



Universidad Politécnica  
de Madrid

**Escuela Técnica Superior de  
Ingenieros Informáticos**



Grado en Ingeniería Informática

Trabajo Fin de Grado

**Lectura Fácil: Aplicación web para  
evaluar adaptaciones automáticas**

Autor: Alejandro Ayuso Expósito

Tutor: MARÍA DEL CARMEN SUÁREZ DE FIGUEROA BAONZA

Madrid, junio - 2024

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

*Trabajo Fin de Grado*  
*Grado en Ingeniería Informática*

*Título: Lectura Fácil: Aplicación web para evaluar adaptaciones automáticas*  
*junio - 2024*

*Autor: Alejandro Ayuso Expósito*  
*Tutor: MARÍA DEL CARMEN SUÁREZ DE FIGUEROA BAONZA*  
*DIA*  
*Escuela Técnica Superior de Ingenieros Informáticos*  
*Universidad Politécnica de Madrid*

# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Motivación del proyecto . . . . .	5
1.2. Contexto del proyecto . . . . .	5
1.3. Objetivos . . . . .	6
1.4. Estructura del Documento . . . . .	7
<b>2. Trabajo relacionado y Estado de la Cuestión</b>	<b>8</b>
2.1. Trabajo Relacionado . . . . .	8
2.1.1. Metodología . . . . .	9
2.1.2. Métricas Utilizadas . . . . .	10
2.2. Métricas relacionadas con el Procesamiento de Lenguaje Natural . . . .	11
2.2.1. Características de las Métricas: . . . . .	11
2.2.2. Aplicaciones Prácticas: . . . . .	12
2.3. Avances Tecnológicos . . . . .	13
2.4. Desarrollo Web y Accesibilidad . . . . .	13
2.5. Tecnologías utilizadas . . . . .	14
2.5.1. Entornos de Programacion y Testing: . . . . .	14
2.5.2. Lenguajes de Programación y Servicios: . . . . .	15
2.5.3. Bibliotecas y Extensiones: . . . . .	15
<b>3. Desarrollo</b>	<b>16</b>
3.1. Diseño del Sistema . . . . .	16
3.1.1. Vision General: . . . . .	16
3.1.2. Componentes del Sistema: . . . . .	17
3.1.2.1. Frontend / Cliente: . . . . .	17
3.1.2.2. Backend / Servidor: . . . . .	18
3.2. Implementacion del Sistema . . . . .	19
3.2.1. Implementación del Frontend: . . . . .	19
3.2.2. Implementación del Backend: . . . . .	24
3.3. Despliegues del Sistema . . . . .	28
3.3.1. Despliegues en el Frontend . . . . .	28
3.3.2. Despliegues en el Backend . . . . .	28
3.4. Pruebas y Validación . . . . .	29
3.4.1. Pruebas Funcionales . . . . .	29
3.4.2. Pruebas de Usabilidad . . . . .	30
3.4.3. Pruebas de Rendimiento . . . . .	30
3.5. Resultados y Análisis . . . . .	31
3.5.1. Análisis Individual . . . . .	31
3.5.2. Análisis General Ponderado . . . . .	32
3.5.3. Resultados de las métricas . . . . .	33

3.5.3.1. BLEU . . . . .	33
3.5.3.2. ROUGE . . . . .	33
3.5.3.3. METEOR . . . . .	33
3.5.3.4. TER . . . . .	33
3.5.3.5. WER . . . . .	34
<b>4. Conclusiones y Líneas Futuras</b>	<b>35</b>
4.1. Impacto del trabajo . . . . .	35
4.2. Conclusiones personales . . . . .	36
4.3. Trabajo futuro . . . . .	37
<b>Bibliografía</b>	<b>38</b>
<b>A. Anexo</b>	<b>39</b>
A.1. Cuadros y tablas . . . . .	39

# Índice de Figuras

2.1. Página principal del proyecto anterior . . . . .	9
2.2. Visualización de resultados . . . . .	10
3.1. Nueva arquitectura del sistema . . . . .	17
3.2. Diagrama caso de uso del Frontend . . . . .	18
3.3. Flujo de trabajo backend . . . . .	19
3.4. Logica colores de metrics.component.ts . . . . .	20
3.5. Tabla de respuesta . . . . .	20
3.6. Cabecera Página de Respuesta . . . . .	21
3.7. Lógica de OnCalculatMetrics . . . . .	22
3.8. Nueva Página Principal . . . . .	23
3.9. Lógica Función CalculateMetrics() . . . . .	24
3.10 Lógica Función evaluateindividualquality() . . . . .	25
3.11 Lógica Función evaluatequality() . . . . .	26
3.12 Lógica Función uploadfile(), segundo endpoint . . . . .	27
3.13 Lógica Función calculatemetrics() . . . . .	28
3.14 Respuesta del servidor después de realizar el comando CURL . . . . .	29
3.15 Respuesta del servidor después de realizar el comando CURL . . . . .	30
3.16 Resultados obtenidos y análisis . . . . .	34

# Índice de Tablas

A.1. Tabla de tests y código asociado . . . . .	39
A.2. Tabla de resultado prueba de carga . . . . .	43

# Resumen

La misión de la Metodología de Lectura Fácil es democratizar la información, hacerla accesible. Todos deberían tener la capacidad de acceder a los textos, de entender documentos que de otro modo permanecerían cerrados, encriptados con terminologías que no son entendibles por personas no formadas en el tema, o escritos en un lenguaje en el cual las personas que tengan alguna dificultad en la comprensión lectora no logren comprender.

Esta metodología va más allá de lo ordinario; proporciona directrices claras y precisas que facilitan la transformación de textos complejos en documentos comprensibles sin sacrificar la integridad del contenido, estos documentos se convierten en herramientas poderosas para aquellas personas que enfrentan barreras intelectuales o lingüísticas. Los principales beneficiados de esta metodología y herramientas, son aprendices de nuevos idiomas, individuos cuyas oportunidades educativas no les prepararon para desentrañar textos sofisticados o personas con un déficit en la comprensión lectora. Esta iniciativa no solo rompe barreras comunicativas, sino que también fomenta una participación más activa y consciente en la sociedad, generando un impacto en documentos gubernamentales, noticias diarias, instrucciones prácticas y materiales educativos los cuales están siendo reescritos o adaptados para que sean claros, accesibles y comprensibles por todos.

En este contexto, el papel de la tecnología es fundamental, se esta automatizando el proceso de adaptación de textos para procesar textos a gran escala y garantizar la accesibilidad universal, esta automatización maximiza la eficiencia, estandariza la calidad y asegura que cada documento adaptado se convierta en un pilar de accesibilidad. Sin embargo, esta tarea es compleja, evaluar la efectividad de las adaptaciones automáticas supone un gran desafío. Simplemente simplificar el léxico no es suficiente, también requiere una reorganización meticulosa y profunda que no solo haga la información legible, sino entendible por el lector.

Este proyecto propone un nuevo avance, introduciendo la evaluación de uno o varios textos haciendo una comparación del texto original con su adaptación, los cuales son proporcionados por el usuario en un archivo CSV. Esta evaluación se realiza mediante el cálculo de varias métricas, encargadas de evaluar la similitud de las palabras usadas, junto con la calidad de estas y la coherencia y la similitud del texto adaptado al original. La Metodología de Lectura Fácil no es solo una herramienta; es una necesidad en nuestra lucha por una sociedad inclusiva. Automatizar este proceso abre nuevos horizontes para la eficiencia y la calidad, sin embargo, nos enfrentamos a desafíos que requieren soluciones innovadoras y creativas. Este trabajo presenta una solución al problema de la evaluación de textos adaptados, proporcionando una herramienta de utilidad para investigadores, educadores y alumnos que quieran comprobar la calidad de sus adaptaciones.

**Palabras Clave:**

- **Metodología de Lectura Fácil:** Un enfoque que simplifica y aclara la información, permitiendo que personas con dificultades de comprensión lingüística o barreras intelectuales accedan y entiendan textos complejos.
- **Democratizar la Información:** El proceso de hacer la información accesible y comprensible para todos, independientemente de sus capacidades lingüísticas o educativas, promoviendo igualdad en el acceso al conocimiento.
- **Integridad del Contenido:** Mantener la veracidad, la calidad y el valor integral del contenido original incluso después de su simplificación.
- **Barreras Intelectuales:** Obstáculos que impiden que personas con limitaciones cognitivas o dificultades de aprendizaje accedan plenamente a la información.
- **Automatización:** Uso de tecnología y sistemas automáticos para realizar tareas que tradicionalmente requieren intervención humana, en este caso, la adaptación de textos a un formato más simple.
- **Evaluación de Efectividad:** Proceso de determinar cuán efectivas son las adaptaciones automáticas en hacer los textos originales accesibles y comprensibles para el público objetivo.
- **Coherencia y Cohesión:** Aspectos de un texto que contribuyen a que sea lógico y fluido, facilitando su comprensión. La coherencia se refiere a cómo se conectan las ideas entre sí, mientras que la cohesión trata sobre cómo se enlazan las palabras y frases dentro del texto.
- **Interfaces de Usuario Intuitivas:** Diseños de software o herramientas que son fáciles de usar y entender, permitiendo a los usuarios interactuar con la tecnología de manera efectiva y sin problemas.

# Abstract

The mission of the Easy Reading Methodology is to democratize information, making it accessible. Everyone should have the ability to access texts and understand documents that would otherwise remain closed, encrypted with terminologies that are not understandable to those not educated in the subject, or written in a language which individuals with reading comprehension difficulties may not comprehend.

This methodology goes beyond the ordinary; it provides clear and precise guidelines that facilitate the transformation of complex texts into understandable documents without sacrificing content integrity. These documents become powerful tools for individuals facing intellectual or linguistic barriers. The primary beneficiaries of this methodology and its tools are language learners, individuals whose educational opportunities did not prepare them to decipher sophisticated texts, or those with deficits in reading comprehension. This initiative not only breaks down communicative barriers but also fosters more active and aware participation in society, impacting governmental documents, daily news, practical instructions, and educational materials which are being rewritten or adapted to be clear, accessible, and understandable to all.

In this context, the role of technology is fundamental. The process of text adaptation is being automated to process texts on a large scale and ensure universal accessibility. This automation maximizes efficiency, standardizes quality, and ensures that each adapted document becomes a pillar of accessibility. However, this task is complex; evaluating the effectiveness of automatic adaptations poses a formidable challenge. Research has shown that simply simplifying the lexicon is not sufficient; it also requires meticulous and deep reorganization that not only makes the information readable but also resonant for the intended reader.

This project proposes a new advancement by introducing the evaluation of one or several texts through a comparison of the original text with its adaptation, provided by the user in a CSV file. This evaluation is carried out by calculating various metrics, tasked with assessing the similarity of words used, their quality, coherence, and the similarity of the adapted text to the original.

In conclusion, Easy Reading Methodology is not just a tool; it is a necessity in our struggle for an inclusive society. Automating this process opens new horizons for efficiency and quality, yet we face challenges that demand innovative and creative solutions. This work presents a solution to the problem of evaluating adapted texts, providing a useful tool for researchers, educators, and students who wish to verify the quality of their adaptations.

**Key Words:**

- **Easy Reading Methodology:** An approach that simplifies and clarifies information, allowing people with linguistic comprehension difficulties or intellectual barriers to access and understand complex texts.
- **Democratize Information:** The process of making information accessible and understandable to everyone, regardless of their linguistic or educational abilities, promoting equality in access to knowledge.
- **Content Integrity:** Maintaining the truthfulness, quality, and overall value of the original content even after its simplification.
- **Intellectual Barriers:** Obstacles that prevent people with cognitive limitations or learning difficulties from fully accessing information.
- **Automation:** The use of technology and automated systems to perform tasks that traditionally require human intervention, in this case, the adaptation of texts to a simpler format.
- **Effectiveness Evaluation:** The process of determining how effective automatic adaptations are in making the original texts accessible and understandable for the target audience.
- **Coherence and Cohesion:** Aspects of a text that contribute to it being logical and fluid, facilitating its comprehension. Coherence refers to how ideas are connected to each other, while cohesion deals with how words and phrases are linked within the text.
- **Intuitive User Interfaces:** Software or tool designs that are easy to use and understand, allowing users to interact with technology effectively and without issues.

# Capítulo 1

## Introducción

En este capítulo, exploramos los aspectos cruciales que dan contexto y proporcionan la justificación necesaria para este proyecto, se destaca la motivación que impulsó el desarrollo del proyecto, contextualizándolo en un marco de referencia y articulando con precisión los objetivos que se quieren alcanzar. Además, se ofrece una visión general de la estructura organizativa del documento, especificando los contenidos que comprenden cada sección para facilitar una comprensión integral del trabajo presentado.

### 1.1. Motivación del proyecto

Hay varias razones que motivaron la realización de este trabajo; en primer lugar, la creciente necesidad del acceso equitativo a la información, actualmente nos encontramos en una sociedad donde la tecnología se está asentando cada vez más en el mundo de la educación y la investigación[1] y el desarrollo web puede proporcionar soluciones a este problema, ya que si se tienen en cuenta las bases de la Lectura Fácil[2] es posible el desarrollo de herramientas capaces de proporcionar un acceso igualitario a esta información, independientemente del dominio de la lengua o las capacidades del usuario.

Además, se han realizado esfuerzos para que la información sea accesible por un público más amplio, incluidas aquellas personas que sufren alguna discapacidad intelectual o problemas del lenguaje. Por lo tanto, proporcionar una guía clara y concisa para interpretar la información es un enfoque práctico y, al utilizar este enfoque en el desarrollo web, los usuarios pueden mejorar su experiencia.

Por lo que enfocarse en la mejora de herramientas de evaluación automática del lenguaje para textos adaptados, podría acelerar la creación de contenido accesible, sino que también podría mejorar su calidad y eficacia. El objetivo de este proyecto es abordar estas necesidades urgentes mediante el desarrollo de una herramienta de evaluación que pueda utilizar métricas lingüísticas para analizar y evaluar textos adaptados.

### 1.2. Contexto del proyecto

En la sociedad contemporánea, caracterizada por una creciente digitalización de la información, el acceso equitativo a esta información es fundamental para garantizar

la inclusión social de todas las personas, especialmente de aquellas que tienen alguna dificultad cognitiva o con el lenguaje. Dentro de este contexto, la metodología de Lectura Fácil[3] proporciona las bases esenciales para adaptar textos haciéndolos entendibles y comprensibles para todas las personas incluidas aquellas con discapacidades intelectuales y para quienes poseen limitaciones lingüísticas o lectoras.

El campo de la Lectura Fácil ha evolucionado significativamente en los últimos años, esto viene propiciado tanto por el avance en las políticas de inclusión como por el desarrollo de nuevas tecnologías. Los trabajos previos en este ámbito han establecido las bases metodológicas para la adaptación de textos[2], pero la aparición de herramientas digitales ha abierto nuevas vías de investigación y desarrollo. En particular, la automatización del proceso de adaptación de textos incluyendo el uso de herramientas tanto como para crearlos como para su posterior análisis y evaluación de calidad.

El presente proyecto busca abordar estas necesidades mediante la actualización de una aplicación web existente que evalúa adaptaciones automáticas a Lectura Fácil[4]. Este trabajo tiene como objetivo implementar una herramienta que use métricas lingüísticas usadas para la evaluación de textos adaptados y rediseñar la interfaz de usuario para optimizar tanto la usabilidad como la precisión en la evaluación de los textos adaptados. Con esto, el TFG no solo contribuye a la tecnología de adaptación de textos sino que también se alinea con los esfuerzos globales por una mayor inclusión digital, garantizando que todos los usuarios, independientemente de sus habilidades cognitivas o lingüísticas, puedan acceder a la información que necesitan en la forma que mejor se adapte a sus capacidades.

