



MÁSTER EN DATA SCIENCE AND BUSINESS INTELLIGENCE
PRESENCIAL

**NOTICIAS CONCISAS
RESÚMENES CON I.A**

TFM elaborado por: Agustina Torres Moray
Tutor/a de TFM: Juan Manuel Moreno Lampero

- Madrid a 23 de Octubre de 2024 -

*A mi familia,
que lejos o cerca,
me apoya e incentiva siempre
a crecer,
ser mejor persona
y perseguir mis sueños.*

RESUMEN:	5
ABSTRACT:	5
INTRODUCCION Y ANTECEDENTES:	6
OBJETIVOS DEL PROYECTO:	6
MATERIAL Y METODOS:	7
Recursos físicos	7
Entorno de desarrollo	7
Librerías y Herramientas	7
Metodologías de trabajo:	7
Otras herramientas	7
EJECUCION DEL PROYECTO /IMPLEMENTACION TECNICA.....	8
Análisis y definición de fuentes de noticias.....	8
Fuente 1- videos de YouTube:	9
Fuente 2 – ficheros PDF:	10
Fuente 3 – Noticieros en línea:	12
Extracción y procesamiento del texto fuente.....	15
Fuente 1- videos de YouTube:	15
Fuente 2- archivos PDF:	15
Fuente 3 – noticias en línea:	15
Estudio, análisis y pruebas de modelos de resumen.	15
Limitaciones técnicas.....	17
Dificultades encontradas en los modelos	17
Selección de modelo/s de resumen	18
Modelos implementados	19
Pruebas y ajuste de parámetros	19
Compilado, limpieza y estandarización del código.	20
Estructura del Código	20
Organización del flujo de trabajo.....	21
Diseño y desarrollo de la interfaz web.....	24
Selección de la herramienta.....	25
Implementación de la interfaz	25
Despliegue y pruebas	26
Guía del usuario	26
Pruebas de testeo y deploy de plataforma.	29
Pruebas de testeo.....	29
Despliegue de la plataforma	30

ANALISIS DE RESULTADOS: 31

Resultados por tipo de origen..... 31

 Resúmenes resultantes de Videos de YouTube 31

 Resúmenes resultantes de ficheros PDFs 32

 Resúmenes resultantes de noticias en línea (Web scraping) 34

 Resultados comunes por tipo de modelo 36

CONCLUSIONES:..... 38

REFERENCIAS BIBLIOGRAFICAS: 39

RESUMEN:

En la presente memoria se detalla el desarrollo de un proyecto que implementa diversas herramientas aprendidas durante el ciclo lectivo, así como otras adquiridas durante la ejecución de este trabajo final, todas asociadas al campo de Data Science e IA.

En concreto, este TFM desarrolla un modelo automático de resumen de noticias que ofrece al usuario final rapidez y facilidad para "ponerse al día" con las novedades deseadas.

El alcance del proyecto abarca desde la planificación y definición de objetivos hasta el desarrollo y publicación de una aplicación web que permita al usuario seleccionar entre tres fuentes diferentes de noticias para obtener un resumen rápido y conciso.

Si bien se alcanzaron los objetivos definidos durante la etapa inicial del proyecto, este último presenta un gran potencial de crecimiento, ya que, incorporando recursos informáticos de mayor memoria y capacidad de cómputo, se podrían indudablemente lograr resúmenes mas precisos.

Palabras claves: ***resumen, noticias, modelos, IA, aplicación, web-scraping, NLP.***

ABSTRACT:

This report outlines the development of a project that implements various tools learned during the academic year, as well as others acquired during the execution of this final project, all related to the fields of Data Science and AI.

Specifically, this Master's Thesis (TFM) develops an automatic news summarization model that provides the end user with speed and ease in "catching up" on desired updates.

The scope of the project covers everything from planning and defining objectives to developing and publishing a web application that allows the user to select from three different news sources to obtain a quick and concise summary.

Although the goals set during the initial phase of the project were achieved, the project has great potential for growth. By incorporating computing resources with greater memory and processing power, more accurate summaries could undoubtedly be achieved.

Keywords: ***summarization, news, models, AI, application, web scraping, NLP.***

INTRODUCCION Y ANTECEDENTES:

Si bien en la actualidad existen numerosas herramientas de NLP e inteligencia artificial que nos ofrecen una infinidad de servicios asociados al lenguaje, dentro de los cuales podemos incluir el resumen de texto, este proyecto no busca “competir” con dichos desarrollos, sino implementar de manera personalizada y concisa algunos de estos en un flujo lineal de trabajo, reflejando los conocimientos adquiridos.

Así mismo, se debe reconocer que la IA esta en pleno auge, debiéndose entonces llamar a este proyecto como “preliminar”, ya que, posiblemente, se desarrollen a corto plazo nuevas tecnologías que reemplacen a las implementadas en el presente Trabajo Final de Máster.

En esta memoria, se documentan los pasos seguidos para implementar el trabajo proyectado, los modelos de IA empleados en este último y los desafíos enfrentados durante el desarrollo, incluyendo pruebas y ajustes realizados. Asimismo, se analizan los resultados obtenidos y se exploran posibles mejoras para futuras investigaciones.

OBJETIVOS DEL PROYECTO:

El objetivo global, o general, del proyecto fue desarrollar una herramienta que permitiese generar resúmenes automáticos de noticias a partir de diversas fuentes de información. Para cumplir con este objetivo general, se definieron, complementariamente, los siguientes objetivos específicos:

- Desarrollar e implementar diferentes sistemas de procesamiento y obtención de texto según las tres posibles fuentes de noticias definidas: videos de YouTube, ficheros PDF y noticieros en línea.
- Incorporar modelos de inteligencia artificial preentrenados para realizar el procesamiento de los textos generados y obtener resúmenes de manera eficiente y precisa.
- Proporcionar una interfaz de usuario interactiva y fácil de usar.

MATERIAL Y METODOS:

Durante el desarrollo del proyecto, se utilizaron diversas herramientas y tecnologías que dieron soporte y fueron complemento del desarrollo propio, necesarias para alcanzar los objetivos planteados.

Recursos físicos

- Ordenador (8GB RAM)

Entorno de desarrollo

El lenguaje de programación elegido para este proyecto fue Python, debido a la gran cantidad de bibliotecas especializadas en procesamiento de lenguaje natural. El entorno de desarrollo integrado (IDE) fue Visual Studio Code (local), en combinación con entornos virtuales para la gestión de dependencias.

Librerías y Herramientas

- PyMuPDF: para la extracción de texto de documentos PDF.
- BeautifulSoup: para el web-scraping de noticias en línea.
- API de YouTube: para la transcripción de vídeos de la misma plataforma.
- LLMs preentrenados de Hugging Face: para la generación de resúmenes automáticos.
- Streamlit: para el desarrollo de la interfaz de usuarios, permitiendo a estos últimos la interacción con la plataforma de manera fácil e intuitiva.

Metodologías de trabajo:

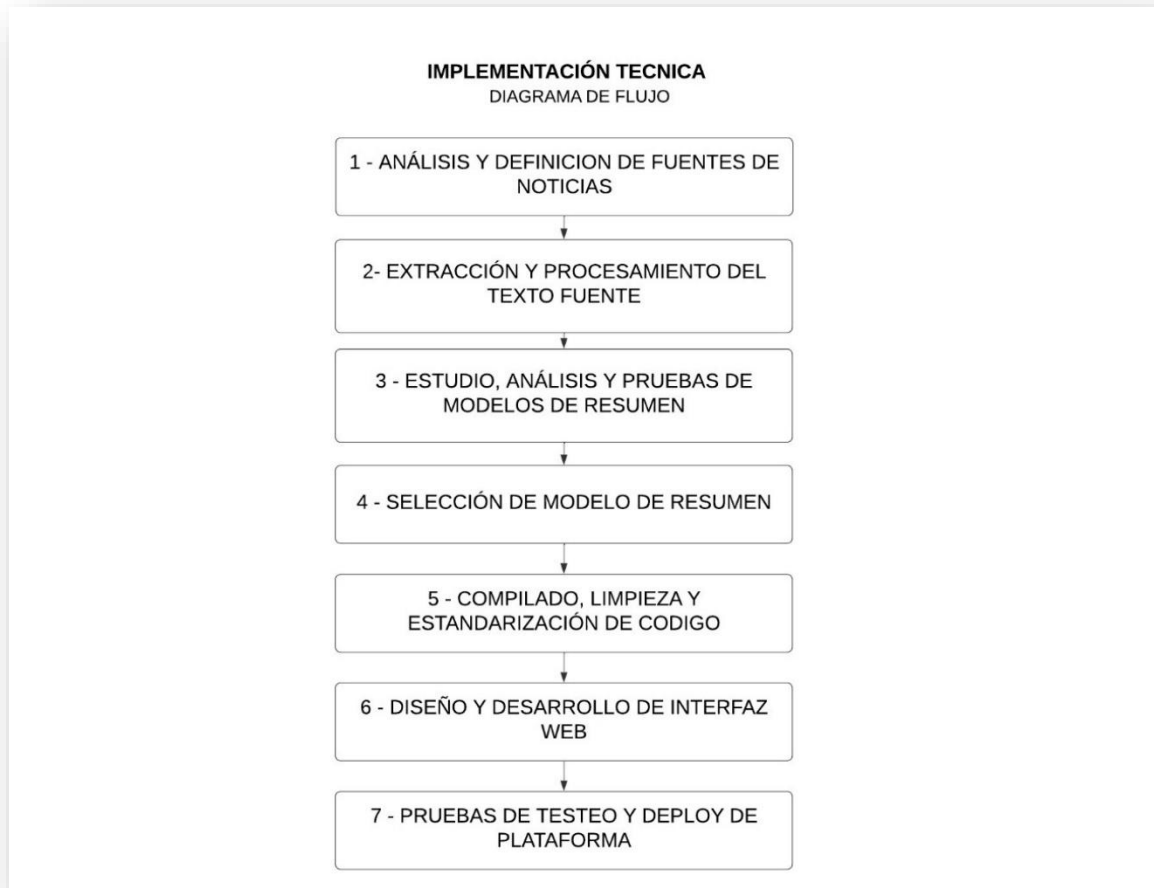
- Planificación y reuniones periódicas con el tutor designado.
- Seguimiento y cumplimiento de objetivos establecidos durante el plan.
- Trabajo individual, pero con resguardo en la nube y versionado en GitHub.

Otras herramientas

- Páginas webs oficiales de librerías instaladas.
- Videos y tutoriales en línea.
- Libros asociados a la IA.

EJECUCION DEL PROYECTO /IMPLEMENTACION TECNICA

El siguiente diagrama (Imagen_1), presenta las etapas comprendidas en la implementación técnica, detalladas a continuación de este.



Imagen_1 - Etapas implementación técnica (fuente propia)

DESARROLLO DE ETAPAS

Análisis y definición de fuentes de noticias.

