es . [Último acceso: 01 Octubre 2022].

[46] S. d. E. Pública, « https://www.gob.mx/,» Secretaría de Educación Pública, 14 Julio 2014. [En línea]. Available: https://www.gob.mx/sep/acciones-y-programas/estandares-nacionales-de-habilidad-lectora-estandares-de-lectura. [Último acceso: 12 Noviembre 2022].

[47] B. Muthukadan, «Selenium with Python,» 2011-2018. [En línea]. Available: https://selenium-python.readthedocs.io/. [Último acceso: 2 Junio 2022].

[48] P. S. Foundation, «Python Software Foundation,» Python Software Foundation, 2022. [En línea]. Available: https://pypi.org/project/selenium/. [Último acceso: 2 Junio 2022].

[49] M. d. C. García Peña, L. M. Gutiérrez Robledo , P. A. Roa Rojas y A. Martínez Ruiz, «La Enfermedad de Alzheimer y otras demencias como problema nacional de salud,» Academia Nacional de Medicina de México (ANMM), México, 2017.

[50] Secretaría de Salud de México, «https://www.gob.mx/,» Secretaria de Salud de México, 05 Octubre 2021. [En línea]. Available: https://www.gob.mx/salud/es/articulos/enfermedad-de-alzheimer-demencia-mas-comun-que-afecta-a-personas-adultas-mayores?idiom=es. [Último acceso: 01 Octubre 22].

[51] R. Mitchell, Web Scraping with Python Collecting. Data from the Modern Web, Sebastopol: 2015, Reilly Media.

[52] K. Malave Polanco y J. L. Beauperthuy Taibo, «“ANDROID” EL SISTEMA OPERATIVO DE GOOGLE PARA DISPOSITIVOS MÓVILES,» *Revista Científica Electrónica Ciencias Gerenciales*, nº 19, pp. 79- 96, 2011.

Anexos 1 Tabla de relación de tiempo con número de sílabas (TTS)

Tabla de procesamiento de palabras monosilábicas con librería Gtts de procesamiento de texto a discurso. (relación audio con número de caracteres y sílabas).

Folio	Palabra	Caracteres	TTS Normal seg	TTS Slow seg
1	do	2	0.624	0.696
2	fa	2	0.624	0.72
3	ir	2	0.552	0.624
4	ni	2	0.648	0.72
5	re	2	0.624	0.696
6	se	2	0.648	0.744
7	si	2	0.696	0.768
8	tu	2	0.576	0.624
9	yo	2	0.6	0.672
10	bar	3	0.624	0.696
11	con	3	0.696	0.768
12	dar	3	0.624	0.696
13	dio	3	0.696	0.792
14	don	3	0.744	0.864
15	doy	3	0.72	0.792
16	fan	3	0.744	0.864
17	fin	3	0.744	0.84
18	gas	3	0.84	0.984
19	gel	3	0.696	0.792
20	gen	3	0.744	0.864
21	gol	3	0.696	0.792
22	luz	3	0.84	0.936
23	mal	3	0.672	0.744
24	mar	3	0.624	0.696
25	mas	3	0.84	0.96
26	mes	3	0.84	0.936
27	mil	3	0.672	0.768
28	muy	3	0.696	0.768
29	paz	3	0.744	0.84
30	pan	3	0.672	0.768
31	par	3	0.576	0.624
32	pez	3	0.792	0.912
33	pie	3	0.624	0.696
34	por	3	0.576	0.648
35	pum	3	0.648	0.72
36	pus	3	0.768	0.864
37	ron	3	0.744	0.84
38	sal	3	0.744	0.816
39	sed	3	0.696	0.768
40	sin	3	0.792	0.888
41	son	3	0.768	0.864
42	sor	3	0.672	0.768
43	soy	3	0.768	0.864
44	sur	3	0.696	0.792
45	tan	3	0.672	0.744
46	tez	3	0.768	0.888
47	tul	3	0.648	0.72
48	ven	3	0.72	0.792
49	ver	3	0.6	0.696
50	vil	3	0.696	0.768

Tabla de procesamiento de palabras bisilábicas con librería Gtts de procesamiento de texto a discurso (relación audio con número de caracteres y sílabas).

Folio	Palabra	Caracteres	TTS Normal seg	TTS Slow seg
0	aire	4	0.744	0.864
1	arca	4	0.888	1.032
2	percha	6	0.936	1.08
3	liebre	6	0.84	0.96
4	social	6	0.984	1.128
5	bordes	6	0.864	0.96
6	ama	3	0.864	0.984
7	antes	5	0.936	1.08
8	gritar	6	0.864	0.984
9	gorro	5	0.936	1.104
10	ratón	5	0.936	1.08
11	banco	5	0.96	1.104
12	soñar	5	0.888	1.008
13	beso	4	0.936	1.08
14	bañar	5	0.84	0.96
15	tabla	5	0.888	1.032
16	Simón	5	0.96	1.104
17	rico	4	0.936	1.08
18	beto	4	0.912	1.056
19	barco	5	0.96	1.128
20	bolos	5	0.96	1.128
21	sentir	6	0.912	1.056
22	rojo	4	0.96	1.104
23	bola	4	0.888	1.056
24	pide	4	0.72	0.792
25	torcer	6	0.864	0.984
26	cortar	6	0.816	0.936
27	rata	4	0.936	1.104
28	bacha	5	1.008	1.152
29	cosa	4	0.912	1.056
30	casa	4	0.912	1.056
31	beber	5	0.768	0.864
32	dulce	5	0.888	1.008
33	corto	5	0.912	1.056
34	azul	4	0.816	0.936
35	atún	4	0.888	1.008
36	cana	4	0.888	1.032
37	cara	4	0.84	0.984
38	contar	6	0.816	0.936
39	capot	5	0.888	1.008
40	taza	4	0.912	1.056
41	largo	5	0.936	1.08
42	caja	4	0.888	1.032
43	vasco	5	0.984	1.128
44	saco	4	1.008	1.152
45	suave	5	0.864	0.984
46	casa	5	0.816	0.936
47	carro	5	0.888	1.008
48	cuatro	6	0.936	1.104
49	cantar	6	0.816	0.936
50	vaso	4	0.96	1.08

Tabla de procesamiento de palabras trisilábicas librería Gtts de procesamiento de texto a discurso (relación audio con número de caracteres y sílabas).

Folio	Palabra	Caracteres	TTS Normal seg	TTS Slow seg
0	Acabar	6	0.888	1.032
1	Aceptar	7	0.984	1.128
2	Acuerdo	7	1.128	1.296
3	Además	6	1.08	1.248
4	Afirmar	7	0.96	1.104
5	Alcanzar	8	1.056	1.2
6	Alguno	6	1.032	1.2
7	Amigo	5	1.032	1.2
8	Androide	8	1.056	1.224
9	Animado	7	1.2	1.416
10	Ansiado	7	1.128	1.32
11	Ansiedad	8	1.008	1.152
12	Anterior	8	0.984	1.128
13	Añora	5	1.08	1.224
14	Apesta	6	1.128	1.296
15	Aplasta	7	1.176	1.392
16	Aragán	6	0.984	1.128
17	Asechas	7	1.176	1.344
18	Aspecto	7	1.152	1.344
19	Asunto	6	1.08	1.248
20	Aumenta	7	1.152	1.344
21	Aumentar	8	1.032	1.2
22	Bellaco	7	1.152	1.344
23	Brillante	9	1.056	1.224
24	Cabeza	6	1.056	1.224
25	Cámara	6	1.032	1.176
26	Camello	7	1.056	1.224
27	Camino	6	1.032	1.176
28	Canela	6	1.032	1.176
29	Cansado	7	1.104	1.296
30	Capataz	7	1.176	1.344
31	Capital	7	1.032	1.176
32	Carreta	7	1.08	1.224
33	Ciruelo	7	1.176	1.368
34	Colegio	7	1.104	1.272
35	Comida	6	1.032	1.176
36	Comenzar	8	1.008	1.152
37	Comercio	8	1.128	1.32
38	Cometa	6	1.056	1.224
39	Condición	9	1.152	1.344
40	Conocer	7	0.96	1.104
41	Conocer	7	0.984	1.128
42	Conseguir	9	1.008	1.176
43	Consejo	7	1.128	1.32
44	Contestar	9	1.056	1.2
45	Contiene	8	1.008	1.176
46	Contrario	9	1.128	1.32
47	Corbata	7	1.128	1.296
48	Corneta	7	1.104	1.272
49	Cornisa	7	1.104	1.296
50	Coronel	7	0.96	1.104

Tabla de procesamiento de palabras tetrasílabico librería Gtts de procesamiento de texto a discurso (relación audio con número de caracteres y sílabas).

