



Universidad de Jaén

Escuela Politécnica Superior de Jaén

(CÓD.: 23/24-4308) DESARROLLO DE UN INTERFAZ WEB PARA EL ANÁLISIS DE CORPUS

Autor: Rubén Aguayo Moreno

Grado: Ingeniería Informática

Tutor: Arturo Montejo Ráez

Fecha: 11/09/2024

Licencia CC



CREEA



Universidad de Jaén

Departamento de Informática

D. Arturo Montejo Ráez, tutor del Trabajo Fin de Grado titulado: '**(CÓD.: 23/24-4308) DESARROLLO DE UNA INTERFAZ WEB PARA EL ANÁLISIS DE CORPUS**', que presenta D. Rubén Aguayo Moreno, otorga el visto bueno para su entrega y defensa en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2024

El alumno:

El tutor:

Rubén Aguayo Moreno

Arturo Montejo Ráez

Agradecimientos

En primer lugar, quisiera agradecer a toda mi familia, en especial a mis padres y a mi hermana, por estar a mi lado en cada paso que doy y por ofrecerme siempre su confianza y apoyo incondicional en todo lo que pueden. Su amor y aliento han sido fundamentales para llegar hasta aquí.

Gracias también a mi pareja, Lucía, quien ha sido un pilar emocional clave, especialmente en los momentos más complicados, animándome a seguir adelante. Hemos compartido muchas tardes trabajando juntos, y su compañía ha hecho que este proceso sea mucho más llevadero.

A mis amigos, tanto los de la infancia como los de la carrera, les agradezco por su apoyo y por hacer de este camino una experiencia más rica y significativa. En especial a Jesús, quien me ha ofrecido valiosos consejos basados en su propia experiencia y me ha brindado su ayuda siempre que la he necesitado.

Finalmente, quiero agradecer a todos mis profesores por haber contribuido, con cada asignatura, a mi formación, y en especial a mi tutor, Arturo, por sus orientaciones y consejos, que han sido de gran valor durante la realización de este TFG.

FICHA DEL TRABAJO FIN DE TÍTULO

Titulación	Grado en Ingeniería Informática
Modalidad	Proyecto de Ingeniería
Especialidad <small>(solo TFG)</small>	Tecnologías de la Información
Mención <small>(solo TFG)</small>	Sin mención
Idioma	Español
Tipo	General
TFT en equipo	No
Autor/a	Rubén Aguayo Moreno
Fecha de asignación	16/04/2024
Descripción corta	<p>Este trabajo consiste en la realización de una Interfaz Web, cuyo objetivo principal es el de poder analizar descriptivamente distintos corpus.</p> <p>Además, debe estar presente la posibilidad de realizar un análisis contrastivo en la que se puedan comparar características de dos colecciones diferentes.</p> <p>Para esto, se utiliza la herramienta TextFlow que proporciona diferentes analizadores, los cuales se emplean para realizar estos análisis.</p>

NORMAS APLICADAS EN ESTE DOCUMENTO

LOCALES

TFT-UJA:2017	Normativa de Trabajos Fin de Grado, Fin de Máster y otros Trabajos Fin de Título de la Universidad de Jaén (Normativa marco UJA aprobada en Consejo de Gobierno)
TFT-EPSJ:2017	Normativa sobre Trabajos Fin de Grado y Fin de Máster en la Escuela Politécnica Superior de Jaén (Normativa EPSJ aprobada en Junta de Escuela)
TFT-EPSJ	Criterios de evaluación y normas de estilo para TFG y TFM de la Escuela Politécnica Superior de Jaén

NACIONALES E INTERNACIONALES

ISO 2145:1978	Documentación - Numeración de divisiones y subdivisiones en documentos escritos
UNE 50132:1994	Traducción de la ISO 2145
APA 6ª edición	Estilo de referencias y citas de APA (American Psychological Association)

NORMAS UTILIZADAS COMO BASE O REFERENCIA

NACIONALES

UNE 157001:2014	Criterios generales para la elaboración formal de los documentos que constituyen un proyecto técnico
UNE 157801:2007	Criterios generales para la elaboración de proyectos de sistemas de información

*Estas normas se han utilizado **como base o referencia** para la inclusión de algunos contenidos y definiciones sobre elaboración de proyectos, entendiendo como **proyecto** la documentación consensuada entre una empresa y un cliente, que da lugar al perfeccionamiento de un contrato para la elaboración de una obra o la prestación de un servicio. Por consiguiente, no debe esperarse la aplicación de estas normas en cuanto a la completitud de los contenidos ni a la organización de los mismos.*

Contenido

1	Especificación del trabajo	13
1.1	Introducción	13
1.2	Objetivos del trabajo	14
1.3	Antecedentes y estado del arte	15
1.3.1	Desarrollo histórico del PLN	15
1.3.2	Análisis de Corpus	19
1.3.3	Visualización y Análisis de Datos	21
1.4	Descripción de la situación de partida	23
1.4.1	Descripción del entorno actual	23
1.4.2	Resumen de las deficiencias y carencias identificadas	24
1.5	Requisitos iniciales	26
1.6	Alcance	27
1.7	Hipótesis y restricciones	28
1.8	Estudio de alternativas y viabilidad	28
1.8.1	Alternativas en Frontend	28
1.8.2	Alternativas en Backend	31
1.8.3	Alternativas en Gestión Base De Datos	33
1.9	Descripción de la solución propuesta	35
1.10	Tecnologías utilizadas	37
1.11	Metodología de desarrollo de software	39
1.12	Estimación del tamaño y esfuerzo	39
1.13	Planificación temporal	40
1.14	Presupuesto	42
2	Diseño inicial	46
2.1	Especificaciones del sistema	46
2.2	Análisis y diseño del sistema	50
2.2.1	Diseño Base De Datos	51
2.2.2	Diagramas de secuencia	54
2.2.3	Diseño de la interfaz	59
3	Desarrollo	65
3.1	Primera Iteración	66
3.1.1	Planificación	66
3.1.2	Implementación	66
3.1.3	Pruebas y Resultado final	69
3.2	Segunda Iteración	71
3.2.1	Planificación	71
3.2.2	Implementación	71
3.2.3	Pruebas y Resultado final	73
3.3	Tercera Iteración	78
3.3.1	Planificación	78
3.3.2	Implementación	78
3.3.3	Pruebas y Resultado final	82
3.4	Cuarta Iteración	86
3.4.1	Planificación	86

3.4.2	Implementación	87
3.4.3	Pruebas y Resultado final.....	90
4	Conclusiones y trabajos futuros.....	93
5	Definiciones y abreviaturas.....	96
6	Manual de despliegue	97
7	Bibliografía	99

Índice de ilustraciones

Ilustración 1 – Explicación Prueba de Turing.....	16
Ilustración 2 – Explicación aprendizaje semi-supervisado.....	17
Ilustración 3 – Herramientas análisis de sentimientos.....	20
Ilustración 4 – Visualización de datos.....	21
Ilustración 5 – Palabras más frecuentes.....	24
Ilustración 6 – Comparación Frontend.....	30
Ilustración 7 – Comparación Backend.....	32
Ilustración 8 – Comparación Gestor Base de Datos.....	34
Ilustración 9 – Arquitectura Aplicación.....	36
Ilustración 10 – CSS en línea.....	37
Ilustración 11 – Diagrama de Gantt.....	40
Ilustración 12 – Caso de Uso. Iniciar Sesión.....	46
Ilustración 13 – Caso de Uso. Registro.....	47
Ilustración 14 – Caso de Uso. Gestionar Corpus.....	47
Ilustración 15 – Caso de Uso. Añadir Colección.....	48
Ilustración 16 – Caso de Uso. Análisis Descriptivo.....	48
Ilustración 17 – Caso de Uso. Descargar Análisis.....	49
Ilustración 18 – Caso de Uso. Análisis contrastivo.....	49
Ilustración 19 – Diseño Base de Datos.....	51
Ilustración 20 – Diagrama de secuencia. Inicio Sesión.....	54
Ilustración 21 – Diagrama de secuencia. Registro.....	55
Ilustración 22 – Diagrama de secuencia. Añadir Colección.....	56
Ilustración 23 – Diagrama de secuencia. Análisis.....	57
Ilustración 24 – Diagrama de secuencia. Ver Datos.....	58
Ilustración 25 – Sketch Inicial.....	59
Ilustración 26 – WireFrame InicioSesión.....	60
Ilustración 27 – WireFrame Registro.....	61
Ilustración 28 – WireFrame Página principal.....	61
Ilustración 29 – WireFrame Añadir Colección.....	62
Ilustración 30 – WireFrame Análisis Descriptivo.....	63
Ilustración 31 – WireFrame Análisis Contrastivo.....	64
Ilustración 32 – Backend Rutas.....	67
Ilustración 33 – Frontend Rutas.....	68
Ilustración 34 – App.js.....	69
Ilustración 35 – Inicio Sesión.....	70
Ilustración 36 – Registro.....	70
Ilustración 37 – Descomprimir Zip.....	72
Ilustración 38 – Leer Archivo.....	72
Ilustración 39 – Cerrar Sesión.....	73
Ilustración 40 – Página Principal (Antes).....	74
Ilustración 41 – Página Principal (Después).....	75
Ilustración 42 – Añadir Colección (Antes).....	75
Ilustración 43 – Añadir Colección (Después).....	76
Ilustración 44 – Ver Colección (Antes).....	76
Ilustración 45 – Ver Colección (Después).....	77

Ilustración 46 – Ver Colección (Después) 2.....	77
Ilustración 47 – Análisis.....	79
Ilustración 48 – Ejecución archivo.....	80
Ilustración 49 – Tabla de metadatos.....	81
Ilustración 50 – Descargar datos.....	82
Ilustración 51 – Análisis Descriptivo Antes.....	83
Ilustración 52 – Análisis Descriptivo Después.....	84
Ilustración 53 – Ver Metadato Antes.....	85
Ilustración 54 – Ver Metadato Después.....	85
Ilustración 55 – Realizar análisis contrastivo.....	87
Ilustración 56 – Tabla análisis contrastivo.....	89
Ilustración 57 – Tabla análisis contrastivo 2.....	89
Ilustración 58 – Análisis Descriptivo Antes.....	91
Ilustración 59 – Análisis Descriptivo Después.....	91
Ilustración 60 – Ver metadato contrastivo Antes.....	92
Ilustración 61 – Ver metadato contrastivo Después.....	92

Índice de tablas

Tabla 1 – Planificación Temporal.....	41
Tabla 2 – Coste Personal.....	43
Tabla 3 – Coste Software.....	43
Tabla 4 – Coste Hardware.....	44
Tabla 5 – Otros costes.....	44
Tabla 6 – Coste Total.....	45

1 ESPECIFICACIÓN DEL TRABAJO

En este capítulo se presenta la especificación del trabajo, con una estructura y contenidos **inspirados** en los criterios y recomendaciones que establece la norma UNE 157801:2007 - “*Criterios Generales para la elaboración de proyectos de Sistemas de Información*”.

A lo largo del documento se utilizarán términos y acrónimos cuya descripción aparecen en el apartado 6 (Definiciones y abreviaturas).

1.1 Introducción

En el mundo actual, existe un especial interés por el análisis de grandes cantidades de texto, también llamado **análisis de corpus**, lo cual es fundamental en campos como la lingüística, la psicología y la inteligencia artificial. Esto, permite realizar comparaciones a gran escala entre los objetos presentes en ellos, lo que se conoce como lectura distante. Esto posibilita observar fenómenos que no siempre son evidentes al leer de manera tradicional, además de poder demostrar hipótesis sobre textos.

Con este Trabajo Fin de Grado se pretende crear una interfaz web que sea fácil de usar y capaz de realizar este análisis. Para la realización de esta, se utilizará el proyecto “**TextFlow**”, impulsado por el Departamento de Ingeniería Informática de la Universidad de Jaén, en el cuál se encuentra el tutor de este TFG, Arturo Montejo Ráez. De forma resumida, esta herramienta realiza un análisis detallado de cada uno de los textos, en el que extrae aspectos como Volumetría, Complejidad o Polaridad, entre otras cosas. Sin embargo, es difícil de utilizar puesto que no tiene una página interactiva definida.

Esta página web permitirá a los usuarios interactuar de forma eficiente, eficaz y productiva con este sistema de análisis de corpus, permitiendo una mayor facilidad en su uso, en el procesamiento y en la visualización de datos textuales. El desarrollo

de esta plataforma no solo va a aumentar la facilidad de acceso y el uso de TextFlow, sino que, a su vez, va a proporcionar la capacidad de poder analizar una gran cantidad de textos de forma más rápida y a su vez, va a permitir la comparación entre estos mediante las características extraídas.

1.2 Objetivos del trabajo

La tarea principal de este trabajo es la de realizar una interfaz web para la herramienta TextFlow de análisis de corpus como ya se ha comentado anteriormente. Sin embargo, para la obtención de esta, el trabajo se va a dividir en la siguiente serie de objetivos:

- **Diseño de una Interfaz de Usuario** que satisfaga las necesidades del usuario, que sea fácil de utilizar y bastante intuitiva. Esta, debe permitir el acceso a los analizadores y funcionalidades de TextFlow.
- **Desarrollar las funcionalidades** más importantes que deben estar presentes. El inicio de sesión o registro de los usuarios, la carga de archivos en formato .zip, la posibilidad de ver el contenido o la capacidad de analizar descriptivamente estos o comparar las características de dos corpus diferentes entre sí son algunas de estas utilidades.
- Permitir la **comunicación entre el backend y el frontend** aportando las funcionalidades que ofrece TextFlow, permitiendo una conexión rápida entre ambos y ofreciendo los servicios necesarios para realizar ambos análisis (descriptivo y contrastivo). A su vez, se debe permitir una gestión eficiente de las peticiones y las respuestas.
- **Desarrollar las funcionalidades del backend.** Para esto, se debe gestionar y poder mantener la gestión de la sesión del usuario, como se guardan las contraseñas o prohibiendo el acceso a páginas no permitidas. Además, la administración y el mantenimiento de la base de datos es otra de estas tareas importantes. En general, se debe de tener una especial atención en la seguridad del sistema.

1.3 Antecedentes y estado del arte

Como en todos los aspectos y especialmente en los que están en auge en los últimos tiempos, entender y comprender su evolución a lo largo del tiempo es una tarea muy útil. Ver en qué consiste el análisis de corpus y cómo este ha ido evolucionando a lo largo de la historia, junto con el uso de interfaces para la realización y la visualización de los patrones extraídos de estos en las aplicaciones de PLN, conforman un papel muy importante en la comprensión y en el desarrollo de este trabajo. Es por esto, que durante este apartado se verán los aspectos más significativos de la evolución de todo esto para así poder justificar ciertas características y decisiones que se han tomado para el desarrollo de la interfaz web.

1.3.1 Desarrollo histórico del PLN

El análisis de corpus, emplea técnicas del PLN, es por eso que antes de centrarnos en el análisis debemos hacer un repaso del Procesamiento del lenguaje natural.

El Procesamiento del Lenguaje Natural [1] es el campo de conocimiento de la Inteligencia Artificial encargado de investigar la forma que tienen de comunicarse tanto las máquinas con las personas como al contrario mediante el uso de lenguas naturales, como el español o el inglés.

Virtualmente, cualquier lenguaje humano puede ser tratado por los ordenadores. Sin embargo, esto no ha sido nunca una tarea sencilla. Lógicamente, algunas limitaciones de interés económico o práctico hacen que solo los idiomas más hablados en el mundo tengan cabida en el mundo digital con aplicaciones en uso.

El PLN es una disciplina que se empezó a trabajar hace muchos años, aunque ya existía trabajos anteriores (pero todavía no considerados PLN). Este concepto, surge sobre los años 50 como una subdivisión o subparte de la Inteligencia Artificial y la Lingüística, teniendo como objetivo la investigación de los desafíos relacionados con la automatización de la generación y comprensión del lenguaje natural hablado por los seres humanos. Anterior a los años 1950 ya existían algunas tareas pero fue

Alan Turing, quien publicó un documento llamado "Inteligencia", donde introdujo lo que actualmente se conoce como la Prueba de Turing para evaluar la inteligencia. Esta prueba consistía en que un humano examinara conversaciones en lenguaje natural entre un humano y una máquina diseñada para generar respuestas similares a las de un humano.



Ilustración 1 – Explicación Prueba de Turing

En sus primeras etapas, el enfoque predominante se basaba en reglas gramaticales que se realizaban a mano. Los sistemas creados usaban patrones lingüísticos y reglas de gramática para analizar y generar texto, lo que hacía cada vez más complicada su capacidad para manejar la complejidad y variabilidad del lenguaje natural. No fue hasta la década comprendida entre los años 90 y los 2000 cuando el campo del PLN experimentó un cambio significativo hacia métodos estadísticos y basados en aprendizaje automático o DL (Deep Learning). Gracias a esto se produjo un aumento en la disponibilidad de grandes cantidades de datos textuales y al desarrollo de algoritmos más potentes para procesar estos datos. Ya no se dependía únicamente de reglas, sino que los sistemas de PLN comenzaron a utilizar modelos estadísticos y de aprendizaje automático para aprender patrones y estructuras del lenguaje de manera automática a partir de la utilización de datos de entrenamiento para su posterior uso con datos de Test.