### 1.3. Objetivos

El objetivo principal de este Trabajo de Fin de Grado es actualizar la aplicación web existente para la evaluación de adaptaciones automáticas a Lectura Fácil. Esto implica el desarrollo de nuevas herramientas y estrategias que se encarguen de realizar una evaluación más precisa y eficiente de los textos adaptados, asegurando que estos sean comprensibles y accesibles para los usuarios finales, especialmente aquellos con discapacidades intelectuales o limitaciones lingüísticas. Los objetivos específicos de este proyecto son:

- **Selección del conjunto de métricas de evaluación a incorporar:** Desarrollar y seleccionar un conjunto de métricas que permitan evaluar de manera efectiva la calidad de las adaptaciones de textos a Lectura Fácil. Este objetivo busca establecer unos criterios claros que reflejen la comprensibilidad y la fidelidad del texto adaptado respecto al original.
- **Diseño de los servicios web:** Crear el diseño técnico de los servicios web que formarán la nueva parte de la aplicación. Esto incluye la definición de la arquitectura de software y la planificación de las interfaces de programación de aplicaciones (APIs) que permitirán la interacción eficiente entre los componentes de la aplicación y los usuarios finales.
- **Implementación de los servicios web:** Codificar y desplegar los servicios web diseñados, asegurando que sean robustos, seguros y capaces de manejar las demandas de tráfico y procesamiento de datos. Este objetivo es crucial para la funcionalidad general del sistema y su capacidad para procesar los archivos que proporcionen los usuarios con los textos a analizar.

- **Re-diseño de la interfaz de usuario web:** Editar la interfaz de usuario de la aplicación web para hacerla más intuitiva, accesible y amigable para los usuarios, añadiendo la nueva funcionalidad la cual se encargará de realizar el análisis de los textos sin influir en la aplicación original.
- **Mejora en la implementación de la interfaz de usuario web:** Refinar y optimizar la implementación de la interfaz de usuario para asegurar su funcionalidad fluida y eficiente en diferentes dispositivos y plataformas. Este objetivo implica ajustes técnicos que mejorarán la respuesta y la estabilidad de la interfaz.
- **Evaluación:** Realizar pruebas exhaustivas de la aplicación para verificar su funcionalidad, usabilidad y precisión en la evaluación de adaptaciones a Lectura Fácil. Este proceso incluirá pruebas unitarias, pruebas de integración y pruebas de usuario para garantizar que la aplicación cumpla con todos los requisitos y expectativas.

### 1.4. Estructura del Documento

El presente documento estará estructurado de la siguiente manera:

- 1 **Introducción:** Esta sección se proporcionan la motivación y el contexto del proyecto, explicando la importancia de la Lectura Fácil y la necesidad de herramientas automáticas para evaluar adaptaciones de textos. Se presenta el objetivo del trabajo y se introduce la estructura del documento.
- 2 **Trabajo Relacionado y Estado de la Cuestión** Un análisis detallado de los estudios y desarrollos previos en el campo de la Lectura Fácil y las tecnologías de adaptación automática. Esta sección ayuda a situar el proyecto dentro del contexto académico y tecnológico actual, identificando las lagunas que el proyecto intenta llenar.
- 3 **Desarrollo:** Descripción del proceso de desarrollo del proyecto, incluyendo la selección y diseño de métricas de evaluación, así como el desarrollo de la arquitectura de software y la implementación inicial de la aplicación. También se proporcionan los detalles sobre la metodología de pruebas empleada para verificar la funcionalidad y eficacia de la aplicación. Incluye pruebas unitarias, pruebas de integración, y pruebas de usabilidad, junto con la evaluación de los resultados obtenidos.
- 4 **Conclusiones y Líneas Futuras:** Una síntesis de los hallazgos del proyecto, resaltando cómo se han alcanzado los objetivos y discutiendo la efectividad de la solución propuesta en base a los resultados de las pruebas. Además del impacto de este proyecto en el campo de la Lectura Fácil y el análisis de textos. Junto con propuestas y recomendaciones para la expansión o mejora futura del proyecto, basadas en los resultados obtenidos y las experiencias adquiridas durante el desarrollo y las pruebas.

**Bibliografía:** Lista completa de todas las fuentes y referencias utilizadas para apoyar el contenido del trabajo, estructurada de acuerdo con las normas académicas apropiadas.

**Anexos:** Material adicional que apoya el contenido principal del trabajo, como las tablas, datos de pruebas, o cualquier otra documentación relevante para la comprensión del proyecto.

## **Capítulo 2**

# **Trabajo relacionado y Estado de la Cuestión**

En este capítulo se analiza y explica el estado actual de la adaptación automática de textos usando la metodología de Lectura Fácil y el desarrollo web. Los diferentes aspectos clave que se van a explorar son:

### **2.1. Trabajo Relacionado**

En esta sección se explica el trabajo realizado por Jesús Vico Castro, titulado "Lectura Fácil: Similitud Semántica"[4] bajo la supervisión de María del Carmen Suárez de Figueroa Baonza e Isam Diab Lozano en la Universidad Politécnica de Madrid en mayo de 2023. El objetivo de este TFG es desarrollar un sistema basado en una aplicación web capaz de identificar y medir la similitud semántica entre diferentes frases en castellano por medio de la implementación de varios modelos de aprendizaje automático. La página principal de esta aplicación se puede observar en la Figura 2.1.

¡Bienvenido a la Calculadora de Similitud de Texto! Esta aplicación te permite cargar archivos CSV que contienen pares de frases y calcular la similitud semántica entre ellas utilizando diferentes modelos de lenguaje.

#### Formato del archivo CSV

El archivo CSV debe tener un formato específico para funcionar correctamente con esta aplicación. Debe contener tres columnas separadas por punto y coma (;): frase\_original, frase\_adaptada y semanticamente\_similares. Por ejemplo:

```
1 frase_original;frase_adaptada;semanticamente_similares
2 El libro fue escrito por mi amigo.;Mi amigo escribió el libro.;SI
3 El libro fue escrito por mi amigo.;El artículo fue escrito por el periodista.;NO
4 El libro fue escrito por mi amigo.;Mi tío escribió el libro.;NO
5 El libro fue escrito por Juan Gómez.;Juan Gómez escribió el libro.;SI
6 El libro fue escrito por Juan Gómez.;El artículo fue escrito por Juan Gómez.;NO
```

#### Modelos de similitud semántica usados

Los modelos de lenguaje utilizados en esta aplicación son los siguientes:

- [Maite89/Roberta\\_finetuning\\_semantic\\_similarity\\_stsb\\_multi\\_mt](#)
- [symanto/sn-xlm-roberta-base-snl-mnli-anli-xnli](#)
- [hiamsid/sentence\\_similarity\\_spanish\\_es](#)

#### Cómo usar la aplicación

1. Carga un archivo CSV. Asegúrate de que esté en el formato adecuado, mencionado anteriormente.
2. Una vez cargado el archivo, presiona el botón "Calcular similitudes".
3. La aplicación comenzará a procesar el archivo y a calcular las similitudes utilizando múltiples modelos de lenguaje. Este proceso puede tardar un poco, dependiendo del tamaño del archivo y del número de pares de frases.
4. Una vez finalizado el cálculo, los resultados se mostrarán en una tabla. En ella, podrás ver las puntuaciones de similitud para cada par de frases.
5. Compara los resultados de los diferentes modelos para obtener una visión completa de la similitud semántica entre las frases.

Seleccionar archivo nada seleccionado

Calcular similitudes

Figura 2.1: Página principal del proyecto anterior

### 2.1.1. Metodología

- 1 **Creación del Corpus:** Se recopiló un corpus especializado con varios pares de frases de diversos tipos que incluyen oraciones en pasiva, con perífrasis verbales, yuxtapuestas, con encadenamiento sintáctico y con acrónimos y diferenciado los pares de frases semánticamente similares o de los que no lo son. Por ejemplo, un par de frases semánticamente similares sería:

- El libro fue escrito por mi amigo.
- Mi amigo escribió el libro.

Y un par de frases que no son semánticamente similares sería:

- El libro fue escrito por mi amigo.
- El artículo fue escrito por el periodista.

- 2 **Desarrollo de la API:** Se implementó una API la cual recibe un archivo donde se encuentran los pares de frases y mediante una llamada al backend se procesa el archivo y se calculan las similitudes semánticas, proporcionando puntuaciones de similitud, predicciones y etiquetas reales.

- 3 **Aplicación web:** Se desarrolló una aplicación web con una estructura de Cliente-Servidor, que permite a los usuarios cargar archivos CSV y visualizar los resultados proporcionados por la API de una manera intuitiva y accesible en un formato de tabla.

- **Cliente:** Se desarrolló un cliente el cual interactúa directamente con el usuario, es donde se cargan los archivos CSV y envía una solicitud al servidor para su procesamiento.

- **Servidor:** El servidor recibe las solicitudes del cliente, es el encargado de procesar los datos utilizando la API y la lógica implementada para calcular las similitudes semánticas en el código y devuelve los resultados al cliente.

4 **Evaluación de los resultados:** Se evaluó la efectividad de los resultados de los modelos de aprendizaje, analizando el rendimiento general de estos modelos y también el rendimiento específico para cada tipo de oración presente en el corpus.

### 2.1.2. Métricas Utilizadas

En este TFG se usaron tres métricas distintas para el análisis de las similitudes, las cuales se muestran en la Figura 2.2, que son:

- **Precisión (Precision):** Mide la exactitud de las predicciones del modelo, es decir, la proporción de verdaderos positivos entre todos los ejemplos que el modelo predijo como positivos.
- **Exhaustividad (Recall):** Mide la capacidad del modelo para identificar todas las oraciones semánticamente similares, es decir, la proporción de verdaderos positivos entre todos los ejemplos que son realmente positivos.
- **Puntuación F1 (F1 Score):** Es la media armónica entre la precisión y la exhaustividad, proporcionando una medida equilibrada del rendimiento del modelo.

[Regresar a la página principal](#)

## Resultados

### Descripción de la evaluación de los modelos

Cada modelo predice si las frases son semánticamente similares usando un umbral de 0.8. Si la puntuación de similitud es mayor que 0.8, las frases se consideran similares (valor de 1). Si es 0.8 o menos, se consideran no similares (valor de 0).

La evaluación de un modelo se realiza comparando las predicciones con las etiquetas reales del conjunto de datos. Si la etiqueta real es 'SI' (las frases son similares), se toma como 1; si es 'NO' (las frases no son similares), se toma como 0.

Se calculan varias métricas para evaluar el rendimiento de cada modelo:

- **Precisión (Accuracy):** Esta es la proporción de predicciones correctas sobre el total. Indica cuántas veces el modelo predijo correctamente si las frases eran o no similares.
- **Precisión (Precision):** Esta es la proporción de predicciones correctas de similitud (valor de 1) entre todas las predicciones de similitud realizadas por el modelo. Indica cuántas de las frases que el modelo pensó que eran similares eran realmente similares.
- **Sensibilidad (Recall):** Esta es la proporción de frases similares que el modelo identificó correctamente. Indica cuántas de las frases que eran realmente similares el modelo pudo identificar.
- **Puntuación F1 (F1 Score):** Esta es la media armónica de la precisión y la sensibilidad, y proporciona un balance entre ambas métricas. Un F1 Score más alto indica que el modelo tiene un buen equilibrio entre precisión y sensibilidad.

### Evaluación de los modelos usados:

#### Modelo 1: Maite89/Roberta\_finetuning\_semantic\_similarity\_stsb\_multi\_mt

Exactitud (Accuracy): 0.7903  
 Precisión (Precision): 0.7624  
 Exhaustividad/Sensibilidad (Recall): 0.9747  
 F1-Score: 0.8556

#### Modelo 2: symanto/sn-xlm-roberta-base-snl1-mnli-anli-xnli

Exactitud (Accuracy): 0.7742  
 Precisión (Precision): 0.7865  
 Exhaustividad/Sensibilidad (Recall): 0.8861  
 F1-Score: 0.8333

#### Modelo 3: hiamsid/sentence\_similarity\_spanish\_es

Exactitud (Accuracy): 0.7823  
 Precisión (Precision): 0.7653  
 Exhaustividad/Sensibilidad (Recall): 0.9494  
 F1-Score: 0.8475

Frase Original	Frase Adaptada	Semánticamente similares	Puntuaciones de similitud de los modelos
El libro fue escrito por mi amigo.	Mi amigo escribió el libro.	Sí	Modelo 1: 0.822 Modelo 2: 0.920 Modelo 3: 0.927

Figura 2.2: Visualización de resultados

### 2.2. Métricas relacionadas con el Procesamiento de Lenguaje Natural

En esta sección se explican y muestran las métricas que se han usado en este proyecto para analizar las adaptaciones de los textos a Lectura Fácil.

#### 2.2.1. Características de las Métricas:

- **BLEU:** BLEU (BiLingual Evaluation Understudy) es una métrica muy utilizada en el procesamiento del lenguaje natural (PLN) y en la traducción automática para evaluar la calidad de los textos generados por una máquina o inteligencia artificial, comparándolos con uno o varios textos de referencia de alta calidad, los cuales han sido adaptados por una persona. El funcionamiento de BLEU se basa en la comparación de n-gramas (secuencias de n palabras consecutivas) entre el texto generado y los textos de referencia originales, BLEU calcula la precisión considerando cuántos n-gramas del texto generado coinciden con los del texto o textos de referencia. Luego, esta puntuación de precisión se ajusta con una penalización por brevedad para evitar que se favorezcan las traducciones más cortas [5].

Se ha demostrado que la puntuación BLEU se correlaciona bien con la evaluación humana de la calidad de la traducción.

- **ROUGE:** La métrica ROUGE se encarga de medir la precisión y la retención de información de un sistema. Ofrece diferentes variantes para proporcionar una medición más precisa de las características de un resumen. A continuación, se explican las dos más comunes, las cuales son las que se han usado en este trabajo. La primera es ROUGE-1. Esta variante calcula la precisión y la retención de información a partir de los unigramas o palabras del resumen generado y del resumen esperado. ROUGE-L, difiere de las demás métricas al buscar la cadena más larga común contenida en ambos textos resumidos o adaptados.[6]

Existen más tipos de métricas ROUGE, pero la ROUGE-1 y otras variantes similares se pueden agrupar en la denominada ROUGE-N, que incluye todos los tipos de n-gramas o caracteres o cadenas de caracteres comunes.[7]

- **METEOR:** METEOR es una métrica de evaluación de traducción automática que considera tanto la precisión como el recall, proporcionando una perspectiva más completa de la calidad de la traducción generada por un modelo. A diferencia de BLEU, METEOR también tiene en cuenta la similitud semántica mediante técnicas como el uso de un stemmer o un buscador de sinónimos, lo que permite emparejar palabras similares incluso si no coinciden exactamente.

En este proyecto se ha utilizado METEOR para superar las limitaciones de BLEU. Una de las ventajas de METEOR frente a BLEU es que se correlaciona mejor con la evaluación humana. Además, METEOR valora tanto la precisión como el recall, aunque da mayor importancia a este último [8].

- **TER<sup>1</sup>:** TER (Translation Edit Rate) es una métrica en la evaluación de traducción automática, que cuenta las ediciones necesarias para cambiar una traducción automática a una de referencia, considera las inserciones, las eliminaciones, las sustituciones, y las transposiciones de palabras. Proporciona una medida clara de cuántas intervenciones editoriales son necesarias para alcanzar la traducción

---

<sup>1</sup><https://kantanmtblog.com/what-is-translation-error-rate-ter/>

## 2.2. Métricas relacionadas con el Procesamiento de Lenguaje Natural

---

ideal; un valor bajo de TER indica una traducción de alta calidad. A diferencia de otras métricas que se centran más en la coincidencia exacta de palabras o n-gramas, TER evalúa la fluidez y la precisión, ofreciendo así una visión más práctica de la utilidad de la traducción. En este contexto, hemos implementado TER para complementar otras evaluaciones, buscando ofrecer un panorama integral de la efectividad de las traducciones automáticas.

- **WER<sup>2</sup>:** WER (Word Error Rate), ampliamente reconocida en áreas como el reconocimiento automático de voz y la traducción automática, evalúa la cantidad de errores en una transcripción o traducción al compararla con un texto de referencia, los errores se dividen en tres tipos: inserciones, eliminaciones y sustituciones. La fórmula para calcular WER es dividir el número total de errores por el número de palabras en el texto de referencia, resultando en una métrica que cuantifica la precisión: un WER bajo significa una mejor precisión. Utilizada principalmente para medir la eficacia de los sistemas de reconocimiento de voz, esta métrica es crucial para nuestro proyecto ya que nos permite determinar la precisión de las transcripciones y traducciones automáticas, orientándonos sobre cómo mejorar la eficiencia y exactitud del sistema.

