
Desarrollo de una herramienta basada en
lenguaje de apoyo a la terapia basada en
reminiscencia
A language-based tool to support reminiscence
therapy



Trabajo de Fin de Grado
Curso 2023–2024

Autor
Marta Vicente Navarro

Director
Gonzalo Mendez Pozo

Grado en Doble grado en Ingeniería Informática y
Matemáticas

Facultad de Informática
Universidad Complutense de Madrid

Desarrollo de una herramienta basada en
lenguaje de apoyo a la terapia basada en
reminiscencia

A language-based tool to support
reminiscence therapy

Trabajo de Fin de Grado en Doble grado en Ingeniería
Informática y Matemáticas

Autor

Marta Vicente Navarro

Director

Gonzalo Mendez Pozo

Convocatoria: *Junio* 2024

Grado en Doble grado en Ingeniería Informática y
Matemáticas

Facultad de Informática

Universidad Complutense de Madrid

6 de abril de 2024

Dedicatoria

A Mamá

Agradecimientos

Resumen

Desarrollo de una herramienta basada en lenguaje de apoyo a la terapia basada en reminiscencia

La enfermedad de Alzheimer es una condición neurodegenerativa que afecta las funciones cognitivas, la memoria, el pensamiento y el comportamiento. Su llegada supone un cambio significativo en la vida de quienes la padecen y en su entorno. Actualmente, se estima que en España 900.000 personas sufren esta y otras formas de demencia, y se proyecta que los casos se duplicarán para el año 2050. Por lo tanto, es de vital importancia desarrollar técnicas que puedan ralentizar el avance de la enfermedad. Aunque no es reversible, existen terapias que pueden mejorar la calidad de vida tanto del paciente como de sus seres queridos.

La terapia de reminiscencia es una modalidad terapéutica que se enfoca en ayudar a las personas a recordar y compartir sus experiencias y recuerdos pasados, especialmente aquellos relacionados con eventos significativos en sus vidas. Aunque es comúnmente empleada con personas mayores, también puede resultar efectiva en otros grupos de edad.

Este proyecto se centra principalmente en el desarrollo de un ChatBot que interactúe con el paciente, recopilando la información necesaria para construir una historia de vida y las imágenes asociadas. Para lograrlo, se ha clonado la API de Gemini y se ha entrenado con datos personalizados, de manera que genere historias de vida específicas y pertinentes.

Este enfoque promete ser una herramienta valiosa para mejorar la calidad de vida de las personas afectadas por Alzheimer y demencias similares. Si necesitas más asistencia en relación a este proyecto, no dudes en decírmelo.

Palabras clave

Reminiscencia, Chatbot, historia de vida, alzhéimer

Abstract

A language-based tool to support reminiscence therapy

Alzheimer's disease is a neurodegenerative condition that affects cognitive functions, memory, thinking, and behavior. Its onset marks a significant change in the lives of those affected and their surroundings. Currently, it is estimated that 900,000 people in Spain suffer from this and other forms of dementia, and it is projected that cases will double by the year 2050. Therefore, it is crucial to develop techniques that can slow down the progression of the disease. Although it is not reversible, there are therapies that can improve the quality of life for both the patient and their loved ones.

Reminiscence therapy is a therapeutic approach focused on helping individuals remember and share their past experiences and memories, especially those related to significant events in their lives. While commonly used with older individuals, it can also be effective with other age groups.

This project primarily focuses on the development of a ChatBot that interacts with the patient, gathering the necessary information to construct a life story along with associated images. To achieve this, the Gemini API has been cloned and trained with customized data, enabling it to generate specific and relevant life stories.

This approach holds promise as a valuable tool for enhancing the quality of life for individuals affected by Alzheimer's and similar dementias.