En esta década, [2] la utilización de modelos estadísticos creció de manera exponencial. Las investigaciones recientes se centran cada vez más en los algoritmos de aprendizaje no supervisados y semi-supervisados. Dichos algoritmos pueden aprender de datos que no se han etiquetado a mano con las respuestas

deseadas, o usando una combinación de datos que están anotados y no anotados. En general y como es evidente, este trabajo es mucho más difícil que el aprendizaje supervisado, y generalmente produce resultados menos precisos para una cantidad dada de datos de entrada. Sin embargo, tiene como ventaja que existe una enorme cantidad de datos no anotados disponibles, incluido, entre otras cosas, todo el contenido de la WWW (World Wide Web).

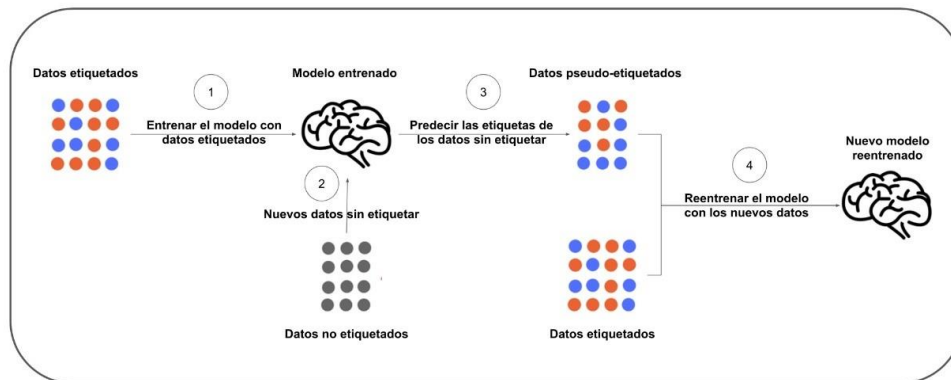


Ilustración 2 – Explicación aprendizaje semi-supervisado

A partir de 2010, nacieron las técnicas basadas en las redes neuronales artificiales y el aprendizaje profundo se generalizó en el procesamiento del lenguaje natural. Este último dio el salto más significativo en términos de precisión y capacidad de generalización. Estos métodos permitieron a los sistemas de PLN realizar tareas más complejas como el reconocimiento de entidades nombradas, la traducción automática, la generación de texto y el análisis de sentimientos con niveles de precisión que rivalizan con los humanos en ciertos casos.

Actualmente, además de avanzar más aún en algoritmos de aprendizaje automático, el PLN ha aprovechado también el procesamiento distribuido y el análisis de big data para manejar grandes volúmenes de datos o textos de manera eficiente. Plataformas como Apache Hadoop y Spark han facilitado el procesamiento paralelo de datos a escala, permitiendo a los sistemas de PLN analizar corpus masivos de textos en tiempos razonables. A pesar de que el PLN está creciendo de forma exponencial, este podría crecer más si no fuese por los siguientes problemas:

- **Ambigüedad:** Esto se refiere a la capacidad que tiene una palabra o una oración de tener diferentes significados dependiendo de quién lo diga o del contexto en el que se utilice. Dentro del campo de estudio del análisis de emociones a través del Procesamiento del Lenguaje Natural (PLN), este concepto conlleva la capacidad de entender una oración o palabra de diversas formas, con el propósito de identificar la interpretación adecuada. La habilidad para transmitir múltiples interpretaciones emocionales dificulta la precisión en la interpretación de las emociones en un texto, impidiendo el crecimiento exponencial en el análisis emocional y generando confusión tanto para humanos como para ordenadores. Existen diferentes tipos de ambigüedades [3], algunas son:
 - *Ambigüedad léxica:* se trata de la ambigüedad de una sola palabra. Para saber el significado de estas determinadas palabras es necesario conocer el contexto sobre el que se está hablando o el idioma en el que se habla. Por ejemplo: el término "frío.
 - *Ambigüedad semántica:* en esta ambigüedad el significado de las palabras pueden malinterpretarse. Estas expresiones pueden transmitir emociones de manera no literal, lo que dificulta su interpretación de forma rápida y eficaz. Por ejemplo: el término "Adolfo Domínguez".
 - *Ambigüedad anafórica:* esta surge debido al uso de anáforas.
 - *Polisemia emocional:* dependiendo del contexto en el que se utilice una determinada palabra, esta puede tener varios significados emocionales y pueden ser tanto positivos como negativos. Por ejemplo: el uso de la palabra "gato".
 - *Ironía y sarcasmo:* a veces los humanos utilizamos la ironía y el sarcasmo, algo que no está plasmado en un texto directamente. Esta es más difícil de analizar, ya que no solo basta con saber el contexto, sino que también hay que saber las intenciones del hablante.
- **Variabilidad cultural:** hace mención a las disparidades en la utilización de las palabras a causa de aspectos como la cultura, el idioma y las

particularidades personales. Esto indica que la manera en que las personas emplean el lenguaje difiere dependiendo de su contexto cultural, idioma nativo y otros factores personales. La variedad de culturas y diferencias lingüísticas dificulta el desarrollo de análisis de emociones en PLN [4], generando un reto para crear modelos universales.

1.3.2 Análisis de Corpus

El análisis de corpus actualmente está teniendo un gran impacto en muchos aspectos, como lo son la lingüística, la informática, las ciencias sociales y muchas otras disciplinas. Un corpus hace referencia a la colección de textos escritos o hablados que se utilizan para ser analizados, ya sea realizar análisis lingüísticos cuantitativos, cualitativos o ambos. Estos textos pueden ser sacados de un gran número de lugares, bien de opiniones en algunas páginas web de productos, comentarios en redes sociales incluso de respuestas a asistentes virtuales.

A medida que ha pasado el tiempo, [5] este análisis ha ampliado su alcance y sofisticación, incluyendo herramientas muy útiles para el estudio y la comprensión del lenguaje humano. Con el avance de la tecnología computacional, el análisis de corpus comenzó a utilizar cada vez más herramientas más preparadas y desarrolladas para la recopilación y análisis de datos. Se empezó a utilizar, un software que permite buscar y analizar palabras y frases específicas en un corpus, algo que fue muy usado por los lingüistas. Con ello, se consiguió un análisis más detallado y preciso de los textos, facilitando estudios sobre colocaciones, patrones sintácticos y semánticos, y variación dialectal. Con esto, no solo se utilizaba en lingüística, sino que el análisis de corpus también se utilizó en la traducción para mejorar su calidad; en educación para mejorar la enseñanza en los colegios, institutos y universidades; en política, para mejorar los discursos que se daban en los medios de comunicación y para mejorar la sociedad en general.

Uno de los ámbitos que más han crecido y en el que más se está trabajando actualmente dentro del análisis de corpus es el análisis de sentimientos y la minería

de opiniones. Con la gran utilización, cada vez más por todas las edades, de las redes sociales y las plataformas de revisión online, los investigadores y las empresas comenzaron a utilizar técnicas de análisis de corpus para evaluar las opiniones y sentimientos expresados por los usuarios. Esta metodología permite identificar automáticamente si un texto expresa una opinión positiva, negativa o neutra, proporcionando una serie de valores que hacen que sea más fácil la toma de decisiones en marketing, desarrollo de productos y gestión de la reputación.

Con esto, las empresas son capaces de mejorar sus productos en base a las críticas de los usuarios en las redes sociales, pueden potenciar algunos de los que obtengan mejores impresiones. Además, se ayuda a los asistentes a mejorar la comunicación con el cliente, sabiendo el estado anímico en el que se puede encontrar entre otras cosas.

Usos de la herramienta de análisis de sentimientos

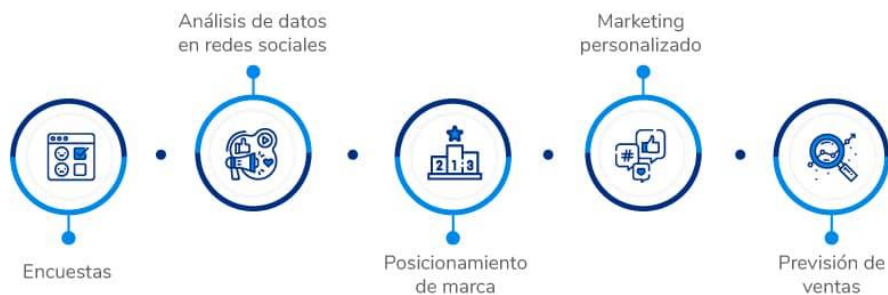


Ilustración 3 – Herramientas análisis de sentimientos

1.3.3 Visualización y Análisis de Datos

Cuando se realiza un análisis de un corpus determinado, el valor que se proporciona no es valor legible y entendible para todos los usuarios que no sean expertos en dicho ámbito. Para eso, los datos deben ser preparados para que sean comprendidos por todos los usuarios.

La visualización de datos es un componente fundamental en el análisis de corpus, ya que se encarga de transformar los datos textuales complejos o difíciles de comprender en representaciones gráficas y tabulares que son más fáciles de entender e interpretar. Esto, ayuda a identificar patrones, tendencias y anomalías que podrían pasar desapercibidos en un análisis puramente textual.

Esta visualización, te permite detectar e interpretar rápida y fácilmente las relaciones e interacciones, así como identificar tendencias que se verían a simple vista si estos datos estuvieran sin procesar. Una representación gráfica bien diseñada, no solo contiene información, sino que aumenta la eficiencia de su percepción, lo que hace que llame la atención y despierte el interés, a diferencia de los datos aislados y documentos que al no tener gráficos o tablas, pueden ser aburrido leerlos ya que son más difíciles de entender y no llaman la atención.

A continuación, se detallan algunas de las herramientas [6] y métodos más utilizados en la visualización de datos para el análisis de corpus:



Ilustración 4 – Visualización de datos

- **Gráficos:** Este es el tipo más común de visualización de datos. Estos, se crean a lo largo de los ejes X e Y. Además, muestran la dependencia de los datos entre sí. Pueden ser de barras o de líneas, teniendo como principal ventaja estos últimos, que permiten observar tendencias temporales y comparaciones a lo largo de una serie continua, facilitando el análisis de cambios dinámicos en el corpus.
- **Diagramas:** Estos, muestran las relaciones existentes entre palabras o entidades en un corpus, representando palabras como nodos y relaciones como enlaces entre ellos. Esta técnica es especialmente útil para visualizar estructuras complejas y relaciones entre múltiples elementos. Se suelen construir alrededor de los ejes. También se les puede representar en sectores o dentro de un sistema de coordenadas polares.
- **Tablas:** Estas, son una de las formas más básicas y efectivas de presentar datos del análisis de corpus. En una tabla, los datos se organizan en filas y columnas, lo que permite una comparación clara y directa de diferentes variables. Las tablas pueden incluir datos de frecuencia, estadísticas descriptivas, relaciones entre términos y más. Que son fáciles de interpretar y pueden ser utilizadas para resumir grandes volúmenes de datos de manera concisa son algunas de las ventajas que estas tienen.

1.4 Descripción de la situación de partida

Actualmente, existen algunas aplicaciones para el análisis de corpus, como es el caso de *AntConc*, la más conocida. Luego, existen muchas aplicaciones creadas por distintas universidades, las cuales no llegan a tener una interfaz bien definida con la que los datos se muestren bien organizados y en la que se puedan realizar comparaciones entre distintos análisis de diferentes corpus. Es esencial considerar que, en un entorno tan competitivo, debemos crear aplicaciones que realmente se diferencien de las demás y que tengan un propósito claro y valioso para justificar su existencia. Para eso, con este trabajo se busca crear una interfaz y que además de analizar, también compare diferentes análisis de estos corpus.

1.4.1 Descripción del entorno actual

Como ya se ha comentado anteriormente, alguna plataforma para analizar corpus es *AntConc*. En la universidad de Jaén, se ha impulsado una biblioteca que se encarga de analizar una serie de características en distintos apartados, ya sea volumetría del texto, emociones o lexicones. Además, este también es capaz de realizar un análisis contrastivo entre distintas etiquetas del mismo corpus e incluso de diferentes corpus. Esta puede ser usada en *Google Colab* separando los distintos algoritmos en celdas y en cualquier otro entorno de Python y es denominada **TextFlow**.

Este algoritmo o conjunto de algoritmos es capaz de mostrar detalladamente algunos aspectos. BDLASE, Complejidad, Emofinder, Emojis, Polaridad Emojis, Polaridad Emoticonos, Emociones, Ironía, ISAL, Lemas, Diversidad Léxica, LIWC, NER (entidades), NGrams, NCR, POS, Perplejidad, Polaridad, SEL, Estilometría y Volumetría son estos aspectos comentados. Al análisis de estos atributos se le denomina análisis descriptivo. Además de esto, es capaz de sacar gráficos y tablas de las diferentes características al realizar una comparación entre distintas etiquetas, la misma etiqueta o entre diferentes corpus. A este se le denomina análisis contrastivo.



Ilustración 5 – Palabras más frecuentes

Pero sin embargo, tiene una serie de carencias que se comentarán a continuación y por las cuales se realiza este trabajo.

1.4.2 Resumen de las deficiencias y carencias identificadas

Como ya he comentado anteriormente, **TextFlow** tiene una serie de deficiencias por las cuales se realiza este trabajo y las cuales hay que mejorar para que sea competente en el mercado de hoy en día.

TextFlow está reconocido como una biblioteca para Python. La forma más fácil de utilizarla hasta ahora es en Google Colab, por lo que tiene una interfaz muy pobre. Esta consta de celdas de código con sus respectivos comentarios explicativos las cuales se van ejecutando una a una. Además, por pantalla se muestran todas las importaciones de librerías realizadas, junto con una gran cantidad de mensajes de error y advertencias, la cual no lo hace muy llamativa de cara al usuario que lo quiera utilizar. Por lo tanto, se deberá realizar una interfaz que ayude en todo esto.

Además de los mensajes comentados, también se muestran los resultados de estos análisis, pero se muestran de forma desordenada, sin saber a qué característica o atributo pertenece cada análisis. Esto es, no se sigue una estructura clara y ordenada en el que se vea claramente a qué pertenece cada valor.

Otra de las carencias que se presenta tiene que ver con la gestión de los corpus. Con la interfaz de **TextFlow** tienes que ir cambiando los valores del código para poder analizar distintos corpus o tendrás que tener un código para todos ellos y varias pestañas abiertas, por lo que para el usuario será más difícil interactuar y al final acabará desistiendo. Además, no se puede ver el contenido de cada fichero del corpus.

Otra de las cosas a tener en cuenta, es que, con esta interfaz, si decides instalarlo localmente, se te instalará **TextFlow** en tu ordenador y te llenará memoria y carga en tu dispositivo, algo que el usuario no desea.

Finalmente, otra de las principales deficiencias es que no se muestra bien diferenciado qué datos se corresponden con el análisis descriptivo y cuáles con el análisis contrastivo.

1.5 Requisitos iniciales

En este apartado, se comentarán los requisitos principales que deben cumplirse para resolver las carencias que TextFlow presenta actualmente, para poder así mejorar su competitividad en el mercado y que sea todo más fácil y accesible para el usuario. Los requisitos que se deben presentar para la interfaz a realizar son los siguientes:

- La interfaz debe ser fácil de usar y permitir a los usuarios que puedan realizar análisis de corpus sin necesidad de tener conocimientos previos en la materia.
- La interfaz debería proporcionar un diseño claro, atractivo y que sea entendido por el usuario, intentando evitar la presentación mensajes de error o advertencias en cuanto a TextFlow.
- Los resultados de los análisis deben presentarse de manera ordenada y clara, con cada resultado asociado a su correspondiente característica o atributo, indicando la colección a la que pertenece.
- Los resultados en la interfaz deben incluir visualizaciones gráficas, en forma de tabla, para facilitar la interpretación y análisis de los datos.
- Los resultados del análisis descriptivo deberían poder descargarse en un .csv por si el usuario que interactuar con ellos para otra cosa diferente.
- La interfaz debe permitir a los usuarios gestionar y seleccionar diferentes corpus sin tener que modificarlos como ocurría en el Google Colab.
- Desde la interfaz, se debería poder visualizar el contenido de los ficheros de cada colección.
- Se debe poder añadir o eliminar colecciones.
- En la interfaz se debe poder diferenciar claramente los datos correspondientes a cada tipo de análisis (descriptivo o contrastivo).
- En el caso del análisis descriptivo, se debería poder eliminar análisis de cada atributo.