### 2.2.2. Aplicaciones Prácticas:

Actualmente, estas métricas son de gran utilidad, y son ampliamente usadas por grandes empresas como Google, Facebook o Microsoft para desarrollar varias de las herramientas que solemos usar en nuestro día a día, varios ejemplos de estas herramientas son:

- **Google Translate<sup>3</sup>:** Es una herramienta comúnmente utilizada para la traducción de textos o palabras, aunque también puede usarse para la simplificación de textos ya que esta herramienta también proporciona sinonimos y frases similares a la introducida. Esta herramienta basa su funcionamiento en BLEU, y la usa para evaluar y mejorar la calidad de las traducciones automáticas.
- **Sumy[9]:** Esta es una de las bibliotecas más usadas en Python, su uso se enfoca principalmente en la extracción de resúmenes de textos junto con el análisis de la calidad de esos resúmenes extraídos, esta biblioteca basa su funcionalidad en la métrica ROUGE.
- **The Stanford Natural Language Processing Group:** Este grupo de estudiantes e ingenieros, han desarrollado software de análisis de calidad de modelos de traducción y otros sistemas de generación de lenguaje natural. Destaca su proyecto para el análisis de traducciones automáticas<sup>4</sup> en el cual una de las métricas usadas para comprobar esa traducción es la de METEOR aunque también se usan métricas como BLEU y TER.
- **Google Speech-to-Text API<sup>5</sup>:** Esta herramienta desarrollada por Google, se encarga de generar texto a partir de la voz y el habla de las personas, una de las métricas que se usan para comprobar la eficacia de esta herramienta es WER, para evaluar y mejorar la precisión de sus transcripciones automáticas.

---

<sup>2</sup><https://martin-thoma.com/word-error-rate-calculation/>

<sup>3</sup><https://cloud.google.com/translate>

<sup>4</sup><https://www-nlp.stanford.edu/projects/mt.shtml>

<sup>5</sup><https://cloud.google.com/translate>

### 2.3. Avances Tecnológicos

En esta sección se analizan los avances recientes en tecnologías similares a las que se desarrollan en este trabajo.

- **Hemingway App**<sup>6</sup>: Esta es una herramienta desarrollada por Hemingway Ltd. se encarga de analizar textos adaptados o sin adaptar y proporcionar un análisis visual de la complejidad de las oraciones y las palabras de estos textos. Dependiendo de la complejidad de la oración, la aplicación la subraya de distintos colores, naranja si es difícil de leer y rojo si es muy difícil. También analiza palabras que tienen alternativas más simples, subrayándolas de morado. Además esta aplicación proporciona una calificación de 0 a 10 sobre la legibilidad del texto y su correspondiente evaluación junto con varias estadísticas como el número de oraciones y palabras o el tiempo necesario de lectura del texto.
- **Lexile Framework for Reading**<sup>7</sup>: Herramienta desarrollada por MetaMetrics, tiene una gran variedad de funcionalidades pero una de las más importantes es la que se encarga de evaluar la capacidad de lectura de las personas y el nivel de complejidad de los textos. Esta herramienta se usa para analizar la complejidad de los textos los desmonta y analiza unas características principales como pueden ser la longitud de las oraciones o la frecuencia de aparición de las palabras, el resultado de este análisis es una medida Lexile, la cual representa una predicción del nivel de comprensión que un lector puede alcanzar. Esta herramienta es de gran utilidad para que padres o educadores puedan emparejar a los estudiantes con textos adecuados a su nivel de lectura.
- **Easy Reading Tool**<sup>8</sup>: Es un proyecto desarrollado por varias instituciones europeas encargada de mejorar la accesibilidad en la lectura. Esta herramienta se encarga de analizar el contenido de una web o texto en línea y ajustarlo a las necesidades del usuario, se puede ajustar el texto o contenido para varios niveles de comprensión, se puede ajustar la disposición de la página y también puede proporcionar una explicación del contenido mediante imágenes, símbolos y vídeos. esta herramienta usa las pautas que se basan en la Lectura Fácil para ajustar y simplificar los textos y poder proporcionar una adaptación y evaluación precisa.

### 2.4. Desarrollo Web y Accesibilidad

El desarrollo web y la accesibilidad son fundamentales para la innovación ética, especialmente en el ámbito de la lectura fácil. Nuestra sociedad, cada vez más digitalizada, se enfrenta a la necesidad de hacer accesibles todas las plataformas en línea. Las personas con discapacidad, los hablantes de lenguas no nativas, las personas mayores y cualquiera que pueda tener dificultades con textos complejos se benefician considerablemente de los esfuerzos por fomentar la inclusión [10].

La Iniciativa de Accesibilidad a la Web del Consorcio World Wide Web esboza las normas de accesibilidad a la web, proporcionando el marco para sitios inclusivos. Estas normas son esenciales, son unas directrices detalladas las cuales describen meticulosamente cómo hacer que los contenidos web sean accesibles para todos, sin excepción [11].

---

<sup>6</sup><https://hemingwayapp.com/>

<sup>7</sup><https://lexile.com/>

<sup>8</sup><https://www.easyreading.eu/>

Incorporar la lectura fácil a este marco exige un manejo experto. Va más allá de la accesibilidad tradicional, refinando la complejidad de los contenidos al tiempo que se preserva la integridad esencial. Las estructuras claras de las frases, el lenguaje directo y los apoyos visuales mejoran la comprensión del mensaje, ampliando la comprensión a un espectro más amplio de usuarios [12].

El principal reto consiste en adaptar los contenidos sin diluir su esencia informativa. Esto es especialmente relevante en campos técnicos o especializados en los que normalmente se usa una terminología compleja. La navegabilidad del sitio, su diseño de interfaz, la estructuración de los menús, las instrucciones para el usuario y la navegación facilitada se convierten en componentes cruciales de la experiencia general del usuario. Una interfaz intuitiva, permite una navegación fluida y sin esfuerzo para todos los usuarios a través de una información simplificada, por lo que al integrar la lectura fácil con las prácticas habituales de desarrollo web, ampliamos el acceso a la información y promovemos una participación más rica y equitativa en nuestra sociedad digital. Estos esfuerzos suponen no sólo un requisito técnico, sino también un firme compromiso con el desarrollo de un espacio digital inclusivo, lo que realza el potencial de la tecnología como catalizador en la inclusión social.

## 2.5. Tecnologías utilizadas

En esta sección se explican las tecnologías que se han utilizado durante la realización de este proyecto.

### 2.5.1. Entornos de Programacion y Testing:

- **Visual Studio Code**<sup>9</sup>: Visual Studio Code, comúnmente conocido como VS Code, es un editor de código fuente desarrollado por Microsoft. Destaca por su naturaleza de código abierto, ligereza y compatibilidad con múltiples sistemas operativos, incluidos Windows, macOS y Linux. Este editor es ampliamente valorado por su interfaz limpia y personalizable, rendimiento optimizado y soporte extensivo para una diversidad de lenguajes de programación como JavaScript, Python, Java, etc.

En el contexto del desarrollo del código para este proyecto, se ha usado Visual Studio Code como la principal herramienta para escribir y editar código.

- **Apache JMeter**<sup>10</sup>: Herramienta open source hecha en Java y desarrollada por Apache que se ha usado básicamente para probar la escalabilidad, el rendimiento y realizar pruebas de carga en aplicaciones web generando múltiples usuarios concurrentes que generan peticiones a una aplicación web base de datos, servidor... Su interfaz gráfica intuitiva permite configurar, ejecutar y visualizar pruebas sin necesidad de conocimientos avanzados en programación, y su extensibilidad mediante plugins permite adaptar las pruebas a necesidades y a aplicaciones específicas. Destaca su uso para poder hacer informes detallados y análisis de métricas clave como son los tiempos de respuesta, el código de la respuesta del servidor. Además también puede ser usado para identificar cuellos de botella en la aplicación.

---

<sup>9</sup><https://code.visualstudio.com/>

<sup>10</sup><https://jmeter.apache.org/>

### 2.5.2. Lenguajes de Programación y Servicios:

- **Python**<sup>11</sup>: Python es un lenguaje de alto nivel el cual tiene una sintaxis clara, sencilla y legible. Creado por Guido van Rossum y lanzado por primera vez en 1991, Python es conocido por su enfoque en la legibilidad del código, lo que permite a los desarrolladores expresar conceptos de manera concisa y efectiva. Su amplio ecosistema de bibliotecas y frameworks, como Django y Flask para el desarrollo web, y Pandas y NumPy para el análisis de datos, lo convierten en una opción preferida en diversos campos, desde la ciencia de datos y el aprendizaje automático hasta el desarrollo web y la automatización de tareas. En este proyecto se ha usado la versión de Python 3.9.5, versión que se lanzó el 3 de mayo de 2021.
- **Angular**<sup>12</sup>: Angular es un framework de desarrollo de aplicaciones web de código abierto, mantenido principalmente por Google, que se utiliza para crear aplicaciones de una sola página (SPA) de manera eficiente. Lanzado por primera vez en 2010 como AngularJS, el framework ha evolucionado significativamente, con una reescritura completa que resultó en Angular 2+ (comúnmente llamado simplemente ".Angular") en 2016. Angular se destaca por su arquitectura basada en componentes, que facilita la creación de interfaces de usuario modulares y reutilizables. Utiliza TypeScript, un superconjunto de JavaScript que agrega tipado estático, lo que mejora la robustez y la mantenibilidad del código. Los usos de Angular son variados pero tratan desde la gestión de datos con servicios, clientes y dependencias, hasta la manipulación del dominio, el enrutamiento y el flujo de datos de la aplicación. Su ecosistema incluye herramientas poderosas como Angular CLI para la generación y gestión de proyectos, y Angular Material para el diseño de interfaces de usuario consistentes y accesibles.

### 2.5.3. Bibliotecas y Extensiones:

- **NLTK**<sup>13</sup>: Biblioteca de Python que se usa para el análisis de lenguaje natural, contiene varios módulos, pero en el presente trabajo, se ha usado el módulo de `nlk.translate`, el cual contiene varias de las librerías que se usarán para el desarrollo del proyecto, para calcular las métricas necesarias para analizar los textos adaptados. Estas librerías son `bleu-score` y `meteor-score`. También para poder tokenizar los textos que se van a analizar se ha importado el módulo de `nlk.tokenize` el cual contiene todas las funciones necesarias para poder tokenizar los textos por palabras o frases.
- **Pyter**<sup>14</sup>: Esta biblioteca de Python que contiene todas las funciones necesarias para poder implementar el cálculo de una de estas métricas, en este caso TER.
- **Jiwer**<sup>15</sup>: Al igual que con `pyter`, esta biblioteca de Python contiene todas las funciones necesarias, se ha importado en el proyecto para calcular una de las métricas necesarias en el análisis de las adaptaciones, en este caso WER.
- **RougeScore**<sup>16</sup>: Esta biblioteca se ha importado en el proyecto para poder calcular la última de las métricas necesarias, en este caso ROUGE.

---

<sup>11</sup><https://www.python.org/about/>

<sup>12</sup><https://angular.dev/overview>

<sup>13</sup><https://www.nltk.org/>

<sup>14</sup><https://pypi.org/project/pyter/>

<sup>15</sup><https://pypi.org/project/jiwer/>

<sup>16</sup><https://pypi.org/project/rouge-score/>

## Capítulo 3

# Desarrollo

En este capítulo se describe el desarrollo que se ha llevado a cabo para implementar una funcionalidad la cual está basada en el análisis de adaptaciones de textos para la lectura fácil según unas métricas lingüísticas dentro de una aplicación previamente desarrollada, que se encargaba de analizar la Similitud Semántica de textos. Los principales objetivos de este capítulo se basan en la explicación de las técnicas y metodologías empleadas para integrar la nueva funcionalidad, la cual nos permite evaluar la calidad general de los textos adaptados según el cálculo de métricas BLEU, ROUGE, METEOR, TER y WER, proporcionando una evaluación detallada y desglosada de cada métrica lingüística involucrada.

La importancia de este desarrollo reside en su capacidad para ofrecer herramientas más robustas y detalladas a los usuarios, facilitando análisis más profundos, específicos y visuales, lo que contribuye significativamente a la investigación y práctica de la adaptación textual. Esto mejora la accesibilidad y comprensión de textos complejos, estos avances son esenciales para validar la efectividad del software y garantizar su utilidad en contextos educativos y profesionales, donde la precisión y la profundidad del análisis lingüístico son fundamentales.

### 3.1. Diseño del Sistema

En esta sección se explica el diseño que tiene la aplicación implementando la nueva funcionalidad, tal y como se puede observar en la Figura 3.1.

#### 3.1.1. Vision General:

El sistema que se ha desarrollado se basa en la ampliación de un trabajo previo, al que se le ha añadido una funcionalidad la cual se enfoca en el análisis de textos adaptados para la Lectura Fácil. Este sistema está formado por dos componentes esenciales que siguen una arquitectura cliente-servidor bien definida. El frontend o cliente, es la parte con la que interactúa el usuario y el backend o servidor, es donde se procesan los datos y se realizan todas las operaciones lógicas y matemáticas. La comunicación entre estos dos componentes se realiza mediante una API, asegurando un flujo de datos y archivos eficiente y seguro. Y tras finalizar el desarrollo, esta nueva funcionalidad se integra en la estructura ya existente, permitiendo un análisis profundo y detallado de un archivo proporcionado por el usuario con los textos originales y su correspondiente adaptación.

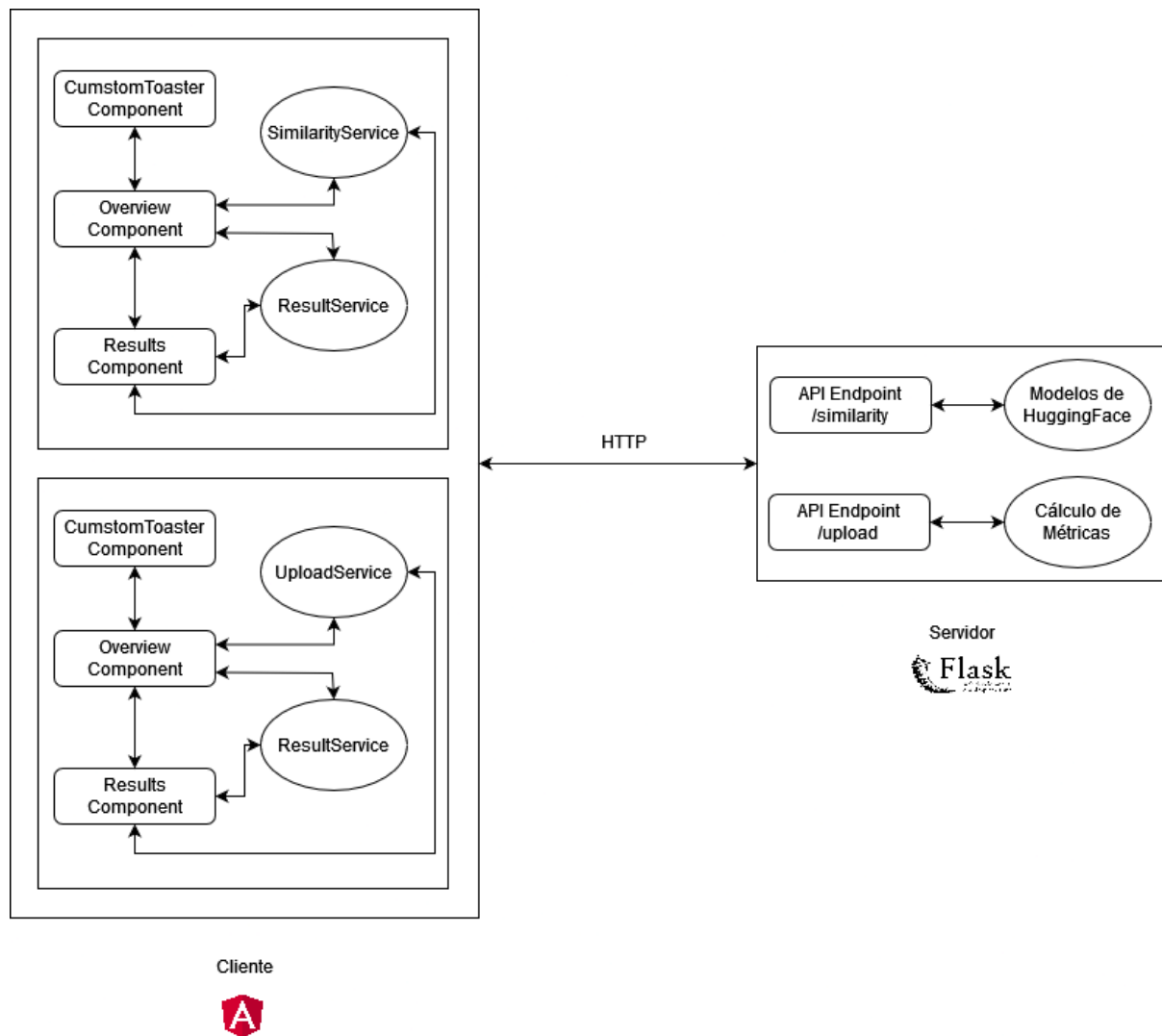


Figura 3.1: Nueva arquitectura del sistema

### 3.1.2. Componentes del Sistema:

#### 3.1.2.1. Frontend / Cliente:

Para la actualización del cliente se han editado algunos componentes y servicios previamente creados en el primer desarrollo de la aplicación, y además, se han añadido nuevos componentes y servicios para la implementación de la nueva funcionalidad, los cuales trabajan de manera conjunta para establecer las bases del sistema. Cabe destacar que para los nuevos componentes se ha mantenido el estilo que se desarrolló originalmente, para mantener así una cohesión con el entorno. Los componentes que ya estaban previamente y se han actualizado son estos:

- 1 **OverviewComponent:** Se ha mantenido la funcionalidad anterior, además de seguir funcionando como el eje principal del cliente, es la parte donde el usuario interactúa con la aplicación y decide entre calcular la similitud o las métricas. En el caso de calcular las métricas, **UploadService**, se encarga de comunicarse con el servidor para calcular las métricas y proporcionar una evaluación de los textos.