Siendo el objetivo principal del presente Trabajo Final de Máster implementar un modelo de inteligencia artificial que permita obtener resúmenes de noticias, la primera pregunta que nos realizamos fue:

¿Qué noticias se podrían resumir?

Como todo proyecto, se definió un alcance inicial, respondiendo la anterior pregunta. Así es como, finalmente, se limitaron y definieron tres posibles orígenes de texto, buscando que estos requiriesen un tratamiento de datos diferentes y le proporcionasen al usuario un abanico, limitado, de posibilidades de resumen.

Dicho esto, a continuación, se detallan los tres canales u orígenes de extracción de texto (noticias) que se contemplaron en el desarrollo, junto a algunas ventajas (beneficios) y/o dificultades enfrentadas al procesarlas.

Fuente 1- videos de YouTube:

En este primer caso, el objetivo fue lograr obtener el texto a resumir de un video de noticia alojado en YouTube. Es decir, que al identificar la url del video deseado, fuésemos capaces de obtener la transcripción en texto de dicho video.

Beneficios:

- ✓ Librerías y APIs especializadas: la API de YouTube permite una fácil integración con herramientas que extraen y manipulan tanto el video como los subtítulos, facilitando el acceso a la transcripción sin necesidad de procesar manualmente los videos.
- ✓ Bajos requerimientos computacionales: no se necesita procesar el video completo ni realizar análisis de video en tiempo real. El acceso a las transcripciones (subtítulos) está optimizado y disponible directamente a través de las APIs, lo que disminuye el costo computacional.
- ✓ Facilidad de implementación: el uso de librerías como youtube-transcript-api y pytube simplifica la extracción de transcripciones desde videos, incluso sin necesidad de autenticar la cuenta del usuario.
- ✓ Multilingüismo: al ser YouTube una plataforma global, la API permite obtener transcripciones en múltiples idiomas o incluso traducir el contenido de manera automática utilizando otras librerías como googletrans para facilitar la comprensión y análisis de videos en idiomas extranjeros.
- ✓ Disponibilidad de videos públicos: al ser una plataforma de fácil acceso, YouTube cuenta con una gran cantidad de contenido público que puede ser utilizado sin mayores restricciones legales, lo que proporciona una fuente ilimitada de información para ser resumida.
- ✓ Versatilidad del formato: los videos contienen tanto audio como visuales, lo que puede ser útil para aplicaciones más avanzadas (ej. análisis de tono, emociones, etc.), aunque esto no se haya implementado en este proyecto.

Dificultades:

- Transcripciones incompletas o sin signos de puntuación: las transcripciones generadas automáticamente suelen carecer de signos de puntuación, lo que dificulta la segmentación del texto en oraciones claras, afectando la calidad del resumen final.
- Ruido en las transcripciones: las transcripciones automáticas tienden a incluir ruidos de fondo, como aplausos, música, interrupciones o comentarios irrelevantes, que afectan la calidad del texto resultante.
- Calidad variable de las transcripciones automáticas: la precisión de las transcripciones automáticas varía considerablemente según la calidad del audio, el acento del hablante, el ruido ambiental y la claridad del discurso.
- Limitaciones en videos sin subtítulos: no todos los videos en YouTube cuentan con transcripciones automáticas o subtítulos generados por el usuario, lo que limita el acceso a ciertos contenidos.
- Tiempos de procesamiento: aunque no es una dificultad técnica severa, en algunos casos, los videos muy largos pueden tardar más tiempo en procesarse, lo que afecta la experiencia del usuario.
- Limitaciones en videos privados o restringidos por región: algunos videos de YouTube no están disponibles para todos los usuarios debido a restricciones geográficas o de privacidad.

Fuente 2 – ficheros PDF:

La segunda alternativa consistió en la extracción de texto a partir de ficheros en formato PDF, con el fin de obtener el contenido completo del documento y posteriormente generar un resumen automático.

Los archivos PDF son un formato comúnmente utilizado para la distribución de artículos académicos, informes y documentos, por lo que esta opción ofrecería una forma de trabajar con grandes cantidades de información escrita de forma eficiente.

Beneficios:

- ✓ Múltiples herramientas para extracción de texto: existen varias librerías en Python, como PyMuPDF y PDFplumber, que permiten extraer texto de manera rápida y precisa. Estas herramientas son ampliamente soportadas, fáciles de integrar y manejan bien la mayoría de los PDFs, lo que facilita el acceso al contenido del documento.
- ✓ Facilidad de implementación: la lectura y extracción de texto desde PDF es un proceso directo con las herramientas adecuadas. El código necesario para lograrlo es relativamente sencillo y las librerías existentes están bien documentadas, lo que facilita su uso incluso en proyectos de gran escala.
- ✓ Bajos requerimientos computacionales: el procesamiento de PDFs generalmente requiere poca potencia de cómputo, ya que la mayoría de los archivos son textos planos o imágenes de texto.
- ✓ Compatibilidad con la mayoría de los documentos: los PDFs son uno de los formatos más comunes para compartir documentos, por lo que este sistema puede aplicarse a una amplia gama de archivos, desde artículos académicos hasta documentos corporativos.
- ✓ Escalabilidad: el proceso de extracción de texto de PDFs puede aplicarse de manera escalable, lo que lo convierte en una solución adaptable para grandes volúmenes de documentos.
- ✓ Portabilidad: los archivos PDF son multiplataforma y se mantienen consistentes independientemente del software utilizado para visualizarlos, lo que asegura que los documentos procesados mantengan su formato original.

Dificultades:

- Necesidad de procesamiento adicional: los textos extraídos de PDFs suelen requerir una limpieza significativa debido a la presencia de elementos no deseados como encabezados, pies de página, números de página, marcas de agua, tablas e imágenes, que pueden interrumpir la coherencia del texto.

- Complejidad en la disposición del texto: la extracción de texto puede volverse complicada dependiendo del formato del PDF, especialmente si contiene múltiples columnas, gráficos, o disposición de texto irregular.
- Archivos escaneados o de baja calidad: en documentos que han sido escaneados o generados a partir de imágenes, la extracción de texto requiere el uso de herramientas más avanzadas, lo que puede aumentar los tiempos de procesamiento y generar errores de reconocimiento.
- Pérdida de formato: a diferencia de otros formatos como HTML, el PDF no está diseñado para ser procesado como texto plano. Por ello, la estructura original del documento (como las negritas, cursivas, tamaño de fuente, etc.) se pierde en el proceso de extracción, lo que puede afectar la legibilidad del texto extraído.
- Incompatibilidad con ciertos PDFs: algunos PDFs pueden estar protegidos con contraseñas o tener restricciones que impiden la extracción de contenido.
- Errores en la interpretación de caracteres: dependiendo de la codificación interna del PDF, es posible que se produzcan errores en la interpretación de ciertos caracteres, especialmente en archivos con caracteres especiales, símbolos matemáticos o textos en idiomas con alfabetos no latinos.
- Posible aumento de los tiempos de procesamiento con documentos muy largos: aunque el procesamiento de PDFs tiene bajos requerimientos computacionales, la extracción de texto en documentos de gran extensión puede resultar en tiempos de procesamiento más prolongados, especialmente si hay gráficos o formatos complejos involucrados.

Fuente 3 – Noticieros en línea:

La tercera y última opción, considerada la de mayor complejidad en términos de desarrollo, permitiría obtener titulares y noticias actualizadas a partir de la URL de un periódico digital utilizando la popular metodología conocida como “web scraping”.

Al ingresar la URL de un determinado sitio web de noticias, se extraería el contenido de interés (barras de navegación, secciones, titulares y cuerpos de las noticias) para su posterior procesamiento.

Dado que cada página web tiene características y estructuras HTML diferentes, fue necesario desarrollar una metodología de extracción personalizada para cada sitio. Esto

implicó ajustar el scrapping según el código HTML y CSS de cada sitio, y hacer frente a desafíos relacionados con la seguridad, accesibilidad y la posible presencia de barreras de suscripción.

Para garantizar consistencia en los resultados y limitar la complejidad del proyecto, se seleccionaron tres periódicos populares que cumplieran con los requisitos de acceso y formato:

- El País (español): <https://elpais.com/>
- Diario.es (español): <https://www.eldiario.es/>
- NBC News (inglés): <https://www.nbcnews.com/>

Beneficios:

- ✓ Acceso a información actualizada: una de las grandes ventajas del web scraping es la capacidad de extraer noticias en tiempo real. Esto garantiza que la información procesada y resumida esté siempre al día, proporcionando resúmenes de noticias actuales al momento de la consulta.
- ✓ Variabilidad de librerías para web scraping: herramientas como BeautifulSoup, Selenium y Scrapy proporcionan múltiples opciones para implementar scraping en diferentes niveles de complejidad. Mientras BeautifulSoup permite un enfoque sencillo y eficiente para extraer contenido HTML estático, Selenium se emplea para interactuar con sitios web dinámicos o protegidos por JavaScript.
- ✓ Rápido acceso a grandes volúmenes de datos: con pocas líneas de código, es posible acceder a múltiples noticias y extraer grandes cantidades de información en cuestión de segundos. Esto hace que el scraping sea una solución altamente eficiente para proyectos donde se requiere trabajar con grandes volúmenes de datos en tiempo real.
- ✓ Personalización en la extracción de datos: al trabajar directamente con el código HTML y CSS de las páginas web, se puede personalizar la extracción de manera precisa, eligiendo qué partes del sitio se desean extraer (titulares, resúmenes, cuerpo de noticias, etc.).
- ✓ Automatización y repetibilidad: una vez configurado el scraping para un sitio web en particular, el proceso puede ser automatizado para extraer contenido de manera regular sin intervención manual. Esto asegura un flujo continuo de

información actualizada, lo que es ideal para aplicaciones que requieren datos frescos constantemente.

- ✓ Aplicación multilingüe: los periódicos en línea están disponibles en muchos idiomas, lo que permite que el sistema trabaje con contenido de diversas regiones del mundo.

Dificultades:

- Código poco estandarizable: cada sitio web tiene su propia estructura HTML y CSS, lo que significa que el código de scraping debe adaptarse individualmente a cada sitio. Esto complica la estandarización del proceso, ya que los selectores de datos (por ejemplo, etiquetas y clases CSS) pueden variar enormemente entre periódicos. Además, si el sitio web cambia su estructura, el código de scraping puede romperse y requiere actualizaciones constantes.
- Ruidos en el texto extraído: durante la extracción de contenido, el scraping también puede capturar elementos no deseados, como anuncios publicitarios, banners de suscripción, pop-ups, o avisos promocionales, lo que introduce "ruido" en los datos y afecta la calidad del texto a procesar.
- Barreras de acceso: algunos sitios de noticias están protegidos por muros de pago, requieren suscripciones para acceder al contenido completo, o implementan medidas de seguridad como CAPTCHAs para evitar el scraping automatizado. Esto puede limitar la cantidad de información que se puede obtener de ciertas fuentes.
- Cambios inesperados en la estructura web: las páginas web pueden actualizarse sin previo aviso, cambiando su estructura HTML y CSS, lo que rompe el código de scraping. Esto requiere un monitoreo constante de las fuentes para asegurarse de que el código de extracción siga funcionando, lo que añade una capa de mantenimiento continuo al proyecto.
- Limitaciones legales y de términos de servicio: algunas páginas web tienen términos de servicio que prohíben el scraping o limitan el uso de sus datos, lo que puede llevar a conflictos legales si no se respetan dichas políticas. Es

necesario asegurarse de que el scraping sea ético y cumpla con los términos y condiciones de cada fuente.