- En el caso del análisis contrastivo, se debe poder elegir el corpus con el que comparar y la característica que se quiere contrastar.
- La interfaz, debe poder diferenciar a los usuarios, teniendo cada uno sus propios corpus subidos y almacenados.
- La aplicación debería ser eficiente en términos de uso de memoria y carga en el dispositivo del usuario, sin tener que instalar TextFlow localmente en el dispositivo.
- La aplicación debería garantizar la seguridad de los datos del usuario y la privacidad de los análisis realizados.

1.6 Alcance

En este apartado se verán los límites del proyecto y los elementos que se entregarán a la finalización y entrega de este trabajo.

- Se entregará la aplicación final ya acabada, con su **código fuente** y un **archivo con la base de datos** creada. Esto, estará en un repositorio de Github, por lo que se entregará un **archivo .pdf** donde estará el **enlace** a este.
- Además, se entregará esta documentación dónde se explica qué se ha creado, todo el proceso de desarrollo, junto con una explicación de las principales funcionalidades que esta tiene y mejoras o consideraciones de cara al futuro.
- Un PowerPoint en formato PDF, donde se incluirán capturas de pantalla de las principales funcionalidades de la interfaz, acabando con un vídeo que muestra todo lo que esta contiene y la forma de utilizarla.

1.7 Hipótesis y restricciones

El TFG se define como una asignatura de 12 créditos, lo que supone que la duración total del proyecto será de 300 horas, incluyendo todas las etapas del ciclo de vida, con la excepción del mantenimiento. Por consiguiente, la principal restricción aplicable es la limitación de la duración del trabajo.

Además de esta, existe una restricción económica. El trabajo realizado se ha podido desarrollar sin necesidad de aportar ninguna cantidad de dinero ya que no se ha expuesto al mercado. En ese caso, habría que aportar una cantidad económica para esto en páginas como Google Cloud, además del posterior mantenimiento y solución de errores que esta tuviera.

1.8 Estudio de alternativas y viabilidad

Como ya se ha comentado anteriormente, la interfaz de **TextFlow** es compleja ya que se divide en líneas de código lo cual hace que la experiencia del usuario no experto disminuya notablemente. Es por esto, por lo que se realiza este trabajo, con el fin de realizar una interfaz intuitiva, fácil y accesible para todo cliente.

Para empezar a realizarla, en primer lugar tenemos que ver todas las alternativas que se nos ofrecen para su diseño y desarrollo. Frontend, Backend y Base de Datos son los 3 aspectos más importantes al tener en cuenta para el desarrollo de la interfaz Web. Para todos existen varios frameworks, lenguajes o librerías, los cuales tienen una serie de características que debemos tener en cuenta y valorar para elegir la mejor opción en cada caso. Se mostrarán las distintas opciones en los siguientes subapartados.

1.8.1 Alternativas en Frontend

A la hora de realizar la interfaz para TextFlow, es muy importante tener en cuenta algunos de los frameworks más importantes, para así evaluar las distintas tecnologías, enfoques y características disponibles para implementar el Frontend. Para saber cuál es la mejor no solo basta con poner especial atención en la

funcionalidad y apariencia del producto final, sino también a factores como el tiempo de desarrollo, la dificultad de aprendizaje del framework utilizado o el tiempo de carga de este. Algunos de los frameworks que he tenido en cuenta a la hora de diseñar como será el frontend son los siguientes:

- **React.js:** Esta es una biblioteca de JavaScript muy utilizada actualmente, que fue desarrollada por Facebook. Fue diseñada para construir interfaces de usuario interactivas de manera eficiente y escalable. Se ha convertido en una de las tecnologías más populares en el desarrollo web ya que presenta un enfoque muy importante en la creación de aplicaciones dinámicas y de alto rendimiento.
 - **Ventajas:** Entre sus ventajas destacan la modularidad debido a que es muy fácil reutilizar el código; cuenta con una gran cantidad de bibliotecas y herramientas, lo que hace que sea muy sencillo integrarse con otras tecnologías. Además, cabe destacar los grandes foros de ayuda para problemas con su utilización.
 - **Desventajas:** Entre las principales desventajas, nos encontramos con su difícil aprendizaje si no tienes mucha experiencia. Además, a la hora de descargarlo y poner en marcha la interfaz localmente requiere de una configuración compleja y muchas dependencias.
- **Angular:** Este es un framework de JavaScript que trabaja en el desarrollo web completo mantenido por Google. Este es de código abierto escrito en TypeScript. Su objetivo principal es desarrollar aplicaciones de una sola página. Está respaldado por Google, por lo que se encarga del mantenimiento y constantes actualizaciones de mejoras para este framework.
 - **Ventajas:** Entre sus ventajas encontramos que es un framework muy útil para el desarrollo de aplicaciones web robustas. Además, está capacitados para elaborar proyectos a gran escala y está respaldado por Google lo que le proporciona un soporte activo.

- **Desventajas:** Entre sus inconvenientes está la complejidad en el aprendizaje de este framework ya que hay muchos conceptos que entender. Además, es muy pesado y poco eficiente para realizar proyectos pequeños.

- **Bootstrap con JQuery:** es un framework de CSS muy demandado para crear interfaces de usuario en aplicaciones web. Además, JQuery es una biblioteca de JavaScript que simplifica las operaciones DOM y las interacciones del usuario. Proporciona una gran cantidad de elementos y plantillas CSS, HTML y JavaScript que cualquiera puede utilizar o modificar de manera gratuita.
 - **Ventajas:** Permiten crear interfaces de forma rápida con una estructura predefinida; son muy simples de utilizar y además se aprenden de forma rápida y sencilla; y son compatibles con una gran cantidad de navegadores, son algunas de las ventajas que este presenta.

 - **Desventajas:** Entre sus principales desventajas nos encontramos con que ofrece menor flexibilidad en la capacidad de personalizar la página web. Además, esta, está empezando a quedar obsoleta, por lo que si se saca al mercado su soporte puede estar muy limitado.



Ilustración 6 – Comparación Frontend

1.8.2 Alternativas en Backend

Otro aspecto muy importante en el desarrollo de una interfaz web es el backend. En el backend podemos utilizar una gran cantidad de librerías, bibliotecas o lenguajes de programación. Es por esto, que al igual que hemos realizado en el frontend, tenemos que analizar algunos de los mejores lenguajes y comprobar las mejores características y lo que más convenga para el trabajo a realizar. A continuación se muestran las ventajas y desventajas de algunos de ellos:

- **Django:** Este es un framework web de alto nivel para Python. Ayuda a realizar un desarrollo rápido, además de un diseño limpio y pragmático. Ofrece una serie de herramientas y funcionalidades integradas para simplificar el desarrollo de aplicaciones web. Este, facilita su trabajo al agrupar las diferentes funciones en distintos módulos más pequeños y mucho más fáciles de usar, a lo que se le denomina marco de aplicación web.
 - **Ventajas:** Entre sus principales ventajas encontramos que contiene una serie de características ya definidas como la autenticación, lo que hace más rápido el desarrollo de este. También contiene ciertas medidas de seguridad integradas lo que ayuda a proteger el sistema de ataques XSS. Otra de las ventajas es que ayuda a realizar aplicaciones que manejen grandes cantidad de carga.
 - **Desventajas:** Entre los principales inconvenientes encontramos que es difícil de aprender debido a su estructura, lo que llevaría mucho tiempo para esto. Además, no es tan flexible y no se puede personalizar libremente.
- **Flask:** es un framework para Python que es ligero y flexible, muy útil para aplicaciones pequeñas y medianas. Este, proporciona una base mínima y permite a los desarrolladores agregar extensiones según sea necesario, por lo que no tiene nada ya definido para usar.
 - **Ventajas:** Entre las ventajas que este framework ofrece está la flexibilidad, ya que deja al usuario que estructure su aplicación de la

mejor manera que considere. Además, es muy ligero y fácil de aprender, al igual que de usar.

- **Desventajas:** Entre las desventajas nos encontramos con que proporciona menos características ya integradas, algo que hace que tengamos que agregar más funcionalidades externas. Además, la seguridad en estos casos está en manos del desarrollador totalmente, a diferencia de Django.
- **Laravel:** es un framework PHP de código abierto que sigue el patrón de arquitectura Modelo-Vista-Controlador (MVC). Este, facilitar el desarrollo de aplicaciones web robustas y modernas, proporcionando una serie de herramientas y características integradas que simplifican el desarrollo, la seguridad y la gestión de bases de datos.
 - **Ventajas:** Ofrece muchas características ya definidas, como es el caso de la autenticación y la autorización; proporciona características de seguridad para evitar ataques del tipo XSS y CSRF, son algunas de las principales ventajas. Además tiene una comunidad grande y activa que ayuda a mejorar las funcionalidades de la aplicación
 - **Desventajas:** Al ser un framework PHP puede ser más difícil de aprender ya que tiene muchas funcionalidades. Además, su rendimiento puede no ser tan alto debido a la sobrecarga que este conlleva.



Ilustración 7 – Comparación Backend

1.8.3 Alternativas en Gestión Base De Datos

Otro de los aspectos muy importantes a la hora de realizar una interfaz web es el de como almacenamos los datos para guardarlos y que puedan ser mostrados en la interfaz. Estos datos, deben poder ser accesibles, modificables, incluso eliminables.

Para gestionar estos datos y las operaciones que se realizan con estos, necesitamos un gestor de base de datos. Al igual que con el frontend y el backend, existen muchos gestores de datos con sus características cada uno, pero debemos elegir el mejor para nuestra interfaz. Algunos de estos son los siguientes:

- **MySQL:** es un sistema de gestión de bases de datos relacional de código abierto muy popular, conocido por su fiabilidad y facilidad de uso. Pertenece de la pila LAMP (Linux, Apache, MySQL, PHP/Python/Perl) y se utiliza ampliamente en aplicaciones web y servicios en línea.
 - **Ventajas:** Entre las principales ventajas, nos encontramos con que es un gestor de código abierto, por lo que se puede utilizar de forma gratuita; ofrece muy buen rendimiento; y es muy útil, tanto para aplicaciones pequeñas como para grandes aplicaciones. Además, tiene una gran comunidad, donde existe mucha documentación sobre este, y en el que podemos preguntar todos los problemas que encontremos. Además, tiene un fácil acceso y una interfaz gráfica desde Xampp.
 - **Desventajas:** Entre los principales inconvenientes, encontramos que no tienen funcionalidades avanzadas integradas.
- **MongoDB:** es un sistema de gestión de bases de datos NoSQL orientado a documentos. Utiliza un formato de documentos BSON (Binary JSON) para almacenar datos, y es conocido por su flexibilidad y escalabilidad horizontal. Se utiliza para almacenar volúmenes masivos de datos. A diferencia de una base de datos relacional SQL tradicional, este no se basa en tablas y

columnas, si no que los datos se almacenan como colecciones y documentos.

- **Ventajas:** Entre las principales ventajas, nos encontramos con que es más fácil la manipulación de los datos no estructuras, tiene una mejor escalabilidad horizontal, lo que favorece al manejo de grandes volúmenes de datos y tráfico. Además es muy útil para aplicaciones que frecuenten muchos cambios.
- **Desventajas:** Entre los inconvenientes, encontramos que puede haber un retraso de la sincronización de los datos y no son tan eficientes en transacciones complejas.
- **SQLite:** es un sistema de gestión de bases de datos relacional ligero y embebido, que almacena datos en un solo archivo de disco. Es ampliamente utilizado en aplicaciones móviles, sistemas embebidos y proyectos pequeños.
 - **Ventajas:** Entre sus ventajas, nos encontramos con que es fácil de configurar y utilizar, ya que es un solo archivo y se puede trasladar fácilmente entre sistemas. Otra de las características, es que es adecuado para proyectos con almacenamiento pequeños y operaciones limitadas.
 - **Desventajas:** Entre los inconvenientes se encuentra, que no está preparado para grandes volúmenes de datos. Además, no tiene características avanzadas de bases de datos más grandes y completas.



Ilustración 8 – Comparación Gestor Base de Datos

1.9 Descripción de la solución propuesta

Al realizar el análisis del trabajo y al empezar a trabajar con el proyecto **TextFlow**, nos damos cuenta de que tiene una interfaz bastante pobre y que es difícil de usar para usuarios no expertos, incluso para algunos expertos. Esto es debido, como ya se ha comentado anteriormente, a su pobre interfaz.

Esta UI es muy modular. Está dividida en varias líneas de código que realizan una serie de operaciones, ya sea importaciones de las librerías necesarias u órdenes que realizan distintos análisis. Sin embargo, esto es una tarea poco productiva y lenta de cara para los usuarios que quieran realizar análisis de sus corpus.

Ante esto, la solución propuesta con este trabajo, consiste en una interfaz gráfica, encargada de mejorar las carencias que el proyecto de TextFlow tiene y así poder mejorar la experiencia del usuario ejecutando sus análisis. Esta, va a hacer que los clientes que la utilicen puedan realizar sus estudios sin necesidad de tener conocimientos previos. Además, se pretende mostrar los resultados de manera legible y ordenada en formato de tabla.

Otro de los aspectos que se va a realizar dentro de la interfaz, es el de poder tener más de un corpus a la vez analizados guardados, sin necesidad de preocupar al usuario de tener que ordenar y gestionar los corpus guardados. Con esto, también se consigue que se pueda ver el contenido de los ficheros, sin necesidad de tener que irlos abriendo uno a uno.

Finalmente, se permitirá el análisis contrastivo entre distintos corpus, comparando cada uno de los metadatos numéricos que estos tienen, consiguiendo así que dicho análisis no esté limitado al mismo corpus.

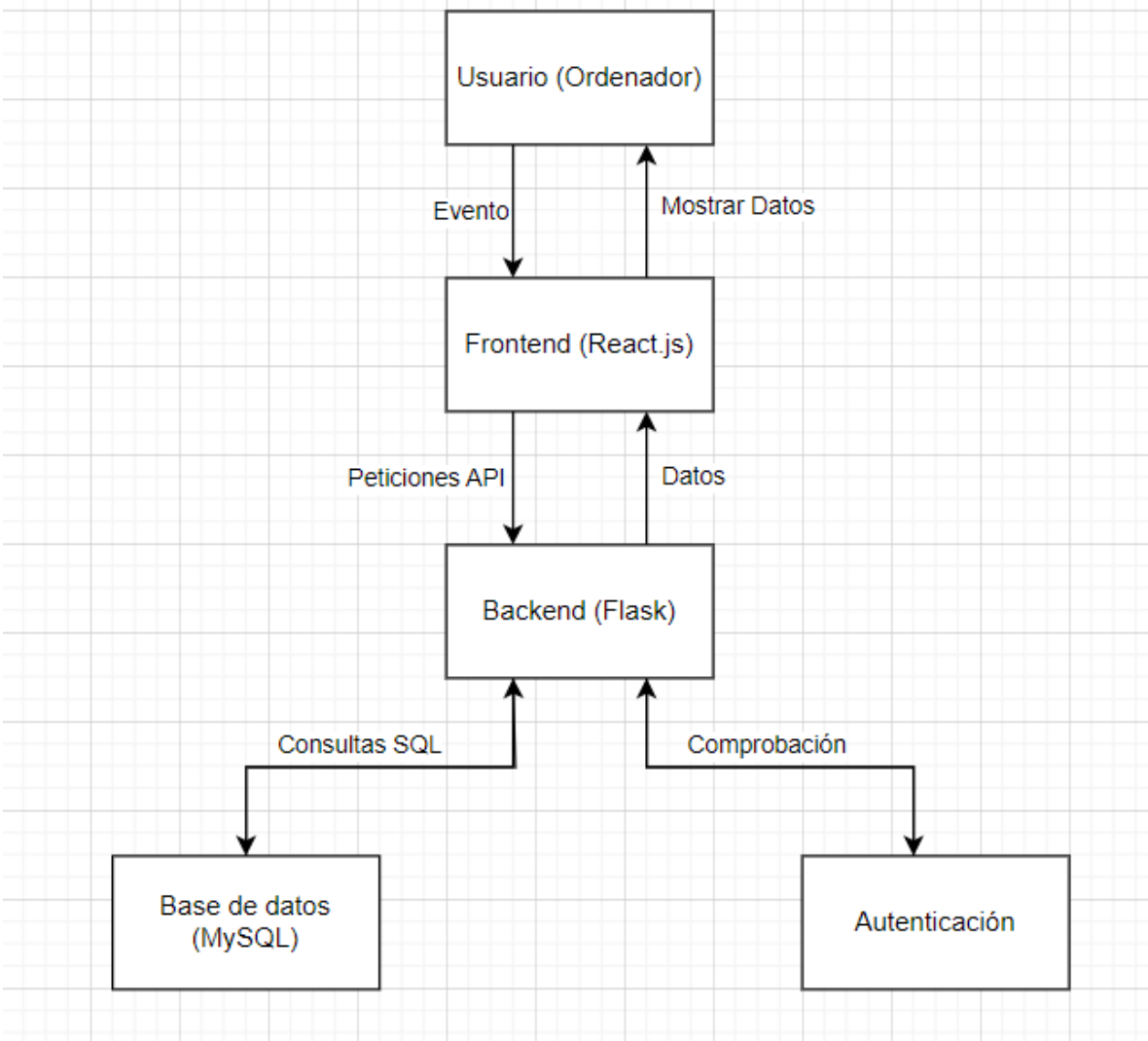


Ilustración 9 – Arquitectura Aplicación

1.10 Tecnologías utilizadas

En el apartado 1.8, hemos visto las distintas alternativas existentes para los elementos más importantes o a tener en cuenta de cara a realizar nuestra interfaz. No todas se pueden utilizar, por lo que se han contrastado y se ha elegido la mejor opción con el fin de que esta UI sea lo más intuitiva, accesible y personalizada posible. Además, al ser un trabajo fin de grado, no es una aplicación demasiado grande por lo que es otro aspecto a tener en cuenta.