Folio	Palabra	Caracteres	TTS Normal seg	TTS Slow seg
0	Alegría	7	1.152	1.344
1	Alteridad	9	1.08	1.248
2	Anticuario	10	1.296	1.512
3	Archivador	10	1.128	1.32
4	Balistica	9	1.272	1.488
5	Barbacoa	8	1.296	1.512
6	Blanquecino	11	1.32	1.536
7	Bolígrafo	9	1.248	1.44
8	Bondadoso	9	1.296	1.512
9	Brutalmente	11	1.296	1.536
10	Calabaza	8	1.248	1.44
11	Carretera	9	1.2	1.392
12	Comestible	10	1.152	1.344
13	Configurar	10	1.08	1.248
14	Decisivo	8	1.248	1.44
15	Democracia	10	1.368	1.584
16	Deportista	10	1.368	1.608
17	Desquiciado	11	1.392	1.632
18	Diccionario	11	1.368	1.584
19	Disciplina	10	1.296	1.488
20	Discoteca	9	1.368	1.584
21	Elefante	8	1.08	1.272
22	Empleado	8	1.224	1.416
23	Emprendedor	11	1.2	1.368
24	Eolico	6	1.2	1.392
25	Esdrújula	9	1.2	1.392
26	Espíritu	8	1.152	1.32
27	Estudiante	10	1.176	1.392
28	Estupendo	9	1.296	1.512
29	Excelente	9	1.176	1.32
30	Explosivo	9	1.344	1.56
31	Fantístico	10	1.368	1.608
32	Felizmente	10	1.248	1.464
33	Futurista	9	1.248	1.464
34	Gallinero	9	1.224	1.416
35	Generoso	8	1.248	1.464
36	Habitación	10	1.224	1.416
37	Herramienta	11	1.272	1.488
38	Honorable	9	1.032	1.2
39	Intimidar	9	1.08	1.248
40	Increíble	9	1.176	1.368
41	Instrumento	11	1.32	1.536
42	Kilogramo	9	1.176	1.392
43	Lapicero	8	1.248	1.464
44	Liberado	8	1.152	1.344
45	Mandarina	9	1.248	1.44
46	Mariposa	8	1.272	1.488
47	Masculino	9	1.272	1.512
48	Menosprecio	11	1.416	1.68
49	Mensajero	9	1.296	1.488
50	Modernismo	10	1.344	1.56

Tabla de procesamiento de palabras pentasílabas librería Gtts de procesamiento de texto a discurso (relación audio con número de caracteres y sílabas).

Folio	Palabra	Caracteres	TTS Normal seg	TTS Slow seg
0	abigarrado	10	1.32	1.536
1	abastecido	10	1.416	1.632
2	abominable	10	1.2	1.392
3	aborrecible	11	1.248	1.464
4	abotonado	9	1.32	1.536
5	absolutismo	11	1.488	1.728
6	absolutorio	11	1.416	1.68
7	abstencionismo	14	1.608	1.872
8	adversamente	12	1.392	1.608
9	alegremente	11	1.32	1.536
10	altisonante	11	1.296	1.512
11	alumbramiento	13	1.464	1.728
12	anticoncepción	14	1.512	1.776
13	apetecible	10	1.296	1.512
14	astrofísica	11	1.44	1.68
15	comunicación	12	1.392	1.608
16	constantemente	14	1.44	1.68
17	culturalmente	13	1.44	1.68
18	disciplinario	13	1.464	1.704
19	dodecaedro	10	1.392	1.608
20	ecología	8	1.296	1.512
21	ecosistema	10	1.392	1.632
22	ecumnico	8	1.296	1.512
23	especialista	12	1.488	1.704
24	especulación	12	1.44	1.68
25	fisioterapia	12	1.464	1.728
26	fotografía	10	1.392	1.656
27	guardaespaldas	14	1.536	1.824
28	hermafrodita	12	1.392	1.632
29	hipocresía	10	1.392	1.632
30	luminiscencia	13	1.488	1.776
31	instantáneo	11	1.392	1.656
32	internacional	13	1.392	1.632
33	invernadero	11	1.368	1.584
34	irresponsable	13	1.416	1.608
35	literatura	10	1.368	1.608
36	matemáticas	11	1.44	1.704
37	misericordia	12	1.464	1.752
38	multicultural	13	1.368	1.608
39	nacionalista	12	1.488	1.728
40	oftalmólogo	11	1.368	1.632
41	onomástico	10	1.368	1.608
42	oprimido	8	1.2	1.392
43	panadería	9	1.248	1.464
44	planificación	13	1.464	1.704
45	positivismo	11	1.44	1.656
46	rinoceronte	11	1.296	1.488
47	racionalismo	12	1.512	1.752
48	rápidamente	11	1.296	1.536
49	tecnología	10	1.368	1.584
50	visibilidad	11	1.224	1.44

Anexos 2 Part-of-speech tagging SPacCy [37]

POS	DESCRIPTION	EXAMPLES	DESCRIPCIÓN	EJEMPLOS
ADJ	*adjective*	*big, old, green, incomprehensible, first*	adjetivo	grande, viejo, verde, incomprensible, primero
ADP	*adposition*	*in, to, during*	adposición	en, a, durante
ADV	*adverb*	*very, tomorrow, down, where, there*	adverbio	muy, mañana, abajo, donde, allí
AUX	*auxiliary*	*is, has (done), will (do), should (do)*	auxiliar	es, ha (hecho), hará (hará), debería (hacer)
CONJ	*conjunction*	*and, or, but*	conjunción	y, o, pero
CCONJ	*coordinating conjunction*	*and, or, but*	conjunción de coordinación	y, o, pero
DET	*determiner*	*a, an, the*	determinante	una una la
INTJ	*interjection*	*psst, ouch, bravo, hello*	interjección	psst, ay, bravo, hola
NOUN	*noun*	*girl, cat, tree, air, beauty*	sustantivo	niña, gato, árbol, aire, belleza
NUM	*numeral*	*1, 2017, one, seventy-seven, IV, MMXIV*	numeral	1, 2017, uno, setenta y siete, IV, MMXIV
PART	*particle*	*’s, not,*	partícula	es, no,
PRON	*pronoun*	*I, you, he, she, myself, themselves, somebody*	pronombre	Yo, tú, él, ella, yo mismo, ellos mismos, alguien
PROPN	*proper noun*	*Mary, John, London, NATO, HBO*	nombre propio	Mary, John, Londres, OTAN, HBO
PUNCT	*punctuation*	*., (,), ?*	puntuación	., (,), ?
SCONJ	*subordinating conjunction*	*if, while, that*	conjunción subordinada	si, mientras, eso
SYM	*symbol*	*$, %, §, ©, +, −, ×, ÷, =, :),* 😝	símbolo	$, %, §, ©, +, −, ×, ÷, =, :), 😝
VERB	*verb*	*run, runs, running, eat, ate, eating*	verbo	corre, corre, corre, come, comió, comiendo
X	*other*	*sfpksdpsxmsa*	otro	sfpksdpsxmsa
SPACE	*space*		espacio	