- Dependencia de la velocidad de respuesta del servidor: el scraping depende de la velocidad y la disponibilidad del servidor del sitio web objetivo. Si el servidor está lento o el sitio web experimenta caídas, el proceso de extracción puede fallar o demorar más de lo esperado, lo que afecta el rendimiento del sistema.
- Limitaciones en el contenido extraído: algunos sitios web sólo permiten acceder a una cantidad limitada de contenido antes de requerir una suscripción o presentar barreras adicionales, lo que puede restringir la cantidad de noticias disponibles para su resumen.

Extracción y procesamiento del texto fuente.

Una vez definidas las fuentes de datos principales (el "raw text" de las noticias) en el apartado anterior, se presentan en esta sección las diferentes alternativas de desarrollo contempladas para la obtención del texto en cada caso.

Fuente 1- videos de YouTube:

- Librería implementada: "youtube_transcript_api", para la extracción de las transcripciones automáticas generadas por YouTube.

Fuente 2- archivos PDF:

- Herramientas de testeo: se testearon diversas alternativas: PyMuPDF, PDFPlumber, y Langchain.
- Librería implementada: "PyMuPDF" para la extracción de texto, debido a su eficiencia en la manipulación de archivos PDF y la calidad de los resultados en la mayoría de los casos.

Fuente 3 – noticias en línea:

- Herramientas de testeo: se probaron dos herramientas para la extracción de contenido web, BeautifulSoup y Selenium.
- Librerías implementadas: BeautifulSoup junto con HTML5 para el scraping y procesamiento del código fuente.

Estudio, análisis y pruebas de modelos de resumen.

Este paso representó uno de los componentes críticos en el desarrollo del proyecto, no solo por constituir el núcleo del TFM, sino también por el desafío que implicó la selección e implementación de un modelo adecuado que cumpliera con los objetivos propuestos,

manteniendo al mismo tiempo la viabilidad en términos de los recursos informáticos disponibles.

En una primera etapa, se exploraron diversos modelos de Transformers ya existentes, diseñados específicamente para tareas de procesamiento de lenguaje natural (NLP), muchos de los cuales estaban enfocados en la generación de resúmenes de texto. A nivel teórico, estos modelos estarían optimizados para realizar resúmenes eficientes, aunque los resultados preliminares obtenidos fueron solo parcialmente satisfactorios.

Si bien algunos de estos modelos proporcionaron resúmenes aceptables en ciertos contextos, no lograron alcanzar el nivel de precisión y coherencia deseado. Esto se evidenció al comparar sus resultados con los modelos de inteligencia artificial generativa más avanzados disponibles actualmente, que son capaces de responder a una variedad de preguntas y generar resúmenes de alta calidad para una amplia gama de textos.

Entre los modelos de Transformers probados en esta primera etapa se encuentran:

- ***facebook/mbart-large-50-many-to-many-mmt*** : modelo multilingüe basado en la arquitectura BART (Bidirectional and Auto-Regressive Transformers) y diseñado para la traducción automática entre 50 idiomas. Su versión "many-to-many" permite la traducción directa entre múltiples pares de idiomas sin necesidad de intermediarios en inglés.
- ***facebook/bart-large-cnn***: modelo basado en la arquitectura Transformer, diseñado para la generación y corrección de texto. La versión "large-cnn" fue entrenada específicamente en datos de noticias de CNN/DailyMail, y es ideal para generar resúmenes de noticias.
- ***google-t5/t5-base***: T5 (Text-to-Text Transfer Transformer) es un modelo de Google que convierte cualquier tarea de procesamiento de lenguaje en un problema de traducción de texto a texto. La versión "base" es una variante más pequeña y ligera que la versión "large", con menos parámetros.
- ***google-t5/t5-large***: al igual que t5-base, T5-large es un modelo text-to-text, pero con un mayor número de parámetros, lo que le permite obtener resultados más precisos en tareas complejas de procesamiento de lenguaje natural, a costa de mayores requerimientos computacionales.

- **google/pegasus-large:** modelo especializado en generación de resúmenes de texto. Fue diseñado por Google y entrenado principalmente en tareas de resumen.

Limitaciones técnicas

Uno de los principales desafíos encontrados durante esta fase fue la limitación computacional, que condicionó la selección de los modelos.

El entorno de trabajo consistía en un ordenador con 8 GB de RAM y sin GPU dedicada, lo que impuso restricciones considerables en cuanto a la capacidad de procesamiento y el manejo de modelos de gran tamaño. Esta limitación fue determinante para elegir modelos más pequeños tanto en esta primera fase como en la fase subsiguiente del desarrollo.

Dificultades encontradas en los modelos

Adicionalmente, se presentaron una serie de dificultades asociadas a los modelos probados, lo que complicó la obtención de mejores resultados. Algunas de estas dificultades incluyeron:

- **Limitación en el número de tokens de entrada:** muchos de los modelos probados tienen un límite en la cantidad de tokens que pueden procesar de una sola vez, lo que restringe la longitud de los textos a resumir. Este factor afectó directamente la calidad de los resúmenes generados para textos extensos.
- Si bien se realizaron pruebas subdividiendo el texto a resumir, proveniente de cualquiera de las fuentes de datos definidas, en “chunks”, para luego pasar dichos “chunks” por el modelo Transformers (limitando de esta manera la cantidad de tokens de entrada), los resultados no fueron validados internamente.
- Modelos disponibles únicamente en inglés: algunos de los modelos evaluados solo estaban entrenados en inglés, lo que requería el uso de pipelines adicionales para traducir los textos de entrada o salida, aumentando la complejidad del proceso y, en algunos casos, afectando la calidad del resumen final.
- **Modelos multilingües con mejor rendimiento en inglés:** a pesar de que algunos modelos soportaban múltiples idiomas, su rendimiento en otros idiomas, como el español, era notablemente inferior en comparación con sus resultados

en inglés. Esto impactó negativamente en la precisión y coherencia de los resúmenes en español.

- **Modelos "pequeños" con parámetros limitados:** dado que algunos de los modelos seleccionados tenían un tamaño reducido debido a las limitaciones computacionales, estaban entrenados con menos parámetros y tipos de texto. Esta limitación afectó la capacidad del modelo para generar resúmenes más ricos y detallados, especialmente en dominios menos representados en los datos de entrenamiento.

Selección de modelo/s de resumen

Finalmente, en la última etapa del proyecto, se consideró la alternativa de implementar un modelo LLM (Large Language Model) de IA generativa, adaptando y limitando este último a la tarea objetivo de resumir noticias.

A diferencia del uso general de estos modelos de inteligencia artificial, donde el usuario puede interactuar libremente con el modelo a través de un "prompt", en este caso, se abstraigo dicho prompt de la interacción visual usuario-modelo, definiéndolo a través de código como una constante fija destinada exclusivamente al resumen de los textos proporcionados como entrada. Esta restricción se implementó con el objetivo de simplificar el uso del modelo y optimizar su rendimiento, considerando las limitaciones computacionales.

Durante esta fase, se probaron varios modelos de LLMs, entre ellos *mistralai/Mistral-7B-v0.1*, pero en muchos casos no se obtuvieron resultados visibles debido a la gran demanda de recursos computacionales de los modelos más avanzados. Cabe destacar que la mayoría de estos modelos requieren una GPU dedicada y al menos 16 GB de memoria RAM para funcionar correctamente, recursos que no estaban disponibles en el entorno de desarrollo, que contaba con 8 GB de RAM y sin acceso a una GPU.

A pesar de estas limitaciones, se optó finalmente por testear dos modelos de IA generativa más ligeros (más pequeños), que, si bien no alcanzarían el rendimiento de los modelos más avanzados, podrían ofrecer resultados aceptables en la tarea de resumen de texto.

Cabe aclarar que, aunque estos modelos no son comparables con los grandes modelos LLM idealmente implementados en aplicaciones de alto rendimiento, se comprobó que podían generar resúmenes con mejor calidad en comparación con los modelos de Transformers probados en una primera instancia. Estos modelos "pequeños" de IA generativa fueron capaces, en algunos casos, de producir resúmenes de mayor precisión y coherencia a pesar de las limitaciones de recursos disponibles.

Modelos implementados

- **EleutherAI/gpt-neo-125m**: modelo de lenguaje entrenado por EleutherAI que forma parte de la familia GPT-Neo. La versión de 125M parámetros es una de las más ligeras de la serie, lo que la hace adecuada para entornos con limitaciones de hardware. Está diseñado para tareas de generación de texto y fue entrenado en una gran cantidad de datos no supervisados, lo que le permite generar texto de forma coherente y fluida.
- **distilbert/distilgpt2**: versión reducida y optimizada de GPT-2, desarrollada por Hugging Face. Al reducir el tamaño del modelo en aproximadamente un 60%, se mantiene el 95% de su capacidad predictiva, lo que lo convierte en una opción viable para tareas que requieren menor consumo de recursos sin sacrificar demasiado la calidad del resultado. Este modelo es especialmente útil en dispositivos con limitaciones de memoria y procesamiento.

Pruebas y ajuste de parámetros

Durante las pruebas con ambos modelos, se ajustaron diversos parámetros para optimizar el rendimiento y los resultados obtenidos en cada caso. Los ajustes se realizaron iterativamente, modificando las configuraciones más relevantes del modelo hasta encontrar la combinación de valores más adecuada para la tarea de resumen.

A continuación (Imagen_2), se listan los parámetros principales ajustados durante las pruebas junto a una breve descripción de estos y los valores por default seleccionados en el deploy del proyecto.

PARÁMETRO	DESCRIPCIÓN	NOMBRE EN STREAMLIT	VALOR DEFAULT
<i>model_name</i>	Nombre del modelo LLM de inteligencia artificial generativa que se utiliza para el resumen.	Modelo de resumen	EleutherAI/gpt-neo-125M
<i>max_new_tokens</i>	Número máximo de tokens (palabras o fragmentos de texto) que el modelo puede generar en una secuencia.	Num. Max. Tokens	100
<i>temperature</i>	Controla la aleatoriedad en la generación de texto. Valores más bajos hacen el texto más preciso, valores más altos aumentan la creatividad.	Aleatoriedad del texto (creatividad)	0.7
<i>mode</i>	Modo en el que el modelo genera texto: "sampling" implica que el modelo elige palabras de manera aleatoria siguiendo probabilidades (creativo), "beam search" es un modo de generación que equilibra entre la precisión y la creatividad (coherente y estructurado)	Modo de generación	beam search

Imagen_2 - Variables parametrizables (fuente propia)

Estos parámetros permitieron ajustar el equilibrio entre precisión, fluidez y creatividad en los resúmenes generados, maximizando la eficiencia de los modelos en función de las limitaciones del entorno de desarrollo.