Para la parte de **frontend**, la tecnología elegida ha sido **React.js**. Esta, ofrece la facilidad de poder organizar el código y poder mantenerlo a largo plazo. Esto es muy útil para TextFlow debido a que si se desarrollan nuevas partes o funcionalidades o se quieren actualizar ciertas partes no tenemos que actualizar o modificar todo el sistema. Además, esta tiene una comunidad muy grande y activa, lo que hace que si ocurre algún problema en la implementación, se puede solucionar haciendo uso de esta, lo que ha hecho que esta tecnología sea utilizada en lugar de Bootstrap la cual está quedando obsoleta y puede no ser una comunidad tan activa. Al haber trabajado un poco con esta tecnología antes, no resulta muy difícil su aprendizaje. No se han elegido otros frameworks como Angular, puesto que este está diseñado para la realización de aplicaciones grandes y complejas, mientras que React.js es mucho más fácil de usar en aplicaciones simples y al ser un proyecto final de grado es innecesaria tanta complejidad.

Aunque no se encuentra, como archivo aparte cabe destacar que para el aspecto y la estructura de la interfaz sea ha utilizado CSS. Este, se ha utilizado debido a su facilidad para realizar diseños y porque tiene una gran capacidad de personalización de cara al desarrollador. Este ha sido integrado **en línea** con la utilización de la palabra “style”, de la siguiente forma tal y como se muestra en la ilustración siguiente:

```
<div style={{ color: 'blue', fontSize: '20px' }}>  
  Hello, World!  
</div>
```

Ilustración 10 – CSS en línea

En el lado de **backend**, el framework elegido ha sido **Flask**. Este, es un framework muy flexible, algo que ha ayudado mucho en la elección debido a que es ideal para un proyecto de estas características, el cual no requiere una estructura interna tan compleja. Otro de los aspectos que han sido muy importantes en la elección ha sido el lenguaje con el que estaba escrito el proyecto de TextFlow. Este, es Python debido a que en aspectos de Procesamiento del Lenguaje Natural o análisis de datos es muy utilizado. Al ser Flask un microframework basado en Python, es mucho más fácil su integración y ofrece la libertad de poder elegir las herramientas o librerías necesarias únicamente. No se ha elegido otro framework como Django, ya que este es mucho más complejo puesto que contiene muchas funcionalidades configuradas como temas de seguridad que ya se han comentado, útiles para grandes interfaces, lo cual no es el caso en este trabajo. Tampoco, se ha utilizado Laravel, ya que nunca lo he utilizado, lo que llevaría un mayor coste de aprendizaje. Además, requiere una gran sobrecarga, algo que no es muy práctico en este proyecto.

Por otro lado, para la parte de la **base de datos** se ha utilizado el gestor **MySQL**. Aunque no sea un trabajo grande y complejo, sí que contiene bastantes tablas con muchos datos a almacenar por lo que necesitaba un gestor que tuviera un buen rendimiento y una gran capacidad para almacenar muchos datos, algo que este tiene como característica. Esto ha sido algo muy importante a la hora de elegir MySQL en lugar de SQLite, la cual tiene menos capacidad de extensión ya que no admite grandes volúmenes de datos. Al utilizar Xampp, este gestor ofrece una interfaz gráfica, lo que facilita mucho, puesto que no hay que realizar consultas en líneas de código lo que hace que sea muy simple de usar. Además, este tiene una gran integración con Flask por lo que así haremos la parte del backend de manera más asequible sin necesidad de integrarlo de manera compleja. No se han usado gestores como MongoDB, ya que necesitaba que los datos estuvieran bien estructurados, algo que está más relacionado con una base de datos SQL que NoSQL.

Finalmente, cabe destacar que se ha utilizado **Visual Studio Code** para la utilización de muchos ficheros, una gran característica que este editor tiene. Además, se ha utilizado **Marvelapp**, para realizar el diseño de la interfaz debido a su facilidad y la gran cantidad de funcionalidades que ofrece.

1.11 Metodología de desarrollo de software

A la hora de realizar una interfaz web, como en muchas otras aplicaciones, elegir una buena metodología es una tarea muy importante para hacer un muy buen proyecto. Con esto, se va a definir una “guía” o seguimiento de trabajo, desde el inicio del trabajo, hasta la entrega final del mismo.

En mi caso, he seguido una metodología **SCRUM**. Esta, es una de las más elegidas y utilizadas en el desarrollo de un software. Tiene un enfoque incremental e iterativo, permitiendo una entrega continua del proceso. Con esta metodología, se busca aprovechar la flexibilidad, transparencia y la mejora continua que esta ofrece, mejorando o solucionando los problemas que se van encontrando.

SCRUM se basa en ciclos de trabajo donde se va realizando una entrega continua, pudiendo ajustar el desarrollo. Este, facilita la transparencia, ya que el equipo revisa después de estos ciclos, el trabajo realizado y planifica lo siguiente a conseguir. Al final de cada uno de estos, se lleva a cabo una valoración de lo completado hasta el momento, para mejorar ciertos aspectos o marcar el objetivo para el siguiente ciclo.

En este enfoque, existen varios roles definidos (en mi caso 2), **el cliente** o **usuario final** que elige y define los requisitos mínimos que tiene que tener el proyecto al ser finalizado, y el **desarrollador** que se encarga de realizar la implementación de las funcionalidades que el cliente ha sugerido. Con esto, se consigue que el proyecto esté siempre en la misma línea con los objetivos y lo que quiere el cliente.

1.12 Estimación del tamaño y esfuerzo

Ya que el presente proyecto es un TFG, no existen restricciones de tipo económico, sino de tipo temporal (en este caso 300 horas). Por consiguiente, los cálculos de tamaño del proyecto están supeditados al tiempo disponible.

En cuanto al esfuerzo, se dispone de tan un solo efectivo (la persona autora del trabajo).

1.13 Planificación temporal

A la hora de realizar un proyecto, es muy importante tener en cuenta la planificación del tiempo, para así poder garantizar que el trabajo se realice de manera ordenada y dentro del plazo establecido. Al ser un TFG (asignatura de 12 créditos), se le dedicará un total de 300 horas.

En este, se ha trabajado con una metodología SCRUM, por lo que se han ido realizando reuniones viendo el desarrollo del proyecto definido. El trabajo fue adjudicado el día 23 de marzo. El diagrama de Gantt se muestra a continuación:

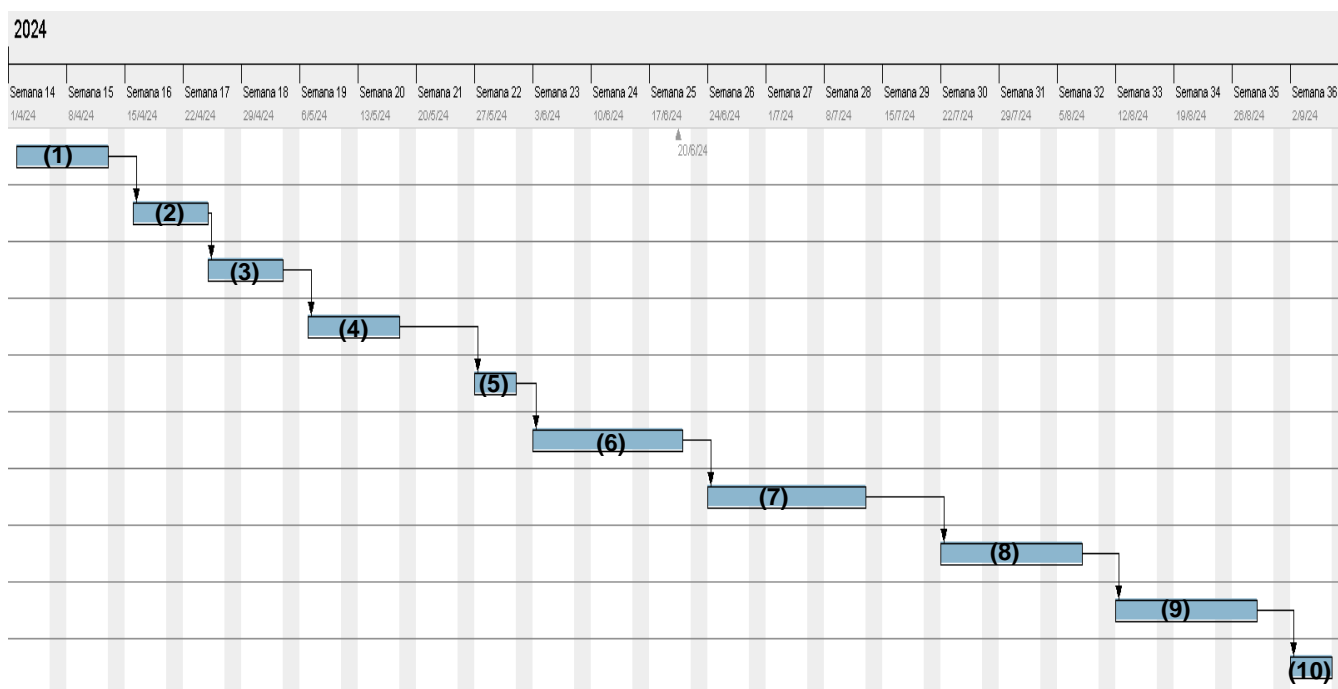


Ilustración 11 – Diagrama de Gantt

La idea inicial era entregarlo para el día 3 de julio, pero debido a una serie de contratiempos no fue posible. Es por esto, que la planificación siguiente mostrará el final para la entrega del 11 de septiembre. Esta división del tiempo, está dividida en tareas que se explicarán a continuación.

Tarea	Fecha Inicio	Fecha Fin
(1) Realización Sketch Inicial	02/04/2024	12/04/2024
(2) Entender Documentación TextFlow	16/04/2024	24/04/2024
(3) Inicio Memoria + Modificación Sketch	25/04/2024	02/05/2024
(4) Implementación (Inicio Sesión, Registro, Añadir Corpus)	07/05/2024	17/05/2024
(5) Implementación (Botones Ver, Analizar, Añadir)	27/05/2024	01/06/2024
(6) Implementación (Lectura Ficheros + Análisis Descriptivo)	02/06/2024	20/06/2024
(7) Corrección Análisis Descriptivo + Detalles de Inicio	24/06/2024	12/07/2024
(8) Implementación (Análisis Contrastivo)	22/07/2024	07/08/2024
(9) Documentación	12/08/2024	28/08/2024
(10) Últimas correcciones	02/09/2024	06/09/2024

Tabla 1 – Planificación Temporal

1.14 Presupuesto

La mayoría de los proyectos informáticos se desarrollan para cubrir una necesidad de alguno de los clientes a corto plazo. Es por eso, que se necesita personal cualificado que lo realice, además de un hardware y un software. Estos elementos tienen un precio y el cual hay que proporcionar en la vida real. El desarrollo de este trabajo de fin de grado no implica demasiado coste, más allá de un ordenador o temas de luz y electricidad.

Sin embargo en el mundo laboral, estos trabajos requieren de una serie de costos, tanto directos como indirectos, que deben ser considerados y justificados. Aunque el desarrollo de software y la investigación no siempre demandan una inversión económica considerable, existen costos asociados a los recursos humanos, el uso de software y hardware, y otros gastos relacionados con el mantenimiento y operación de la infraestructura necesaria para completar el proyecto. A continuación, se simulará lo que costaría este Trabajo Fin de Grado. Cabe destacar, que las personas implicadas **solamente pondrán su conocimiento**, lo demás será proporcionado por el jefe del proyecto.

En primer lugar, se hablará sobre los **costes de personal**. Para este aspecto, vamos a contar con que tendremos un desarrollador de frontend, un desarrollador de backend y un gestor de proyectos. Cabe destacar que para todos vamos a tener en cuenta el tiempo de inversión. En primer lugar para el **desarrollador frontend**, vamos a tener en cuenta que de media cobra 38.750€, lo que equivale a unos 20€ por hora aproximadamente. Para el **desarrollador backend**, tendremos en cuenta que su salario medio es de 36.000€, algo igual a 20€ por hora, otra vez aproximado. Finalmente, con el **gestor de proyectos**, tenemos que el sueldo medio es de 39.000€, que es otros 20€ la hora. Este último será encargado del diseño y de la realización de la documentación. Además, voy a tomar que disponemos de 300 horas para la finalización del proyecto, por lo que para el diseño se utilizarán 50 horas, para la parte de implementación unas 175 horas y para la parte de documentación unas 75 horas. Dentro de la parte de implementación vamos a suponer, unas 100 horas para la parte de backend, debido a que es la parte más a desarrollar con TextFlow y para el frontend unas 75 horas.

Rol	Desarrollador Frontend	Desarrollador Backend	Gestor de Proyectos	Total
Horas	75h	100h	125h	300h
Precio/hora	20€	20€	20€	20€
Total	1500€	2000€	2500€	6000€

Tabla 2 – Coste Personal

A continuación, hablaré de los costes **software**. Para este apartado, se van a suponer versiones mejoradas de ciertas aplicaciones y herramientas para que el proyecto se haga de manera completa. Para eso, se utilizará la Licencia de Windows 10 (pago único), la gestión de base de datos de MySQL en la nube (pago mensual), Github mejorado para subir y gestionar el proyecto (pago mensual), más todas las librerías que son gratuitas. Además, para Github y Windows 10 tenemos en cuenta que es para las 3 personas que intervienen en el proyecto, mientras que para MySQL solo hace falta una suscripción.

Herramientas	Windows Pro	MySQL en la nube	Github	Librerías y VS Code	Total
Precio	150€	50€	3€	0€	203€
Personas	3 personas	1 persona	3 personas	3 personas	3 personas
Total precio personas	450€	50€	9€	0€	509€
Tiempo de uso	4 meses	4 meses	4 meses	4 meses	4 meses
Total	450€	50€	36€	0€	536€

Tabla 3 – Coste Software

Para el aspecto **hardware**, para el aspecto hardware vamos a tener en cuenta que se trabaja para cada uno de las personas que participan en el proyecto con el mismo ordenador. Este es el Dell XPS 1 Intel Core i7 con 16 GB de RAM y 512 MB de Almacenamiento, valorado en 1572 €. Además, utilizaremos un ratón de 35€, un monitor de 27” de 400€ y un teclado por 60€. Como solo se va a usar 4 meses, lo que se hace es prorratear el costo a estos meses. Para esto, dividimos el coste de cada uno entre 12 meses que tiene el año y posteriormente multiplicamos por 4. **El coste total será el precio por mes x las personas x los meses que se utiliza.**

Herramientas	Ordenador	Ratón	Monitor	Teclado	Total
Precio	1572€	35€	400€	60€	2067€
Precio/mes	131€	2.92€	33.33€	5€	172.25
Personas	3 personas	3 personas	3 personas	3 personas	3 personas
Meses	4 meses	4 meses	4 meses	4 meses	4 meses
Total	1572€	35€	400€	60€	2067€

Tabla 4 – Coste Hardware

Finalmente, existen otros aspectos a tener en cuenta en el presupuesto. Estos son los **costes indirectos** y la **publicación de la aplicación en Google Cloud**. En cuanto a los primeros vamos a tener en cuenta solo y exclusivamente la electricidad (la cual contaremos que el precio por mes aproximado es de 3 euros, para la utilización del ordenador) y la conexión a Internet (que contaremos lo que vale la fibra para cada mes que son 30 euros) algo que será un pago único para las 3 personas. Para el segundo, cabe destacar que solo se tendrá en cuenta el coste de publicación y el alojamiento de la base de datos (pago único), y no costes de soporte y mantenimiento.

	Electricidad	Conexión a Internet	Publicación	Total
Precio	3€	30€	15€	48€
Meses	4 meses	4 meses	4 meses	4 meses
Total	12€	120€	15€	147€

Tabla 5 – Otros costes

De cara al presupuesto total, vamos a realizar la suma de todos los costes totales anteriores.