2 **ResultService:** Además de mantener todo el proceso que se desarrolló anteriormente, se ha añadido el envío de archivos al servidor y la recepción de resultados, permitiendo una actualización dinámica de la interfaz de usuario con los resultados de las métricas y los análisis.

Los componentes nuevos que se han añadido son:

1 **MetricsComponent:** Este nuevo componente es el encargado de mostrar los resultados de los análisis de las métricas en los textos, recoge los datos de ResultsService y presenta las métricas calculadas en un formato de tabla, también se ha añadido el uso de funciones específicas para asignar colores a las métricas haciendo los resultados más visuales y comprensibles.

Para mostrar cómo un usuario interactúa con la aplicación podemos observar la Figura 3.2.

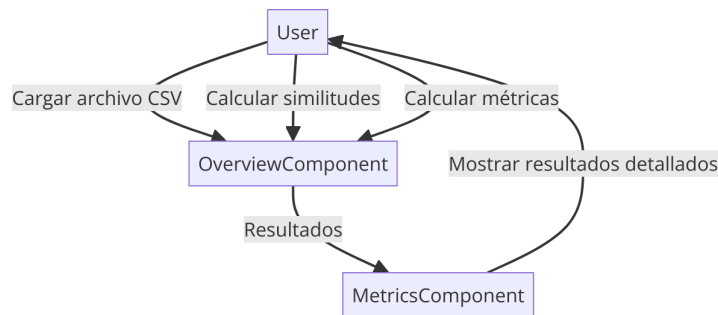


Figura 3.2: Diagrama caso de uso del Frontend

**3.1.2.2. Backend / Servidor:**

En la parte del backend, se ha decidido añadir una segunda ruta en la API llamada */upload*, se ha hecho de esta manera para no interferir con la funcionalidad anterior y separar cada aspecto de la plataforma. Esta separación, permite una mejora en la organización del código y las tareas que se realizan y también facilita la mantenibilidad y escalabilidad de la aplicación, permitiendo expandir o modificar funcionalidades en el futuro sin grandes reestructuraciones del código existente, ya que cada ruta se usa para funciones específicas y permite probar el código de manera más focalizada. Además esta estructura, permite al usuario una experiencia con la aplicación más fluida al proporcionarle respuestas de manera más eficaz.

El proceso que se ha llevado a cabo para desarrollar el cálculo de las métricas BLEU, ROUGE, METEOR, TER y WER, es mediante el uso de sus correspondientes librerías implementadas en Python. Primero, se carga un archivo con los textos originales y su correspondiente adaptación, posteriormente se analizan y mediante los resultados obtenidos usando estas métricas se hacen dos valoraciones, una individual, es decir, se devuelve un valor específico para cada métrica y su correspondiente evaluación mostrada de manera más visual y comprensible, y la segunda general, la cual, mediante una ponderación de estas métricas, se calcula un valor general que se muestra por pantalla junto con su evaluación presentada de manera visual.

Este flujo de trabajo se puede observar en la Figura 3.3.

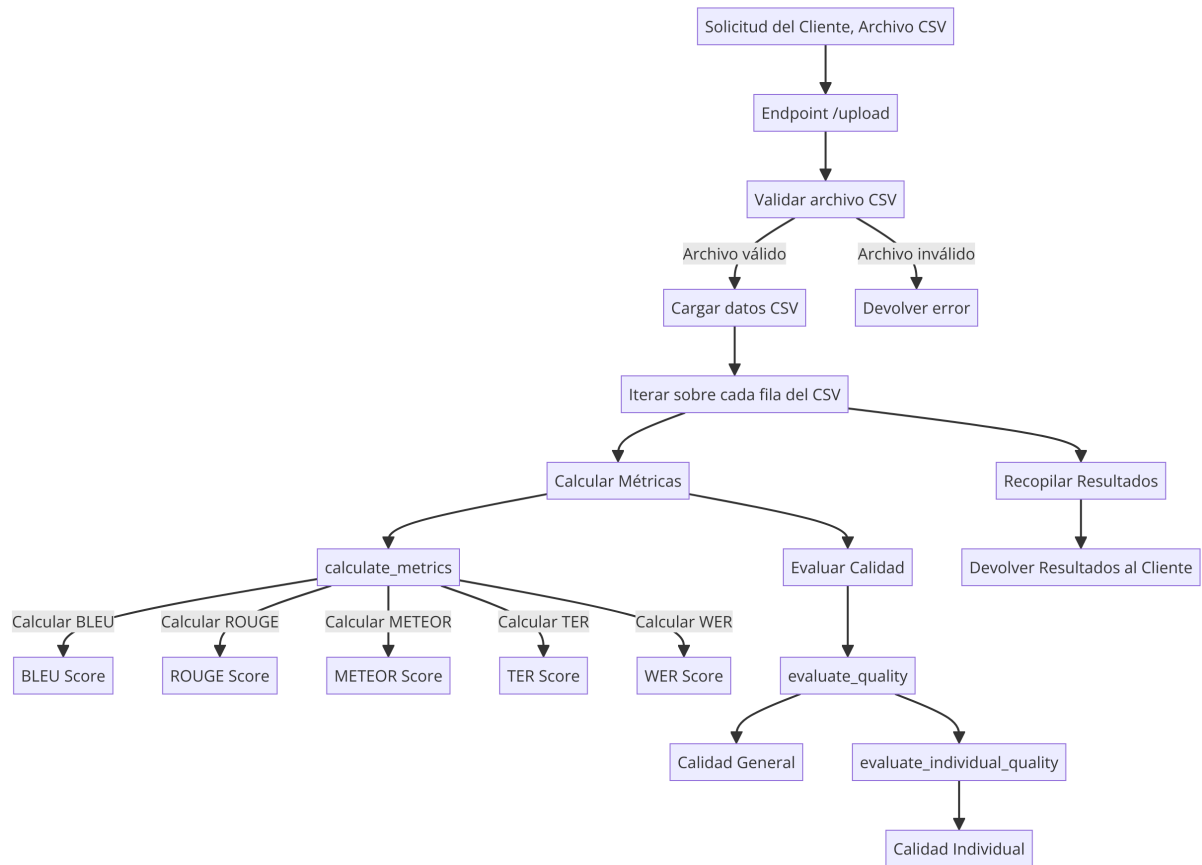


Figura 3.3: Flujo de trabajo backend

## 3.2. Implementacion del Sistema

En esta sección se explica el procedimiento que se ha seguido para la implementación del nuevo código y edición de código ya presente para desarrollar la nueva funcionalidad de la aplicación.

### 3.2.1. Implementación del Frontend:

La edición del código del Frontend (Cliente), se ha realizado de la siguiente manera, primero, se implementaron las partes de código que no interfirieran con el código original, se desarrollaron los siguientes componentes:

- **MetricsComponent:** Nuevo componente añadido para mostrar los resultados del análisis de métricas. Utiliza ResultsService para recuperar los datos y los presenta en un formato de tabla. Primero, se desarrolló el *metrics.component.ts* donde se implementa toda la lógica para la gestión de los resultados obtenidos después de aplicar las métricas, los resultados son obtenidos mediante la suscripción al */ResultService* el cual se encarga de recuperar las métricas del backend. Esta implementación incluye métodos para la asignación de colores a las evaluaciones de calidad, facilitando una interpretación visual rápida y efectiva, esto puede ser observado en la figura 3.4.

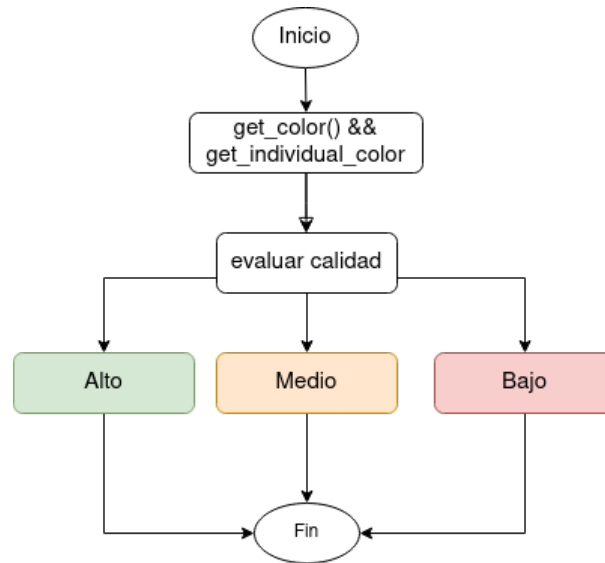


Figura 3.4: Logica colores de metrics.component.ts

Posteriormente se implementaron los archivos *metrics.component.html* y *metrics.component.css* que definen la estructura y el estilo visual, El archivo HTML utiliza la directiva *\*ngFor* de Angular para iterar sobre cada conjunto de métricas y mostrarlos en una tabla estructurada, cada fila de la tabla muestra la frase original y adaptada, junto con las métricas calculadas y sus evaluaciones de calidad tanto individuales como generales. Y el archivo CSS es el encargado de proporcionar los estilos para la visualización de la página. El resultado de esto se puede observar en la Figura 3.5.

Frase Original	Frase Adaptada	Métricas Calculadas	Evaluación Individual de la Calidad	Evaluación General de la Calidad
La biodiversidad es fundamental para el mantenimiento de los ecosistemas del mundo, pero está siendo amenazada por la deforestación y el cambio climático.	La biodiversidad ayuda a mantener nuestros ecosistemas. Está en peligro por la tala de árboles y los cambios en el clima.	BLEU: 0.04 ROUGE-1: 0.48 ROUGE-L: 0.35 METEOR: 0.34 TER: 0.87 WER: 0.84	BLEU: <b>LOW</b> METEOR: <b>MEDIUM</b> ROUGE-1: <b>MEDIUM</b> ROUGE-L: <b>LOW</b> TER: <b>LOW</b> WER: <b>LOW</b>	<b>LOW</b>
Los economistas sugieren que el incremento en las tasas de interés podría ayudar a estabilizar la economía inflacionaria actual.	Algunos expertos creen que subir los intereses puede calmar la economía que tiene mucha inflación.	BLEU: 0.03 ROUGE-1: 0.26 ROUGE-L: 0.21 METEOR: 0.19 TER: 1.06 WER: 0.90	BLEU: <b>LOW</b> METEOR: <b>LOW</b> ROUGE-1: <b>LOW</b> ROUGE-L: <b>LOW</b> TER: <b>LOW</b> WER: <b>LOW</b>	<b>LOW</b>
Las teorías de la relatividad de Einstein, que transformaron nuestra comprensión del espacio y el tiempo, siguen siendo parte fundamental de la física moderna.	Las ideas de Einstein cambiaron cómo entendemos el espacio y el tiempo y son muy importantes para la física hoy.	BLEU: 0.12 ROUGE-1: 0.41 ROUGE-L: 0.41 METEOR: 0.28 TER: 0.81 WER: 0.63	BLEU: <b>LOW</b> METEOR: <b>MEDIUM</b> ROUGE-1: <b>MEDIUM</b> ROUGE-L: <b>MEDIUM</b> TER: <b>LOW</b> WER: <b>LOW</b>	<b>LOW</b>

Figura 3.5: Tabla de respuesta

En los archivos HTML y CSS además de definir el estilo y forma de la tabla, también se define y se da forma a la página en la que aparece esta tabla, se crea una cabecera en la cual se puede ver que nos encontramos en la página de Evaluación de Métricas y en la esquina superior izquierda un botón que nos devolvería a la página de inicio, debajo de esto, se puede observar un breve resumen dividido en dos columnas, la de la izquierda tiene una breve explicación de las métricas y la de la derecha la manera en la que se evalúa la calidad del texto adaptado en función de las métricas, tal y como se puede ver en la Figura

### 3.6.

[Regresar a la página principal](#)

## Métricas de Evaluación

### Explicación de Métricas

**BLEU:** BLEU compara el texto traducido con textos de referencia considerados ideales. Evalúa la precisión de n-gramas, es decir, la aparición de secuencias consecutivas de palabras, ajustadas por la longitud del texto para evitar penalizar traducciones breves. Es ampliamente usado en la evaluación automática de traducciones.

**ROUGE:** Esta métrica es esencial para evaluar resúmenes automáticos. ROUGE mide cuántos n-gramas, o grupos de palabras consecutivas, el resumen generado tiene en común con uno o varios textos de referencia. Favorece la inclusión de contenido relevante sobre su precisión literal.

**METEOR:** METEOR mejora sobre BLEU al considerar sinónimos y la estructura gramatical de las oraciones. Evalúa la calidad de las traducciones no solo por la coincidencia exacta de términos, sino también por su similitud semántica, haciendo sus evaluaciones más alineadas con la interpretación humana.

**TER:** La Tasa de Edición de Traducción (TER) cuenta las operaciones mínimas necesarias para que un texto traducido coincida con una referencia. A diferencia de otras métricas, en TER, un puntaje más bajo indica una mayor precisión, ya que requiere menos modificaciones.

**WER:** La Tasa de Error por Palabra (WER) se usa predominantemente para evaluar sistemas de reconocimiento de voz. Mide el porcentaje de palabras que necesitan ser corregidas en la transcripción automática comparada con un texto de referencia.

### Evaluación de Calidad

Para evaluar la calidad de los textos adaptados, se opta por un análisis individual, el cual muestra para cada métrica el resultado y la evaluación de este y una evaluación general. La evaluación general de la calidad se determina utilizando un sistema de ponderación basado en la importancia de cada métrica. Esto asegura que las métricas más relevantes tengan un mayor impacto en la evaluación final de la calidad.

**Alta Calidad:** Se alcanza alta calidad cuando el promedio ponderado de las métricas supera un umbral ajustado, reflejando la alta precisión y fidelidad del texto adaptado con respecto al original. Los umbrales han sido diseñados para ser rigurosos pero alcanzables, garantizando que solo los textos de la más alta calidad cumplan con estos criterios.

**promedio\_ponderado > 0.75 (considerando pesos específicos para cada métrica)**

**Calidad Media:** Un texto se considera de calidad media cuando cumple con umbrales que indican una adaptación adecuada pero no excepcional. Estos umbrales son menos exigentes y permiten una mayor flexibilidad, asegurando que textos razonablemente adaptados sean reconocidos.

**0.5 < promedio\_ponderado ≤ 0.75**

**Baja Calidad:** Los textos que no alcanzan el mínimo establecido para la calidad media son clasificados como de baja calidad. Esto indica que las adaptaciones tienen desviaciones significativas del texto original, lo cual podría afectar la comprensibilidad o la fidelidad del contenido adaptado.

**promedio\_ponderado ≤ 0.5**

**Fórmula de Ponderación:**

**promedio\_ponderado = (Σ(value \* weight) / Σ(weights))**

Donde 'value' es el valor normalizado de cada métrica y 'weight' es el peso asignado basado en la importancia de esa métrica.