Tabla Named Entity Recognition SpaCy [37]

NER	Description	Descripción
PERSON	*People, including fictional.*	Personas, incluidas las ficticias.
NORP	*Nationalities or religious or political groups.*	Nacionalidades o grupos religiosos o políticos.
FAC	*Buildings, airports, highways, bridges, etc.*	Edificios, aeropuertos, carreteras, puentes, etc.
ORG	*Companies, agencies, institutions, etc.*	Empresas, agencias, instituciones, etc.
GPE	*Countries, cities, states.*	Países, ciudades, estados.
LOC	*Non-GPE locations, mountain ranges, bodies of water.*	Ubicaciones no GPE, cadenas montañosas, cuerpos de agua.
PRODUCT	*Objects, vehicles, foods, etc. (Not services.)*	Objetos, vehículos, alimentos, etc. (No servicios.)
EVENT	*Named hurricanes, battles, wars, sports events, etc.*	Nombrado huracanes, batallas, guerras, eventos deportivos, etc.
WORK_OF_ART	*Titles of books, songs, etc.*	Títulos de libros, canciones, etc.
LAW	*Named documents made into laws.*	Documentos nombrados convertidos en leyes.
LANGUAGE	*Any named language.*	Cualquier idioma con nombre.
DATE	*Absolute or relative dates or periods.*	Fechas o periodos absolutos o relativos.
TIME	*Times smaller than a day.*	Tiempos menores a un día.
PERCENT	*Percentage, including ”%“.*	Porcentaje, incluyendo ”%“.
MONEY	*Monetary values, including unit.*	Valores monetarios, incluida la unidad.
QUANTITY	*Measurements, as of weight or distance.*	Medidas, como de peso o distancia.
ORDINAL	*“first”, “second”, etc.*	“primero”, “segundo”, etc.
CARDINAL	*Numerals that do not fall under another type.*	Números que no pertenecen a otro tipo.
PER	*Named person or family.*	Persona o familia nombrada.
LOC	*Name of politically or geographically defined location (cities, provinces, countries, international regions, bodies of water, mountains).*	Nombre de la ubicación definida política o geográficamente (ciudades, provincias, países, regiones internacionales, cuerpos de agua, montañas).
ORG	*Named corporate, governmental, or other organizational entity.*	Entidad corporativa, gubernamental u otra organización nombrada.
MISC	*Miscellaneous entities, e.g. events, nationalities, products or works of art.*	Entidades misceláneas, por ejemplo, eventos, nacionalidades, productos u obras de arte.
PER	*Named person or family.*	Persona o familia nombrada.

Anexos 3 Publicación

Publicación realizada en el 5to. Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT) y 1er. Congreso Internacional de Mecatrónica, Control e Inteligencia (CIMCIA)

Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial.

Jaime Emilio Zavala Barrios, Víctor Manuel Zamudio Rodríguez
División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/ IT de León
León Gto. México
{M2020116, vic.zamudio}@leon.tecnm.mx

Carlos Lino Ramírez and David Asael Gutiérrez Hernández
División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/IT de León
León Gto. México
{carloslino, david.gutierrez}@leon.tecnm.mx

***Resumen*— En México se estima que aproximadamente un millón 300 mil personas padecen la enfermedad de Alzheimer, cuya cifra representa ser entre el 60 y 70 por ciento de los mexicanos diagnosticados con demencia. El Alzheimer afecta con mayor frecuencia a las personas mayores de 65 años. [1] Actualmente no existe una cura para la enfermedad. En esta investigación se plantea la implementación del procesamiento del lenguaje natural para el análisis de recuerdos y la extracción de información con las técnicas de identificación de partes de discurso y reconocimiento de identidades, con la finalidad de preservar los recuerdos en un repositorio digital y enriquecer las historias con la sincronización de material multimedia.**

Palabras Clave—machine learning, data mining, text mining, storytelling, tokenization y named-entity recognition.

I. INTRODUCCIÓN

En el presente artículo se presenta un trabajo en progreso relativo al almacenamiento de información (basado en recuerdos) en repositorio digitales, apoyado con estrategias de procesamiento de lenguaje natural y sincronización de contenido multimedia. La motivación de la investigación en desarrollo tiene como finalidad la preservación de recuerdos de usuarios con algún tipo de demencia (incluyendo Alzheimer), a través de un repositorio digital. Este repositorio permite la aplicación de técnicas de procesamiento de lenguaje natural para la clasificación y extracción de la información contenida.

El Alzheimer es un tipo de demencia que afecta directamente a la memoria, el comportamiento y el pensamiento, al ser una enfermedad crónica los síntomas en la última fase avanzan hasta el punto de que logran interferir con las tareas cotidiana.

A inicios del siglo XX, se describió por primera vez por Alois Alzheimer las lesiones neuropatológicas asociadas con un tipo particular de demencia senil. Este hecho fue la pauta para establecer un nuevo concepto etiológico que en la actualidad se le conoce como demencia por enfermedad de Alzheimer [1].

Dicha enfermedad aumenta gradualmente, por lo cual desde su detección se observa como los síntomas de demencia empeoran progresivamente durante varios años. Una de las características en etapas tempranas es la pérdida de memoria sobre detalles pequeños y es considerada como una pérdida de memoria tenue. Sin embargo, las personas que llegan a padecerla en las etapas más avanzadas desarrollan la pérdida de capacidad para mantener una conversación y posteriormente, en la última etapa de la enfermedad, la posibilidad de responder a su entorno [2].

En promedio, una persona con Alzheimer vive de 4 a 8 años después del diagnóstico, pero con una probabilidad de hasta 20 años, dependiendo de múltiples factores como el cuidado, la detección temprana de la enfermedad, el uso de los medicamentos entre otros, pero como factor principal la evolución de la enfermedad [3].

Es entonces que este trabajo gira en torno a la preservación de la memoria de las personas diagnosticas con la enfermedad, mediante la ejecución de estrategias digitales e implementación de algunos algoritmos de inteligencia artificial para el análisis del contenido, extracción de información, sincronización de contenido multimedia con la narrativa automatizada (videos, fotos, grabaciones, etc.), así como el mapeo de relaciones de datos.

La conservación de la memoria de una persona es un tema muy relevante en el contexto del Alzheimer, pues tal como se abordó con antelación, en las primeras etapas de la enfermedad, la persona comienza a olvidar detalles menores de su día y mientras que la enfermedad avanza, de manera progresiva, aquello que parecía claro y fácil de recordar ahora cuesta completar o recuperar fragmentos y datos de su memoria [1].

La memoria puede sufrir alteraciones por diferentes factores tales como los emocionales [4], entendiéndose estos como aquellos que protegen a la persona de una mala experiencia, por otro lado, el tiempo también permite agregar o remover detalles del suceso original [5].

La memoria puede estimularse y recordar fragmentos olvidados gracias a estímulos sensoriales, o bien con ayuda de material audiovisual, por ejemplo; al observar una fotografía, escuchar una canción o audios, así como ver un algún video pues estos materiales funcionan como detonantes que permiten revivir o recordar experiencias [6].

Con el avance de las herramientas digitales, se puede plantear la concepción y generación de nuevas estrategias para el almacenamiento digital, permitiendo a la persona almacenar y proteger el momento. Aunque el eje de la investigación se concentra en pacientes con Alzheimer, la herramienta estará disponible para cualquier persona, sin importar si es o no diagnosticada con esta enfermedad.