A su vez, es importante aclarar que los resultados obtenidos dependieron en gran medida de la calidad del texto obtenido en pasos previos, es decir, de las diferentes fuentes de transcripción definidas en pasos anteriores y sus outputs.

Debido al reconocimiento de la alta variabilidad de resultados posibles a causa de dichos factores (mecanismo de extracción, calidad del texto transcrito, tipo de modelo y combinación de parámetros posibles) se desarrolló una aplicación web flexible, donde el usuario pudiese probar, seleccionando el tipo de modelo generativo deseado y ajustando manualmente ciertos parámetros asociados a este último, diferentes resultados, optando por el que mejor le rindiese (desarrollado mas adelante en el apartado “6 – Diseño y desarrollo de la interfaz web”).

Compilado, limpieza y estandarización del código.

Una vez que se probaron por separado las distintas metodologías de extracción de texto y los modelos de resumen seleccionados, se procedió a compilar, limpiar y estandarizar el código para asegurar un flujo de trabajo eficiente y coherente. El objetivo en esta fase fue consolidar el desarrollo en un código estructurado y modular, permitiendo que todas las etapas del proceso, desde la extracción hasta la generación del resumen, se ejecutaran de forma fluida y sin interrupciones.

Se diseñaron pipelines de trabajo para garantizar un seguimiento lineal entre cada fase, de manera que los outputs generados en cada paso pudieran ser utilizados como inputs en las siguientes etapas. Este enfoque facilitaría la transición entre los distintos métodos de extracción de texto (videos de YouTube, archivos PDF, web scraping) y los modelos de IA generativa seleccionados para la tarea de resumen.

Estructura del Código

El código fue desarrollado en Python y gestionado, como ya se hizo alusión previamente en esta memoria, en el entorno de desarrollo Visual Studio Code.

Para mantener el código organizado y modular, se distribuyó este último en diferentes scripts especializados, cada uno enfocado en una tarea específica del flujo de trabajo. Esto no solo mejoraría la mantenibilidad del proyecto, sino que también permitiría realizar ajustes o mejoras en cada módulo sin afectar el funcionamiento global.

La estructura del código refleja una jerarquía de carpetas y archivos, donde cada archivo cumple una función específica y se invoca de manera secuencial según las necesidades de los pipelines.

A continuación (Imagen_3), se muestra la estructura del proyecto, diferenciando los diferentes archivos (scripts) y sus carpetas contenedoras, así como todo otro material requerido durante el desarrollo.

```
1  Listado de rutas de carpetas para el volumen Sistema SSD
2  El n-mero de serie del volumen es 3C3B-7249
3  C:.\
4  @   .gitignore
5  @   App.py
6  @   estructura_directorios.txt
7  @   functions.py
8  @   pipelines.py
9  @   requirements.txt
10 @   summary_llms.py
11 @   summary_transformers.py
12 @
13 +---assets
14 +---output_data
15 @   scrapping_text.txt
16 @   uploaded_pdf.pdf
17 @   video_text.txt
18 @
19 +---processed_data
20 @   pdf_text.txt
21 @   scrapping_text.txt
22 @   video_text.txt
23 @
24 +---raw_data
25 @   uploaded_pdf.pdf
26 @
27 +---venv
28 @   @   pyvenv.cfg
29 @   @
```

Imagen_3 - Estructura código (fuente propia)

Organización del flujo de trabajo

Scripts principales

- **App.py:** este es el script principal de la aplicación desarrollada en Streamlit, desde donde se ejecuta la interfaz gráfica. Contiene el código que conecta las distintas partes del sistema y se encarga de llamar a los demás scripts en función

de la opción seleccionada por el usuario (procesamiento de video, PDF o scraping de noticias web)(Imagen_4).

```
# Importar librerías
> import streamlit as st...

# Configurar el título principal y la disposición de la página
st.set_page_config(page_title="TRABAJO FINAL DE MASTER - Torres Moray, Agustina", layout="wide")

# ----- DISEÑO DE PAGINAS -----

# Definir estilo de fondo
> page_bg_img = ''...

# Aplicar el estilo de fondo
st.markdown(page_bg_img, unsafe_allow_html=True)

# Mostrar iniciales en la esquina superior derecha . simila logo personalizado
st.markdown('<div class="initials">ATM</div>', unsafe_allow_html=True)

# Definir estilos comunes
> menu_styles = {...

# ----- MENU DESPLEGABLE -----

# Menú vertical desplegable
> with st.sidebar:...

# ----- ESTADO GLOBAL - PARAMETRIZACION MODELOS -----

# Inicializar 'ajustes' en session_state si no existe
> if "ajustes" not in st.session_state:...
```

Imagen_4 – App.py (fuente propia)

- **Functions.py:** contiene todas las funciones auxiliares relacionadas con el preprocesamiento, extracción y limpieza de texto. Incluye tanto funciones comunes a todas las fuentes de noticias como aquellas que son específicas para cada tipo de fuente (por ejemplo, manejo de transcripciones de videos de YouTube o limpieza de texto extraído de PDFs) (Imagen_5).

```
1  # Importar librerías
2  > from youtube_transcript_api import YouTubeTranscriptApi, TranscriptsDisabled, NoTranscriptFound...
14
15  #----- FUNCIONES PDF -----
16
17  > def save_pdf_file(pdf_file, input_path):...
20
21  > def extract_text_from_pdf(pdf_path):...
46
47  > def extract_text_from_pdf_plumber(pdf_path):...
65
66  > def extract_text_from_pdf_with_columns_and_footer_filter(pdf_path):...
93
94  #----- FUNCIONES VIDEO -----
95
96  > def get_video_code(url):...
98
99  > def fetch_transcript(video_code, languages=['es', 'en']):...
112
113  > def transcribe_youtube(transcript):...
120
121  #----- DATA Y FUNCIONES WEB SCRAPING -----
122
123  # Lista de periodicos web junto a sus características individuales requeridas
124  > news_sites = [...
174
175  > def get_user_web_selection(selected_news_site):...
179
180  > def scrape_web_header(url, header_tag, header_class):...
193
194  > def scrape_news(user_section_selection, user_web_selection):...
```

Imagen_5 -Functions.py (fuente propia)

- **Pipelines.py:** se encarga de gestionar los diferentes pipelines de procesamiento, uno para cada tipo de fuente de datos (video de YouTube, PDF o scraping de noticias). Cada pipeline agrupa las funciones necesarias para procesar el texto extraído de cada fuente, asegurando un flujo de trabajo eficiente y secuencial (Imagen_6).

```

1  # Importar librerias
2  > from functions import get_video_code, fetch_transcript, transcribe_youtube, remove_non_alphanumeric, remove_li
6
7  # Carpetas donde se guardan los inputs y outputs.
8  input_folder = "raw_data"
9  transcription_folder = "processed_data"
10 summary_folder = "output_data"
11
12 # Fichero de pdf a adjuntar y textos obtenidos de las transcripciones/resúmenes finales.
13 pdf_file_name = "uploaded_pdf.pdf"
14 pdf_transcription_name = "pdf_text.txt"
15 video_transcription_name = "video_text.txt"
16 scrapping_transcription_name = "scrapping_text.txt"
17
18 # Paths para guardar el pdf y textos transcritos.
19 input_path = os.path.join(input_folder, pdf_file_name)
20 output_path_pdf = os.path.join(transcription_folder, pdf_transcription_name)
21 output_path_video = os.path.join(transcription_folder, video_transcription_name)
22 output_path_scrapping = os.path.join(transcription_folder, scrapping_transcription_name)
23
24 # Paths para guardar los textos resumidos.
25 summary_path_pdf = os.path.join(summary_folder, pdf_file_name)
26 summary_path_video = os.path.join(summary_folder, video_transcription_name)
27 summary_path_scrapping = os.path.join(summary_folder, scrapping_transcription_name)
28
29 > def pdf_pipeline(pdf_file, model_name): ...
63
64 > def video_pipeline(video_url, model_name): ...
96
97 > def scrapping_pipeline(raw_text, model_name, new_site): ...

```

Imagen_6 – Pipelines.py (fuente propia)

- **Summary_llms.py:** almacena las funciones que llaman a los modelos de IA generativa utilizados para generar los resúmenes finales. Estos modelos incluyen aquellos probados y seleccionados para la implementación final del proyecto: GPT-Neo y DistilGPT2 (Imagen_7).

```

# Importar librerias
> from transformers import GPTNeoForCausalLM, GPT2Tokenizer, AutoTokenizer, AutoModelForCausalLM...

> def summarize_with_gptneo(text): ...

> def summarize_with_distilgpt2(text): ...

```

Imagen_7 – Summary_llms.py (fuente propia)

- **Summary_transformers.py**: contiene las funciones relacionadas con los modelos de Transformers probados en las etapas iniciales del proyecto, pero que finalmente no fueron implementados. Sin embargo, se conservan aquí como referencia y para posibles mejoras futuras (Imagen_8).

```
# Importar librerías
import re
from transformers import MBartForConditionalGeneration, MBart50TokenizerFast, pipeline, AutoTokenizer, T5Token
import os
from IPython.display import display, Markdown

> class MultilingualSummarizer: ...

> class T5Summarizer: ...

> def split_text(text, max_length): ...

> def split_text_tokenizer(text, tokenizer, chunk_size): ...

> def summarize_text(text, summarizer, max_length, chunk_size): ...
```

Imagen_8 – Summary_transformers.py (fuente propia)

Carpetas de datos

- **Assets**: almacena archivos estáticos, como imágenes, íconos, y cualquier otro recurso necesario para la interfaz gráfica de la aplicación.
- **Raw_data**: se almacenan los archivos PDF cargados por los usuarios a través de la plataforma web. Estos archivos se guardan aquí antes de ser procesados y convertidos en texto.
- **Processed_data**: guarda los textos extraídos en crudo de cualquier fuente (videos, PDFs y/ o noticias obtenidas por web scrapping) antes de ser procesados por los modelos de IA. Estos archivos representan una etapa intermedia en el pipeline.
- **Output_data**: almacena los resúmenes finales generados por los modelos de IA. Estos resúmenes son el resultado del proceso completo y están listos para ser presentados al usuario.

Cada uno de estos scripts se invoca en el orden correspondiente según el flujo del pipeline, asegurando un procesamiento eficiente y evitando redundancias en el código. El desarrollo modular también facilita la integración de nuevas funcionalidades o la mejora de procesos en etapas futuras.