Keywords

reminiscence, chatbot, life story, Alzheimer's

Índice

1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	2
1.3. Plan de trabajo	2
1.4. Estructura de la memoria	2
Introduction	3
1.5. Motivation	3
1.6. Goals	4
1.7. Work Plan	4
2. Estado de la Cuestión	5
2.1. Enfermedad de Alzheimer	5
2.2. Historia de vida	6
2.3. Terapias de reminiscencia	6
2.4. Trabajo de Cristina	7
3. Objetivos y planteamiento de la solución	9
3.1. APIs	9
3.2. Software y herramientas	9
3.2.1. Entorno de ejecución	9
3.2.2. Librerías utilizadas	9
3.2.3. RDF	9
3.2.4. Uso de RDFLib en Python	9
3.2.5. Arquitectura del sistema	10
3.3. Procesamiento del lenguaje	10
3.3.1. Bibliotecas de Python	11
3.3.2. APIs de procesamiento del lenguaje natural	11
3.3.3. Gemma	13
3.3.4. Gemini	13
3.4. Almacenamiento de la información	13
3.4.1. Conclusión final	13

4. ChatBot de apoyo a la terapia de Reminiscencia	15
4.1. Desarrollo de las funciones	15
5. Interfaz y resultados finales	17
6. Conclusiones y Trabajo Futuro	19
Conclusions and Future Work	21
A. Título del Apéndice A	23
B. Título del Apéndice B	25

Índice de figuras

Índice de tablas

Introducción

“Hay enfermos incurables, pero ninguno incuidable”
— Francesc Torralba

El avance en el campo de la medicina en el último siglo ha permitido un notable aumento en la esperanza de vida a nivel mundial. Sin embargo, este alargamiento de la vida también ha traído consigo un aumento en las enfermedades relacionadas con la vejez, como el Alzheimer. A medida que vivimos más tiempo, enfrentamos un mayor riesgo de desarrollar estas condiciones.

1.1. Motivación

A pesar de los avances en la investigación médica, todavía no se ha encontrado una cura definitiva para el Alzheimer. La mayoría de los tratamientos se centran en aliviar los síntomas y ralentizar la progresión de la enfermedad. Entre estos tratamientos, la terapia de reminiscencia ha surgido como una opción no farmacológica destacada.

La terapia de reminiscencia se centra en estimular los recuerdos del pasado del paciente, lo que puede tener beneficios significativos en su bienestar social, mental y emocional. Ayuda a las personas a recordar y compartir experiencias pasadas, lo que puede ser reconfortante y estimulante, especialmente para aquellos que enfrentan el desafío de la pérdida de memoria asociada con el Alzheimer.

Durante el período académico 2023-2024, se desarrolló la aplicación YayoBot con el objetivo de facilitar a los terapeutas la realización de terapias basadas en reminiscencia, simplificando y agilizando el proceso. Este desarrollo parte desde cero y busca crear un chatbot funcional y útil, capaz de mantener una conversación útil con el paciente para extraer la información necesaria.

Este Trabajo de Fin de Grado tiene como propósito asistir a terapeutas, familiares o amigos de pacientes con demencia en la obtención del material necesario para llevar a cabo terapias de reminiscencia. Se busca mejorar la eficacia de estas terapias y, en consecuencia, la calidad de vida tanto de los pacientes como de sus familiares.

1.2. Objetivos

Este trabajo busca desarrollar un chatbot que permita a los terapeutas aplicar terapia de reminiscencia a sus pacientes. En concreto, se centra en la primera etapa de búsqueda de la información con el objetivo de, más adelante generar las historias de vida y aplicar la terapia.

Para cumplir este objetivo general, se abordan los siguientes objetivos específicos:

- Desarrollar un primer chatbot básico capaz de realizar preguntas predefinidas y almacenar las respuestas de forma eficiente.
- Estudiar las técnicas modernas de procesamiento del lenguaje como los LLMs, bibliotecas como NLTK y spaCy. Además, de las diferentes APIs de para el desarrollo de chatbots.
- Mejorar el primer chatbot haciendolo más inteligente y capaz de analizar las respuestas, identificar la información omitida y hacer preguntas específicas para obtener la información faltante.
- Hacer una versión final del chatbot que sea capaz de analizar las repuestas y sepa tirar del hilo. También que sea capaz de generar preguntas adecuadas.