II. JUSTIFICACIÓN

"La demencia es un nombre colectivo para los síndromes cerebrales degenerativos progresivos que afectan la memoria, el pensamiento, el comportamiento y las emociones" [7].

A partir de los datos oficiales se observa como México se está convirtiendo en un país con más personas mayores que infantes menores de 5 años, se prevé que para el año 2050, la proporción de las juventudes de 15 años en la población total disminuirá de 30.0% a 17.4%. De este modo, el incremento en la población de personas adultas mayores iría en aumento con la utilización de los servicios de salud destinados a la atención de enfermedades propias del envejecimiento como las demencias [8].

La demencia es uno de los factores que más contribuyen a la discapacidad ya la dependencia en las personas adultas mayores. La Enfermedad de Alzheimer (EA), es el tipo de demencia más frecuente y actualmente se conoce que existe una prevalencia del 7.3% y una incidencia de 27.3 (1000 personas/año) de la población adulta mayor mexicana para la EA [9]. En el sitio oficial de la secretaria de salud de México se indica que existe aproximadamente un millón 300 mil personas padecen la enfermedad de Alzheimer, cifra que representa entre 60 y 70 por ciento de los diagnósticos de demencia en el país y que afecta con mayor frecuencia a las personas mayores de 65 años [1]. La enfermedad de Alzheimer se genera debido a cambios en el cerebro por la presencia de la proteína llamada beta amiloide, que se acumula frecuentemente en el lóbulo temporal. Dicha toxina provoca inflamación y muerte progresiva de neuronas [3]. La Organización Mundial de la Salud (OMS) calcula una cifra aproximada de 60 millones de personas a nivel global que viven con Alzheimer, de las cuales 8.1 por ciento son mujeres y 5.4 por ciento, hombres mayores de 65 años [3].

Para una atención adecuada de la persona con enfermedad de Alzheimer, es necesario el diagnóstico temprano y tratamiento integral de alta especialidad que se basa en medicamentos que estimulan y ayudan a prolongar la vida de las neuronas de la memoria para mejorar su calidad de vida. [3].

En el presente trabajo, el procesamiento de recuerdos permitirá a pacientes con Alzheimer poder acceder a recuerdos aun con en el avance de la enfermedad, extraer información de las transcripciones que serán enriquecidas con material multimedia para estimular experiencias y hacerlas más vívidas. Las memorias almacenadas estarán disponibles para ser compartidas con familiares y descendencia, de tal modo que generaciones distintas de una familia puedan conectarse, siempre que la persona dé su consentimiento.

Al preservar las memorias del paciente, a pesar de la perdida de detalles sobre sus recuerdos, podrá revivirlos por medio de la aplicación, como la primera vez que fueron subidos al ser complementados con el contenido multimedia.

El procesamiento de lenguaje natural es una técnica indispensable que permite analizar las transcripciones de los recuerdos, extraer la información y complementarlos con material multimedia relacionado a los datos extraídos. Estos contenidos multimedia pueden ser enriquecidos con archivos subidos de otras experiencias previas, información disponible en la red o con la que contenida en el sistema.

Al almacenar las memorias de la persona dentro de un repositorio digital, se evitará que el paso del tiempo u otros factores como enfermedad, medicamentes, lesiones modifiquen la historia y el recuerdo.

El procesamiento de lenguaje natural se llevará a cabo valiéndose de las transcripciones en formato texto de los audios, controlándose desde la aplicación el formato de en mp4, y con la captura de recuerdos de manera escrita se aceptarán archivos en diferentes formatos: PDF, Text. y. Doc, para su posterior conversión en formato de texto plano.

La conservación del audio permitirá validar la precisión de las transcripciones escritas con respecto al audio original e identificar palabras que al sistema le cuesta diferenciar por la velocidad del habla de cada persona.

III. MARCO TEORICO

A. Parte del discurso / (Part of Speech Tagging - POS)

Parte del discurso se refiere a la categoría de palabras o los términos léxicos en un idioma.

El Español contiene 8 grupo de clases de palabras que hacen referencia a estos términos léxicos: sustantivo, verbo, adjetivo, adverbio, pronombre, preposición, conjunciones y nombres. En la siguiente tabla se muestra un ejemplo de clases de palabras, su abreviación basada en "T*he Penn TreeBank Tagset"* [3] y en la tercera columna se logra apreciar un conjunto de palabras de cada categoría.

TABLA DE EJEMPLO DE ASIGNACIÓN DE ETIQUETAS.

Término léxico	Etiqueta estándares	Ejemplos
Sustantivo	NN	Algo, nada
Verbo	VB	Comer, correr
Preguntas	WRB	Cómo, Cuándo
Artículos	DT	El, la, los

El Penn Treebank, en sus ocho años de funcionamiento (1989–1996), produjo aproximadamente 7 millones de palabras de texto etiquetado de parte del discurso, 3 millones de palabras

de texto analizado esqueléticamente, más de 2 millones de palabras de texto analizado para la estructura de argumento predicativo y 1,6 millones de palabras de texto hablado transcrito y anotado por falta de fluidez en el habla [4].

TABLA I. THE PENN TREEBANK TAGSET. ADAPTADO DE [27]

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund, or present participle
VBN	Verb, past participle
VBP	Verb, non-3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Wh-determiner
WP	Wh-pronoun
WP$	Possessive wh-pronoun
WRB	Wh-adverb

B. Reconocimiento de entidad nombrada.

Con la tarea de identificación de identidades, se pueden reconocer patrones numéricos para ser categorizados como entidades de tiempo, formulas químicas, entidades monetarias, expresiones de porcentajes y patrones de fechas.

Anterior a los modelos de aprendizaje profundo se utilizaba modelos probabilísticos tales como los Modelos Ocultos de Markov [11] y Campos Aleatorios Condicionales [12].

A partir de la implementación del Aprendizaje profundo o Deep Learning se han ido desarrollando diferentes enfoques para poder resolver la tarea del reconocimiento de entidades nombradas, algunos son implementando modelos con incrustaciones de vectores Word2Vec, Glove, Senna y FastText [13].

IV. METODOLOGÍA EXPERIMENTAL PROPUESTA

El desarrollo de la aplicación se llevó por medio del software Android Studio, utilizando el SDK Software Development Kit Flutter, el cual es de código fuente abierto creado por Google [14].

Se desarrolló una aplicación híbrida para sistemas operativos de Android y IOS. Esta aplicación busca conservar los recuerdos por medio del discurso oral y escrito del usuario contemplando en el desarrollo de la interfaz, las brechas [15] generacionales para simplificar sus funciones.

El sistema operativo de Apple iOS, y diseñado exclusivamente para el hardware producido por dicha compañía fue el primero en implementar el interfaz usuario multitouch y, en buena medida, se puede ver como el responsable de la explosión y universalización en el uso de dispositivos móviles. Al igual que el sistema operativo que emplean para sus equipos de escritorio, MacOS X, iOS está basado en el núcleo Darwin, derivado de FreeBSD, un sistema libre tipo Unix [16].

El sistema operativo Android Diseñado por la compañía Google, basa la mayor parte de su operación en software libre (un núcleo Linux, máquina virtual Java, y muchas de las bibliotecas de sistema comunes en sistemas Linux), agregando una capa de servicios propietarios. La estrategia de Google ha sido inversa a la de Apple: en vez de fabricar sus propios dispositivos, otorga licencias para el uso de este sistema operativo a prácticamente todos los fabricantes de hardware, con lo que la amplia mayoría de los modelos de teléfonos inteligentes y tabletas corren sobre Android [16].