Costes	Total
Costes Humanos	6000€
Costes Software	536€
Costes Hardware	2067€
Otros Costes	147€
Total coste	<u>8750€</u>

Tabla 6 – Coste Total

Como hemos visto, el presupuesto total es de **8.750 euros** para este proyecto. Este, supone realizar una inversión significativa en desarrollo, infraestructura y servicios. Esto es así, debido a que hemos contratado al personal comentado, pero estos solo aportaran el conocimiento y todo lo demás irá por parte del jefe. Los costos principales se distribuyen entre los salarios del equipo de desarrollo, la infraestructura en Google Cloud, y el hardware necesario, lo que garantiza que el proyecto cuente con los recursos humanos y tecnológicos necesarios para su buen funcionamiento. Podemos destacar que una planificación económica bien estructurada para el desarrollo y mantenimiento de la aplicación es muy importante.

2 DISEÑO INICIAL

Este trabajo se realiza con el objetivo de diseñar y desarrollar una Interfaz Web que mejore la de Google Cloud dividida en celdas de código, como ya se ha comentado anteriormente. Para realizar esto, en primer lugar, hemos definido los requisitos iniciales y mínimos que debe tener esta aplicación. A continuación hay que mostrar los casos de uso o los prototipos o wireframe de cómo quedaría la aplicación. En este apartado, se recogen todos estos aspectos clave en el diseño del sistema y que facilitarán la tarea de cara al desarrollo del proyecto.

2.1 Especificaciones del sistema

En el apartado 1.5 se han mencionado los requisitos funcionales y no funcionales de manera resumida. En este apartado, se van a comentar estos requisitos de manera más completa, siendo acompañados por los respectivos diagramas de caso de uso. Es importante destacar para estos últimos que solo hay un actor principal que es el usuario que va a interactuar con nuestro sistema. Los requisitos funcionales principales que nos encontramos son los siguientes:

- **Inicio de Sesión:** el usuario debe ser capaz de escribir su usuario y su contraseña para poder autenticarse y acceder a la página de inicio. También, tiene que tener la posibilidad de poder cerrar sesión.

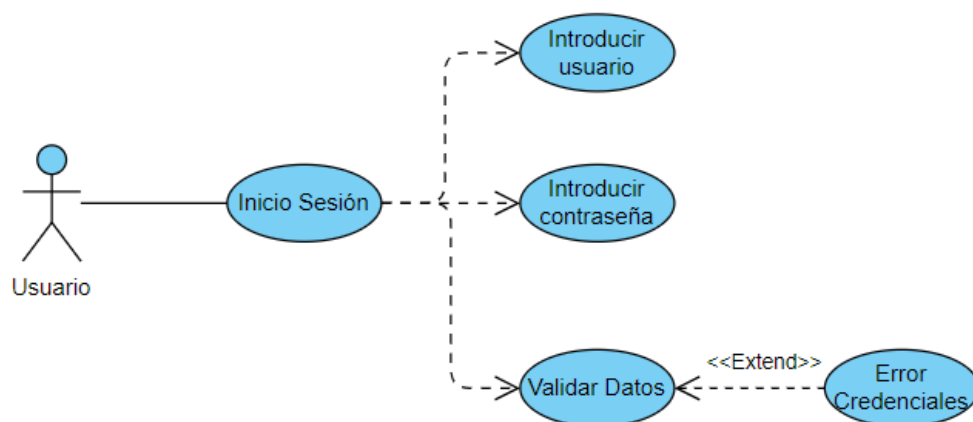


Ilustración 12 – Caso de Uso. Iniciar Sesión

- **Registrarse:** el usuario debe ser capaz de enviar sus credenciales en el sistema para su futuro inicio de sesión. Este debe introducir el nombre, correo, contraseña e usuario.

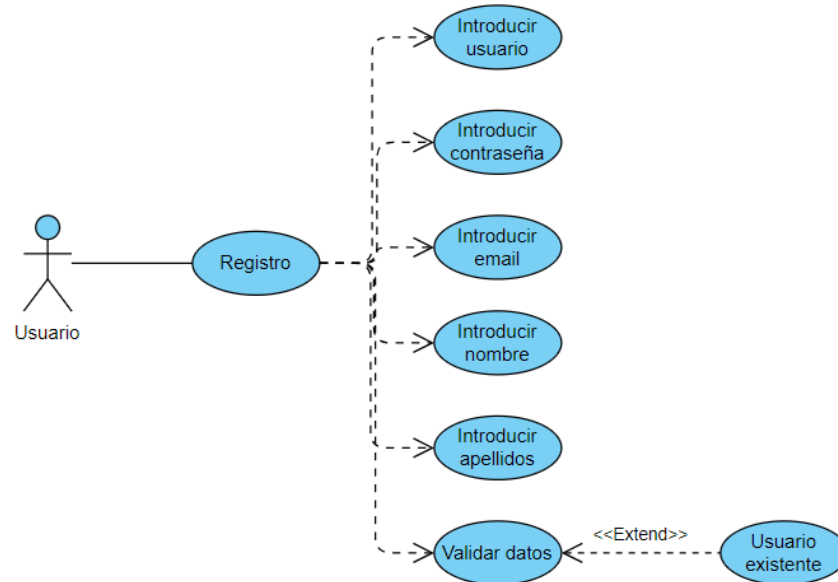


Ilustración 13 – Caso de Uso. Registro

- **Gestionar sus corpus:** el usuario autenticado tiene que poder añadir nuevos corpus, ver el contenido de sus corpus, eliminar los mismos y analizarlos.

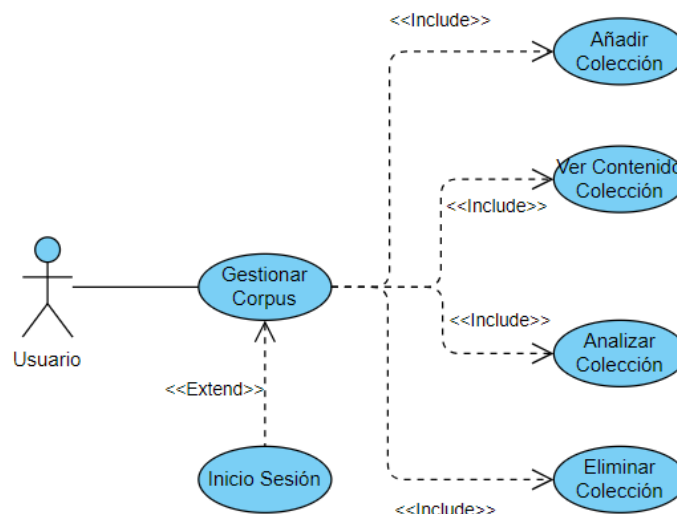


Ilustración 14 – Caso de Uso. Gestionar Corpus

- **Añadir corpus:** este usuario debe tener la capacidad de elegir un nombre para el corpus, que debe ser el mismo del fichero que se suba, y de elegir la carpeta comprimida que se quiere subir para su posterior visualización en la página principal.

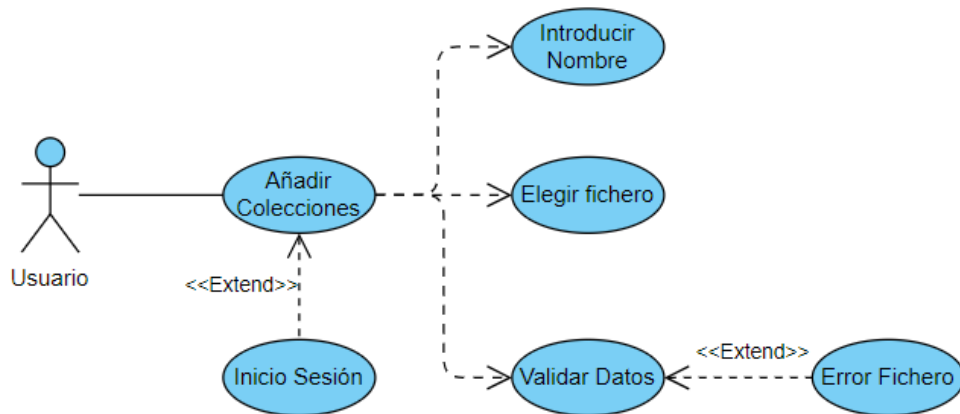


Ilustración 15 – Caso de Uso. Añadir Colección

- **Análisis Descriptivo:** el usuario debe de ser capaz de analizar descriptivamente su corpus. Para eso debe poder elegir los metadatos que quiera visualizar y eliminar los análisis de estos metadatos no deseados.

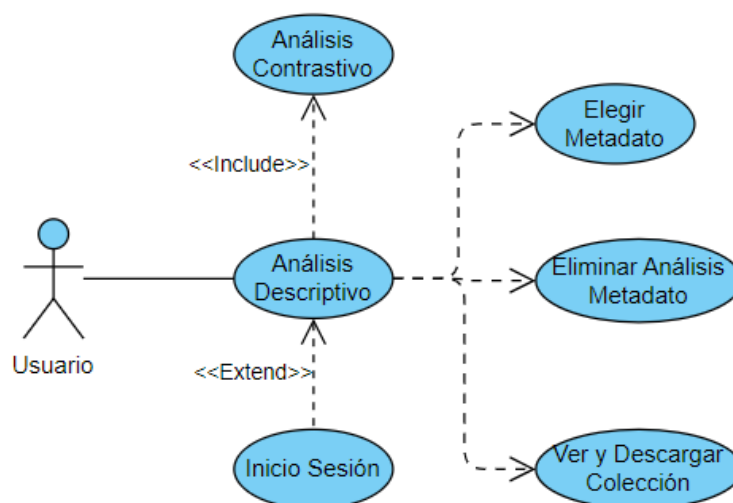


Ilustración 16 – Caso de Uso. Análisis Descriptivo

- **Descargar Análisis:** Tras la visualización de estos metadatos los cuales deben estar ordenados en formatos tabulares, el usuario debe tener la capacidad de descargar sus respectivos análisis, para poder seguir trabajando de forma independiente a la aplicación.

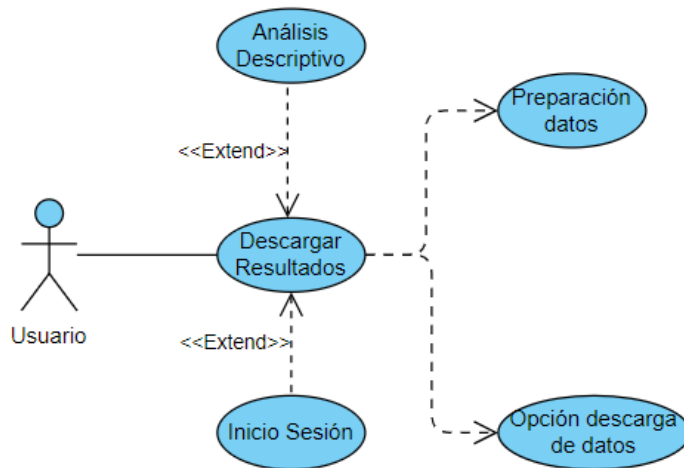


Ilustración 17 – Caso de Uso. Descargar Análisis

- **Análisis Contrastivo:** El usuario debe tener la capacidad de comparar dos corpus distintos que ya tenga analizados descriptivamente previamente para poder encontrar diferencias entre ambos. Para eso, este debe poder elegir el corpus con el que quiere contrastar el activo y el metadato que se va a querer comparar.

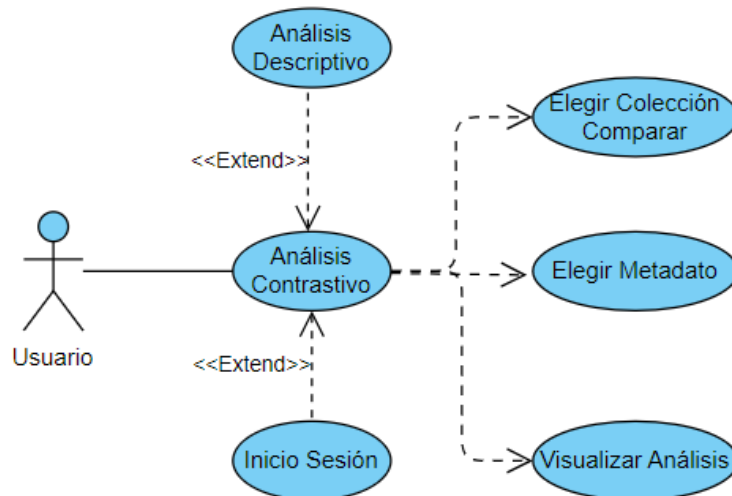


Ilustración 18 – Caso de Uso. Análisis contrastivo

A continuación, se mostrarán de forma más detallada los requisitos no funcionales de mi aplicación web, que suele coincidir en su mayoría con el del resto de interfaces web.

- **Interfaz de usuario intuitiva:** La aplicación debe ser fácil de usar. Debería permitir a los usuarios interactuar con ella de manera sencilla sin necesidad de tener un conocimiento avanzado en ella.
- **Privacidad de los datos:** La interfaz debe proporcionar la privacidad en la contraseña o en los análisis realizados. Además, los datos de usuario deberán estar protegidos a accesos no autorizados.
- **Tolerancia a fallos:** Debe ser capaz de manejar los errores, recuperándose, sin la pérdida de este.
- **Escalabilidad:** Debe ser posible añadir nuevas funcionalidades al sistema, sin necesidad de realizar grandes esfuerzos o tener que modificar gran parte de la interfaz.

2.2 Análisis y diseño del sistema

A la hora de diseñar un proyecto hay que empezar desde 0, por lo que cada decisión marcará el futuro de nuestro trabajo. Para eso, debemos realizar un diseño mirando al futuro y que esté bien planificado. En este apartado va a presentar una serie de diagramas que representarán un análisis detallado y un diseño estructurado de la interfaz web que se va a realizar. El objetivo de este se mostrará una visión completa y ordenada del sistema propuesto desde el principio. Se inicia con el análisis del sistema, que incluye la evaluación de los requisitos y la identificación de mejoras necesarias para satisfacer las necesidades del usuario. En este se presentará el diseño de la base de datos, lo cual facilitará la gestión y el mantenimiento de los datos. A continuación se definirán los diagramas de secuencia para indicar como se comunican los distintos elementos que participan en el sistema. Finalmente, se mostrará el diseño inicial de cómo queremos que sea la interfaz, iniciando desde el diseño en papel hasta un wireframe o prototipo de este.

2.2.1 Diseño Base De Datos

En esta subapartado se presenta el diseño de la base de datos utilizada para la interfaz web. Este diseño es un aspecto muy importante para garantizar un almacenamiento eficiente, además de conseguir la integridad y la accesibilidad de los datos necesarios para el funcionamiento de la aplicación. En este, se muestran las estructuras de las tablas, las relaciones entre ellas, y las restricciones aplicadas para asegurar la consistencia de la información. Además, se explican las decisiones tomadas durante el diseño, basadas en los requisitos del sistema y en la necesidad de optimizar el rendimiento y la escalabilidad de la aplicación. Cabe destacar que en algunas entidades no se muestran todos sus atributos puesto que no entrarían bien en la fotografía, simplemente se muestran algunos de estos en este caso.