Diseño y desarrollo de la interfaz web

Una vez finalizado el desarrollo del código y obtenidos resultados preliminares, el siguiente paso fue la creación de una interfaz gráfica que permitiese a los usuarios interactuar fácilmente con el sistema.

El objetivo principal planteado para esta interfaz gráfica fue que los usuarios pudiesen seleccionar la noticia que desearan resumir, especificar la fuente de información y, de manera opcional, ajustar algunos parámetros variables, como ser el modelo LLM de resumen a implementar y algunos de sus parámetros.

Todo esto debía ser realizado de forma rápida y automática, sin necesidad de conocimientos de programación ni interacción con el código subyacente.

Selección de la herramienta

Con los requisitos claros, se evaluaron diferentes librerías y frameworks asociados a Python que permitiesen el desarrollo de la interfaz gráfica. Tras analizar varias opciones, se seleccionó “Streamlit” como el framework más adecuado para el proyecto.

¿Qué es Streamlit?

Streamlit es un framework de código abierto diseñado para crear aplicaciones web interactivas de manera rápida y sencilla utilizando Python. Su enfoque intuitivo permite transformar scripts de Python en aplicaciones de datos totalmente funcionales con solo unas pocas líneas de código. Streamlit destaca por su simplicidad y facilidad para generar interacciones visuales como botones, menús desplegables y gráficos, sin la necesidad de conocimientos avanzados de desarrollo web.

Implementación de la interfaz

Para implementar la interfaz web, se adaptó el código existente y se integraron varias funcionalidades propias de Streamlit.

Entre los componentes clave que se añadieron se incluyen:

- **Elementos de interacción:** desplegables, botones, spinners (indicadores de carga) y mensajes de resultados para guiar al usuario a lo largo del proceso de selección de la noticia y generación del resumen.
- **Estilos personalizados:** aunque Streamlit proporciona un estilo base, se añadieron estilos CSS personalizados para mejorar la apariencia y usabilidad de la aplicación, ajustando colores, tipografía y otros elementos visuales para ofrecer una experiencia más amigable.
- **Lógica de compilación:** se ajustó la lógica interna del código para compilar las diferentes partes del proyecto de manera eficiente, asegurando que cada paso (selección de fuente de noticias, extracción de texto, procesamiento y resumen) funcione de forma secuencial e integrada.

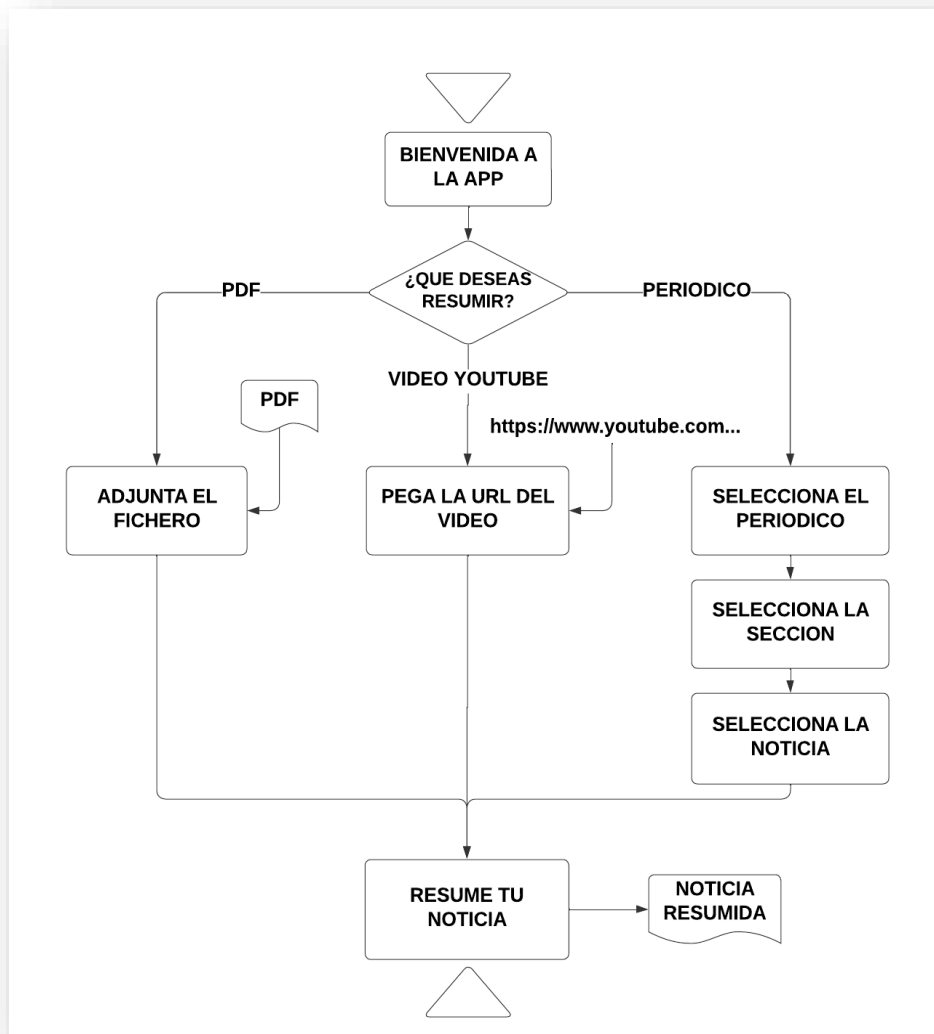
Despliegue y pruebas

Aunque el desarrollo inicial y las pruebas se realizaron en una máquina local, en sincronización con un repositorio remoto en GitHub, una vez validada la funcionalidad de la interfaz, se desplegó la aplicación en “Community Cloud”, propia de Streamlit, para facilitar el acceso del usuario final.

El uso de un host público en la nube garantiza que el acceso a la aplicación sea sencillo y no requiera instalación de software adicional por parte del usuario final, lo que optimiza la experiencia de evaluación.

Guía del usuario

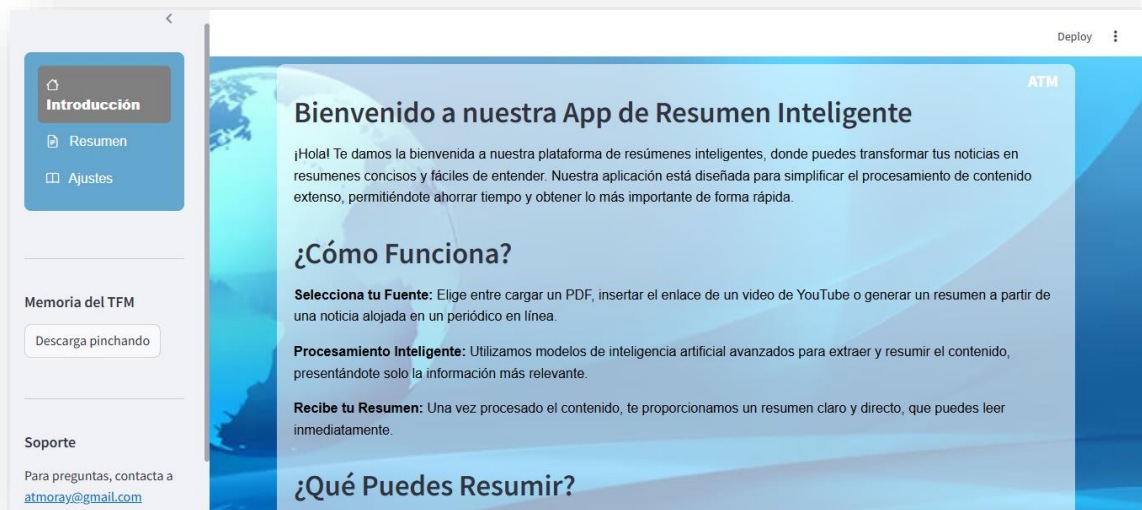
A continuación (Imagen_9), se presenta un diagrama de flujo que resume los pasos que debiera seguir el usuario para utilizar la aplicación e interactuar fácilmente con el sistema obteniendo resúmenes de noticias de forma eficaz.



Imagen_9 - Flujograma interacción Usuario – App (fuente propia)

En el apartado siguiente, se incluyen imágenes (capturas de pantalla de la aplicación web tomadas durante el testeo local), que permiten observar cómo como se ejecutarían dichas etapas al momento de utilizar la herramienta en línea.

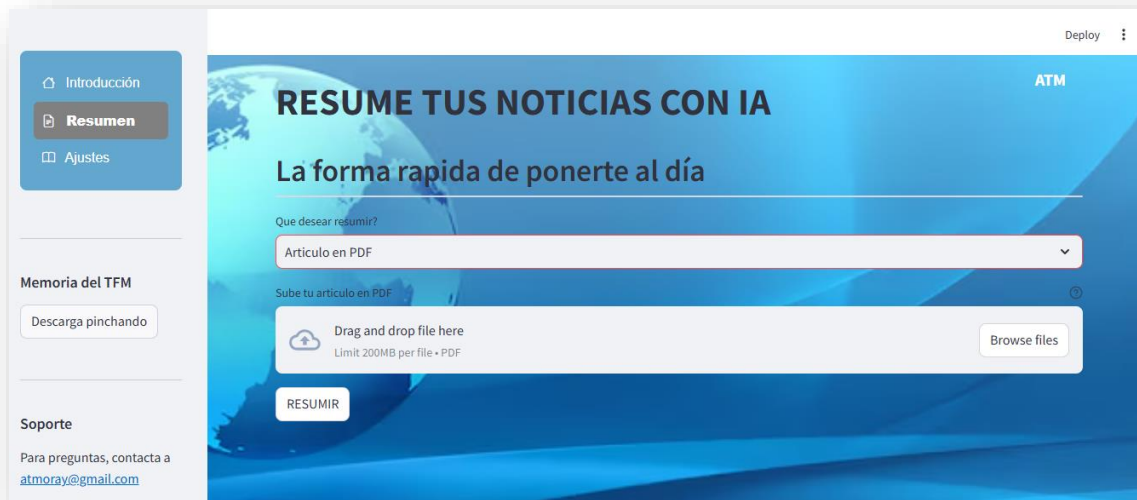
Imágenes de la plataforma web



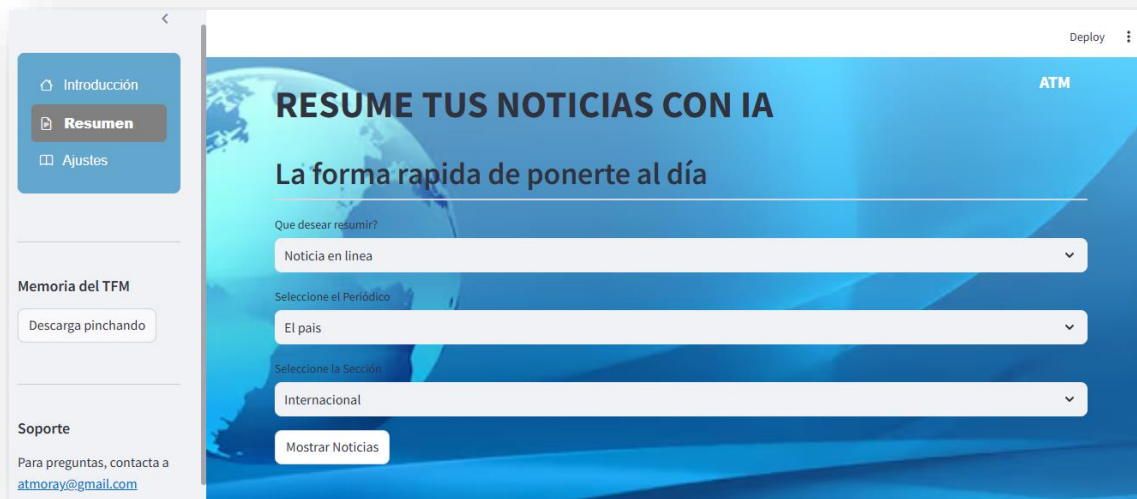
Imagen_10 – Página de Introducción -Streamlit (fuente propia)