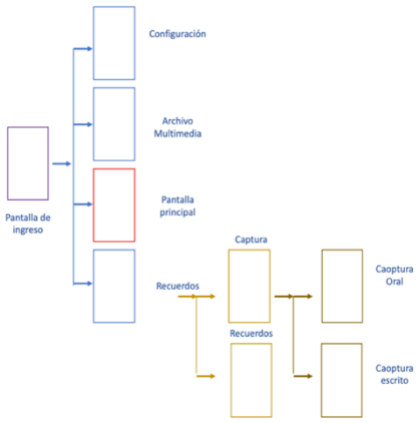


Fig. 1. Diagrama de estructura de aplicación.

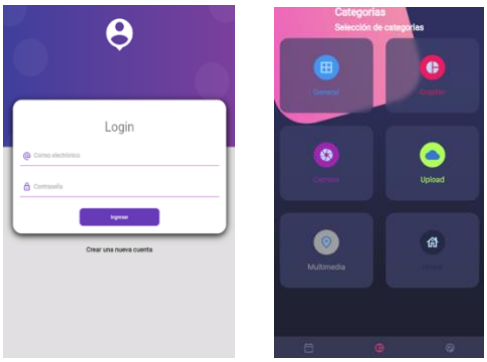


Fig. 2. Pantalla de ingreso a la aplicación y categorias.

Flutter funciona con Dart, un lenguaje optimizado para aplicaciones rápidas en cualquier plataforma que permite que desde una única base de código se la aplicación se pueda implementar en múltiples dispositivos: dispositivos móviles, web, de escritorio e integrados [14].

La aplicación móvil desarrollada en la investigación busca ser la herramienta que funcione como puente entre la parte de la captura de los recuerdos en su versión oral o escrita, con el repositorio digital y el sistema de narrativa inteligente.

Entre las características que se diseñaron para la aplicación se encuentran aquellas que toman en cuenta al paciente:

- La aplicación debe ser responsiva (se adapta a las diferentes dimensiones de equipos).
- Debe contar con características de usos intuitivo para el usuario.
- Tener las funciones integradas para poder actualizarse de manera remota.
- La herramienta debe estar diseñada para ser compatible con el mayor número de dispositivos Android y permitir el uso de código para el sistema IOS.

El primer paso del proyecto se enfoca en la captura de los recuerdos de las personas para la extracción y generación de transcripciones en texto plano ".txt" para su posterior procesamiento, como se muestra en la Figura 3.

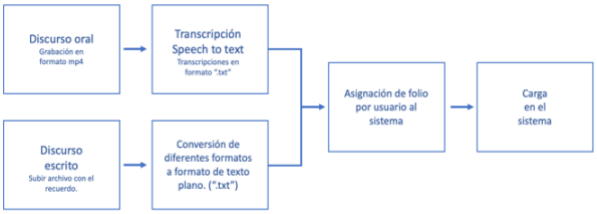


Fig. 3. Diagrama de captura de recuerdo.

En la Figura 3. Se observa la captura de los recuerdos por parte de grabaciones, las cuales se almacenarán ya que son un medio de validar si las transcripciones se están realizando correctamente.

En la parte de captura, el proyecto busca mantener la integridad de la historia en el caso de que la persona suba una igual posteriormente, por lo cual se considerará la primera historia subida al sistema, como la original. A partir de este relato se pueden realizar comparaciones con cargas posteriores, cuyo objetivo es detectar las variaciones entre ellas, como cambios, la disminución de detalles, la falta de participantes entre otros aspectos.

Teniendo en formato de texto plano los recuerdos o textos a procesar se puede implementar las técnicas de identificación de partes de discurso y reconocimiento de entidades.

Entender la identificación de discurso como la generación de un etiquetado, que generalmente se modela como una tarea estructurada: la etiqueta de la primera palabra puede depender de la etiqueta de la tercera, pero se puede aproximar bastante bien clasificada cada palabra de forma aislada en una etiqueta ("part-of-speech" en inglés, siglas POS), basada en una ventana de dos palabras a cada lado de la palabra. Si se etiquetan las palabras en un orden fijo, por ejemplo, de izquierda a derecha, también se puede condicionar cada predicción de etiquetado a las predicciones de etiquetas realizadas en etiquetas anteriores [17].

En la NER se proporciona un documento y se necesita encontrar entidades nombradas, así como categorizarlas en un conjunto predefinido de categorías como locación, organización, personas y otras categorías [17]. Algunas palabras pueden recibir una categoría diferente de acuerdo con el contexto en el que se emplean dentro del texto.

Se optó por un arreglo que almacene la información para cada palabra. Esta información va a estar separada por categorías relacionada a la implementación en las tareas de sincronización como se aprecia en la Figura 4.

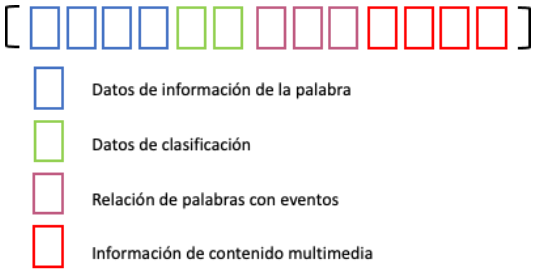


Fig. 4. Diagrama de estructura de cada elemento del texto, excluyendo signos de puntación.

En la Figura 5 Se desglosa más a detalle las funciones dentro de cada una de las categorías.

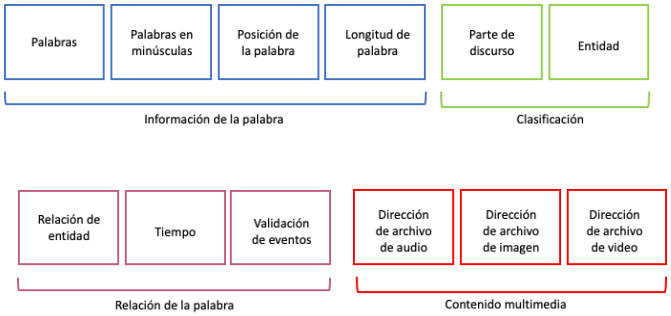


Fig. 5. Desglozado de procesamiento de cada palabra dentro del recuerdo.

Para algunas de las funciones de la aplicación y del sistema se consideró el uso de la librería SpaCy.

SpaCy es una biblioteca gratuita de código abierto para el procesamiento avanzado del lenguaje natural (NLP) en Python, diseñada para ayudar a crear aplicaciones NLP, cuyo uso no esté restringido por cuota de palabras o eventos, haciendo que no sea un servicio consumible y siendo una opción gratuita [18].

Por medio de la librería referida, se pueden construir sistemas de extracción de información o comprensión del lenguaje natural, procesar y entender grandes volúmenes de texto.

En lo que corresponde a la parte experimental, se utilizó el modelo en español de nivel medio, sin desestimar las palabras vacías ("*stop words*" en inglés) las cuales son palabras muy frecuentes en el idioma, pero cuyo uso no genera un significado importante al contenido.

Se asume que el conjunto de textos fue revisado y ajustado para que su composición tenga sentido y cumpla con las reglas gramaticales del idioma implementado.

El proceso anterior hace referencia a la corrección de errores en la escritura o transcripción de determinadas palabras para proceder con el procesamiento y extracción de información de los documentos.

Las palabras son posteriormente separadas en elementos individuales denominados tokens mismos que se obtienen al separar cada frase en sus elementos individuales sin considerar los signos de puntación. Esto se puede llevar a cabo por medios de ciclos que recorran e identifiquen cada espacio vacío, o la aparición de un signo de puntación en el caso del español.

En la siguiente Figura 7 se puede apreciar un diagrama del recorrido que permitirá identificar cada token y encapsularlo en frases o párrafos según la función. Como se ha mencionado anteriormente, se debe de ajustar el proceso para considerar las reglas gramaticales del idioma trabajado.