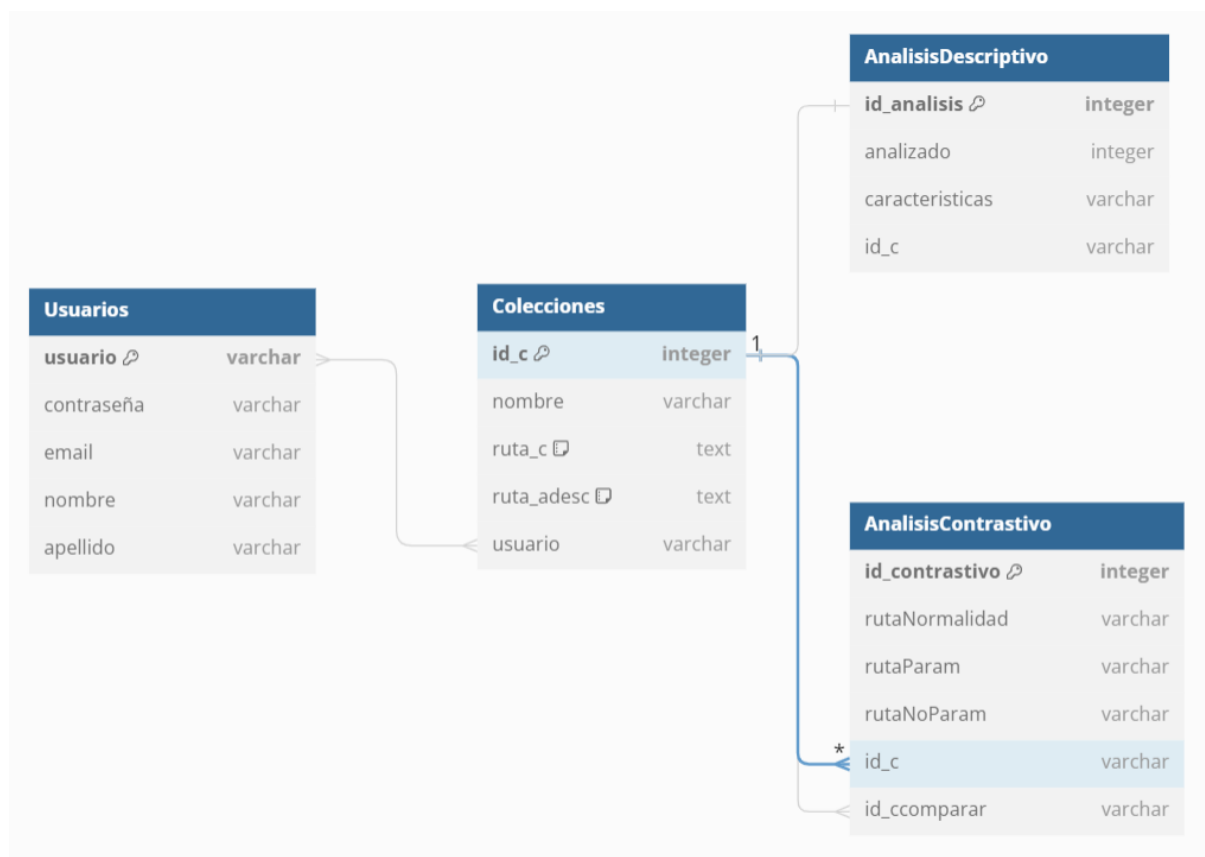


Ilustración 19 – Diseño Base de Datos

Nota: Como vemos en la relación marcada, la flecha representa n y la rayita representa 1.

La base de datos estará formada por 24 entidades que se relacionarán entre ellas. Las entidades que encontramos son las siguientes:

- **Usuarios:** representan a las personas que se registran en la App con sus datos personales.
- **Colecciones:** se corresponde con las colecciones que se añaden desde la interfaz. Están asociadas a un usuario concreto, además uno de sus atributos a destacar es la ruta del análisis contrastivo.
- **Análisis Descriptivo:** se mostrarán las entidades de cada uno de los análisis de los metadatos existentes. Los atributos de cada uno de estos son las características que se pueden analizar. Además, están asociados a la colección que se compara y tiene un atributo para saber si se quiere mostrar el contenido o no. Cabe destacar que en el diseño se ha creado una única tabla denominada así, pero que representa a cada una de las entidades que luego habrá en la implementación, ya que todas tienen la misma relación y solo cambian los atributos. Las entidades son las siguientes:
 - **Bdlase**
 - **Complejidad**
 - **Diversidad léxica**
 - **Emociones**
 - **Emofinder**
 - **Emojis**
 - **Estilometría**
 - **Ironía**
 - **ISAL**
 - **Lemas**
 - **LIWC**
 - **NCR**

- **NER**
- **NGrams**
- **Perplejidad**
- **Polaridad**
- **Polaridad Emojis**
- **Polaridad Emoticonos**
- **POS**
- **SEL**
- **Volumetría**
- **AnálisisContrastivo:** esta entidad representa al análisis contrastivo realizado sobre la colección que lleva asociada y la colección con la que se compara. Además, presenta como atributo las rutas de los tres tipos de test que se pueden realizar.

La relación existente entre usuarios y colecciones es de **n a m**, esto es, una colección puede estar en más de un usuario y un usuario puede tener más de una colección. En cuanto a la relación entre AnálisisContrastivo y colecciones, esta tiene dos relaciones. Estas son una relación **n a 1**. Ya que una colección puede tener varios análisis de este tipo, mientras que el mismo análisis solo puede estar en uno. Por último, tenemos las relaciones entre los distintos metadatos del análisis descriptivo. La relación de estos es de **1 a 1** ya que una colección suele puede tener un análisis de bdlase por ejemplo, y el mismo análisis solo puede estar en una colección. El nombre de las relaciones son los siguientes:

- Relación entre Usuario y colecciones: **Gestiona**
- Relación entre AnálisisContrastivo y colecciones: **Tiene**
- Relación entre Metadatos análisis descriptivo y colecciones: **Realiza**

2.2.2 Diagramas de secuencia

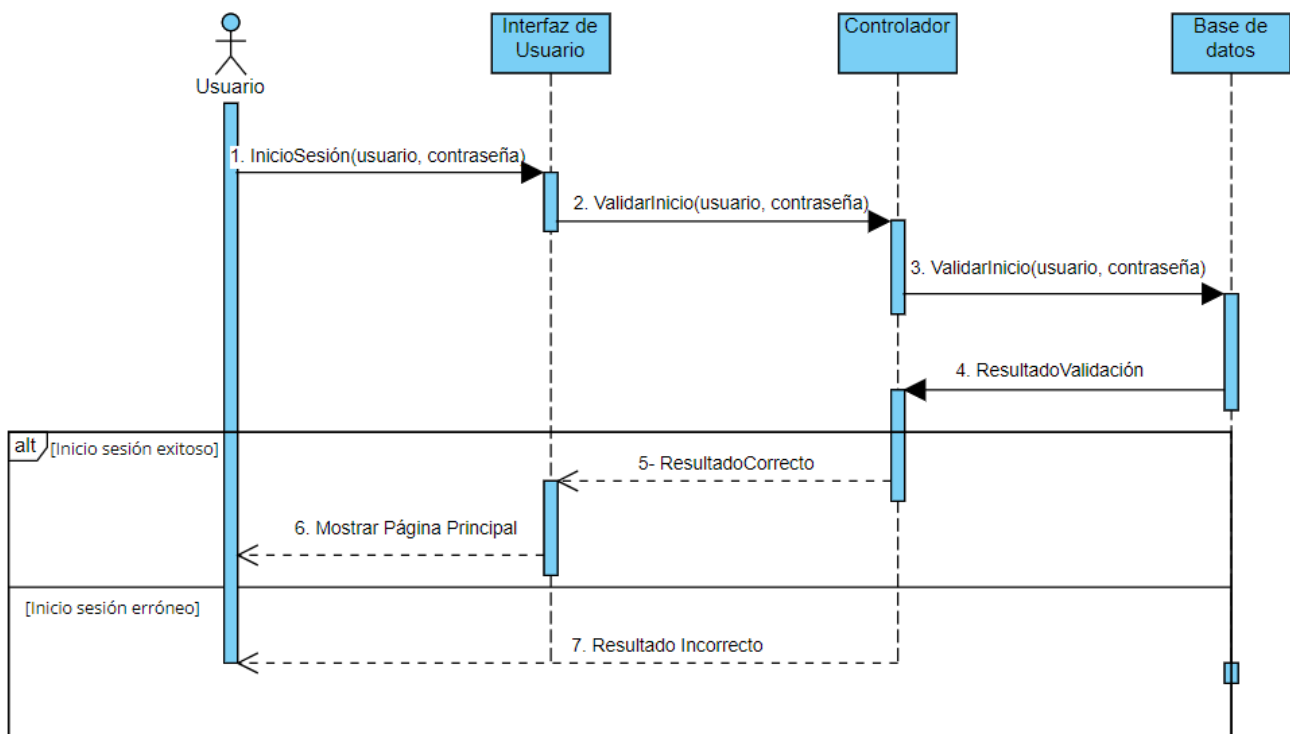


Ilustración 20 – Diagrama de secuencia. Inicio Sesión

En el anterior diagrama se muestra el funcionamiento de la página a la hora de iniciar sesión.

En primer lugar el usuario introduce su usuario y contraseña en la página web. Tras esto, la página solicita comprobar si los datos introducidos son correctos. Para esto, se manda una petición a la base de datos con el usuario y la contraseña. A partir de aquí, existen dos alternativas. Si el usuario y la contraseña existen, se devolverá la página principal al usuario. En caso contrario se le mostrará un mensaje de que o el usuario o la contraseña no son correctos.

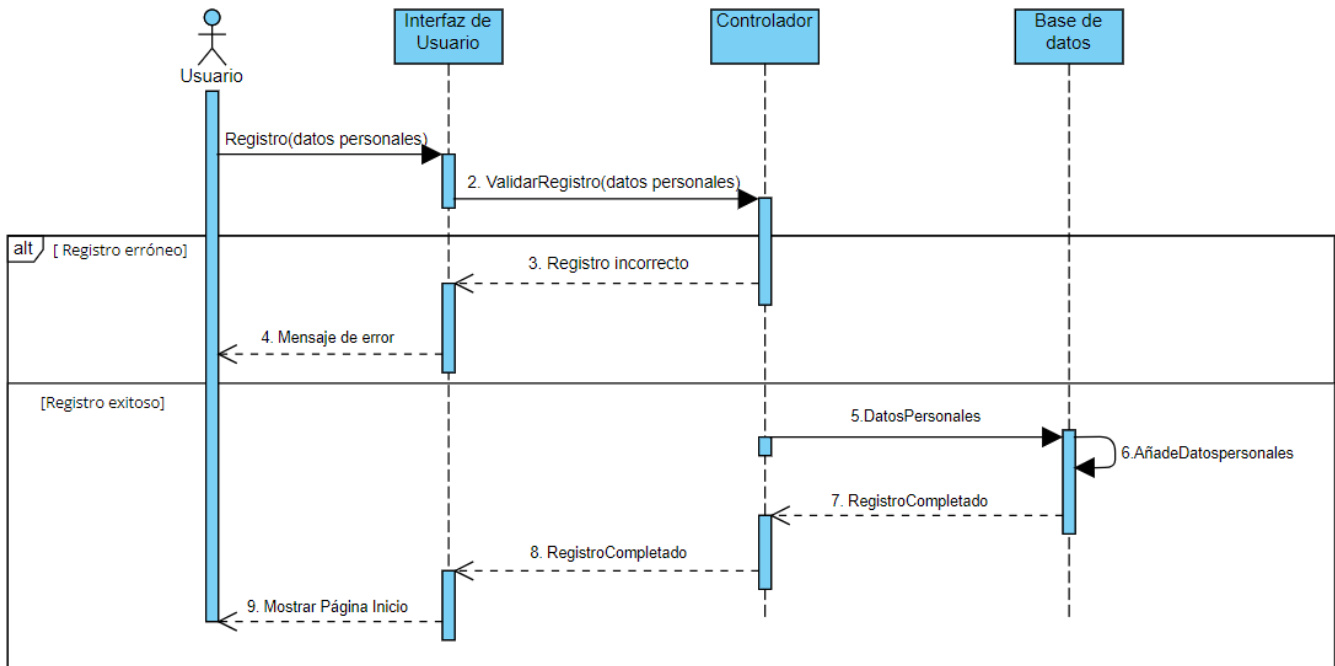


Ilustración 21 – Diagrama de secuencia. Registro

En el diagrama anterior se muestra el funcionamiento de la página a la hora de registrar a un usuario que interactúa por primera vez con esta.

En primer lugar el usuario introduce su usuario, contraseña, email, nombre y apellido en la página web. Tras esto, la página solicita comprobar si los datos introducidos son correctos. Para esto, en primer lugar, comprueba que todo esté en un formato correcto. Si esto no es así, muestra un mensaje de error al usuario. En caso de si ser correcto todo, se manda una petición a la base de datos con estos datos personales y se guardan en ella. A partir de aquí, la página web le muestra al usuario la página principal, en el caso de que todo haya ido bien.

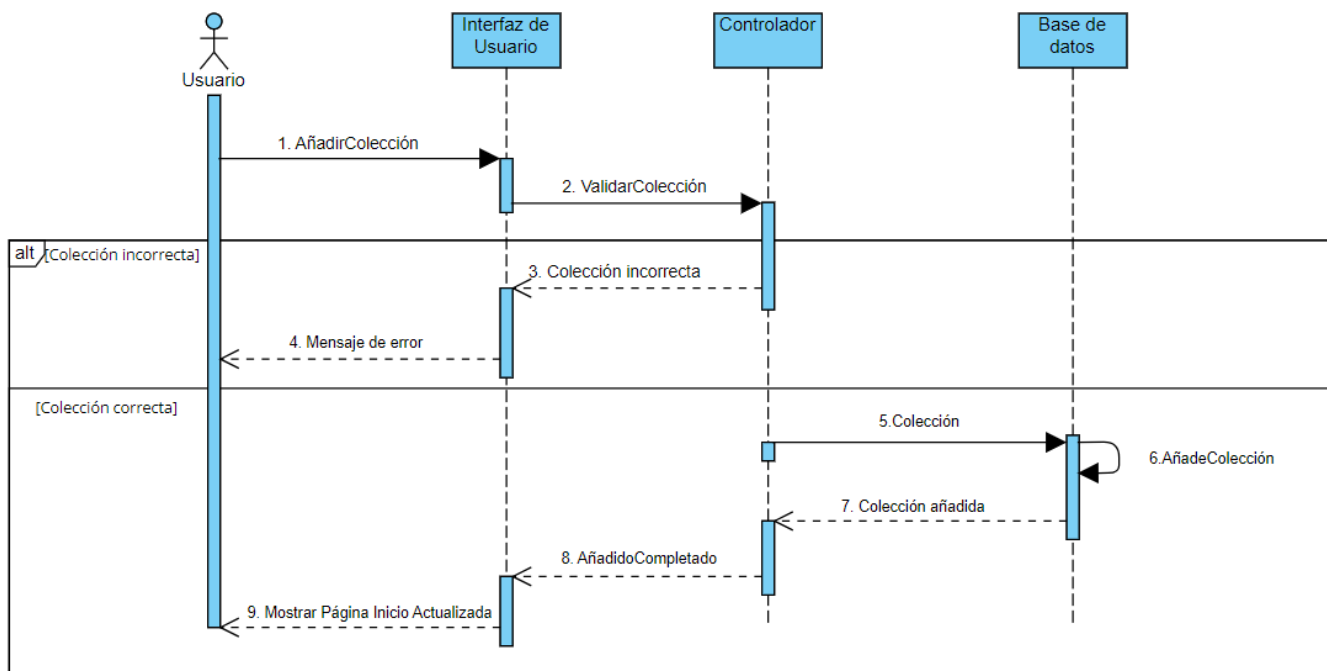


Ilustración 22 – Diagrama de secuencia. Añadir Colección

En el diagrama anterior se muestra el funcionamiento de la página a la hora de un usuario querer añadir una colección. También sirve para eliminar una colección cambiando una serie de aspectos que a continuación se comentarán

En primer lugar el usuario añade la colección que desee incluyendo una serie de aspectos. Tras esto, la página solicita comprobar si los datos introducidos son correctos. Para esto, en primer lugar, comprueba que todo esté en un formato correcto y que el nombre sea válido. Si esto no es así, muestra un mensaje de error al usuario. En caso de si ser correcto todo, se manda una petición a la base de datos con estos datos y se guardan en ella. A partir de aquí, la página web le muestra al usuario la página principal ya actualizada con la colección en ella, en el caso de que todo haya ido bien.

Esto también puede servir para eliminar colecciones, cambiando la palabra añadir por eliminar, puesto que el proceso es muy similar.