1.3. Plan de trabajo

El plan de trabajo para obtener alcanzar los objetivos desarrollados anteriormente será:

1. Desarrollar la primera versión del chatbot antes del 15 de Noviembre, habiendo trabajado para entonces la introducción de la memoria.
2. Tener para finales de Febrero, la segunda versión desarrollada así como un 85 % de la memoria del proyecto, principalmente los capítulos centrales.
3. Desarrollar para Mayo la versión final. Tener el código cerrado y la memoria con todas las conclusiones, resultados y trabajo futuro.

Para llevar el control de versiones utilizaremos el repositorio de github:
<https://github.com/NILGroup/TFG-2324-ChatbotCANTOR>.

1.4. Estructura de la memoria

Introduction

“There are incurable patients, but none that are beyond care.”
— Francesc Torralba

In this first chapter, the reasons that have led me to undertake this project and the objectives sought to be achieved from the outset are presented.

1.5. Motivation

Currently, memory loss is a problem that affects a large portion of the population, from individuals with mild cognitive impairment to those suffering from more severe forms of dementia, such as Alzheimer’s. This condition not only diminishes the patient’s quality of life but also impacts the well-being of their family and loved ones. In Spain alone, according to the Spanish Society of Neurology, there are 800,000 people afflicted by this disease.

Non-pharmacological techniques have shown highly positive results in preserving memory, maintaining cognitive abilities, and retaining memories, thereby contributing to delaying cognitive decline. Among these, approaches based on reviewing the individual’s own Life History have proven to be highly effective. This involves the patient, even one suffering from dementia, personally recording the most significant experiences, people, and places in their life. The aim is to stimulate the patient to discuss various topics, past events, and historical occurrences. It has been found that these exercises help preserve skills such as reasoning, self-esteem, confidence, and social abilities. Up until now, therapists implementing reminiscence therapies have manually created patients’ life stories and relied on printed documents to prepare the sessions.

During the academic period of 2023-2024, the ReminiscenceBot application was developed with the purpose of facilitating therapists in conducting reminiscence-based therapies, streamlining and simplifying the process. This development starts from scratch and aims to create a functional and useful chatbot capable of engaging in a productive conversation with the patient to extract the necessary information.

This Bachelor’s Thesis arises with the intention of assisting therapists, family members, or friends of dementia patients in obtaining the necessary material to apply reminiscence therapies on the patient. Thus, the goal is to enhance the effectiveness of these therapies and improve the quality of life for both the patients and their

families.

1.6. Goals

This work aims to develop a chatbot from scratch using the Bard API that is capable of:

1. Developing an initial basic chatbot capable of responding to predefined questions and storing answers.
2. Enhancing the previous chatbot to make it more intelligent and capable of analyzing responses, identifying omitted information, asking specific questions to obtain missing information, etc.
3. Creating a final version of the chatbot that can analyze responses and engage in a meaningful conversation. It should also be able to generate appropriate questions.

1.7. Work Plan

The work plan to achieve the previously outlined objectives will be as follows:

1. Develop the first version of the chatbot by November 15th, with a focus on drafting the project introduction.
2. By the end of February, have the second version developed, along with 85
3. Aim to complete the final version by May. Ensure that the code is finalized and the report includes all conclusions, results, and future work.

To manage version control, we will use the GitHub repository at: <https://github.com/NILGroup/TFG2324-ChatbotCANTOR>.

Estado de la Cuestión

2.1. Enfermedad de Alzheimer

El Alzheimer, una enfermedad neurodegenerativa progresiva y devastadora, afecta a millones de personas en todo el mundo. Se caracteriza por la pérdida gradual de la memoria y otras funciones cognitivas, lo que eventualmente conduce a la incapacidad para llevar a cabo las actividades diarias más básicas. Su curso clínico se divide en varias etapas distintas, cada una con sus propias características y desafíos.

- Etapa Temprana o Leve:

En esta fase inicial, los síntomas pueden pasar desapercibidos o atribuirse a simples descuidos. La persona afectada puede experimentar dificultades para recordar nombres, eventos recientes o encontrar las palabras adecuadas en conversaciones. A pesar de estos desafíos, generalmente conservan la capacidad de realizar tareas cotidianas con cierta independencia. Sin embargo, es posible que comiencen a perder interés en actividades previamente disfrutadas.