Figura 3.6: Cabecera Página de Respuesta

Y posteriormente se editaron todos los componentes necesarios para que el frontend quedara completamente actualizado añadiendo la nueva funcionalidad a la que ya estaba previamente implementada. Estos componentes son:

- **OverviewComponent:** Dentro de este componente, el cual se encarga de crear la pagina de inicio y ser el eje principal de la aplicacion, se le ha añadido una nueva función `OnCalculateMetrics()` la cual se encarga de manejar los archivos CSV y el cálculo de métricas a través de `/ResultService`. Verifica primero si un archivo ha sido seleccionado; si no hay archivo, notifica al usuario que seleccione uno; si hay un archivo, procede a invocar el servicio para calcular las métricas y maneja la respuesta o los errores correspondientes, esta lógica se puede observar en la Figura 3.7.

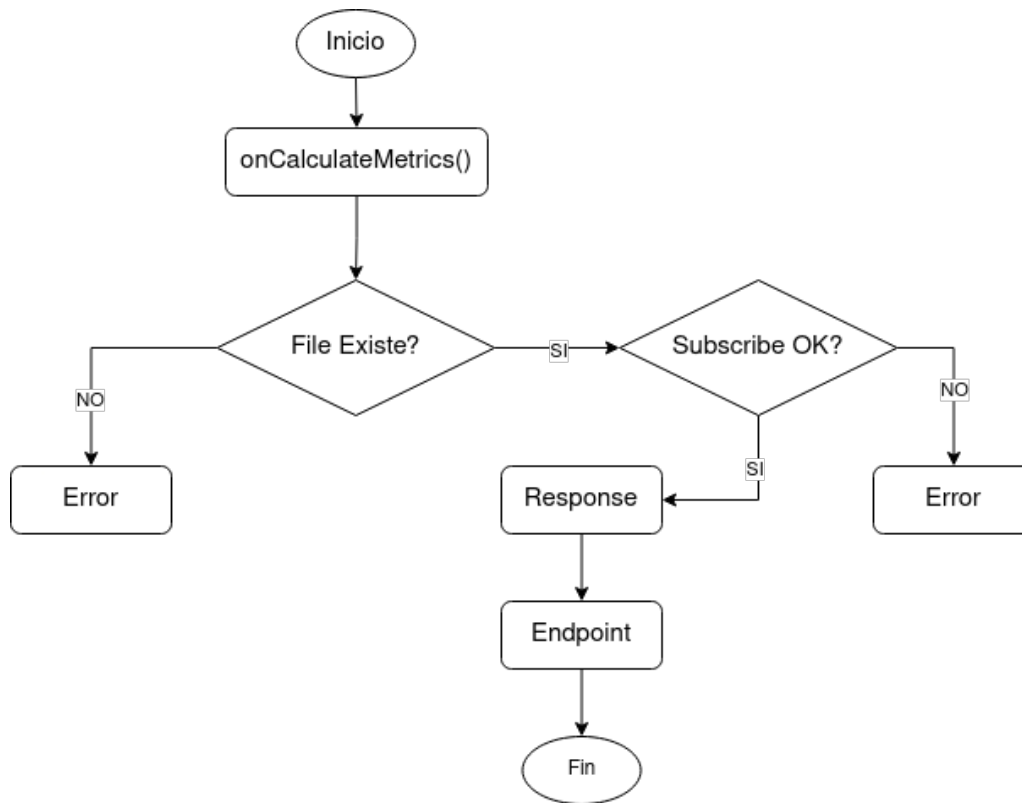


Figura 3.7: Lógica de OnCalculatMetrics

Posteriormente, se modificaron los archivos HTML y CSS, en los cuales se añadieron dos cosas, la primera fue la modificación del texto de la introducción de la página de inicio, tal y como se puede observar en la Figura 3.8, se añadieron las métricas usadas y dentro de la columna que había en la derecha se hizo una subdivisión, la primera mantuvo el texto para subir los archivos para calcular las similitudes, y en la segunda se añadió el nuevo texto donde se explica cómo subir los archivos para calcular las métricas de los textos adaptados. Y el segundo cambio que se realizó fue la creación de un segundo botón de color azul para diferenciarlo del que está justo arriba que era el que estaba creado previamente, este nuevo botón será el encargado de llevarnos a la página de resultados y ejecutar las funciones para calcular las métricas. Los cambios pueden observarse en la Figura 3.8.

¡Bienvenido a la Calculadora de Similitud y métricas de Textos! Esta aplicación te permite cargar archivos CSV que contienen pares de frases y calcular la similitud semántica entre ellas utilizando diferentes modelos de lenguaje. O también puedes cargar un archivo CSV que contiene frases y su correspondiente adaptación y calcular la calidad de su adaptación.

**Formato del archivo CSV**

El archivo CSV debe tener un formato específico para funcionar correctamente con esta aplicación. Debe contener tres columnas separadas por punto y coma (;): frase\_original, frase\_adaptada y semanticamente\_similares. Por ejemplo:

```
1 frase_original;frase_adaptada;semanticamente_similares
2 El libro fue escrito por mi amigo;mi amigo escribió el libro;SI
3 El libro fue escrito por mi amigo;El artículo fue escrito por el periodista;NO
4 El libro fue escrito por mi amigo;mi tío escribió el libro;NO
5 El libro fue escrito por Juan Gómez;Juan Gómez escribió el libro;SI
6 El libro fue escrito por Juan Gómez;el artículo fue escrito por Juan Gómez;NO
```

**Modelos de similitud semántica usados**

Los modelos de lenguaje utilizados en esta aplicación son los siguientes:

- [Maite89/Roberta\\_finetuning\\_semantic\\_similarity\\_stsb\\_multi\\_mt](#)
- [symanto/sn-xlm-roberta-base-snl1-mnli-anli-xnli](#)
- [hliamsid/sentence\\_similarity\\_spanish\\_es](#)

**Métricas usadas**

- [BLEU](#)
- [ROUGE](#)
- [METEOR](#)
- [TER](#)
- [WER](#)

**Cómo usar la aplicación**

1. Carga un archivo CSV. Asegúrate de que esté en el formato adecuado, mencionado anteriormente.

**Para calcular similitudes**

- Una vez cargado el archivo, presiona el botón "Calcular similitudes".
- La aplicación comenzará a procesar el archivo y a calcular las similitudes utilizando múltiples modelos de lenguaje. Este proceso puede tardar un poco, dependiendo del tamaño del archivo y del número de pares de frases.
- Una vez finalizado el cálculo, los resultados se mostrarán en una tabla. En ella, podrás ver las puntuaciones de similitud para cada par de frases.
- Compara los resultados de los diferentes modelos para obtener una visión completa de la similitud semántica entre las frases.

**Para calcular métricas de las adaptaciones**

- Una vez cargado el archivo, presiona el botón "Calcular métricas".
- El sistema evaluará métricas específicas basadas en criterios preestablecidos para análisis lingüístico avanzado.
- Los resultados se muestran en una tabla. Incluyen los dos textos evaluados, la puntuación de similitud (ROUGE-1, ROUGE-L y WER), la puntuación de coherencia (METEOR), la puntuación de cohesión (ROUGE-L), la puntuación de redacción (TER) y la puntuación de fluidez (BLEU). Junto a una evaluación de la calificación de estas métricas.
- Finalmente observa los resultados obtenidos para obtener una visión general de la calidad de los textos.

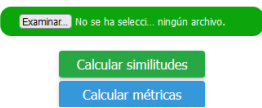


Figura 3.8: Nueva Página Principal

Y finalmente se añadió el nombre y el correo del autor de estos cambios en el pie de página. En el archivo CSS se añadió todo el código necesario para subdividir la columna de la derecha en dos, aumentar el espacio que ocupa el texto en la página y el estilo de los bordes, color y sombras del botón.

- **ResultService:** Este componente es el encargado de gestionar los resultados de ambas operaciones, las de calcular las similitudes y la de calcular las métricas. La funcionalidad que se ha añadido es la segunda, para ello se tuvo que añadir en el constructor de *ResultService* la configuración del backend lo cual establece la URL base para todas las comunicaciones con el backend, usar una variable centralizada para la URL facilita cambios futuros en la configuración del endpoint sin necesidad de modificar múltiples líneas de código. También se tuvo que añadir el constructor de e inyector de dependencias, esto se hace para poder inyectar el *HttpClient* de Angular, lo cual es crucial para poder gestionar peticiones http. Y finalmente se creó la función de *calculateMetrics()*, esta función se encarga de gestionar el envío de archivos como parte de los datos de un formulario, lo cual es necesario para la carga de archivos en el backend, su lógica se puede observar en la Figura 3.9, se añade el archivo seleccionado al objeto *FormData*, etiquetado con una clave 'file', que es como el servidor espera recibirlo y finalmente envía una solicitud POST a la url configurada, permitiendo que los componentes de la aplicación se suscriban y reaccionen cuando la respuesta esté disponible.

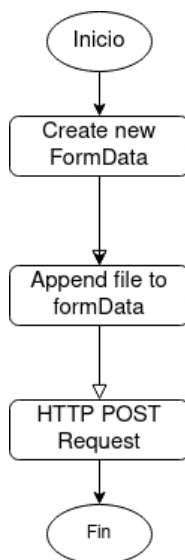


Figura 3.9: Lógica Función `CalculateMetrics()`

### 3.2.2. Implementación del Backend:

Para actualizar la aplicación sin influir en la funcionalidad que ya estaba previamente y para proporcionar una mejor escalabilidad a la aplicación se decidió crear una segunda ruta `/upload`, aquí se realizan los cálculos necesarios para realizar el análisis de las métricas. Se ha mantenido el formato para subir los archivos, aunque también se pueden proporcionar archivos sólo con dos columnas, la de el texto original y texto adaptado. La nueva funcionalidad del backend introduce nuevas funciones las cuales se encargan de realizar el análisis de las métricas, una para el análisis individual y otra para el análisis general ponderado, además de la nueva función que se encarga de gestionar todos los procesos necesarios de comunicación con el frontend. Estas funciones permiten evaluar de manera automatizada la calidad de textos adaptados mediante distintas métricas lingüísticas, facilitando así la interacción entre el frontend y los algoritmos de procesamiento de lenguaje natural. Las nuevas funciones añadidas al backend están diseñadas para calcular varias métricas de texto, como *BLEU*, *ROUGE*, *METEOR*, *TER*, y *WER*, y luego evaluar la calidad de los textos adaptados basada en estos resultados. Estas métricas ofrecen una visión completa de cómo los textos originales han sido modificados y adaptados, proporcionando una medida cuantitativa de su similitud y calidad.

- **`evaluateindividualquality()`** Esta función es la encargada de analizar las métricas de manera individual, en base a unos `thresholds`, devuelve el nivel de la adaptación con sus correspondientes colores (low-rojo, medium-amarillo, high-verde). La lógica usada y el código puede ser observados en la Figura 3.10.

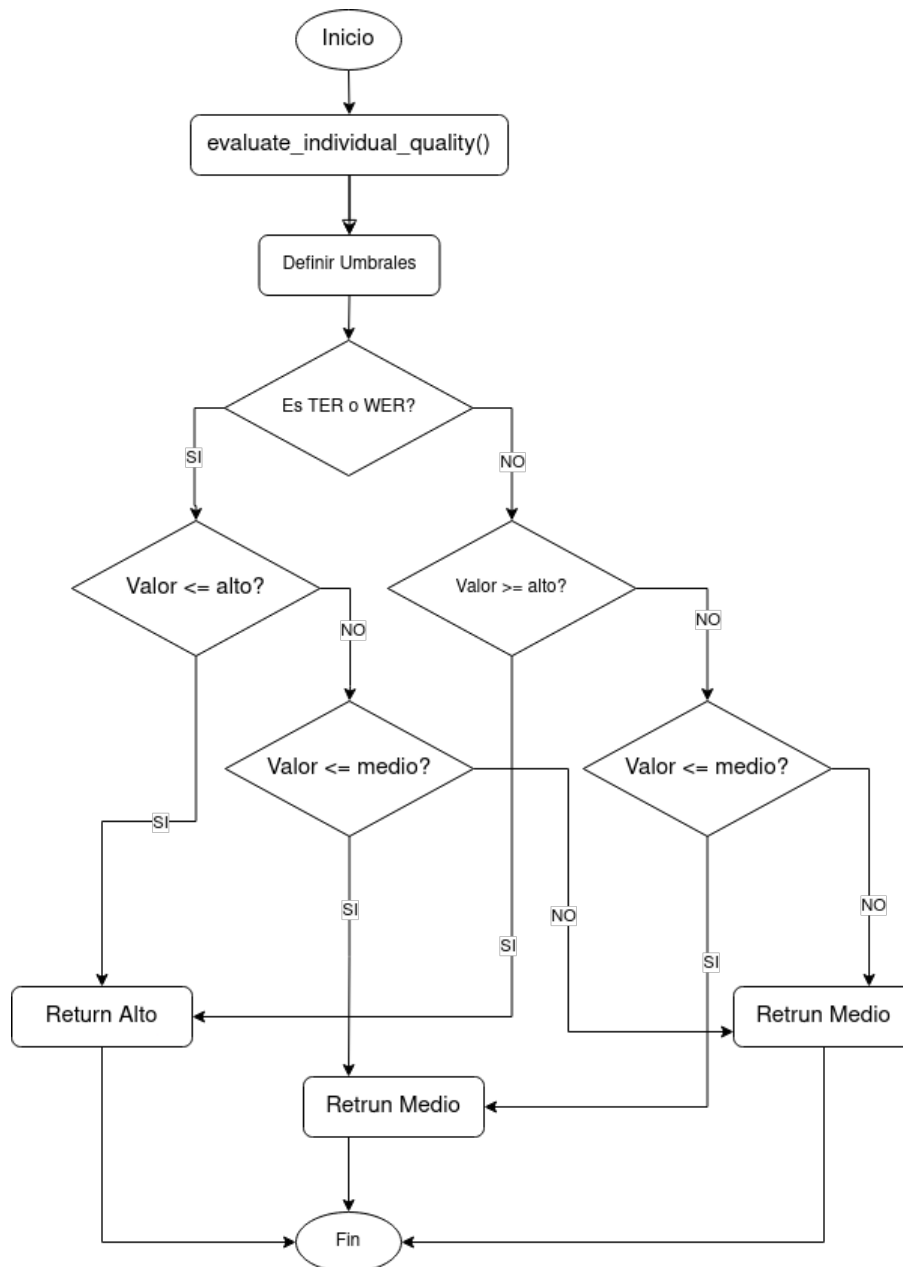


Figura 3.10: Lógica Función `evaluateindividualquality()`

- **evaluatequality()** Esta función es la encargada de analizar las métricas de forma general, pero en base a unos pesos y ponderaciones, devuelve el nivel de la adaptación con sus correspondientes colores (low-rojo, medium-amarillo, high-verde). Esta función está presentada en la Figura 3.11.