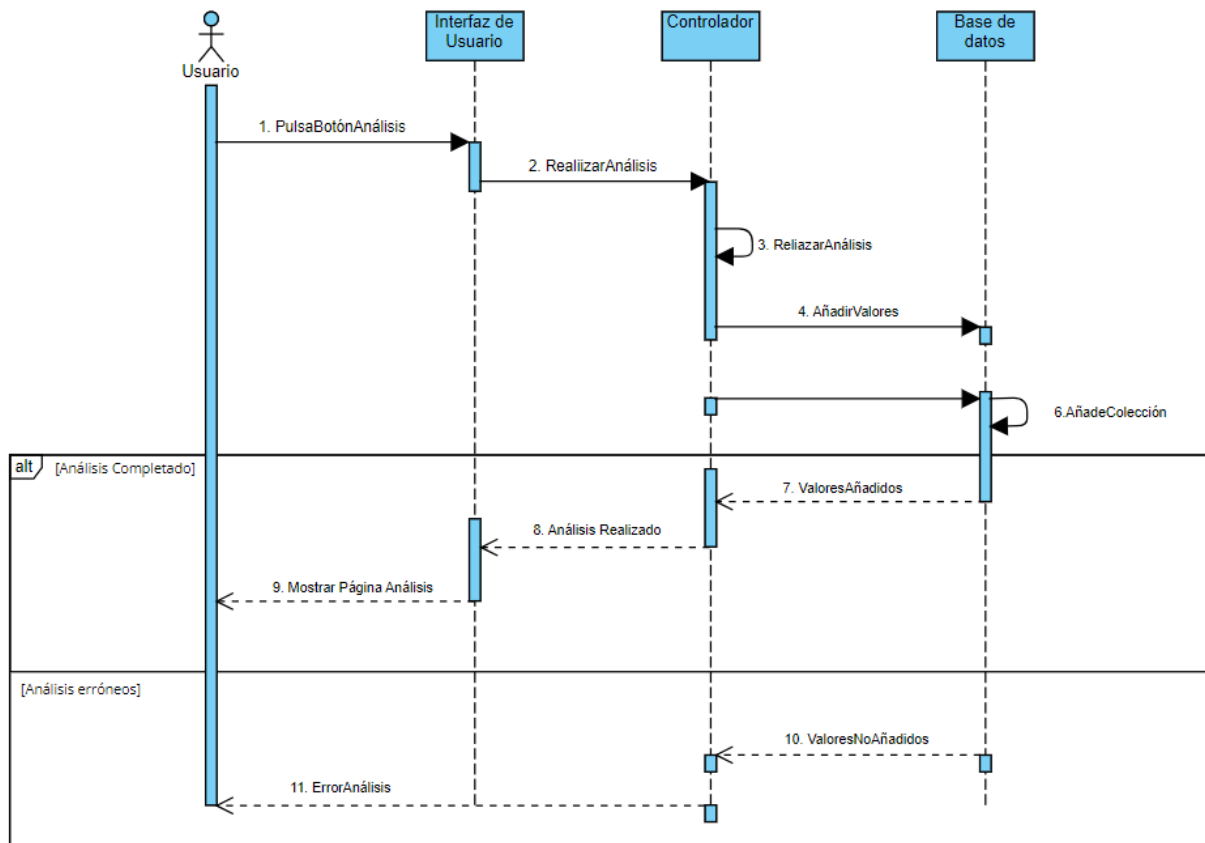


Ilustración 23 – Diagrama de secuencia. Análisis

En el diagrama anterior se muestra el funcionamiento de la página a la hora de realizar un análisis ya sea descriptivo como contrastivo de una determinada colección.

En primer lugar el usuario pulsa en el botón de analizar un determinado corpus. Tras esto, la página solicita realizar el análisis e interiormente se realiza este. Terminado este proceso, los valores obtenidos se añaden a la base de datos. Si ha ocurrido algún problema en este proceso de inserción de los datos, se mostrará un mensaje de que ha ocurrido un error en el proceso. En caso contrario, se mostrará la página para poder seleccionar y ver los determinados análisis.

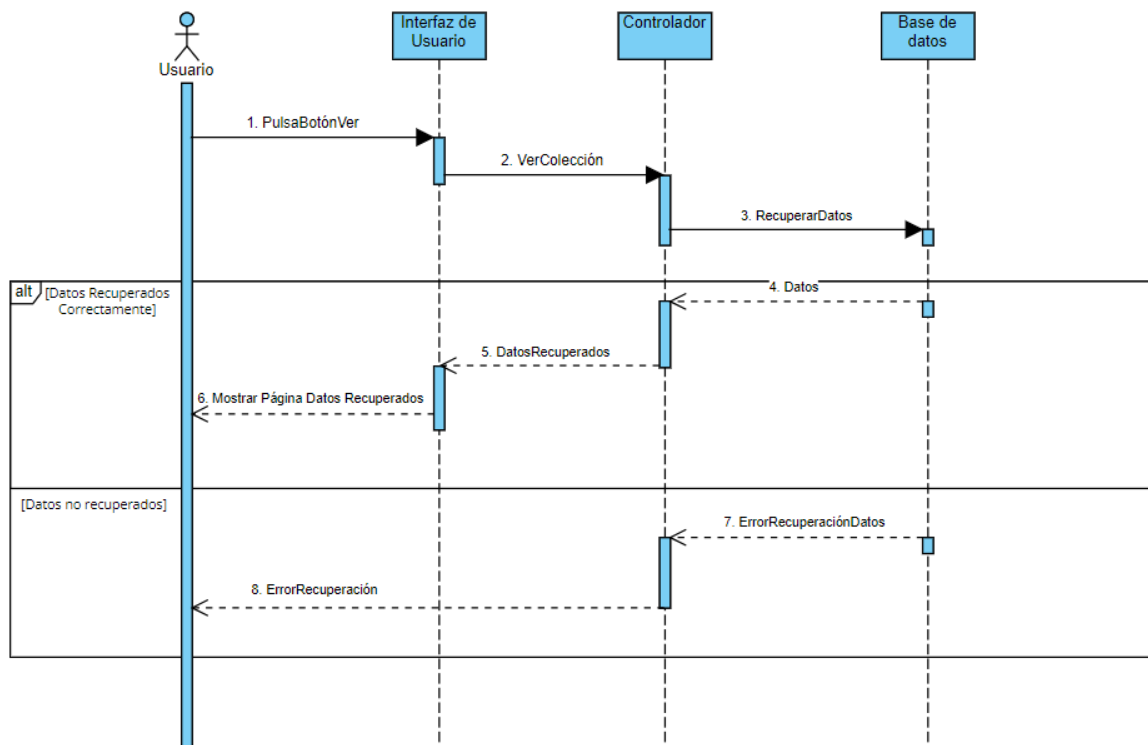


Ilustración 24 – Diagrama de secuencia. Ver Datos

Por último, se muestra el diagrama donde se puede visualizar el funcionamiento de la página a la hora de ver un determinado análisis ya sea contrastivo o descriptivo. Este diagrama también puede servir para descargar los datos localmente.

En primer lugar el usuario pulsa en el botón de ver el análisis de un determinado corpus. Tras esto, la página solicita ver los datos y para esto se realiza una solicitud a la base de datos. Si ha ocurrido algún problema en este proceso de recopilación de los datos se muestra un mensaje de error. En caso contrario, se mostrará la página para poder ver los datos del análisis elegido concretamente.

Esto también puede servir para descargar datos puesto que el proceso es similar cambiando el pulsador del botón de ver por el de descargar.

2.2.3 Diseño de la interfaz

En este subapartado se mostrará, en primer lugar, el **sketch inicial** realizado junto con el usuario final (en este caso el tutor de este TFG, Arturo Montejo) donde se muestra una visión general de lo que se quiere que contenga la página y como debe estar distribuida. De esta primera visualización se puede sacar como conclusión, que se quiere una página para iniciar sesión, otra para registrarse, una página principal donde mostrar los corpus y una serie de botones para poder ver el contenido, analizarlo o eliminar alguno de estos. Luego queremos tener dos aspectos, uno con el análisis descriptivo y otro con el análisis contrastivo donde podamos ver los valores de ambos.

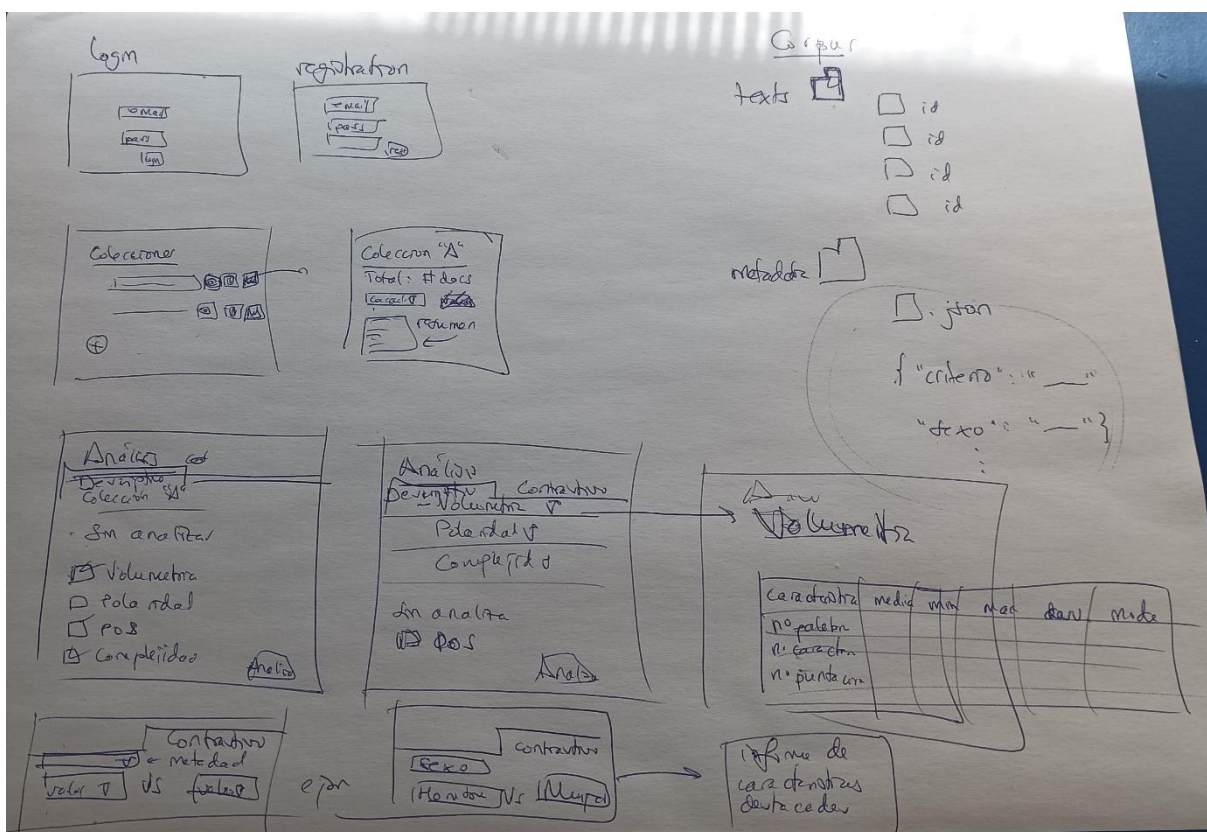


Ilustración 25 – Sketch Inicial

A continuación se mostrarán los distintos **Wireframes** de las diferentes pantallas con las correspondientes funcionalidades que estas deben tener. Cabe destacar que solo se mostrarán las pantallas más importantes y a tener en cuenta. En el siguiente enlace se pueden ver todas las pantallas y todo lo relacionado con el diseño de los wireframes: <https://marvelapp.com/prototype/31e2g2dg>. Además, a continuación, también se muestran algunas de las páginas:

Inicio de Sesión

The wireframe shows a blue header with the text 'Analiza Tus Corpus'. Below the header is the title 'Iniciar Sesión'. There are two input fields: 'Email' and 'Contraseña'. Below the 'Contraseña' field is a link 'Registrarse' with the text 'Si no tiene cuenta.' to its left. At the bottom center is a blue button labeled 'Iniciar sesión'. At the bottom right is a link 'Activar Win' with the text 'Ve a Configurac' below it.

Ilustración 26 – WireFrame InicioSesión

Entre las funcionalidades de la página anterior, encontramos que al clickar en *Iniciar sesión* se nos muestra la página de Inicio y al clickar en registrarse, nos manda a la página de registro que es la siguiente.

Registro

Analiza Tu Corpus

Registro

Nombre

Apellidos

Email

Contraseña

Registrarse

Ilustración 27 – WireFrame Registro

Lo importante a destacar en esta página es que a la hora de clicar en registrarse, se nos manda a la página de Inicio de sesión que es la siguiente.

Página Principal

Analiza Tus Corpus

Colecciones Guardadas

Colección "X"

Colección "X"

+

Ilustración 28 – WireFrame Página principal

En la página anterior podemos ver varias funcionalidades. La primera de ellas es al clicar en el ojo, la cual nos redirige a la página de ver el contenido de la colección. La del gráfico nos manda a la página del análisis descriptivo/contrastivo. Si pulsamos sobre la papelera, se nos elimina la colección y finalmente si clickamos sobre el + podremos añadir más colecciones. A continuación, se muestran algunas de estas.

Añadir Colección



Analiza Tus Corpus

Añadir Colección

Nombre de la Colección

Añade la colección que quieras subir.
Solo se aceptan archivos .zip



Añadir Colección

Ilustración 29 – WireFrame Añadir Colección

En esta página, se debe introducir el nombre de la colección (que deberá ser igual que el de la colección que se sube) y solo se pueden subir archivos .Zip. Además, al clicar en añadir colección se nos mostrará la página principal vista anteriormente con el corpus añadido.

Análisis Descriptivo

Analiza Tus Corpus

Análisis Descriptivo Análisis Contrastivo

Colección "X" < Volver a página principal

Volumetría	➤
Complejidad	➤

Sin analizar

POS

Polaridad

[Analizar](#)

Ilustración 30 – WireFrame Análisis Descriptivo

En esta página, se pueden observar varias funcionalidades. En primer lugar vemos dos partes una de metadatos analizados y otros sin analizar. Si clickamos sobre algunos de estos últimos y le damos a analizar, estos se añadirán a la parte de analizados. Una vez estos se encuentren aquí, podremos clickar sobre la flechita y ver el contenido de estos. También, vemos una opción de volver a la página principal que realiza lo que el propio nombre indica. Finalmente, podemos pulsar sobre análisis contrastivo y se nos da la opción de realizar el análisis contrastivo con otra colección.

Análisis Contrastivo

Analiza Tus Corpus

Análisis Descriptivo | **Análisis Contrastivo**

Inserte nombre del Analisis

< Volver a página principal

Metadato

Valor1 VS Valor2

Analizar

Ilustración 31 – WireFrame Análisis Contrastivo

Finalmente, en esta página se puede realizar el análisis contrastivo. Entre las funcionalidades presentes en esta, nos encontramos con que podemos elegir la característica a comparar pulsando en metadato. Además, podemos elegir el corpus con el que lo queremos comparar marcando sobre la opción de valor2. Al igual que en el análisis anterior, también podemos volver a la página principal como el propio nombre del botón indica. Tras clicar en analizar, podremos ver el análisis que hemos realizado. Finalmente, si clickamos sobre el botón de análisis descriptivo se podrá optar por realizar un análisis descriptivo (página explicada anteriormente).

3 DESARROLLO

En este apartado se va a describir el proceso de desarrollo de la interfaz web destinada a la gestión y análisis de corpus lingüísticos. Como ya se ha comentado anteriormente, se ha seguido la metodología ágil SCRUM, que permite un desarrollo iterativo e incremental, asegurando que cada fase del trabajo se centre en la implementación de funcionalidades nuevas, con ajustes y mejoras continuas de las funcionalidades anteriores.

Como se ha comentado también anteriormente, la interfaz, está realizada con tecnologías como Flask para la parte del backend, React.js en el frontend, y MySQL para la gestión de bases de datos. Con esto se quiere conseguir que esta se desarrolle con el objetivo de ofrecer una herramienta robusta y accesible. En este apartado se va a abordar cada iteración del proyecto. Al ser una metodología ágil esta va a estar compuesta por varias iteraciones. Estas son las siguientes:

- **Primera Iteración:** En esta iteración se realiza la implementación de funcionalidades básicas para el usuario como son el inicio de sesión y el registro.
- **Segunda Iteración:** En esta segunda fase, se realizará la página principal propiamente dicha, junto con las funcionalidades de añadir colecciones, ver la colección y eliminarla, junto con sus botones asociados además del de analizar.
- **Tercera Iteración:** Para esta iteración, se va a realizar la implementación del análisis descriptivo, añadiendo las funcionalidades de añadir análisis de un metadato y eliminarla, además de poder ver y descargar el análisis.
- **Cuarta Iteración:** Para esta última, se realizará la implementación del análisis contrastivo, añadiendo la funcionalidad de poder ver dichos análisis, además de cuestiones como elegir solo las colecciones analizadas para comparar y los metadatos analizados por ambos.

Como se explica, la planificación ágil también se refleja en cómo se ha abordado la implementación: desde el manejo de la autenticación de usuarios hasta la ejecución y presentación de los análisis descriptivos y contrastivos. En cada iteración, se han realizado ajustes y optimizaciones que han permitido evolucionar el proyecto, asegurando que se satisfagan las necesidades del usuario.

3.1 Primera Iteración

3.1.1 Planificación

Como ya se ha comentado anteriormente, en esta primera iteración se van a realizar las páginas de **inicio de sesión y registro**. La planificación o diseño inicial de estas páginas, se realizó en el apartado anterior en el subapartado del diseño de la interfaz. Estas secciones, serán las primeras que nos encontraremos al entrar en esta página, puesto que sin esto, no se podrá acceder a las distintas páginas que esta interfaz web contiene. Entre las funcionalidades que tienen que tener estas páginas nos encontramos con un botón de inicio de sesión, que al clickar nos devuelve la página principal si todo es correcto; un botón de registro que añadirá al usuario a la base de datos y un botón de registrarse en la parte del inicio de sesión que nos redirige a la página de Registro

3.1.2 Implementación

Como ya se ha comentado anteriormente, para la realización de la página web decidí utilizar React.js para frontend, Flask para backend y MySQL para la gestión de la base de datos. Por lo tanto, en primer lugar, antes de empezar había que entender los lenguajes de programación.