Dentro del español se implementa un recorrido de izquierda a derecha, cada espacio en blanco delimita a una palabra, al igual que la presencia de ciertos signos de puntuación: puntos, comas, entre otros y dentro del texto plano se reconoce a la combinación de ".\n" como saltos de párrafo y a la presencia de "\n" como un salto de línea, que puede o no removerse dentro de los procesos o implementarse para dar una estructura.

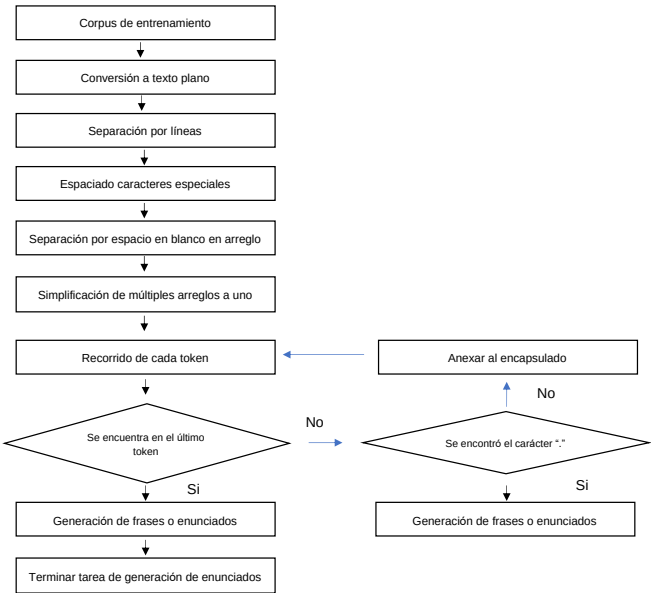


Fig. 6. Diagrama de procesamiento de texto.

En las siguientes líneas de seudocódigo se observa el desglose de los pasos para la selección y sincronización del material de las transcripciones.

Es importante mencionar que no todas las categorías de POS en que serán clasificadas las palabras requieran una sincronización con el material multimedia, como son los casos de artículos, verbos, conjunciones entre otras

categorías, pero la presencia anterior o posterior puede aumentar el tiempo del contenido a mostrarse de la palabra que si requiere su sincronización. Ello se refleja en el código de la Figura 7.

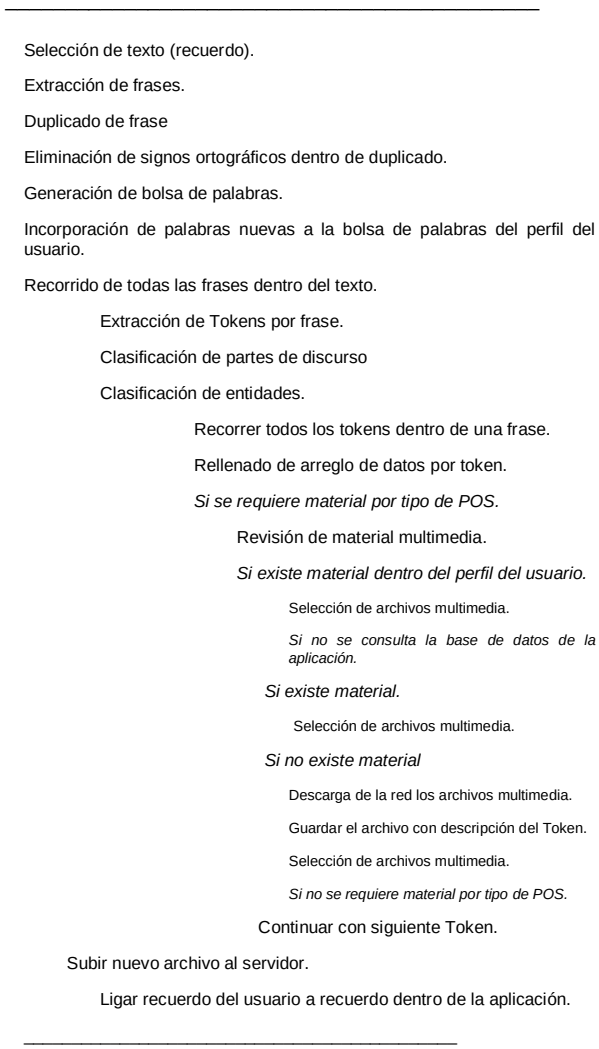


Fig. 7. Diagrama de procesamiento de recuerdos y sincornización.

Los recorridos por cada frase permiten identificar las expresiones regulares por medio de n-gramas, controlando la cantidad de palabras que se agrupan.

Un n-grama es un conjunto de n elementos consecutivos en un documento de texto, que puede incluir palabras, números, símbolos y puntuación. Los modelos de n-gramas son útiles en muchas aplicaciones de análisis de texto en que la secuencia de palabras es pertinente, tales como análisis de sentimiento, clasificación de texto y generación de texto. El modelado de n-gramas es una de las técnicas utilizadas para convertir texto de un formato no estructurado a un formato estructurado [19].

Una de las ventajas de la identificación de las expresiones regulares, es que se puede poseer una mayor precisión sobre el dominio de patrones con los que se está trabajando. Entre más recuerdos va ingresando la persona, más se podrá personalizar el sistema. Se puede determinar la frecuencia de algunas palabras, la extensión de su vocabulario, los patrones del habla, etc., lo cual permitirá facilitar la extracción de información.

El proceso permite identificar una entidad que va agrupada en una expresión regular, aumentando la certeza de que se ha hecho una clasificación con una precisión más elevada. Se puede llegar a aumentar la precisión de que todas las entidades que se capturan coincidan con determinado patrón, son del mismo tipo.

Aunado a esto, el hacer adaptable el sistema al uso de expresiones ahorra los reentrenamientos constantes del modelo de reconocimiento.

El uso de las expresiones regulares se considera para patrones individuales de una persona ya que es susceptible a patrones definidos y no llegan a ser reglas que puedan repetirse en cada caso.

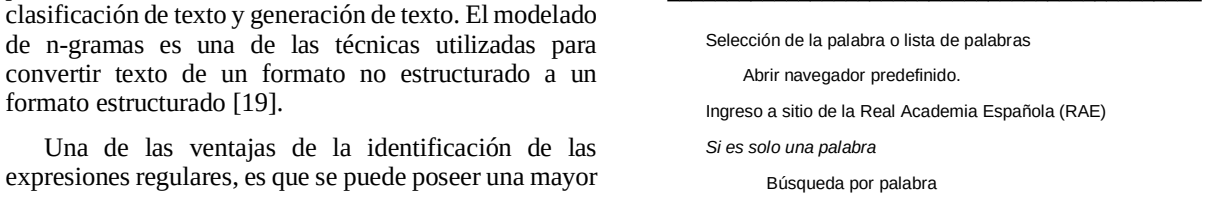
Es importante destacar que los n gramas son susceptibles a palabras con: caracteres extras, repetición de letras, errores ortográficos entre otros, por eso se hace énfasis en la revisión y correcciones como pre-procesamiento de los textos para posteriormente hacer su análisis y extracción de información.

Mediante la creación de un proyecto de reconocimiento de entidades con nombre personalizado, se pueden etiquetar datos de forma iterativa, entrenar, evaluar y mejorar el rendimiento del modelo, ayudando al mapeo y extracción de datos.

La obtención de bolsa de palabras por el procesamiento de cada memoria permitirá realizar una actualización recurrente dentro del perfil del usuario. Y al segregar a usuarios de la misma zona geográficas se podrá detectar regionalismos dentro de un mismo idioma e incorporar estas nuevas palabras al sistema para mejorar los niveles de precisión.

Para todas aquellas palabras desconocidas se optó por implementar un modelo de extracción de datos de la web que ingresa a la página oficial de la Real Academia de la Lengua Española (RAE), buscando la palabra deseada y extrayendo la implementación de la palabra en el español. El código mostrado en la Figura 8 ejemplifica la secuencia de pasos para realizar esta tarea.