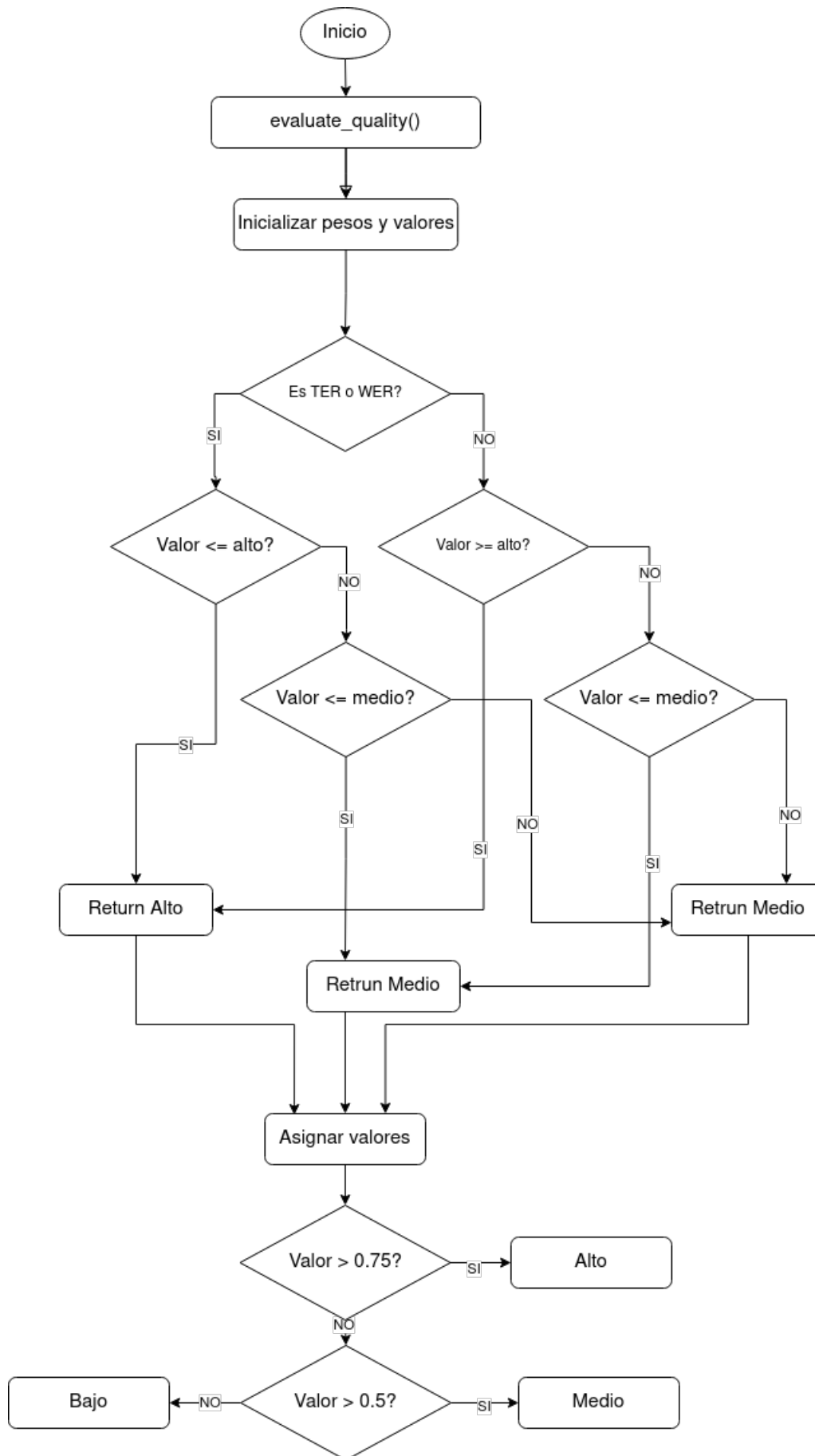


Figura 3.11: Lógica Función evaluatequality()

- **uploadfile()** Esta función es la encargada de comunicarse con el frontend, recibir el archivo CSV y enviar los resultados. Su lógica puede observarse en la imagen 3.12.

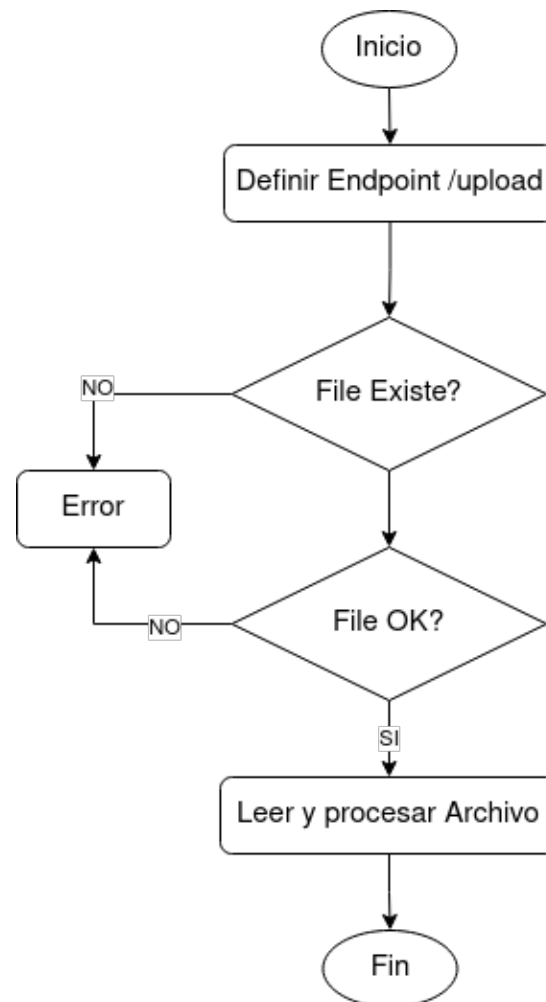


Figura 3.12: Lógica Función `uploadfile()`, segundo endpoint

- **calculatemetrics()** Esta es la función encargada de mediante las funciones de las métricas, calcular todos los resultados de los textos, tokenizando los textos para las métricas que se necesiten, excepto WER, que necesita un tratamiento especial. Esta lógica puede observarse en la Figura 3.13.

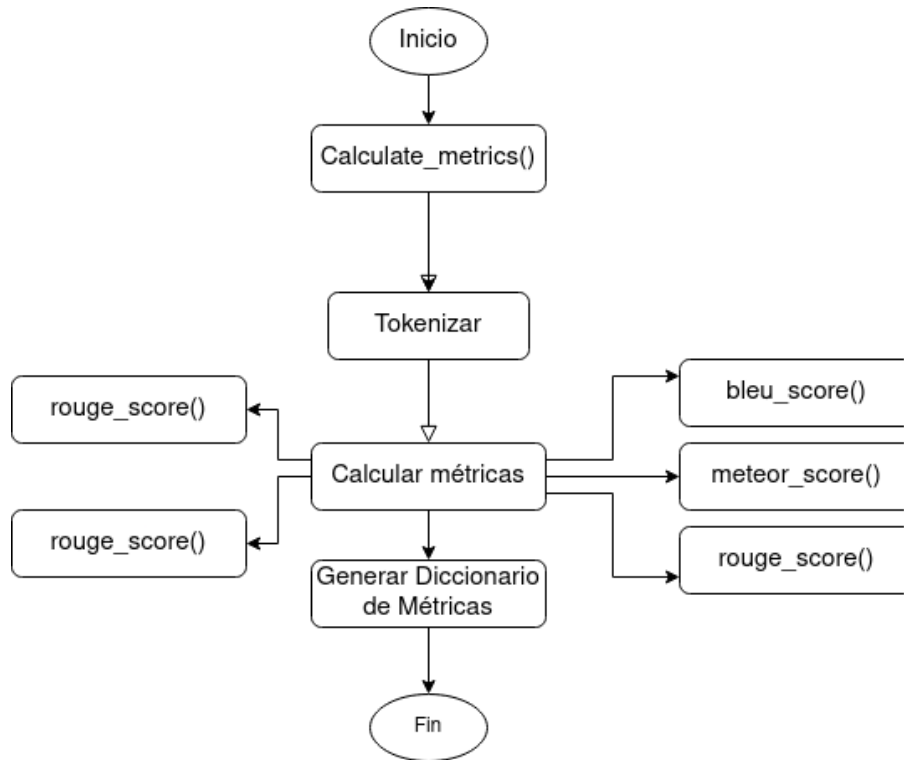


Figura 3.13: Lógica Función calculatemetrics()

- **calculatewer()** Esta función es la encargada de tokenizar y convertir a string los textos y posteriormente calcular la métrica.

### 3.3. Despliegues del Sistema

En esta sección se explica el proleso que se ha llevado a cabo para realizar los despliegues del sistema los cuales se han realizado de dos maneras, relaizando los despliegues de Frontend y Backend por separado.

#### 3.3.1. Despliegues en el Frontend

Los despliegues en frontend se han realizado de la siguiente manera:

- 1º **Desarrollo de Código:** Primero se desarrolla el código que se quiere desplegar y se revisan errores tipográficos para prevenir errores simples en la compilación.
- 2º **Compilación del Código:** En la terminal, en el repositorio del `/front` se realiza el comando `flask run` este comando intentará xompilar y desplegar el código del frontend, si hay algún error nos avisará y deberemos de corregirlo
- 3º **Revisión de los Cambios:** Cuando esté el código compilado, se comprueba en la url `http://localhost:4200/-componente-` el estado de los cambios y si son correctos.

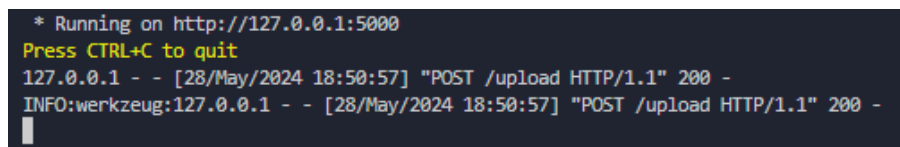
#### 3.3.2. Despliegues en el Backend

- 1º **Desarrollo de Código:** Al igual que en el Frontend, primero se desarrolla el código que se quiere desplegar y se revisan errores tipográficos para prevenir

errores simples en la compilación.

2º **Compilación del Código:** En la terminal, en el repositorio del */back* se realiza el comando *ng serve* este comando intentará compilar y desplegar el código del backend, si hay algún error nos avisará y deberemos de corregirlo.

3º **Revisión de los Cambios:** Cuando esté el código compilado, hay dos opciones de comprobar el código, manualmente accediendo a la url del front y probando la funcionalidad que se ha implementado o mediante la terminal, lo cual es más recomendable ya que además de visualizar la respuesta, también se pueden visualizar los códigos de estado que devuelve servidor, mediante el comando *curl -X POST -F 'file=@C:dir/a/tu/archivo.csv' http://localhost:5000/-endpoint-* tal y como se puede observar en la Figura 3.14.



```
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [28/May/2024 18:50:57] "POST /upload HTTP/1.1" 200 -
INFO:werkzeug:127.0.0.1 - - [28/May/2024 18:50:57] "POST /upload HTTP/1.1" 200 -
```

Figura 3.14: Respuesta del servidor después de realizar el comando CURL

## 3.4. Pruebas y Validación

En este apartado se muestran todos los tipos de pruebas que se han hecho, y se explicará el procedimiento que se ha llevado a cabo para relajarlas, además del objeto de cada una de ellas.

### 3.4.1. Pruebas Funcionales

Estas quince pruebas se realizan para comprobar si las características del código escrito cumple con los requisitos que se piden, también se comprueba que el funcionamiento de la aplicación sea correcto, y el comportamiento ante errores o situaciones que no están contempladas en los requisitos sea correcto. Cabe destacar que la totalidad de las pruebas fue completada con éxito.

Para este tipo de pruebas se han usado tests unitarios, los cuales cubren los casos comentados anteriormente. Estos tests se han realizado para cubrir la funcionalidad implementada en el Backend (Servidor), su lógica y funciones. Se han realizado un total de quince tests unitarios de los cuales quince han finalizado correctamente estos se agrupan en tres tipos de tests:

- **Tests para subir el archivo:** Se han realizado una serie de tests, concretamente seis tests unitarios para comprobar el funcionamiento y las respuestas de la aplicación ante una serie de casos; intentar calcular las métricas sin proporcionar ningún archivo, en este caso debería de saltar un error y pedir al usuario que proporcione un archivo y también se prueba si el endpoint maneja correctamente archivos vacíos o que no tengan la estructura correcta. Se comprueba si la respuesta del servidor es la que debería de llegar, recibir un estado 400 o 500.
- **Pruebas de evaluación general:** Este conjunto de cuatro pruebas unitarias se realizan para comprobar si la evaluación que se realiza de los textos de forma general, con la ponderación de las métricas, se realiza correctamente, se simula la subida de un archivo CSV, con textos y su adaptación, y se comprueba la

si la respuesta y el resultado de la evaluación es correcta y corresponde con lo esperado.

- **Pruebas de evaluación individual:** Este conjunto de cinco pruebas unitarias se realizan para comprobar el análisis individual de los textos. Al igual que en el conjunto de pruebas anterior, se simula la subida de el CSV y se comprueba que el resultado individual de las métricas sea el esperado.

En la figura 3.15 se puede observar individualmente el estado de finalización de los tests y que el 100% han concluido correctamente, también el código de estas pruebas puede observarse en el cuadro A.1.

```
PS C:\Users\aleja\OneDrive\Documents\TFG\lectura-facil\back> python -W ignore -m unittest -v tests/test_app.py
[nltk_data] Downloading package punkt to
[nltk_data] C:\Users\aleja\AppData\Roaming\nltk_data...
[nltk_data] Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data] C:\Users\aleja\AppData\Roaming\nltk_data...
[nltk_data] Package wordnet is already up-to-date!
test_upload_file_invalid_csv (tests.test_app.AdditionalTests) ... ok
test_upload_file_no_file_part (tests.test_app.AdditionalTests) ... ok
test_upload_file_no_selected_file (tests.test_app.AdditionalTests) ... ok
test_upload_file_success (tests.test_app.AdditionalTests) ... ok
test_evaluate_individual_quality_high (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_individual_quality_high_ter (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_individual_quality_low (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_individual_quality_low_ter (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_individual_quality_medium (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_quality_high (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_quality_low (tests.test_app.EvaluationTestCase) ... ok
test_evaluate_quality_medium (tests.test_app.EvaluationTestCase) ... ok
test_get_similarity (tests.test_app.FlaskTestCase) ... ok
test_upload_file_invalid_csv (tests.test_app.FlaskTestCase) ... ok
test_upload_file_no_file_part (tests.test_app.FlaskTestCase) ... ok
-----
Ran 15 tests in 1.145s

OK
```

Figura 3.15: Respuesta del servidor después de realizar el comando CURL

#### 3.4.2. Pruebas de Usabilidad

Las pruebas de usabilidad se han realizado desde el frontend, y se ha probado todas las funcionalidades que han sido implementadas. Primero se probó el funcionamiento y la eficacia de la página de inicio con los dos botones, demostrando que es una interfaz eficiente y bien implementada. Luego se probó, el botón de la página de resultados, que también resultó ser intuitivo, y finalmente el análisis de la tabla de resultados, que al tener una variedad de colores muy significativos, resultó ser muy intuitiva y fácil de interpretar.

#### 3.4.3. Pruebas de Rendimiento

Las pruebas de rendimiento se realizaron con JMeter, configurando el thread group que es el número de usuarios virtuales que accederán a la aplicación simultáneamente, se hicieron pruebas con 0, 25, 50, 100, 125, 175 y 200 usuarios concurrentes, tiempo de ram-up o tiempo que estarán activos estos usuarios, configurado a 1 segundo como máximo y el número de ciclos, que es el número de veces que intentarán la prueba estos usuarios, también configurado a 1; También se configuró el endpoint al que se iba a acceder y los métodos http que se usarán con su correspondiente contenido, siendo el endpoint /upload para probar la funcionalidad de las métricas y un método POST, con un archivo de prueba con varios textos y su adaptación; y

finalmente se configuraron las aserciones o códigos de respuesta esperados. Después de hacer esta prueba, podemos afirmar que la aplicación soporta cargas moderadas de usuarios concurrentes (hasta 100 usuarios) sin ningún error. El resultado de esta prueba con 100 usuarios concurrentes se muestra en el cuadro A.2.

### 3.5. Resultados y Análisis

En esta sección se analizan los resultados obtenidos con la implementación de la nueva funcionalidad de la aplicación. Además de todas las métricas usadas, la manera que se han implementado y los pesos y la ponderación usada para calcular los resultados en el análisis general.

La aplicación ha sufrido una mejora en su funcionalidad y ligeramente en su aspecto, desde la primera versión que se creó hasta la versión que está en uso actualmente, la cual presenta una mejora significativa en varios aspectos clave; ampliando su funcionalidad y la interfaz de usuario. Estos cambios han tenido un gran impacto en la eficiencia operativa de la aplicación y en la experiencia del usuario.

Anteriormente, los usuarios solo podrían calcular la similitud semántica entre textos, ahora además de esto, si tenemos un texto con su correspondiente adaptación a Lectura Fácil, con la nueva funcionalidad que se ha implementado en esta aplicación, que influye significativamente tanto en la parte funcional como en la parte técnica, el usuario recibe una evaluación de la calidad del texto adaptado. Esta evaluación consta de dos partes, la integración de una funcionalidad que permite a los usuarios realizar un análisis general ponderado y otra individual, por cada métrica usada, esta nueva herramienta facilita un análisis profundo de los textos, que se devuelve de una manera muy intuitiva y fácil de comprender, aumentando así la utilidad práctica de la aplicación para fines educativos y de investigación.