- Etapa Intermedia o Moderada:

A medida que la enfermedad progresa, los síntomas se vuelven más evidentes y problemáticos. La pérdida de memoria se vuelve más pronunciada, con dificultades para reconocer a familiares y amigos cercanos. Además, pueden surgir problemas de orientación en tiempo y espacio, lo que puede resultar en desorientación incluso en entornos familiares. Las habilidades de comunicación también se ven afectadas, con dificultades para seguir conversaciones o expresar pensamientos de manera coherente.

- Etapa Avanzada o Severa:

En esta etapa tardía, el Alzheimer alcanza su punto más devastador. La pérdida de memoria es profunda y completa, con una incapacidad para recordar incluso eventos recientes o reconocer caras familiares. La persona afectada puede experimentar cambios significativos en la personalidad y el comportamiento, volviéndose agitada, ansiosa o incluso agresiva en ocasiones. La capacidad para realizar actividades básicas de la vida diaria, como vestirse o alimentarse, se ve seriamente comprometida, y la supervisión constante se vuelve esencial.

El desarrollo del Alzheimer se asocia con cambios físicos y químicos en el cerebro, incluida la acumulación de placas de proteínas llamadas beta-amiloide y ovillos neurofibrilares compuestos de proteína tau. Estas alteraciones provocan la muerte de células nerviosas y la disrupción de las conexiones entre ellas, lo que resulta en la progresiva pérdida de funciones cognitivas y conductuales.

Aunque no existe cura para el Alzheimer, existen tratamientos farmacológicos y terapias no farmacológicas que pueden ayudar a aliviar los síntomas y mejorar la calidad de vida de los pacientes en las etapas tempranas y moderadas de la enfermedad. Sin embargo, a medida que avanza la enfermedad, el enfoque se centra más en la atención y el apoyo integral, tanto para la persona afectada como para sus cuidadores y familiares.

El Alzheimer es una enfermedad desgarradora que afecta no solo a quienes la padecen, sino también a sus seres queridos. La investigación continua es fundamental para comprender mejor sus mecanismos subyacentes, desarrollar tratamientos más efectivos y, en última instancia, encontrar una cura para esta enfermedad que roba la memoria y la identidad de quienes la sufren.

2.2. Historia de vida

Las Historias de Vida son registros detallados de los aspectos más relevantes de la vida de un paciente, o de personas significativas para él. Son especialmente importantes en las primeras fases del Alzheimer, cuando la memoria aún es relativamente intacta. Estas historias proporcionan dignidad al paciente y permiten a quienes lo rodean conocer mejor su identidad, a medida que las pérdidas de memoria comienzan a ser significativas.

La creación de estas historias es vital para preservar la identidad del paciente. Pueden incluir detalles sobre la familia, la carrera profesional, los viajes y otros aspectos importantes de la vida del individuo. La elaboración puede realizarse de diversas formas, como escribir un libro, crear collages de fotos, producir una película o usar una caja de memoria".

Es fundamental que estas historias reflejen la perspectiva personal del paciente, incluyendo emociones, sentimientos e interpretaciones. No se trata simplemente de relatar hechos cronológicos, sino de capturar la esencia única de la persona.

Un terapeuta suele encargarse de recopilar y estructurar los eventos importantes de la vida del paciente para crear la Historia de Vida. La estructura puede variar según las necesidades del paciente y los objetivos terapéuticos.

La participación de familiares o conocidos puede facilitar la evocación de recuerdos y enriquecer la Historia de Vida. Además, fortalece la comunicación entre el paciente y sus seres queridos, facilitando el proceso terapéutico.

2.3. Terapias de reminiscencia

La reminiscencia, según la definición de la CEAFA (Confederación Española de Asociaciones de Familiares de personas con Alzheimer y otras demencias), es una

técnica que busca evocar recuerdos en las personas, especialmente aquellos relacionados con eventos importantes de su vida.

En las terapias de reminiscencia se emplean diversos materiales, como fotografías, vídeos, noticias de periódicos, audios y objetos significativos, con el objetivo de estimular el recuerdo y activar la memoria a través de las emociones que despiertan estos elementos en el paciente.