La RAE, es una institución cultural dedicada a la regularización lingüística entre el mundo hispanohablante [20].



Extracción de categorías del discurso de las palabras

Extracción de la entidad.

Si no es solo una palabra

Recorrido de toda la lista de palabras.

Búsqueda por palabra

Extracción de categorías del discurso de las palabras

Extracción de la entidad.

Generación de bolsa de palabras.

Incorporación de palabras nuevas a la bolsa de palabras del perfil del usuario.

Recorrido de todas las frases dentro del texto.

Extracción de Tokens por frase.

Clasificación de partes de discurso

Clasificación de entidades.

Cerrar navegador.

Fig. 8. Seudocódigo de busqueda de categorías y entidades por medio de Spacy.

En la Figura 9 se muestra el proceso de sincronización de contenido multimedia con la información obtenida de cada una de las palabras al ir recorriendo cada una de ellas de izquierda a derecha.

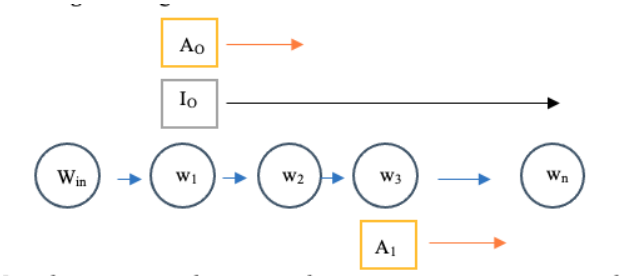


Fig. 9. Sincornización de elementos con narrativa.

Explicado más a detalle, los algoritmos deberán ser capaces de generar un preprocesamiento y llevar a cabo el análisis del contenido dentro del recuerdo antes de comenzar de que la persona comience la narrativa, se deben estimar los tiempos de duración de los elementos a aparecer dentro del recuerdo y de manera automática los llamados.

Se puede hacer una comparación al proceso de separar la oración, que se convertirá de discurso escrito a oral en palabras individuales y considerarlas como estados independientes de otros dentro de un arreglo secuencial. Al visitar cada uno de estos estados individualmente, y de manera ordenada se puede llamar a los eventos que contienen el estado de manera paralela a otros eventos generados por otros estados o pausar un evento que inicio en un estado qx. de la lista de palabras que conforman una frase como en la Figura 10.

Fig. 10. Diagrama secuencial de una frase

Nota: cada uno de los estados “q” hace referencia a una palabra de una frase, siendo el estado $q_n$la palabra final o el signo especial “.” que indica el final de la oración, esto puede variar si el diseño contempla a los signos de puntación como estados. Dicha secuencia se aprecia en la Figura 11.

El llamado de los elementos multimedia debe ser un proceso paralelo asíncrono ejemplificado en la Figura 10. De tal manera que no dependan el llamado de un elemento de la aparición de otro si no del contexto y el momento en el que se encuentra la historia así pudiendo complementar los contenidos de imágenes, audio o video sin requerir que estén todos u otro elemento, como se muestra en la Figura 11.

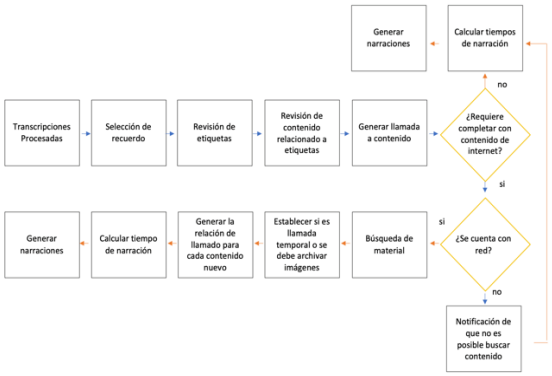


Fig. 11. Diagrama de flujo de procesamiento de recuerdo

A partir de este punto al generar la narrativa del recuerdo, una imagen puede ser mostrada en la pantalla del dispositivo sin pausar la narrativa y a su vez, otro evento puede solicitar un efecto sonoro o arrancar de una pista audiovisual sin opacar la voz principal o parar los otros dos eventos, siendo esto calculado por el sistema.

Los eventos se pueden poner a dormir o concluir según indique el arreglo la configuración predefinida para cada palabra de acuerdo con el texto. Por lo cual, la manera que se comparta un elemento relacionado a una palabra en texto va a variar al contexto de la historia y a los otros elementos que se encuentran dentro de ella.

La opción de complementar las historias con elementos fuera de los archivos, el usuario o de los archivos generales cargados al momento queda a discreción de usuario. De igual modo queda a discreción pues no se requiere la función del llamado a elementos multimedia para la narración por defecto.

Al activarse la función de llamada a elementos multimedia, el algoritmo será capaz revisar con el primer

preprocesamiento los elementos con los que cuenta en su base de datos interna del dispositivo que tengan una relacionados al recuerdo, complementarlo con elementos del repositorio general o permitir una descarga de elementos de la red tal como imágenes, videos, elementos de audio, etc., para complementar los vacíos de elementos.

Si el dispositivo cuenta con una conexión a la red y estos vacíos son de carácter universal como ciudades, países, lugares públicos, por medio de algoritmos de raspado de web ("scraping" en inglés) se podrá descargar los elementos que complementen la historia y añadirlos a una base de datos temporal de donde serán llamados. Si, por el contrario, son de carácter privado no se llamará a ningún archivo o elemento haciendo referencia a fotografía personales con las que el sistema no cuente.

El raspado web es la práctica de recopilar datos a través de cualquier otro medio que no sea un programa que interactúa con una API también considerado a través de un humano que usa un navegador web. Comúnmente se logra este proceso al escribir un programa automatizado que consulta un servidor web determinado, solicita los datos ayudándose de identificadores de lenguaje HTML y de otros archivos que compongan la página o páginas web seleccionadas y luego analiza los datos para extraer la información necesaria [21].

El web scraping abarca una amplia variedad de técnicas y tecnologías de programación, como el análisis de datos y la seguridad de la información [21].

RESULTADOS PRELIMINARES.

Como puede apreciarse, existen numerosas tareas y múltiples aproximaciones para el procesamiento del lenguaje natural y la implementación de contenido multimedia en las historias.

Los primeros desarrollos de la aplicación se enfocaron en realizar el proceso de transcripciones en tiempo real de los recuerdos, pero la configuración del uso de librerías generó limitantes. La primera es que la aplicación, al detectar una pausa el relato, detenía por completo la transcripción y se debía iniciar de nuevo, por ende, se optó por la captura de un audio en formato MP4 con la aplicación y por medio de código escrito en Python generar la conversión a WAV para poder aplicar la librería de Google de conversión de discurso hablado a escrito.

La preservación del formato de audio permitirá medir la precisión en el proceso de conversión del discurso oral al escrito en futuros trabajos ("speech to text", en inglés). Es entonces que se limitó la duración de las grabaciones a un máximo de 30 min para los archivos de audio.

Las transcripciones y los textos ingresados como recuerdos se convierten a texto plano por medio de la librería de Python Tika, la cual detecta el formato en el que se encuentra y lo convierte a texto plano. Uno de los inconvenientes detectados fue le necesidad de conexión a internet y generar una conexión con el servidor que aloja Tika, lo cual requiere un análisis a mayor detalle para reforzar la seguridad en el manejo de datos personales.

En la parte de sincronización al utilizar un estándar de 150 palabras por minutos y una relación de 7.5 ms por silaba, con un espacio entre las palabras de 8 ms de espera, se logró mostrar el material en forma línea, pero en algunos momentos existía un retraso en lo que sé que se apreciaba con la grabación que contenía la conversión del texto plano a discurso oral.