También, al decidir que se iba a dividir la estructura de la aplicación y crear un nuevo endpoint, se deja entrever el impacto positivo que tiene para la futura escalabilidad de la aplicación, al separar los endpoints, también separa carga de trabajo al servidor, permitiendo unas llamadas a los endpoints más ligeras y que tendrán una respuesta más rápida. Esto también mejora la robustez en la carga de archivos, dando un mejor servicio durante la gestión de errores y asegurando que los datos se carguen y procesen correctamente.

En cuanto al impacto visual, la aplicación no ha sufrido grandes cambios, se han añadido los textos necesarios para que el usuario sea capaz de poder usar fácilmente la nueva funcionalidad, un nuevo botón en la página principal, el cual se encarga de calcular las métricas y redirigir al usuario a la página de respuesta correspondiente, en la cual se muestran los resultados en una tabla, y un encabezado previo para poder comprender que evalúan estas métricas y cómo se hacen estas evaluaciones.

#### 3.5.1. Análisis Individual

Para calcular el análisis individual se ha seguido el siguiente procedimiento:

- Primero se calculan los resultados de las métricas comentadas en la sección 2.2.
- Posteriormente con el resultado obtenido, podemos clasificar si según esa métrica la adaptación del texto tiene un nivel alto, medio o bajo:

- **Alto:** Este nivel se mostraría como HIGH y en color verde. Según las métricas que se calculan ( BLEU, ROUGE [ROUGE-1 Y ROUGE-L], METEOR, TER y WER ) se considera como una adaptación con nivel alto (HIGH) a cualquiera que obtenga un resultado mayor que 0.5 en BLEU y METEOR, 0.6 en ROUGE-1 y ROUGE-L, menos que 0.4 en TER y 0.3 en WER.
- **Medio:** Este nivel se mostraría como MEDIUM en color amarillo. Para considerar la adaptación como media (MEDIUM) a cualquiera de las adaptaciones que obtengan un valor mayor que 0.3 y menor que 0.5 en BLEU, un valor mayor que 0.35 y menor que 0.6 en ROUGE-1 y ROUGE-L, un valor mayor que 0.25 y menor que 0.5 en METEOR, un valor menor que 0.5 y mayor que 0.4 en TER y un valor menor que 0.4 y mayor que 0.3 en WER.
- **Bajo:** Este nivel se mostraría como LOW en color rojo. Finalmente si para cada una de las métricas de BLEU, ROUGE y METEOR es menor que cualquiera de los valores menores de estas se consideraría de baja calidad, al igual que si supera al máximo de los valores de TER y WER.

#### 3.5.2. Análisis General Ponderado

Para realizar el análisis general se ha seguido el siguiente procedimiento:

- Primero se debe realizar una asignación de pesos a las métricas a calcular, los pesos se asignan de menor a mayor influencia en el resultado final:
  - Los más influyentes serían **METEOR y ROUGE**, ya que la evaluación que realiza METEOR está más alineada con la percepción de los seres humanos y la evaluación que realiza ROUGE sobre resúmenes y generación de texto es más completa que el resto de las métricas, a estas dos métricas se les asigna un peso de 3.
  - Después a **TER y WER** se les asigna un peso de 2, ya que estas dos métricas no tienen tan en cuenta la fluidez y se centran más en la corrección técnica.
  - Finalmente a **BLEU** se le asigna un peso de 1, ya que BLEU y METEOR realizan tareas parecidas y al realizar METEOR un análisis más completo que BLEU, recibe un peso menos relevante en el resultado.

Después de haber asignado los pesos, se realiza el cálculo del resultado ponderado de la siguiente manera:

- Se realiza el cálculo de cada métrica, con el mismo procedimiento que en el análisis individual.
- Dependiendo de su resultado se suma al total ponderado en su respectiva categoría.
  - **Alta:** Se suman todos los elementos que estén en esta categoría y el resultado final se multiplica por 1.
  - **Media:** Se suman todos los elementos que estén en esta categoría y el resultado final se multiplica por 0.5.
  - **Baja:** Se suman todos los elementos que estén en esta categoría y el resultado final se multiplica por 0.25.
- Se realiza un sumatorio de los resultados ponderados.

- Finalmente se divide por el total de los pesos, y el resultado se clarificará como alto, medio o bajo, de igual manera que en el análisis individual.

### 3.5.3. Resultados de las métricas

Todas las cinco métricas son usadas para evaluar la calidad de los textos adaptados, cada una con sus ventajas y aplicaciones. Podemos afirmar que BLEU y ROUGE son métricas muy útiles para el análisis de textos resumidos, ya que aprecia la retención de información relevante, pero carece de un análisis semántico. METEOR, mejora sobre las anteriores al incluir sinónimos y la estructura gramatical, este proporciona una evaluación más alineada con la percepción humana, para esta métrica se ha usado ROUGE-1 y ROUGE-L, la primera se basa en la coincidencia de palabras individuales entre el sistema y los textos adaptados de referencia y la segunda se basa en la subsecuencia común más larga, detecta los grupos de palabras más largos que aparecen seguidos en el texto adaptado. Y, finalmente TER y WER las cuales se usan para medir la cantidad de ediciones necesarias y la precisión a nivel de palabras, estas dos métricas no evalúan la calidad semántica de los cambios.

#### 3.5.3.1. BLEU

Esta métrica es de gran utilidad en caso de necesitar un análisis rápido, de cuántas palabras o cuántas frases del texto inicial se mantienen iguales en el texto adaptado. El principal problema que tiene BLEU es que no tiene en cuenta sinónimos ni la estructura gramatical del texto, esto puede llevar a penalizar adaptaciones que son válidas pero al no tener palabras o frases que se mantengan con un significado equivalente al original, penalizando textos más cortos o más largos.

#### 3.5.3.2. ROUGE

ROUGE es especialmente efectivo cuando se necesita medir la calidad de contenido relevante que se ha mantenido al hacer una adaptación automática es de gran importancia, esta métrica mide la coincidencia de n-gramas o subsecuencia común más larga proporcionando una medida de la calidad del contenido retenido. Al igual que BLEU, ROUGE no contempla sinónimos ni la estructura gramatical del texto, por lo que la calidad del resultado de esta métrica puede verse afectada en el análisis de textos muy cortos o muy largos ya que depende de la coincidencia de las cadenas.

#### 3.5.3.3. METEOR

Esta métrica es de gran importancia cuando la flexibilidad y el contexto semántico son de gran importancia, como pueden ser textos educativos o adaptaciones literarias. METEOR es más preciso que BLEU y ROUGE en casos como el anterior, ya que este si que tiene en cuenta sinónimos, la estructura gramatical y usa alineaciones a nivel de frase, por lo que se alinea más con las evaluaciones realizadas por humanos. El principal inconveniente que tiene esta métrica es que al tener en cuenta más recursos literarios que las anteriores requiere de un poder computacional mayor y su cálculo consume la mayoría de los recursos y puede ser más lento.

#### 3.5.3.4. TER

TER es útil para la evaluación de la cantidad de trabajo que se necesitará para la adaptación de un texto, especialmente en procesos que requieran una revisión y

evaluación, en traducciones y en la evaluación de herramientas de post edición de traducciones automáticas. Mide el número de ediciones que se necesitarían para convertir para covertir el texto original en el texto adaptado, mostrando claramente el esfuerzo de edición requerido. El principal problema que tiene esta métrica es que no considera la calidad de la adpatación ni si los cambios que se han hehco al texto original lo mejoran o lo empeoran.

#### 3.5.3.5. WER

Finalmente WER es de gran utilidad para evaluar transcripciones o adaptaciones donde la precisión a nivel de palabra es de gran importancia, se suele usar para evaluar textos generados pro voz y sus posteriores adaptaciones. Mide los errores que se generan en estas transcripciones a nivel de palabra, mostrando un resultado detallado de las inexactitudes en el texto adaptado. El principal inconveniente que tiene esta métrica es similar a la de TER, no considera si los cambios son semánticamente correctos, es capaz de penalizar palabras que no afectan el en sentido del texto, pero que son erróneas, lo que generaría adaptaciones que no son correctas.

Todo esto, se puede observar en la Figura 3.16, donde podemos ver los resultados de cada métrica y también los análisis individuales y el general ponderado.

Métricas Calculadas	Evaluación Individual de la Calidad	Evaluación General de la Calidad
BLEU: 0.04 ROUGE-1: 0.67 ROUGE-L: 0.33 METEOR: 0.57 TER: 1.17 WER: 0.88	BLEU: <b>LOW</b> METEOR: <b>HIGH</b> ROUGE-1: <b>HIGH</b> ROUGE-L: <b>LOW</b> TER: <b>LOW</b> WER: <b>LOW</b>	<b>MEDIUM</b>

Figura 3.16: Resultados obtenidos y análisis

## Capítulo 4

# Conclusiones y Líneas Futuras

En este capítulo se explican las conclusiones finales y personales sobre este proyecto, reflejando los hallazgos más significativos y el aprendizaje obtenido, también, se tratan posibles líneas de investigación y trabajos que pueden derivarse de este proyecto en el futuro.

### 4.1. Impacto del trabajo

En esta sección se analiza el impacto de este trabajo, se evalúan los aportes de este proyecto ha realizado al ámbito de la adaptación de textos a la Lectura Fácil, y su relevancia en el ámbito académico y de la investigación.

La implementación de esta nueva funcionalidad en la antigua aplicación, ha tenido un impacto de gran magnitud en la eficiencia operativa y en la experiencia del usuario al interactuar con la aplicación. Al introducir una nueva herramienta la cual se encarga de analizar textos adaptados a Lectura Fácil, ha permitido ofrecer al usuario final una evaluación eficaz, completa, detallada y fácil de comprender ya que ofrece una respuesta visual a esta evaluación, esto no solo ha mejorado la precisión de la aplicación, sino que también ofrece una mayor comprensión y mejora la adaptación de los textos proporcionados. Ahora los usuarios, pueden acceder a una evaluación extra de textos, no solo calcular su nivel de similitud, lo cual es fundamental para asegurar que los textos adaptados cumplan con los estándares necesarios para ser considerados de Lectura Fácil como puede ser un Lenguaje claro y simple o que el texto mantenga un diseño visual accesible, estos estándares están definidos Standards for easy-to-read materials - IFLA [2].

Si tenemos en cuenta un punto de vista más técnico, esta nueva implementación ha hecho que la aplicación mejore su respuesta con el nuevo endpoint ante el manejo de grandes volúmenes de datos, incrementando su escalabilidad y aumentando el rendimiento de la aplicación ante un gran número de usuarios concurrentes. Al introducir la evaluación de métricas textuales como son BLEU, ROUGE, METEOR, TER y WER, se ha proporcionado una nueva base para poder realizar evaluaciones cuantitativas y precisas de textos adaptados, cada una de estas métricas cubren diferentes aspectos sobre la calidad textual, o mejoran las capacidades unas sobre otras. Estas métricas ofrecen a los usuarios una perspectiva más visual, permitiéndolos evaluar tanto la calidad o exactitud de la adaptación como la naturalidad o la fluidez de estos textos de una manera más fácil de entender para el usuario como para el público objetivo.

La aplicación aporta a investigadores y educadores una herramienta útil y versátil para el análisis de textos, proporcionando una plataforma que puede ser utilizada para varias finalidades como pueden ser la comprobación de la calidad de textos adaptados por usuarios individuales, profesionales o máquinas o también el cálculo de la similitud entre dos textos escritos. Esta nueva capacidad de realizar análisis detallados de textos abre nuevas oportunidades para investigaciones o estudios académicos centrados en mejorar la accesibilidad de la información a personas con capacidades reducidas, lo que abre una puerta al desarrollo de nuevas metodologías para el análisis de textos avanzados. Además la mejora en la escalabilidad y la flexibilidad del sistema aseguran que se puedan incorporar en la aplicación nuevos avances tecnológicos, nuevas métricas o nuevas metodologías de una manera sencilla y efectiva, permitiéndola una fácil adaptación a las necesidades que engasnan los usuarios.

Desarrollando este TFG se ha logrado mejorar la funcionalidad y usabilidad de la aplicación y también ha proporcionado un marco sólido para futuras investigaciones y desarrollos encaminados hacia la Lectura Fácil y el análisis de textos adaptados. Estas mejoras funcionales y técnicas que se han implementado comentadas anteriormente, han generado una herramienta más robusta y eficaz, capaz de ofrecer evaluaciones detalladas y precisas las cuales son de gran importancia para la adaptación efectiva de textos, lo cual representa un gran avance en la creación de herramientas accesibles y eficientes para la adaptación de textos, contribuyendo no solo al campo de la Lectura Fácil sino también al ámbito académico y de la investigación.

## 4.2. Conclusiones personales

En esta sección se exploran las conclusiones personales sobre este TFG, sobre su desarrollo, los desafíos que se han enfrentado y las lecciones aprendidas durante su realización, también se comparten observaciones sobre la experiencia de investigación y del proceso de desarrollo, destacando los aspectos más gratificantes y aquellos que podrían mejorarse en futuros trabajos.

En cuanto al desarrollo personal, este TFG ha proporcionado una visión de unas necesidades que no se tenían en cuenta en el momento de escribir y desarrollar textos como puede ser que estos textos sigan las bases de la Lectura Fácil y sean sencillos de leer y comprender. Además, se han mejorado las habilidades personales en el ámbito de la programación web, se ha adquirido una visión de desarrollo a futuro, esto quiere decir que en todo momento hay que ser consciente de los posibles cambios que puede sufrir una aplicación, y programarla teniendo esto en cuenta, así se facilita el trabajo de comprensión, implementación y despliegue para futuros desarrollos.

Por una parte, el principal desafío o inconveniente que se han encontrado durante el desarrollo de este TFG ha sido la integración de las diferentes métricas de evaluación de textos, ya que cada una tiene un uso diferente, la necesidad de equilibrar su influencia en la evaluación general con su usabilidad mediante la ponderación, ha requerido un profundo conocimiento técnico de las métricas BLEU, ROUGE, METEOR, TER y WER. Otro desafío encontrado fue la decisión de añadir un segundo endpoint a la aplicación pudiendo mantener la funcionalidad anterior y el proceso de carga de archivos intacto, realizando todo el procesamiento en el nuevo endpoint, frente a la posibilidad de desarrollarlo todo junto en el endpoint original, esto podría parecer más fácil en el momento del desarrollo, pero posteriormente podría generar una gran cantidad de conflictos con los archivos o con la elección del cálculo a realizar, también empeoraría la organización del código, la respuesta de la aplicación y finalmente las

pruebas de la nueva funcionalidad también generarían conflictos con las antiguas.

Por otra parte, uno de los aspectos más gratificantes fue observar cómo las mejoras implementadas incrementaron significativamente la respuesta de la aplicación y su utilidad. Al añadir esta nueva funcionalidad, el alcance de la aplicación aumentó, ofreciendo un análisis más detallado de los textos adaptados. Esto mejora considerablemente lo que el usuario puede hacer con la aplicación, proporcionando un avance significativo en la creación de herramientas que mejoran la accesibilidad de los usuarios a una mayor cantidad de textos adaptados y con una mayor calidad en la adaptación, contribuyendo positivamente al campo de la Lectura Fácil y al ámbito académico y de la investigación.

### 4.3. Trabajo futuro

Esta sección se dedica a sugerir los siguientes pasos que podrían tomarse para expandir y mejorar el desarrollo realizado en este Trabajo de Fin de Grado, además de nuevas líneas de investigación.

Posiblemente, añadir una nueva funcionalidad en la aplicación sería de gran ayuda. Actualmente, con el auge de las inteligencias artificiales, la implementación de una nueva herramienta, un modelo de lenguaje o inteligencia artificial generativa que se base en los estándares de la Lectura Fácil [2], crearía una herramienta de gran utilidad en campos de investigación y educativos. Esto facilitaría las tareas de adaptación de textos y, junto a las dos funcionalidades existentes que calculan la similitud entre textos y la calidad de los textos adaptados, podrían crearse adaptaciones de gran calidad en un tiempo significativamente menor.

También sería beneficioso para los usuarios a ambos lados, tanto para los usuarios que se encargan de realizar las adaptaciones, como para los que necesitan estas adaptaciones, la implementación de un sistema de feedback, permitiendo a los usuarios proporcionar comentarios sobre las adaptaciones generadas por otros usuarios, lo cual podría usarse para entrenar los modelos de inteligencia artificial, asegurando que las adaptaciones sean más precisas con el paso del tiempo.