Además de estimular los cinco sentidos para provocar el recuerdo, también se utiliza la narración de historias conocidas por el paciente. Por lo tanto, es crucial conocer las experiencias pasadas del individuo afectado para adaptar los materiales utilizados en las terapias.

La participación de personas cercanas al paciente, como familiares y cuidadores, en ejercicios de reminiscencia puede mejorar significativamente su calidad de vida.

Es recomendable construir relatos basados en las historias de vida del paciente para estos ejercicios de memoria. Las terapias de reminiscencia se consideran como parte de un proceso más amplio en el que el individuo intenta recuperar y vincular recuerdos que abarcan gran parte de su vida.

2.4. Trabajo de Cristina

En el curso 2021-2022, la compañera María Cristina Alameda Salas, en su trabajo de fin de grado, Generación de historias de vida usando técnicas de Deep Learning, desarrolló un sistema basados en técnicas de Deep Learning que de soporte a la generación de historias de vida. Partiendo de unos datos de entrada en forma de datos estructurados de tipo biográfico, ese trabajo permite la construcción de un sistema de generación de lenguaje natural, transformador de los datos de entrada a un escrito fluido y coherente, que abarque la representación de los datos de partida de manera completa, sin incorrecciones y lo más cercana posible a una redacción humana. Nuestro objetivo ahora sería el desarrollo de un programa que interactuara con el usuario y nos permitiera obtener toda esa información bibliográfica que da lugar a las historias de vida.

Objetivos y planteamiento de la solución

En este capítulo, se abordarán los objetivos y requisitos esenciales del sistema propuesto en este trabajo, así como las restricciones que deben tenerse en cuenta. A continuación, se presentará el planteamiento inicial del sistema, incluyendo un análisis de la arquitectura general y la identificación de las herramientas necesarias para su desarrollo. Además, se llevará a cabo una evaluación exhaustiva de las diferentes opciones disponibles para la implementación del sistema, junto con las consideraciones que respaldan las decisiones de implementación adoptadas.

3.1. APIs

3.2. Software y herramientas

3.2.1. Entorno de ejecución

3.2.2. Librerías utilizadas

3.2.3. RDF

El Resource Description Framework (RDF) es un modelo estándar para describir recursos en la web semántica. En RDF, la información se representa en forma de tripletas, que consisten en un sujeto, un predicado y un objeto. Estas tripletas se pueden usar para modelar datos de una manera semántica y enlazada, lo que facilita su interpretación por parte de las máquinas.

En el Listado , se muestra un ejemplo de tripletas RDF en formato Turtle. Aquí, se describe que "Juan.^{es} amigo de "María"z ambos son de tipo "Persona".

3.2.4. Uso de RDFLib en Python

Para trabajar con tripletas RDF en Python, podemos utilizar la biblioteca RD-FLib. A continuación, se muestra un ejemplo de cómo crear y manipular un grafo RDF utilizando RDFLib:

`g = Graph()`

Uno de los beneficios clave de las tripletas RDF es su flexibilidad en la representación de datos. No están limitadas a un esquema predefinido, lo que significa que pueden adaptarse fácilmente a diferentes tipos de conversaciones y contextos terapéuticos. Esto permite que el chatbot capture una amplia gama de información de manera estructurada y significativa.

Las tripletas RDF promueven la interoperabilidad y la reutilización de datos al utilizar estándares abiertos y enlazados. Esto significa que la información almacenada en un formato RDF puede ser compartida, integrada y enriquecida con datos externos de manera más eficiente. En el contexto de la terapia de reminiscencia, esto podría traducirse en la capacidad de acceder a una amplia gama de recursos y experiencias compartidas por diferentes individuos o comunidades.

La estructura de tripletas RDF facilita la consulta y el análisis de datos, lo que permite al chatbot realizar búsquedas avanzadas, generar estadísticas y ofrecer recomendaciones personalizadas. Esto puede mejorar significativamente la experiencia del usuario al proporcionar respuestas relevantes y contextualizadas durante las sesiones de terapia de reminiscencia.