La mayoría de las pruebas de procesamiento se han realizado con cuentos cortos y fabulas que no contienen material similar entre sí para observar el proceso de clasificación de cada texto por separado y consolidados en un solo documento.

Las primeras fases no tuvieron los resultados deseados al momento de implementar SpaCy, pero tras algunos ajustes se logró mejorar la identificación de identidades y partes del discurso para completar el esquema de guardar la información de la palabra, por medio de un arreglo dividido en áreas de implementación. A pequeña escala permitió llamar los eventos de manera separada, aunque todavía quedan pendiente las pruebas de los tres elementos al mismo tiempo. Todas estas pruebas se realizaron por medio de código de Python, fuera de la aplicación de Flutter, por tanto, quedan pendientes las pruebas dentro de la aplicación.

CONCLUSIONES Y FUTUROS TRABAJOS.

En el presente trabajo (actualmente en investigación) se plantea el desarrollo de una estrategia que permite la recolección de historias, el análisis de sus contenidos, la extracción de información y la sincronización con contenido multimedia almacenado en bases de datos.

La identificación y extracción con técnicas de procesamiento natural es susceptible a la interpretación en el proceso de generación de transcripciones del discurso hablado al escrito por la velocidad y modo de pronunciación.

El cálculo en los tiempos de aparición por palabra con una relación de 7.6 ms por silaba permitió sincronizar de manera funcional los tokens que se sincronizaron con contenido.

Gracias a la realización de este trabajo se pudieron implementar y obtener resultados en análisis de textos junto con la sincronización, aunque limitada de los recuerdos.

En trabajos futuros se implementará un analizador de imágenes para asegurar que en el proceso de la generación de repositorios de imágenes de manera automática con contenido de la red sea confiable. Se plantea entrenar un modelo que asegure que la imagen descargada se encuentre relacionada al token que se busca descargar y de esta manera, evitar imágenes incorrectamente etiquetadas en la red para estandarizar folios dentro del sistema.

AGRADECIMIENTOS

J.E.Z.B. agradece al CONACYT el apoyo brindado para la realización de estudios de Maestría. Los autores agradecen al Tecnológico Nacional de México/ IT León las facilidades brindadas para la realización de este proyecto.

REFERENCIAS

[1] M. d. C. García Peña, L. M. Gutiérrez Robledo , P. A. Roa Rojas y A. Martínez Ruiz, «La Enfermedad de Alzheimer y otra Fig. 8. Seudocódigo de busqueda de categorías y entidades por medio de Spacy.s demencias como problema nacional de salud,» Academia Nacional de Medicina de México (ANMM), México, 2017.

[2] F. A. ESPAÑA, «tp://www.alzfae.org/,» 13 04 2015. [En línea]. Available: http://www.alzfae.org/actualidad/mundo-alzheimer/1158/30-congreso-annual-de-alzheimrer-europe-la-demencia-en-un-mundo-cambiante-congreso-virtual-celebrado-entre-20-22-de-octubre-de-2020. [Último acceso: 02 10 2022].

[3] S. d. S. d. México, «https://www.gob.mx/,» Secretaría de Salud , 05 Octubre 21. [En línea]. Available: https://www.gob.mx/salud/es/articulos/enfermedad-de-alzheimer-demencia-mas-comun-que-afecta-a-personas-adultas-mayores?idiom=es. [Último acceso: 01 Octubre 22].

[4] N. Justel, M. Psyrdellis y E. Ruetti, «Modulación de la memoria emocional: una revisión de los principales factores que afectan los recuerdos,» Suma Psicológica, vol. 20, nº 2, 2013.

[5] E. Echeburúa y P. J. Amor, «Memoria traumática: estrategias de afrontamiento adaptativas e inadaptativas,» Terapia psicológica, vol. 37, nº 1, p. 2, 2019.

[6] D. A. Muñoz González, «La estimulación cognitiva como estrategia para la atención psicogerontológica a los adultos mayores con demencia,» Revista Cubana de Salud Pública, vol. 44, nº 3, 2018.

[7] C. d. I. p. D. e. S. Pública, «Instituto Nacional de Salud Pública,» Instituto Nacional de Salud Pública, 21 Septiembre 2020. [En línea]. Available: https://www.insp.mx/avisos/hablemos-de-demencia#sup3. [Último acceso: 04 Octubre 2022].

[8] T. Álvarez Cisneros, S. Torres Castro, B. Mena Montes y . N. M. Torres Carrillo, «Factores de protección para la demencia en adultos mayores en México: ¿dónde estamos? En el libro: Gerontología para la vida activa.,» Género y salud en cifras, vol. 15, nº 3, p. 13, 2016.

[9] M. d. L. Reyna Carrizales, M. V. González Rubio, F. J. López Esqueda y G. R. González González , «Demencias, Una visión panorámica,» de V Simposio de Medicina Geriátrica, San Luis Potosí, 2014.

[10] A. Taylor, B. Santorini y M. Marcus, «The Penn Treebank: An overview,» Springer, Dordrecht, 2003.

[11] D. Jurafsky y . J. H. Martin, Hidden Markov Models, Palo Alto: Stanford, 2021.

[12] R. Cabeza Ruiz, «Aplicando los Campos Aleatorios Condicionales en la Segmentación de Textos por Idiomas,» de Conditional Random Fields in Text Segmentation by Language, Holguin, 2019.

[13] S. K. Siencnik, «Adapting word2vec to named entity recognition,» Proceedings of the 20th nordic conference of computational linguistics, vol. 20, p. 239–243., 2015.

[14] Google, «flutter.dev,» Google, 01 Octubre 2022. [En línea]. Available: https://flutter.dev/. [Último acceso: 01 octubre 2022].

[15] A. M. M. S. B. Taylor, «The Penn Treebank: An Overview.,» Springer, Dordrecht., Vols. %1 de %2In: Abeillé, A. (eds) Treebanks. Text, Speech and Language Technology, vol 20. , nº 20, p. 18, 2003.

[16] G. Wolf, E. Ruiz, F. Bergero y E. Meza, FUNDAMENTOS DE SISTEMAS OPERATIVOS, Ciudad de México: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2015.

[17] Y. Goldberg, Neural Network Methods for Natural Language Processing, Toronto: Graeme Hirst, University of Toronto, 2017.

[18] Spacy, «Spacy.io,» 01 Octubre 2022. [En línea]. Available: https://spacy.io/usage/spacy-101. [Último acceso: 02 Octubre 2022].

[19] mathworks, «mathworks,» 01 Octubre 2022. [En línea]. Available: https://la.mathworks.com/discovery/ngram.html. [Último acceso: 01 Octubre 2022].

[20] R. A. Española, « Real Academia Española,» 01 Octubre 2022. [En línea]. Available: https://www.rae.es . [Último acceso: 01 Octubre 2022].

[21] R. Mitchell, Web Scraping with Python Collecting. Data from the Modern Web, Sebastopol: 2015, Reilly Media.

Universidad Nacional Autónoma de México
Facultad de Estudios Superiores Cuautitlán
Departamento de Ingeniería

CEIAAIT

Otorgan la presente

Constancia

A: Jaime Zavala Barrios, Víctor Zamudio Rodríguez and Carlos Lino Ramírez

Por su participación como ponente del tema

Evaluación e implementación de estrategias de asociación de voz o texto con contenido multimedia para la generación de repositorios digitales mediante técnicas de Inteligencia Artificial.

En el **5° Congreso Estudiantil de Inteligencia Artificial Aplicada a la Ingeniería y Tecnología (CEIAAIT)**, realizado del 9 al 11 de noviembre de 2022.

"POR MI RAZA HABLARÁ EL ESPÍRITU"
Cuautitlán Izcalli, Estado de México, noviembre de 2022.

Folio: 202200422

Dr. David Quintanar Guerrero
Director