Finalmente, otra línea de investigación interesante podría ser la implementación de adaptaciones de textos a otros idiomas, aumentando el alcance de la aplicación a nivel internacional, la implementación de un sistema que reconozca varias lenguas con el mismo nivel de precisión permitiría evaluar y adaptar textos en varios idiomas, aumentando así la accesibilidad de la información a una audiencia global.

# Bibliografía

- [1] Saeed Alshahrani, Ejaz Ahmed y Rupert Ward. «The influence of online resources on student–lecturer relationship in higher education: a comparison study». En: *Journal of Computers in Education* 4.2 (2017), págs. 87-106. URL: <https://rdcu.be/dMf9J>.
- [2] International Federation of Library Associations e Institutions. «Guidelines for Easy-to-Read Materials». En: (2011). URL: <https://www.ifla.org/publications/ifla-professional-reports-120>.
- [3] Inclusion Europe. «Información para todos: Las reglas europeas para hacer información fácil de leer y comprender». En: *Inclusion Europe* (2017). URL: [https://www.inclusion-europe.eu/wp-content/uploads/2017/06/ES\\_Information\\_for\\_all.pdf](https://www.inclusion-europe.eu/wp-content/uploads/2017/06/ES_Information_for_all.pdf).
- [4] Jesús Vivo Castro. «Lectura fácil: similitud semántica». Trabajo de Fin de Grado. E.T.S. de Ingenieros Informáticos (UPM), mayo de 2023.
- [5] Traceloop. «Demystifying the BLEU Metric: A Comprehensive Guide to Machine Translation Evaluation». En: (2023). URL: <https://www.traceloop.com/demystifying-the-bleu-metric/>.
- [6] Chin-Yew Lin. «ROUGE: A Package for Automatic Evaluation of Summaries». En: *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*. 2004, págs. 74-81. URL: <https://aclanthology.org/W04-1013.pdf>.
- [7] Towards AI. «ROUGE Metrics: Evaluating Summaries in Large Language Models». En: (2023). URL: <https://towardsai.net/p/nlp/rouge-metrics-evaluating-summaries-in-large-language-models>.
- [8] Satanjeev Banerjee y Alon Lavie. «METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments». En: (2005). URL: <https://www.cs.cmu.edu/~alavie/METEOR/pdf/Banerjee-Lavie-2005-METEOR.pdf>.
- [9] Abhilasha Sharma, Raghav Aggarwal y Raghav Alawadhi. «A Comparative Study of Text Summarization using Gensim, NLTK, Spacy, and Sumy Libraries». En: *Journal of Xi'an Shiyou University, Natural Science Edition* 19.4 (2023), págs. 1038-1043. URL: <https://www.xisdxjxsu.asia/V19I04-90.pdf>.
- [10] W3C Web Accessibility Initiative. «Introduction to Web Accessibility». En: *World Wide Web Consortium (W3C)*. 2024. URL: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.
- [11] World Wide Web Consortium (W3C). «Web Content Accessibility Guidelines (WCAG) Overview». En: *W3C* (2024). URL: <https://www.w3.org/WAI/standards-guidelines/wcag/>.
- [12] EMB Global. «Web Accessibility in 2024: Building Inclusive and Universal Web Experiences». En: *EMB Global Blog* (2024). URL: <https://blog.emb.global/web-accessibility-in-2024>.

# Apéndice A

## Anexo

### A.1. Cuadros y tablas

Cuadro A.1: Tabla de tests y código asociado

Tipo de Tests	Código
Test de subida de archivo con éxito	<pre>1 @patch('app.pd.read_csv') 2 def test_upload_file_success(self, mock_read_csv): 3     # Mock del DataFrame 4     mock_read_csv.return_value = pd.DataFrame({ 5         'frase_original': ['Hola Como estas'], 6         'frase_adaptada': ['Hola Que tal'] 7     }) 8 9     # Creación del archivo CSV falso 10    data_csv = """ 11    frase_original;frase_adaptada 12    Hola Cmo est s?;Hola Qu tal? 13    """ 14    mock_csv = (BytesIO(data_csv.encode()), "mock.csv") 15 16    # Llamada al endpoint '/upload' con el archivo CSV 17    # falso 18    response = self.app.post( 19        '/upload', 20        content_type='multipart/form-data', 21        data={'file': mock_csv} 22    ) 23 24    # Comprobamos que el código de estado de la 25    # respuesta es 200 (OK) 26    self.assertEqual(response.status_code, 200) 27    response_data = response.get_json() 28    self.assertIn('original', response_data[0]) 29    self.assertIn('adapted', response_data[0]) 30    self.assertIn('metrics', response_data[0]) 31    self.assertIn('overall_quality', response_data[0]) 32    self.assertIn('individual_qualities', response_data[0])</pre>

Continúa en la próxima página

**Cuadro A.1 – continúa de la página anterior**

Tipo de Tests	Código
Test sin parte de archivo	<pre> 1 def test_upload_file_no_file_part(self): 2     response = self.app.post('/upload', content_type='       multipart/form-data') 3     self.assertEqual(response.status_code, 400) 4     self.assertEqual(response.get_json()['error'], "No file       part") </pre>
Test de archivo CSV inválido	<pre> 1 @patch('app.pd.read_csv') 2 def test_upload_file_invalid_csv(self, mock_read_csv): 3     mock_read_csv.side_effect = pd.errors.ParserError("Error       parsing CSV") 4 5     data_csv = "this is not a valid CSV file" 6     mock_csv = (BytesIO(data_csv.encode()), "mock.csv") 7 8     response = self.app.post( 9         '/upload', 10        content_type='multipart/form-data', 11        data={'file': mock_csv} 12    ) 13 14    self.assertEqual(response.status_code, 500) 15    response_data = response.get_json() 16    self.assertIn('error', response_data) 17    self.assertEqual(response_data['error'], "Error parsing       CSV") </pre>
Test de subida de archivo sin parte de archivo	<pre> 1 def test_upload_file_no_file_part(self): 2     response = self.app.post('/upload', content_type='       multipart/form-data') 3     self.assertEqual(response.status_code, 400) 4     self.assertEqual(response.get_json()['error'], "No file       part") </pre>
Test de subida de archivo sin archivo seleccionado	<pre> 1 def test_upload_file_no_selected_file(self): 2     response = self.app.post( 3         '/upload', 4         content_type='multipart/form-data', 5         data={'file': (BytesIO(b''), '')} 6     ) 7     self.assertEqual(response.status_code, 400) 8     self.assertEqual(response.get_json()['error'], "No       selected file") </pre>

Continúa en la próxima página

Cuadro A.1 – continúa de la página anterior

Tipo de Tests	Código
Test de evaluación de calidad alta	<pre>1 def test_evaluate_quality_high(self): 2     metrics = { 3         "BLEU": 0.6, 4         "ROUGE-1": 0.7, 5         "ROUGE-L": 0.7, 6         "METEOR": 0.6, 7         "TER": 0.3, 8         "WER": 0.2 9     } 10    weights = { 11        "BLEU": 1, 12        "ROUGE-1": 1, 13        "ROUGE-L": 1, 14        "METEOR": 1, 15        "TER": 1, 16        "WER": 1 17    } 18    result = evaluate_quality(metrics, weights) 19    self.assertEqual(result, "high")</pre>
Test de evaluación de calidad media	<pre>1 def test_evaluate_quality_medium(self): 2     metrics = { 3         "BLEU": 0.4, 4         "ROUGE-1": 0.5, 5         "ROUGE-L": 0.5, 6         "METEOR": 0.3, 7         "TER": 0.4, 8         "WER": 0.3 9     } 10    weights = { 11        "BLEU": 1, 12        "ROUGE-1": 1, 13        "ROUGE-L": 1, 14        "METEOR": 1, 15        "TER": 1, 16        "WER": 1 17    } 18    result = evaluate_quality(metrics, weights) 19    self.assertEqual(result, "medium")</pre>

Continúa en la próxima página

Cuadro A.1 – continúa de la página anterior

Tipo de Tests	Código
Test de evaluación de calidad baja	<pre> 1 def test_evaluate_quality_low(self): 2     metrics = { 3         "BLEU": 0.1, 4         "ROUGE-1": 0.2, 5         "ROUGE-L": 0.2, 6         "METEOR": 0.1, 7         "TER": 0.6, 8         "WER": 0.5 9     } 10    weights = { 11        "BLEU": 1, 12        "ROUGE-1": 1, 13        "ROUGE-L": 1, 14        "METEOR": 1, 15        "TER": 1, 16        "WER": 1 17    } 18    result = evaluate_quality(metrics, weights) 19    self.assertEqual(result, "low") </pre>
Test de evaluación individual de calidad alta	<pre> 1 def test_evaluate_individual_quality_high(self): 2     result = evaluate_individual_quality(0.6, "BLEU") 3     self.assertEqual(result, "high") </pre>
Test de evaluación individual de calidad media	<pre> 1 def test_evaluate_individual_quality_medium(self): 2     result = evaluate_individual_quality(0.4, "BLEU") 3     self.assertEqual(result, "medium") </pre>
Test de evaluación individual de calidad baja	<pre> 1 def test_evaluate_individual_quality_low(self): 2     result = evaluate_individual_quality(0.2, "BLEU") 3     self.assertEqual(result, "low") </pre>
Test de evaluación individual de calidad alta para TER	<pre> 1 def test_evaluate_individual_quality_high_ter(self): 2     result = evaluate_individual_quality(0.3, "TER") 3     self.assertEqual(result, "high") </pre>
Test de evaluación individual de calidad baja para TER	<pre> 1 def test_evaluate_individual_quality_low_ter(self): 2     result = evaluate_individual_quality(0.6, "TER") 3     self.assertEqual(result, "low") </pre>

Cuadro A.2: Tabla de resultado prueba de carga

<b>elapsed</b>	<b>label</b>	<b>responseCode</b>	<b>responseMessage</b>	<b>success</b>	<b>bytes</b>	<b>sentBytes</b>
2217	HTTP Request	200	OK	true	27204	12862
2229	HTTP Request	200	OK	true	27204	12871
2908	HTTP Request	200	OK	true	27204	12847
3878	HTTP Request	200	OK	true	27204	12844
4781	HTTP Request	200	OK	true	27204	12844
4813	HTTP Request	200	OK	true	27204	12859
5370	HTTP Request	200	OK	true	27204	12868
5720	HTTP Request	200	OK	true	27204	12865
6642	HTTP Request	200	OK	true	27204	12853
7280	HTTP Request	200	OK	true	27204	12850
7386	HTTP Request	200	OK	true	27204	12874
7957	HTTP Request	200	OK	true	27204	12850
8100	HTTP Request	200	OK	true	27204	12865
9082	HTTP Request	200	OK	true	27204	12847
9531	HTTP Request	200	OK	true	27204	12850
10210	HTTP Request	200	OK	true	27204	12853
10920	HTTP Request	200	OK	true	27204	12847
11204	HTTP Request	200	OK	true	27204	12871
11314	HTTP Request	200	OK	true	27204	12844
11910	HTTP Request	200	OK	true	27204	12856
12046	HTTP Request	200	OK	true	27204	12847
12819	HTTP Request	200	OK	true	27204	12865
13305	HTTP Request	200	OK	true	27204	12871
13525	HTTP Request	200	OK	true	27204	12856
13589	HTTP Request	200	OK	true	27204	12850
14072	HTTP Request	200	OK	true	27204	12865
14467	HTTP Request	200	OK	true	27204	12847
14986	HTTP Request	200	OK	true	27204	12847
15640	HTTP Request	200	OK	true	27204	12850
15693	HTTP Request	200	OK	true	27204	12874
15900	HTTP Request	200	OK	true	27204	12862
16510	HTTP Request	200	OK	true	27204	12850
16523	HTTP Request	200	OK	true	27204	12865
16532	HTTP Request	200	OK	true	27204	12847
16728	HTTP Request	200	OK	true	27204	12853
17675	HTTP Request	200	OK	true	27204	12850
18739	HTTP Request	200	OK	true	27204	12859
19281	HTTP Request	200	OK	true	27204	12856
19946	HTTP Request	200	OK	true	27204	12850
19952	HTTP Request	200	OK	true	27204	12844
20449	HTTP Request	200	OK	true	27204	12844
20917	HTTP Request	200	OK	true	27204	12847
20994	HTTP Request	200	OK	true	27204	12853
21454	HTTP Request	200	OK	true	27204	12844
21873	HTTP Request	200	OK	true	27204	12847
22766	HTTP Request	200	OK	true	27204	12865
23554	HTTP Request	200	OK	true	27204	12862

Continúa en la próxima página

Cuadro A.2 – continua de la página anterior


elapsed	label	responseCode	responseMessage	success	bytes	sentBytes
23611	HTTP Request	200	OK	true	27204	12850
23978	HTTP Request	200	OK	true	27204	12859
24331	HTTP Request	200	OK	true	27204	12856
24351	HTTP Request	200	OK	true	27204	12850
25043	HTTP Request	200	OK	true	27204	12847
25676	HTTP Request	200	OK	true	27204	12853
26054	HTTP Request	200	OK	true	27204	12847
26081	HTTP Request	200	OK	true	27204	12862
26333	HTTP Request	200	OK	true	27204	12874
26653	HTTP Request	200	OK	true	27204	12862
26910	HTTP Request	200	OK	true	27204	12874
27239	HTTP Request	200	OK	true	27204	12874
27691	HTTP Request	200	OK	true	27204	12850
28446	HTTP Request	200	OK	true	27204	12856
28877	HTTP Request	200	OK	true	27204	12850
29819	HTTP Request	200	OK	true	27204	12862
29932	HTTP Request	200	OK	true	27204	12862
29969	HTTP Request	200	OK	true	27204	12856
30611	HTTP Request	200	OK	true	27204	12856
31473	HTTP Request	200	OK	true	27204	12853
31997	HTTP Request	200	OK	true	27204	12862
32414	HTTP Request	200	OK	true	27204	12871
33068	HTTP Request	200	OK	true	27204	12868
33294	HTTP Request	200	OK	true	27204	12844
33344	HTTP Request	200	OK	true	27204	12874
33980	HTTP Request	200	OK	true	27204	12862
34671	HTTP Request	200	OK	true	27204	12868
35320	HTTP Request	200	OK	true	27204	12874
35394	HTTP Request	200	OK	true	27204	12844
35470	HTTP Request	200	OK	true	27204	12850
36357	HTTP Request	200	OK	true	27204	12871
36422	HTTP Request	200	OK	true	27204	12844
36920	HTTP Request	200	OK	true	27204	12874
36958	HTTP Request	200	OK	true	27204	12844
36950	HTTP Request	200	OK	true	27204	12844
36941	HTTP Request	200	OK	true	27204	12859
36931	HTTP Request	200	OK	true	27204	12853
36921	HTTP Request	200	OK	true	27204	12871
36907	HTTP Request	200	OK	true	27204	12847
36884	HTTP Request	200	OK	true	27204	12856
36885	HTTP Request	200	OK	true	27204	12874
36907	HTTP Request	200	OK	true	27204	12850
36878	HTTP Request	200	OK	true	27204	12874
36867	HTTP Request	200	OK	true	27204	12868
36851	HTTP Request	200	OK	true	27204	12853
36862	HTTP Request	200	OK	true	27204	12844
36815	HTTP Request	200	OK	true	27204	12850
36821	HTTP Request	200	OK	true	27204	12865

Continúa en la próxima página

**Cuadro A.2 – continua de la página anterior**

<b>elapsed</b>	<b>label</b>	<b>responseCode</b>	<b>responseMessage</b>	<b>success</b>	<b>bytes</b>	<b>sentBytes</b>
36836	HTTP Request	200	OK	true	27204	12847
36806	HTTP Request	200	OK	true	27204	12865
36838	HTTP Request	200	OK	true	27204	12865
36795	HTTP Request	200	OK	true	27204	12847
36790	HTTP Request	200	OK	true	27204	12871

Este documento esta firmado por



<b>Firmante</b>	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
<b>Fecha/Hora</b>	Sun Jun 30 18:14:56 CEST 2024
<b>Emisor del Certificado</b>	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
<b>Numero de Serie</b>	561
<b>Metodo</b>	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)