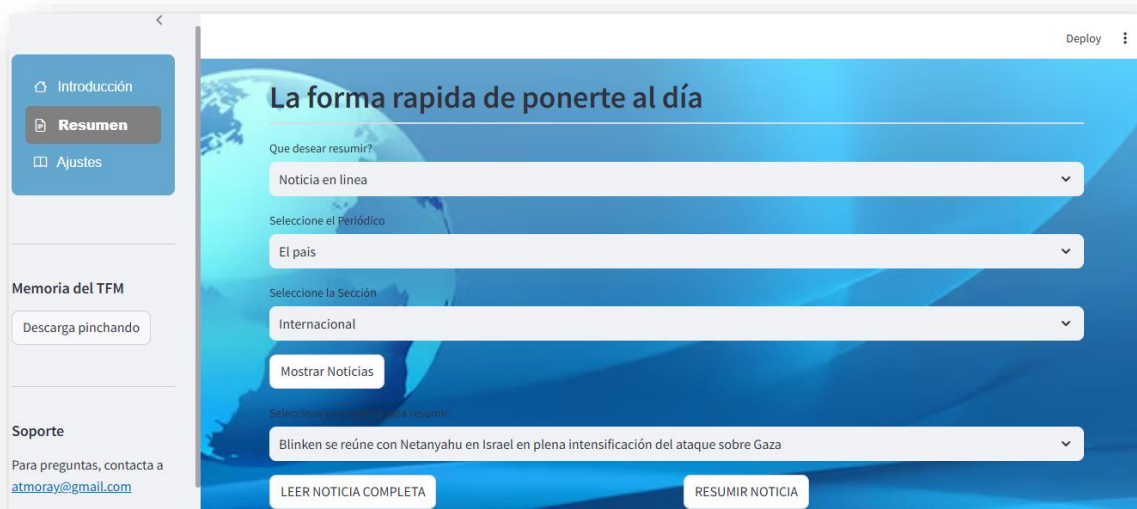
Imagen_11 – Página de resumen – Resumen de videos de YouTube (fuente propia)



Imagen_12 – Página de resumen – Resumen de PDF (fuente propia)



Imagen_13 – Pagina de resumen – Resumen de noticias en línea (fuente propia)



Imagen_14 – Página de resumen – Selección de noticia en línea (fuente propia)



Imagen_15 – Página de ajustes (fuente propia)

Pruebas de testeo y deploy de plataforma.

Con la interfaz web y los modelos de resumen implementados, el siguiente paso crítico fue realizar pruebas exhaustivas de testeo para garantizar el correcto funcionamiento de la plataforma y asegurar que los resultados obtenidos fueran coherentes y precisos. Además, se llevó a cabo el despliegue de la plataforma para permitir su uso por parte de los evaluadores y usuarios finales.

Pruebas de testeo

En esta fase, el objetivo fue validar que todas las funcionalidades de la plataforma estuvieran correctamente integradas y que el flujo de trabajo fuera coherente y sin errores. Para ello, se realizaron las siguientes pruebas:

- **Testeo de extracción de texto:** Se probaron las tres fuentes principales de datos (videos de YouTube, archivos PDF y noticias mediante web scraping) para verificar que el texto fuera extraído correctamente. Esto incluyó la validación de formatos y contenidos para asegurarse de que el sistema pudiera manejar diferentes tipos de entradas de manera eficiente.
- **Testeo de modelos de resumen:** Se probaron los modelos de IA generativa (GPT-Neo y DistilGPT2) con distintos textos para verificar la calidad de los resúmenes generados. Se ajustaron parámetros clave como la longitud del resumen, el número de tokens de entrada y la coherencia en la generación de texto.
- **Pruebas de interacción con la interfaz:** Se realizaron pruebas para garantizar que los elementos de la interfaz (desplegables, botone, sliders, entre otros)

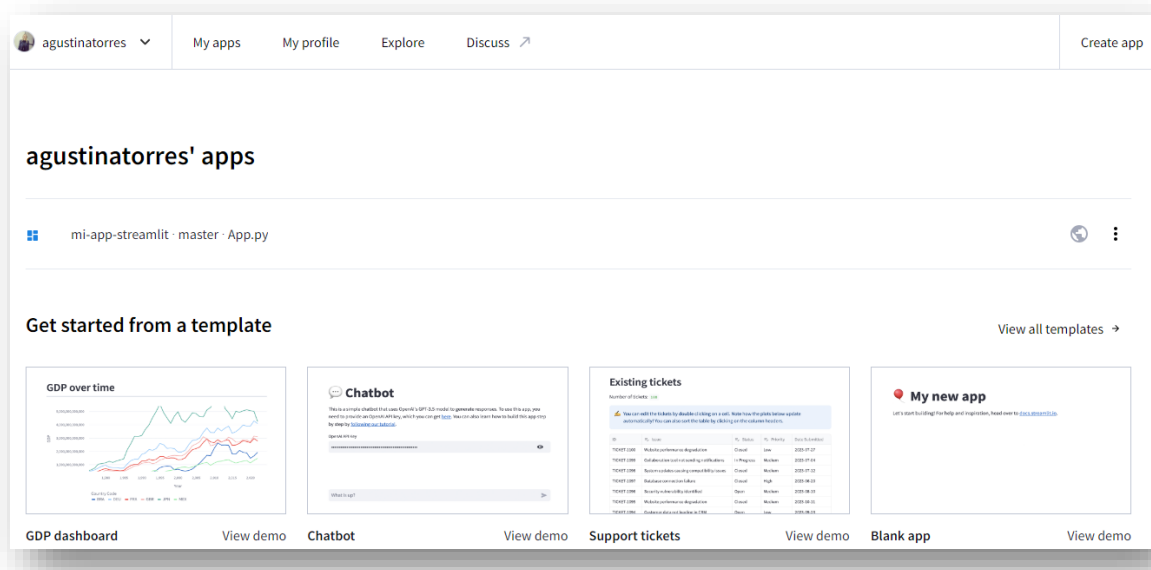
funcionaran correctamente y permitieran al usuario interactuar de manera intuitiva con la plataforma.

Despliegue de la plataforma

Una vez finalizadas las pruebas locales, se procedió al despliegue de la plataforma para que pudiese ser accesible por los evaluadores y usuarios finales de manera remota. Se utilizó un hosting gratuito para el despliegue, lo que permitió realizar una validación adicional del comportamiento de la aplicación en un entorno de producción.

Pasos del despliegue

- **Elección del servicio de hosting:** Se optó por utilizar **Streamlit Cloud**, un servicio de hosting especializado para aplicaciones de Streamlit, que permite desplegar y compartir aplicaciones con facilidad (Imagen_16).
- **Configuración del entorno de producción:** Para garantizar que el sistema funcionara correctamente en el entorno de despliegue, se replicaron las configuraciones y dependencias del entorno local en el servicio de hosting seleccionado. Esto incluyó la instalación de las librerías necesarias y la configuración de los archivos de dependencia (como requirements.txt).
- **Pruebas en entorno de producción:** Después de desplegar la aplicación, se realizaron pruebas adicionales para asegurar que la plataforma respondiera adecuadamente a las solicitudes de los usuarios y que la interacción con el sistema fuera fluida.



Imagen_16 - Cuenta de Streaming Cloud – deploy de APP (fuente propia)

ANÁLISIS DE RESULTADOS:

En este apartado se analizan los resultados obtenidos a partir de las distintas fuentes de datos (videos de YouTube, archivos PDF y noticias web). Se identifican los errores observados tanto en los textos transcritos, como en los resúmenes finales, y se proponen posibles mejoras y futuros desarrollos que podrían optimizar los resultados.

Resultados por tipo de origen

Resúmenes resultantes de Videos de YouTube

En esta sección se evalúa el desempeño de la transcripción y resumen de noticias obtenidas de videos de YouTube (ver ejemplo Imagen_17 e Imagen_18)

Resultados obtenidos:

La transcripción de los videos es exitosa en cuanto a la conversión de audio a texto, especialmente en videos con buena calidad de sonido y en aquellos donde el locutor habla claramente.

- El texto generado por la transcripción contiene el contenido hablado de manera precisa, aunque con ausencia de puntuación y problemas con el reconocimiento de nombres propios o términos técnicos.

Errores detectados:

- **Falta de puntuación:** el texto transcrito carece de signos de puntuación como comas, puntos y mayúsculas, lo que afecta la legibilidad y coherencia del resumen final.
- **Ruido de fondo:** en algunos casos, la transcripción incluye ruidos irrelevantes como aplausos, música de fondo o comentarios ininteligibles.
- **Errores en palabras:** dificultades en la transcripción de nombres propios, términos técnicos o palabras en idiomas distintos al español, lo que introduce incoherencias en el texto.

Posibles mejoras:

- **Uso de bibliotecas de puntuación automática:** Implementar modelos de puntuación o usar redes neuronales para limpiar el texto transcrito, mejorando la fluidez y la comprensión del resumen.
- **Filtrado de ruido:** Mejorar los filtros de transcripción para eliminar elementos innecesarios como música o aplausos, y priorizar solo la voz del locutor.