En conclusión, las tripletas RDF ofrecen una solución sólida y versátil para el almacenamiento de información en el contexto de la terapia de reminiscencia. Su capacidad para modelar semánticamente los datos, su flexibilidad en la representación, su interoperabilidad y su facilidad de consulta hacen que sean una opción ideal para el desarrollo de chatbots de ayuda a la terapia de reminiscencia. Al aprovechar estas ventajas, se puede mejorar la calidad y efectividad de la interacción entre el chatbot y los usuarios, contribuyendo así a mejorar su bienestar emocional y cognitivo.

3.2.5. Arquitectura del sistema

3.3. Procesamiento del lenguaje

El Procesamiento del Lenguaje Natural (NLP) es una rama de la informática que se enfoca en proporcionar a los sistemas informáticos la capacidad de entender y procesar textos y palabras habladas de manera similar a los seres humanos. Combina la lingüística computacional con modelos estadísticos, de machine learning y deep learning para permitir a las máquinas comprender el significado completo, la intención y el sentimiento detrás del lenguaje humano, ya sea en forma de texto o voz.

NLP impulsa una amplia gama de aplicaciones, desde la traducción automática hasta la respuesta a órdenes habladas, pasando por la atención al cliente mediante chatbots. Además, está ganando cada vez más popularidad, tanto a nivel particular como empresarial, médico, etc.

Hasta hace unos años, el procesamiento del lenguaje en Python se hacía utilizando diferentes bibliotecas como NLTK o Spacy. Sin embargo, en los últimos años se han desarrollado numerosos modelos de chatbots.

3.3.1. Bibliotecas de Python

3.3.1.1. NLTK

Python, un lenguaje de programación popular, cuenta con una amplia variedad de herramientas y bibliotecas para abordar tareas específicas de NLP. Una de las bibliotecas más destacadas es el kit de herramientas de lenguaje natural (NLTK), una colección de recursos de código abierto que facilita la creación de programas de NLP.

NLTK ofrece una amplia gama de funcionalidades, incluyendo el análisis de oraciones, segmentación de palabras, etiquetado gramatical, radicación y lematización, así como el reconocimiento de voz y la generación de lenguaje natural. También proporciona herramientas para el razonamiento semántico y la comprensión del texto. Esta biblioteca es ampliamente utilizada en la comunidad de NLP debido a su versatilidad y facilidad de uso en proyectos de procesamiento de lenguaje.

3.3.1.2. SpaCy

SpaCy es una biblioteca de procesamiento del lenguaje natural (NLP) en Python, lanzada en 2015 por Matt Honnibal. Ofrece soporte para más de 70 idiomas y cuenta con modelos preentrenados como BERT. Sus características incluyen tokenización, reconocimiento de entidades nombradas y análisis de dependencia. Comparada con NLTK, spaCy es más rápida y precisa, mientras que Gensim y TensorFlow/Keras complementan sus funcionalidades. Sus capacidades lingüísticas abarcan etiquetado de partes de discurso, análisis de dependencia, reconocimiento de entidades y más.

3.3.2. APIs de procesamiento del lenguaje natural

Las APIs de procesamiento del lenguaje son conjuntos de herramientas y servicios que permiten a los desarrolladores integrar capacidades de procesamiento del lenguaje natural (NLP) en sus aplicaciones y sistemas. Estas APIs ofrecen funcionalidades como análisis de sentimientos, reconocimiento de entidades nombradas, etiquetado de partes del discurso, traducción automática, resumen de texto, entre otros. Al utilizar una API de procesamiento del lenguaje, los desarrolladores pueden aprovechar las capacidades avanzadas de NLP sin necesidad de desarrollar desde cero sus propios algoritmos y modelos. Esto facilita la implementación de características de procesamiento del lenguaje en una amplia variedad de aplicaciones, desde chatbots hasta análisis de redes sociales y sistemas de recomendación.

Las APIs modernas suelen contar con todas las funcionalidades que contaban las bibliotecas de Python mencionadas anteriormente. En concreto, para el desarrollo de este proyecto se estudiaron las APIs que se presentan a continuación.