Imagen_17 – Ejemplo resumen noticia de video de YouTube (fuente propia)

Resumen de la Noticia:

ieno escuchamos al Vocero presidencial de la sala de conferencias de la casa de gobierno donde El Vocero presidencial Manuel adorni ya tenemos la imagen va a comenzar en instantes nada más la conferencia de prensa habitual en una jornada marcada por supuesto por lo que ha sido el discurso del presidente en la jornada de ayer en el congreso de la nación escuchamos a dor bueno Buen día a todos buen arranque de semana Todo bien sentate No está bien bueno bueno en primer lugar destacar algunos puntos sobresalientes de lo que fue ayer la presentación del presupuesto 2025 que ha realizado el presidente de la nación en el Congreso Nacional es la primera vez que un primer mandatario argentino Explica el presupuesto ante el congreso Y por supuesto Esto está relacionado con el valor que le damos a la palabra por supuesto siendo lo opuesto a lo que estamos acostumbrados en la Argentina que son las mentiras de los de los políticos cuando ocurren este tipo de cuestiones parece una anécdota pero no es menor jamás sea por falta de argumentos o por desconocimiento un presidente le explicó a los pagadores de impuestos o sea a todos nosotros Cómo se va a distribuir el fruto de ese de Nuestro esfuerzo es la primera vez también que se blinda el déficit fiscal el presupuesto estará anclado a los ingresos repito por sí no quedó claro efectivamente este va a ser un presupuesto donde los gastos van a estar anclados a los ingresos y esto es más o menos como funciona el sector privado no en el sector privado una empresa que tiene déficit a lo largo del tiempo quiebra en el caso del Estado el estado No quiebra pero si lo hace lo que hace es multiplicar pobres que es lo que nos ha pasado no décadas y décadas y décadas de déficit fiscal es cierto el estado no quebró como hubiese quebrado una empresa privada pero sí lo que ocurrió fue que nos han llevado a las más profundas de las miserias O al menos eh O al menos eh los niveles de pobreza de indigencia y de atraso que ha demostrado la Argentina efectivamente se puede comparar con lo de una empresa privada cuando sufre sistemáticamente durante años déficit las empresas quiebran los países generan pobreza y mucha mucha mucha miseria eh A partir de ahora bueno y por por supuesto No por al de alguna manera es que Hemos llegado a un 2% de inflación a un 60 55 por de pobres a un 10 o 12 o 15% de indigencia bueno Esto no fue por nada no Esto fue efectivamente porque las cosas se han hecho mal y porque lo que para nosotros es la columna vertebral que es tener las cuentas en orden no se ha cumplido en la mayoría de los años del último del último siglo a partir de ahora el equilibrio fiscal va a ser Norma No solo por el presupuesto 2025 sino para todo lo que venga en la Argentina independientemente de la coyuntura económica y a cualquier gesto de buena intención que puedan tener en el congreso promulgando leyes eh todas las leyes vamos a pretender que toda aquella que implique un aumento de gasto explique tal cual marca la ley pero que explique qué partida se va a reducir para poder afrontarlo eh en otro orden de cosas contarles que los contribuyentes que se adhieran al régimen de regularización de activos podrán pagar el impuesto obligatorio en dólares con una transferencia bancaria internacional para quienes participan de esta primera etapa de regularización eh les recordamos que el pago debe hacerse antes del 30 de septiembre todos los argentinos quedan invitados a regularizar sus activos sin importar el monto todos los detalles de esto lo repito una vez más se encuentran en la página de afip Tal como corresponde Por último señalar que hoy se publicó el decreto 830 del 2024 que desregula dos aspectos muy importantes del transporte por un lado se desregulan los servicios de oferta libre de transporte público de pasajeros para el área metropolitana de Buenos Aires por lo que a partir de ahora serán los transportistas y las empresas quienes determinen la cantidad de servicios los recorridos las tarifas y demás cuestiones esto lo que va a ha ser indefectiblemente es aumentar la oferta para todos los pasajeros y por otro y esto es tal vez lo más importante del decreto se modificó el registro único del transporte automotor conocido como ruta que reglamenta el transporte de cargas en la Argentina no solo a partir de ahora va a ser electrónico sino que va a ser sin costo sin la necesidad de presencialidad Y además que esto Tal vez sea lo máximo posible que sea para los científicos deberían hablar sobre este punto de vista.

Imagen_18 -Ejemplo resumen de noticia obtenido de url de video Imagen_17 (fuente propia)

Resúmenes resultantes de ficheros PDFs

En este caso, se evalúan los resultados obtenidos al procesar archivos PDF que contienen noticias o artículos (ver ejemplo Imagen_19 e Imagen_20)

Resultados obtenidos:

- La extracción de texto desde PDFs es exitosa cuando el archivo PDF es simple y tiene un formato estándar (una columna, texto claro y sin imágenes).

Errores detectados:

- **Ruido en el texto:** en archivos con múltiples columnas, encabezados, pie de página o numeración de páginas, el texto extraído contiene muchos elementos no deseados que interfieren con la calidad del resumen.

- **Imágenes o gráficos:** el sistema no puede procesar adecuadamente PDFs que contienen **imágenes** o gráficos, lo que genera problemas cuando estos elementos contienen información relevante para el texto.
- **Formateo inconsistente:** algunos textos extraídos carecen de la estructura correcta, con saltos de línea o mezcla de contenido de diferentes partes del documento.

Posibles mejoras:

- **Mejora en el procesamiento de PDFs complejos:** implementar librerías y/o modelos que puedan manejar archivos PDF con formatos más complejos (varias columnas o imágenes), utilizando herramientas como OCR avanzado (Reconocimiento Óptico de Caracteres) que identifiquen la disposición de los elementos en el documento.
- **Filtrado de ruido:** optimizar las funciones de limpieza de texto para eliminar encabezados, pies de página y otros elementos que no sean parte del contenido principal.



Imagen_19 – Ejemplo resumen de noticia de fichero PDF (fuente propia)

Resumen de la Noticia:

97 Anuario CEIPAZ 2023-2024 En su obra 100 mitos sobre Oriente Próximo, el profesor Fred Halliday desmontaba la creencia extendida de que la crisis del Mundo Árabe deriva del impacto negativo que ha tenido el conflicto con Israel en los procesos de cambio social y de democratización. Aunque es "indiscutible que el conflicto entre Palestina e Israel ha servido para reafirmar el autoritarismo de los estados vecinos: Egipto, Jordania, Líbano y Siria", escribía, este enfrentamiento "es una explicación parcial y, a menudo, poco más que un pretexto para la persistencia de los gobiernos autoritarios en los Estados árabes" (Halliday, 2006). Y concluía que "en cualquier caso, los regímenes árabes siempre han tratado de utilizar la 'prioridad urgente' del conflicto con Israel para acallar la crítica dirigida a los aspectos más represivos y antidemocráticos de sus gobiernos". El impacto del conflicto de Gaza en la región Rosa Meneses Periodista de El Mundo especializada en Oriente Medio y el Magreb 98 De Egipto a Siria, de Irán al Golfo, el conflicto palestino-israelí vuelve a ser utilizado como "prioridad urgente" para reforzar la represión de los gobiernos y enrocarse en el poder. La guerra que estalló el 7 de octubre de 2023 con el ataque terrorista de milicianos del grupo islamista palestino Hamas a Israel y la fulminante ofensiva del ejército israelí contra la Franja de Gaza vuelve a presentar un pretexto como oportunidad de reafirmación autoritaria para los países de la región. De Egipto a Siria, de Irán al Golfo, el conflicto palestino-israelí vuelve a ser utilizado como "prioridad urgente" para reforzar la represión de los gobiernos y enrocarse en el poder. Al tiempo que supone un problema real de escalada violenta regional con consecuencias imprevisibles. En el momento en que se escribe este texto (abril de 2024), la guerra en Gaza continúa y con ella si guen evolucionando sus consecuencias en Oriente Próximo, por lo que los análisis también se encuentran en proceso de transformación según los nuevos acontecimientos que se vayan sucediendo. Devastación y contagio La ofensiva militar israelí contra Gaza ha provocado como consecuencia la peor catástrofe humanitaria que ha sufrido una pequeña banda de territorio de la dimensión de la Franja (de apenas 365 kilómetros cuadrados densamente poblados) en lo que llevamos de siglo, por el nivel de muerte y devastación causado y su rápido deterioro en el tiempo. Un dato publicado por la Agencia de la ONU para los Refugiados Palestinos (UNRWA) lo certifica: en los primeros cuatro meses de conflicto murieron más niños y niñas palestinos que en los últimos cuatro años de guerras a nivel mundial (Lazzarini, 2024). Cuando se cumplían seis meses de confrontación los muertos palestinos (la inmensa mayoría de ellos, civiles) superaban los 33.000, a los que había que sumar las más de 8.000 personas desaparecidas (en su mayor parte, muertas bajo los escombros de los edificios bombardeados) y las más de 76.000 heridas. De los 2,3 millones de palestinos residentes en la Franja de Gaza, el 75% habían perdido sus hogares y medios de vida. A este alto grado de destrucción se une el efecto contagio del conflicto, con un potencial devastador que ya ha sido probado en la historia reciente. La violencia ha extendido sus tentáculos al Líbano, Siria, Irak y Yemen, países con una importante presencia de milicias armadas aliadas de Irán que han protagonizado enfrentamientos con el ejército israelí y que han sido objeto de ataques de éste y de las fuerzas de Estados Unidos. Irán –cómodo en un principio a la sombra de sus proxies– también acabó viéndose afectado directamente, cuando se traspasaron los seis meses de guerra. Jordania y Egipto –primeros países árabes firmantes de sendos tratados de paz con Israel– miran ilegalmente a los exploradores del mundo.

Imagen_20 – Ejemplo resumen de noticia obtenido Imagen_19 (fuente propia)

Resúmenes resultantes de noticias en línea (Web scraping)

En el caso de noticias extraídas directamente de sitios web a través de web scraping, los resultados dependen en gran medida del diseño y formato de cada página web (ver ejemplos Imagen_21, Imagen 22, Imagen_23 e Imagen_24)

Resultados obtenidos:

- El scraping de noticias es efectivo en su mayoría, obteniendo los titulares y textos completos de las secciones seleccionadas en las páginas web.
- En páginas con una estructura HTML clara y sin restricciones de acceso, los textos obtenidos son consistentes y adecuados para el resumen.
- Para el caso del periódico internacional "NBC", se incluyeron en el pipeline de procesamiento correspondiente funciones de traducción "inglés – español", las que implementan modelos de traducción. Estos últimos, si bien tienen una buena performance, tienen mayores limitaciones en los tokens de entrada, por lo que es necesario dividir el texto completo en "chunks", resumir dichos textos de menor longitud y concatenar finalmente toda la traducción.

Errores detectados:

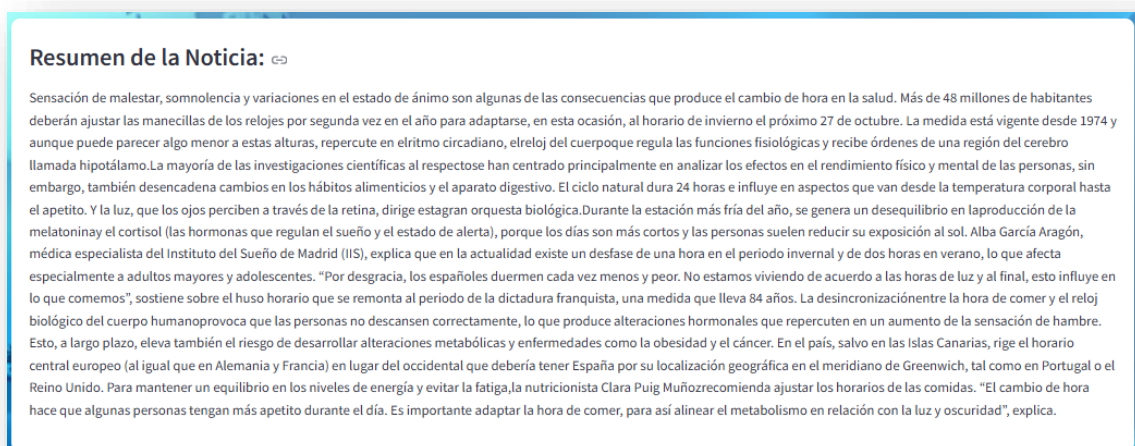
- **Mensajes de suscripción o paywalls:** algunas páginas muestran mensajes de suscripción, lo que interfiere con la extracción de noticias completas o se adicionan dichos mensajes a la noticia en sí.

Posibles mejoras:

- **Adaptación continua del scraping:** mejorar el scraping usando librerías que se adapten automáticamente a cambios menores en el HTML, o emplear herramientas más avanzadas como Selenium para manejar páginas más dinámicas.
- **Mejora del filtrado de contenido:** desarrollar un sistema de filtrado más robusto para eliminar anuncios, secciones irrelevantes o elementos dinámicos que puedan interferir en la calidad del texto extraído.



Imagen_21 – Ejemplo resumen noticia en línea (fuente propia)



Resumen de la Noticia: ⇄

Sensación de malestar, somnolencia y variaciones en el estado de ánimo son algunas de las consecuencias que produce el cambio de hora en la salud. Más de 48 millones de habitantes deberán ajustar las manecillas de los relojes por segunda vez en el año para adaptarse, en esta ocasión, al horario de invierno el próximo 27 de octubre. La medida está vigente desde 1974 y aunque puede parecer algo menor a estas alturas, repercute en el ritmo circadiano, el reloj del cuerpo que regula las funciones fisiológicas y recibe órdenes de una región del cerebro llamada hipotálamo. La mayoría de las investigaciones científicas al respecto han centrado principalmente en analizar los efectos en el rendimiento físico y mental de las personas, sin embargo, también desencadena cambios en los hábitos alimenticios y el aparato digestivo. El ciclo natural dura 24 horas e influye en aspectos que van desde la temperatura corporal hasta el apetito. Y la luz, que los ojos perciben a través de la retina, dirige esta gran orquesta biológica. Durante la estación más fría del año, se genera un desequilibrio en la producción de la melatonina y el cortisol (las hormonas que regulan el sueño y el estado de alerta), porque los días son más cortos y las personas suelen reducir su exposición al sol. Alba García Aragón, médica especialista del Instituto del Sueño de Madrid (IIS), explica que en la actualidad existe un desfase de una hora en el periodo invernal y de dos horas en verano, lo que afecta especialmente a adultos mayores y adolescentes. "Por desgracia, los españoles duermen cada vez menos y peor. No estamos viviendo de acuerdo a las horas de luz y al final, esto influye en lo que comemos", sostiene sobre el huso horario que se remonta al periodo de la dictadura franquista, una medida que lleva 84 años. La desincronización entre la hora de comer y el reloj biológico del cuerpo humano provoca que las personas no descansen correctamente, lo que produce alteraciones hormonales que repercuten en un aumento de la sensación de hambre. Esto, a largo plazo, eleva también el riesgo de desarrollar alteraciones metabólicas y enfermedades como la obesidad y el cáncer. En el país, salvo en las Islas Canarias, rige el horario central europeo (al igual que en Alemania y Francia) en lugar del occidental que debería tener España por su localización geográfica en el meridiano de Greenwich, tal como en Portugal o el Reino Unido. Para mantener un equilibrio en los niveles de energía y evitar la fatiga, la nutricionista Clara Puig Muñoz recomienda ajustar los horarios de las comidas. "El cambio de hora hace que algunas personas tengan más apetito durante el día. Es importante adaptar la hora de comer, para así alinear el metabolismo en relación con la luz y oscuridad", explica.

Imagen_22 – Ejemplo resumen noticia en línea obtenido de Imagen_21 (fuente propia)

Imagen_23 – Ejemplo resumen de noticia en línea – NBC (fuente propia)

Resumen de la Noticia:

Perfil Sections tv Featured More From NBC Follow NBC News Alerts No hay nuevas alertas en este momento Rusia e Irán pueden tratar de incitar a la violencia o protestas disruptivas en Estados Unidos después de las elecciones, dicen funcionarios de inteligencia estadounidenses. Una evaluación no clasificada publicada el martes dice que la comunidad de inteligencia estadounidense "está cada vez más segura de que los actores rusos están considerando —y en algunos casos implementando— una amplia gama de esfuerzos de influencia cronometrados con las elecciones. La Oficina del Director de la Inteligencia Nacional, que coordina la respuesta de las agencias de inteligencia a las campañas de influencia extranjera, ha reiterado en los últimos meses que Irán está tratando de influir en los estadounidenses. En 2020, Estados Unidos acusó a Irán de crear un sitio web en el que En diciembre de 2020, Irán casi con toda seguridad fue responsable de la creación de un sitio web que contenía amenazas de muerte contra funcionarios electorales estadounidenses, dice la evaluación. La oficina del director de inteligencia ha advertido constantemente que tres adversarios estadounidenses han llevado a cabo una persistente ópera propagandística y una campaña en los medios sociales, haciéndose pasar por estadounidenses derechistas, pidiendo que varias figuras públicas vinculadas a esa elección sean asesinadas. Los tres tienen como objetivo denigrar el proceso democrático, aunque sus preferencias presidenciales son diferentes: Rusia apoya al ex presidente Donald Trump para presidente, Irán apoya al presidente Vice Kamala Harris, y China no tiene un claro favorito. Irán y China generalmente han negado las irregularidades. RT, un medio de comunicación dirigido por el Kremlin y acusado por el Departamento de Justicia, ha emitido declaraciones Portavoz del embajador de Irán ante las Naciones Unidas, la Embajada de China en Washington y el Ministerio de Relaciones Exteriores de Rusia no respondieron a las solicitudes de comentarios. En un memorando desclasificado, parcialmente redactado, fechado el 8 de octubre, también publicado el martes, la oficina del director de inteligencia encontró que los tres países están "mejor preparados para explotar las oportunidades de ejercer influencia en las elecciones generales de Estados Unidos después de que las urnas cierran el día de las elecciones debido a las lecciones extraídas del 2020. "Evaluamos que Irán está tratando de fomentar la discordia social, avivar la violencia y socavar la confianza en el proceso democrático de Estados Unidos, independientemente de quién gane las elecciones", añade. "En una llamada de prensa que previó el memorando, un funcionario de inteligencia dijo a los periodistas que Estados Unidos estaba particularmente preocupado por el intento de Rusia de alentar la violencia en las protestas estadounidenses después de las elecciones. los actores pueden percibir una ventana de vulnerabilidad para empresa, con la forma de los comunes de Irán.

Imagen_24 – Ejemplo de resumen de noticia en línea obtenido de Imagen_23 (fuente propia)

Resultados comunes por tipo de modelo

Si bien los resultados finales obtenidos dependen de la calidad de los textos transcritos y extraídos de las diferentes fuentes definidas de noticias, el resumen varía también de acuerdo con los ajustes parametrizables:

- Tipo de Modelo.
- Largo del resumen
- Aleatoriedad
- Modo de generación

Posibles mejoras:

- **Mejora en la calidad de los resúmenes generados:** implementar modelos de IA generativa más avanzados, como mistralai/Mistral-7B-v0.1 o Llama, para obtener resúmenes más precisos y naturales, siempre y cuando se dispongan de mayores recursos computacionales.
- **Despliegue escalable:** para un uso más amplio, sería recomendable desplegar la aplicación en una infraestructura más robusta, como un servidor dedicado o una solución en la nube escalable (por ejemplo, AWS, Google Cloud) que permita atender a un mayor número de usuarios simultáneamente.

CONCLUSIONES:

Finalmente, haciendo referencia a los objetivos planteados inicialmente en esta memoria, concluimos que, si bien las herramientas y metodologías implementadas fueron consistentes, los resultados son variables y dependen de numerosos factores. Dentro de estos últimos, resaltamos la participación principal de los modelos de IA generativa incorporados, que, si bien son poderosos, resultaron, en nuestro caso, sensibles a la calidad del texto con el que se alimentan. Teniendo luego estas últimas consideraciones en cuenta, resolvemos que los mejores resultados se lograron con textos obtenidos mediante web scraping, en comparación con los provenientes de PDFs o videos.

El contenido extraído de sitios web de noticias, al estar mayormente estructurado y limpio, permitió a los modelos generar resúmenes más coherentes y precisos. Por otro lado, las transcripciones de videos y los textos extraídos de PDFs presentaron mayores dificultades debido al "ruido" inherente: errores de transcripción, fragmentos irrelevantes o la ausencia de una estructura clara. Este ruido afectó negativamente la capacidad de los modelos para procesar y generar resúmenes de calidad, lo que sugiere que los modelos LLMs pequeños de IA generativa, como los utilizados en este proyecto, funcionan óptimamente con texto bien estructurado y libre de interferencias.

Considerando estos hallazgos, un área clave de mejora sería la optimización del procesamiento previo del texto en casos como transcripciones de video o extracción de PDF, así como la prueba e implementación de modelos de mayor capacidad.

Este proyecto sienta una base sólida, pero a medida que se integren técnicas de preprocesamiento más avanzadas y se continúe refinando el uso de estos modelos LLMs, se podrían obtener resúmenes más precisos y útiles, incluso cuando las fuentes de texto no sean del todo limpias o estructuradas.

El vertiginoso avance de la inteligencia artificial nos sitúa en un momento crucial para aprovechar todo el potencial que estas tecnologías ofrecen. El uso de la IA no solo está transformando la manera en que procesamos la información, sino que también está redefiniendo los límites de lo posible. En este contexto, el futuro promete nuevas herramientas capaces de superar las limitaciones actuales y enfrentar los retos de una manera cada vez más inteligente y precisa. Aprovechar estas tecnologías no solo nos permitirá resolver problemas más complejos, sino también explorar nuevos horizontes que aún no hemos imaginado.

REFERENCIAS BIBLIOGRAFICAS:

- Hugging Face, Inc. (s.f.). <https://huggingface.co>
- YouTube. (s.f.). <https://youtubetranscript.com>
- PyMuPDF. (s.f.). <https://pymupdf.readthedocs.io/en/latest/>
- BeautifulSoup. (s.f.). <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>
- Selenium. (s.f.). <https://www.selenium.dev/>
- LangChain. (s.f.). <https://www.langchain.com/>