3.3.2.1. Bard

En primer lugar, para realizar la tarea de procesamiento de la información y análisis del lenguaje se utilizó la API de Bard. La API de Bard ofrece a los desarrolladores una herramienta poderosa para el procesamiento de lenguaje natural y

la generación de texto. Sus características clave incluyen la capacidad de producir texto coherente y relevante, adaptado al contexto específico de la aplicación. Además, permite una personalización flexible de la salida de texto para satisfacer las necesidades individuales de cada usuario. Gracias a su integración con modelos de lenguaje avanzados, la API de Bard es capaz de generar resultados de alta calidad en múltiples idiomas y estilos de escritura. Su interfaz fácil de usar y su versatilidad hacen que sea una opción ideal para una amplia gama de aplicaciones de NLP.

En diciembre de 2023, Google fortaleció la capacidad de Bard al incorporar Gemini Pro en inglés, brindando habilidades más avanzadas de comprensión, razonamiento, resumen y codificación. Posteriormente, en febrero de 2024, se anunció la expansión de Gemini Pro a más de 40 idiomas y se oficializó el cambio de nombre de Bard a Gemini, con lo que se tuvo que descartar el primer modelo del proyecto desarrollado en Bard, y tampoco se pudo volcar a Gemini ya que, por el momento, no está disponible en España.

3.3.2.2. Rasa

Entre las opciones que se barajaron para seguir desarrollando el proyecto se encuentra Rasa. Rasa es una plataforma de código abierto diseñada para el desarrollo de chatbots y asistentes virtuales conversacionales. Utilizando técnicas de procesamiento de lenguaje natural (NLP) y aprendizaje automático, Rasa permite a los desarrolladores crear sistemas de diálogo inteligentes y personalizados. Una de las principales ventajas de Rasa es su flexibilidad y personalización, ya que los desarrolladores tienen control total sobre el comportamiento y la lógica de sus chatbots. Además, Rasa proporciona herramientas robustas para la gestión del diálogo, la comprensión del lenguaje natural y la integración con otros sistemas. Sin embargo, una posible desventaja de Rasa es su curva de aprendizaje, ya que requiere un conocimiento sólido de NLP y aprendizaje automático para aprovechar al máximo su potencial. Además, debido a su naturaleza de código abierto, puede requerir más tiempo y recursos para implementar y mantener en comparación con otras soluciones comerciales.

3.3.2.3. GPT

La API de ChatGPT ofrece una solución altamente adaptable y fácil de usar para la generación de texto conversacional, proporcionando respuestas coherentes y contextualmente relevantes que imitan de manera natural las interacciones humanas. Su versatilidad la hace adecuada para una amplia gama de aplicaciones, desde chatbots de atención al cliente hasta asistentes virtuales y juegos de texto, y su escalabilidad y medidas de seguridad robustas garantizan un rendimiento confiable y protección de datos en cualquier entorno de implementación. Sin embargo para obtener el comportamiento que se necesitaba en este trabajo debía ser entrenada, y debido a las limitaciones hardware esto suponía una cantidad de tiempo inviable.

3.3.3. Gemma

Gemma es una herramienta de código abierto que ofrece capacidades avanzadas para la generación de texto y el procesamiento de lenguaje natural (NLP). Durante su evaluación, se probaron los modelos Gemma-2b y Gemma-7b, descargados de Hugging Face, en un entorno de sistema operativo Linux para optimizar su rendimiento. Aunque Gemma se destacó como la mejor opción disponible hasta el momento, se encontraron limitaciones relacionadas con el tamaño del modelo y algunas restricciones adicionales.

3.3.4. Gemini

Finalmente, se decidió utilizar la API de Gemini descartada en un primer momento por no estar disponible en España. Por tanto, para su uso, y en consecuencia para hacer funcionar el proyecto es necesario tener una VPN conectada a uno de los países en los que Gemini está disponible. Una vez conectado se ha de obtener una API KEY y configurar el proyecto adecuadamente. Durante el resto de uso del programa, sigue siendo necesario estar conectado a la VPN.

3.4. Almacenamiento de la información

3.4.1. Conclusión final

Capítulo 4

ChatBot de apoyo a la terapia de Reminiscencia

4.1. Desarrollo de las funciones

Capítulo 5

Interfaz y resultados finales

PyQt, wxPython y Kivy son opciones populares para la implementación de interfaces gráficas, cada una con sus propias ventajas y desventajas.

PyQt es conocido por su completo conjunto de widgets, lo que te permite crear interfaces gráficas complejas y altamente personalizadas. Sin embargo, puede tener una curva de aprendizaje más pronunciada debido a su complejidad y sintaxis más verbosa.

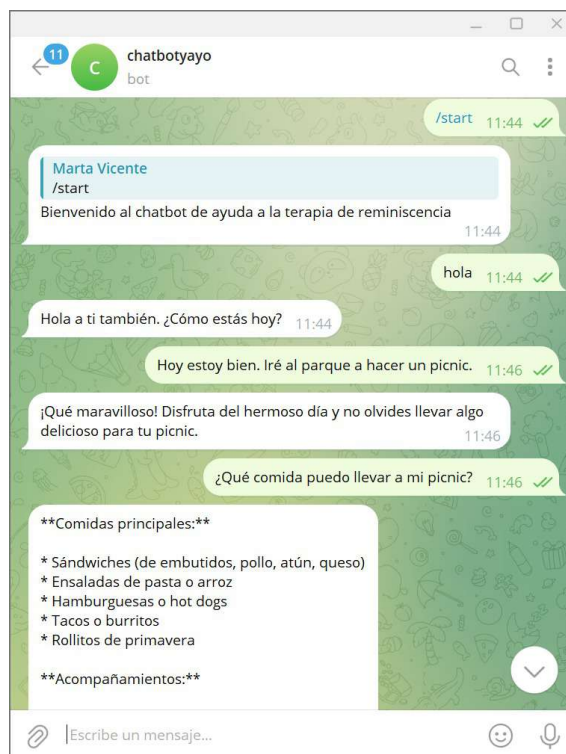
Por otro lado, wxPython ofrece una sintaxis más simple y fácil de entender, lo que puede ser beneficioso si estás empezando o prefieres un enfoque más directo. Aunque tiene menos widgets y funcionalidades avanzadas que PyQt, sigue siendo una opción sólida con una comunidad activa que proporciona soporte.

Kivy destaca por su diseño adaptable, diseñado para crear aplicaciones con interfaces gráficas que funcionan en una amplia gama de dispositivos. Utiliza un lenguaje de marcado declarativo que permite definir la interfaz de usuario de manera intuitiva y separada del código Python. Sin embargo, puede tener menos documentación y recursos disponibles en comparación con PyQt y wxPython.

Sin embargo, y pese a las grandes funcionalidades de todas estas alternativas nos hemos decantado por hacer una interfaz con telegram desarrollando nuestro propio chatbot usando rasa.

Para configurarlo, en primer lugar hay que ejecutar el programa `mibot.py` estando en telegram en la conversaa conversación de telegram el comando `/start` para comenzar la conversación y ya.

Con esto la interfaz se ve de la siguiente forma:



Capítulo 6

Conclusiones y Trabajo Futuro

Conclusiones del trabajo y líneas de trabajo futuro.

Antes de la entrega de actas de cada convocatoria, en el plazo que se indica en el calendario de los trabajos de fin de grado, el estudiante entregará en el Campus Virtual la versión final de la memoria en PDF.

Conclusions and Future Work

Conclusions and future lines of work. This chapter contains the translation of Chapter 6.

Apéndice A

Título del Apéndice A

Los apéndices son secciones al final del documento en las que se agrega texto con el objetivo de ampliar los contenidos del documento principal.

Apéndice	B
----------	----------

Título del Apéndice B

Se pueden añadir los apéndices que se consideren oportunos.

Este texto se puede encontrar en el fichero Cascaras/fin.tex. Si deseas eliminarlo, basta con comentar la línea correspondiente al final del fichero TFGTeXiS.tex.

*–¿Qué te parece desto, Sancho? – Dijo Don Quijote –
Bien podrán los encantadores quitarme la ventura,
pero el esfuerzo y el ánimo, será imposible.*

*Segunda parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

*–Buena está – dijo Sancho –; fírmela vuestra merced.
–No es menester firmarla – dijo Don Quijote–,
sino solamente poner mi rúbrica.*

*Primera parte del Ingenioso Caballero
Don Quijote de la Mancha
Miguel de Cervantes*