Para Flask no necesité mucho tiempo, puesto que es un lenguaje que ya tenía familiarizado de algunas asignaturas en la carrera. Con respecto a MySQL tampoco necesité demasiado tiempo puesto que también fue usado en una asignatura y simplemente hacía falta instalar Xampp para su ejecución. Además, la tarea de

relacionar la base de datos con la parte backend de nuestra interfaz tampoco es de demasiada dificultad porque únicamente hay que crear una instancia de la base de datos con “*mysql = MySQL(app)*” habiendo antes configurado la propia app (una instancia de flask con *app.config['MYSQL_HOST']*, *app.config['MYSQL_USER']* y *app.config['MYSQL_DB']*). Tras haber puesto estos datos correctamente, nuestro backend estará unido correctamente con la base de datos. Cabe destacar que anterior a estos pasos, debemos haber instalado una librería para poder unir flask con la base de datos. Esto se realiza con el comando *pip install flask-mysqldb*.

Sin embargo, para la parte de React.js, necesité un poco más de tiempo sobre todo para la parte de entender cómo comunicar la parte del backend con la parte del frontend. Para esto, tras haber realizado una serie de instalaciones previas, simplemente había que utilizar la ruta para cada función de la parte del backend. A continuación se muestra un ejemplo de lo comentado en este apartado. Hay que tener en cuenta también que para que la página web funcione debemos tener iniciada siempre la parte del backend y para eso hay que ejecutar *npm start*.

```
@app.route('/login', methods=['POST'])  
def login():
```

Ilustración 32 – Backend Rutas

La imagen anterior se corresponde con la parte del backend. En este caso, se puede comprobar como para Iniciar Sesión, hemos utilizado una ruta `/login` con la que podremos comunicarnos con la parte del frontend de la siguiente forma:

```
fetch('http://localhost:5000/login', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({
    usuario: usuario,
    contraseña: contraseña
  })
})
```

Ilustración 33 – Frontend Rutas

En esta imagen, vemos como con `fetch` y la ruta podremos acceder a la parte del backend y que estos estén comunicados.

Como hemos visto en la anterior ilustración, de esta manera comunicamos la parte de frontend con la de backend. Pero además se puede comprobar también como desde el frontend se mandan el usuario y la contraseña introducidos por el usuario a la parte de backend donde se realizará la comprobación de si estos datos se encuentran en la base de datos. Una cosa a tener en cuenta en esta parte, es que la contraseña no se almacena en la base de datos en claro, si no que se hace cifrada. Esto es así, debido a que si la base de datos sufre una irrupción por parte de un atacante, podemos evitar que se publiquen contraseñas de los usuarios que usen nuestra página web. Es por eso que a la hora de comprobar la contraseña en primer lugar tenemos que descifrarla y para eso se ha utilizado la siguiente función: `bcrypt.check_password_hash(hashed_password, contraseña.encode('utf-8'))` mediante la cual se descifra la contraseña y posteriormente se compara con la que el usuario ha introducido. Además, en esta parte de inicio de sesión, si esta contraseña no ha sido correcta, se devuelve una alerta indicado que la contraseña no es correcta, igual que pasaría si fuese el usuario el que es incorrecto.

Con respecto a la parte de Registro, se ha comentado que la contraseña se guarda cifrada y esto se realiza con la función `bcrypt.generate_password_hash(datos_usuario['contraseña']).decode('utf-8')`, con la cual se recogen el valor de la contraseña y se codifica para después guardarla en la base de datos.

Finalmente, para comunicar la parte de Iniciar sesión con la parte de Registro, igual que se hará con el resto de páginas, se debe comunicar utilizando un fichero auxiliar, `App.js` en mi caso, donde se recogen todas las rutas de las diferentes páginas, para poder acceder a ellas. Dentro de la parte de iniciar sesión, para acceder a la página comentada, se debe hacer con la siguiente función: `<Link to="/registro">Regístrate aquí</Link>`. Con esto, vemos que al clicar en regístrate aquí, se nos redirige a la parte del registro y se nos muestra esta página.

En la imagen posterior, se muestra como se realizan los distintos enlaces en la página `App.js` comentada.

```
<Route path="/" element={<InicioSesion />} />  
<Route path="/registro" element={<Registro />} />
```

Ilustración 34 – App.js

3.1.3 Pruebas y Resultado final

Tras esta fase de implementación, donde se han realizado las funcionalidades más importantes de la fase del Inicio de sesión y del Registro, se realizan una serie de pruebas para ver si todo ha ido correctamente.

Una vez probado el funcionamiento de estas páginas, se optó porque todo se realizaba correctamente, tras una serie de pruebas. Tras esto, fue revisada por el tutor, el cual indicó que la implementación era buena y que se podía seguir con la iteración 2 sin necesidad de modificar nada de esta primera iteración.

Es por esto que el resultado final de ambas páginas es el siguiente:

Analiza tu corpus

Iniciar Sesión

Usuario:

Contraseña:

 Iniciar Sesión

Si no tienes cuenta, [Regístrate aquí](#).

Activar Windows
Ve a Configuración para activar Windows.

Ilustración 35 – Inicio Sesión

Analiza tu corpus

Registrarse

Usuario:

Email:

Contraseña:

Nombre:

Primer Apellido:

 Registrarse

Activar Windows
Ve a Configuración para activar Windows.

Ilustración 36 – Registro

3.2 Segunda Iteración

3.2.1 Planificación

En esta segunda fase del proyecto se debe corregir los errores cometidos anteriormente y que se han hablado con el tutor. Como en la primera iteración, no ha habido sugerencias de cambios, pasamos directamente a implementar lo nuevo.

En esta segunda iteración, procedemos a implementar la página principal donde se mostrarán todas las colecciones, además de unos correspondientes botones para poder ver, analizar y eliminar estas. También, se pondrá un botón, tal y como se indicaba en el WireFrame de esta página, donde se mostrará una página para añadir colecciones a la página principal y que se almacenarán en la base de datos. Utilizando los botones nos mandarán a las distintas páginas. Por ejemplo, se clickamos sobre la página de ver se nos mostrará el contenido de la colección, o si clickamos sobre eliminar se borrará esa colección de la base de datos.

3.2.2 Implementación

En esta fase, no hay que centrarse en enlazar y comunicar los distintos elementos que participan en la interfaz web, puesto que ya se han comentado en la iteración anterior. Es por esto, que pasaremos a implementar directamente la página principal, propiamente dicha.

En esta segunda interacción hay que incluir funcionalidades algo más complejas que en la anterior. En primer lugar, se ha creado la página principal, en la cual se muestran las distintas colecciones que se vayan añadiendo. Inicialmente, esta página está vacía y simplemente contiene un botón de más para añadir colecciones. Al clickar sobre este, se nos muestra una página donde podemos añadir un nombre y subir un fichero, el cual queremos que sea solo .Zip. Para esto añadimos la siguiente comprobación en la parte de frontend: `accept=".zip"`. Sin embargo, para guardarlo en la parte de la base de datos añadiremos la ruta de la carpeta del archivo sin el .zip puesto que así será más fácil gestionarlo y trabajar con él, sin necesidad de tener que descomprimirlo todas las veces. Esto lo realizamos de la siguiente manera:

```
with zipfile.ZipFile(archivo, 'r') as zip_ref:
    # Extraer todo el contenido del archivo ZIP
    directorio_destino = os.path.join(usuario_directorio) # Elimina la extensión .zip
    os.makedirs(directorio_destino, exist_ok=True)
    zip_ref.extractall(directorio_destino)
```

Ilustración 37 – Descomprimir Zip

De esta manera podremos guardar el archivo en una carpeta interna para posteriormente guardar la ruta de esta en la base de datos.

Tras esto, el archivo se guarda en la base de datos y se nos dirige a la página principal de igual forma que se comentaba en la iteración anterior. Una vez, aquí se mostrará el nombre de la colección. Es importante destacar, que anteriormente había que incluir el nombre y debía ser igual que el del fichero, puesto que así garantizamos un mayor procesamiento del mismo.

Posteriormente, con esta colección podemos hacer varias cosas. En primer lugar, podemos ver el contenido de la misma, mediante la cual, al pulsar sobre el botón correspondiente se nos mandará a esta página. En esta podemos ver su contenido. Para esto, en la parte del backend se abrirá el archivo, se leerá su contenido y se mandará a la parte del frontend donde será mostrado, de la siguiente forma dividiendo entre .csv y .json:

```
if nombreArchivo.endswith('.csv'):
    with open(filepath, 'r', encoding='utf-8') as file:
        csv_content = file.read()
        return jsonify({'filename': nombreArchivo, 'type': 'csv', 'content': csv_content})
elif nombreArchivo.endswith('.json'):
    with open(filepath, 'r', encoding='utf-8') as file:
        json_content = json.load(file)
        return jsonify({'filename': nombreArchivo, 'type': 'json', 'content': json_content})
```

Ilustración 38 – Leer Archivo

Finalmente, otras de las funcionalidades que se van a realizar en esta iteración es la de eliminar una colección. Para eso, se creará un botón rojo con el nombre de “Eliminar”, mediante el cual al pulsar mandaremos una solicitud al backend indicando que debe eliminar esa tupla de la colección de la base de datos. Tras esto, se devuelve un mensaje de acierto o error en función de si se ha podido o no eliminar la colección. Tras esto, la página no se actualizaría y seguiría apareciendo la colección. Para mejorar esto, volvemos a cargar la página con los archivos que haya en la página y por lo tanto aparecerá la página principal sin este corpus eliminado.

Cabe destacar que en esta fase, también se creará un botón de analizar, que ahora mismo no tendrá ninguna función. Además, también se creará un botón de cerrar sesión el cual pues redirigirá al usuario a la página de iniciar sesión y se cerrará la sesión del usuario sin poder acceder a las funcionalidades que esta interfaz web ofrece. Para esto, en la parte del backend, simplemente pondremos la variable de *usuario_activo* a *none* y se redirigirá a inicio de sesión en la parte del frontend.

```
@app.route('/logout', methods=['POST'])
def logout():
    global usuario_activo
    usuario_activo = None
    return jsonify({'message': 'Sesión cerrada correctamente'}), 200
```

Ilustración 39 – Cerrar Sesión

3.2.3 Pruebas y Resultado final

Tras esta segunda fase de implementación, donde se han realizado las funcionalidades más importantes de la página principal, tales como la página de ver una determinada colección, eliminarla o añadir una nueva. Además, también se ha realizado el botón de analizar sin funcionalidad actualmente y el de cerrar sesión.

Una vez probado el funcionamiento de estas páginas, se llegó a la conclusión de que funcionaba correctamente. Tras esto, fue revisada por el tutor, el cual indicó que la implementación de las funcionalidades estaba realizada correctamente, pero

que en tema de diseño había que realizar una serie de retoques para que quedara mejor y la aplicación diera un salto de calidad.

En cuanto a la página de ver, sería necesario meter un scroll para no tener que bajar todos los archivos existentes con sus correspondientes datos. De esta manera, hacemos que para el usuario sea mucho más cómodo volver a la página principal, sin tener que bajar hasta el final. Además, sería necesario meter un buscador de archivos, puesto que de esta manera no tenemos que bajar al final de todos los archivos y andar buscando por todos ellos para encontrarlo, si no que con un nombre ya podemos dar con él de manera fácil y cómoda.

Fuera de esta página, también sería necesario añadir iconos en todos los botones para así mejorar la aplicación y que esta sea mucho más fácil e intuitiva.

A continuación se verá un antes y un después de estos cambios, aunque estos se realicen en la iteración posterior. De esta manera, podemos ver de forma más clara los cambios realizados:

Página Principal (Antes)



Ilustración 40 – Página Principal (Antes)

Página Principal (Después)



Ilustración 41 – Página Principal (Después)

Añadir Colección (Antes)

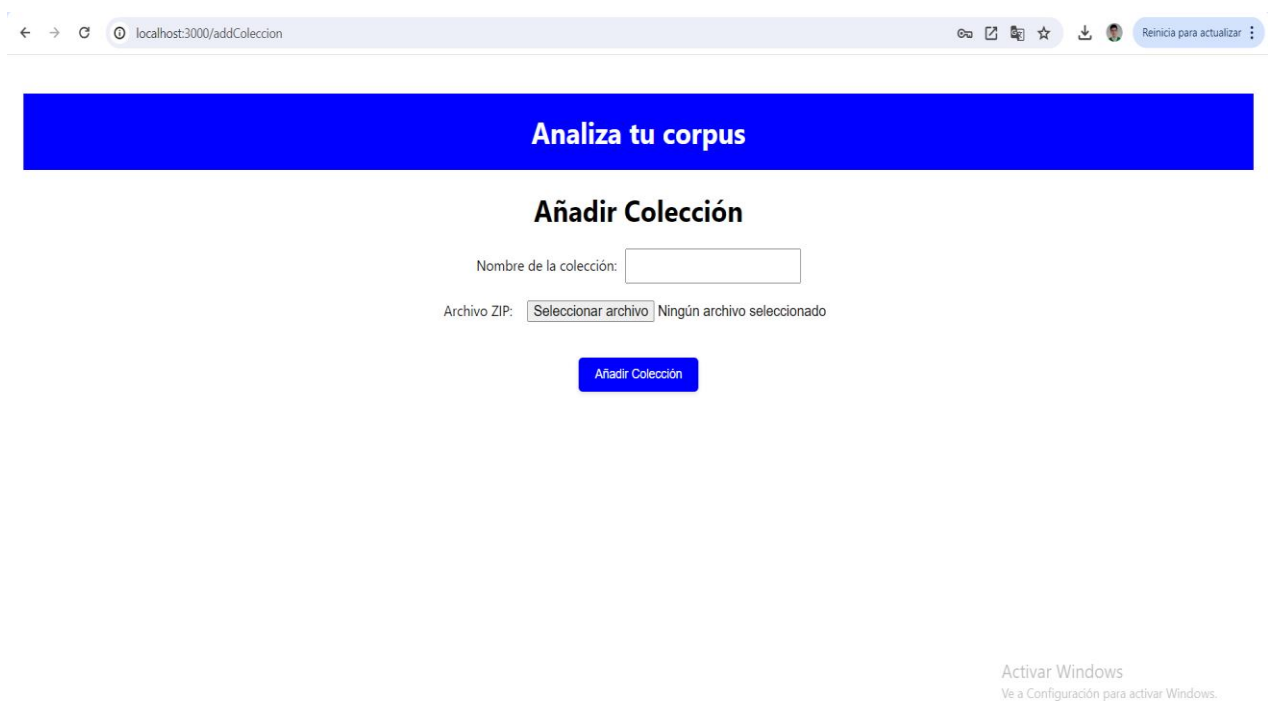


Ilustración 42 – Añadir Colección (Antes)

Añadir Colección (Después)

Analiza tu corpus

Añadir Colección

Nombre de la colección:

Archivo ZIP: Ningún archivo seleccionado

Nota: El nombre de la colección debe coincidir con el nombre del archivo ZIP.

Activar Windows
Ve a Configuración para activar Windows.

Ilustración 43 – Añadir Colección (Después)Ver Colección (Antes)

Analiza tu corpus

Colección "ED"

Número de archivos: 336 docs

Listado de archivos:

- gold_label.csv
- subject1.json
- subject10.json
- subject100.json
- subject101.json
- subject102.json
- subject103.json
- subject104.json
- subject105.json
- subject106.json
- subject107.json
- subject108.json
- subject109.json
- subject11.json
- subject110.json

Ilustración 44 – Ver Colección (Antes)

Ver Colección (Después)

Analiza tu corpus

Colección "ED"

Número de archivos: 336 docs

Listado de archivos:

subject3

- [subject3.json](#)
- [subject30.json](#)
- [subject300.json](#)
- [subject301.json](#)
- [subject302.json](#)
- [subject303.json](#)

Activar Windows
 Ver configuración para activar Windows.
[← Volver Página Principal](#)

Ilustración 45 – Ver Colección (Después)

Archivo "subject3.json"

```
[
  {
    "date": "2020-04-09 04:58:32",
    "id_message": 34558803124,
    "message": "hola soy nueva 😊"
  },
  {
    "date": "2020-04-09 05:16:00",
    "id_message": 43677758135,
    "message": "ok ya me presentó"
  },
  {
    "date": "2020-04-09 05:16:52",
    "id_message": 35015343844,
    "message": "hola soy azzu soy Ana Peso : 55 meta : 46 tengo 18 años"
  },
  {
    "date": "2020-04-09 05:18:11",
    "id_message": 20214870549,
    "message": "gracias ♥ "
  },
  {
    "date": "2020-04-09 05:35:53",
    "id_message": 75880739408,
    "message": "yo no cené así que me comí una manzana jeejeje"
  },
  {
    "date": "2020-04-09 05:38:37",
    "id_message": 26124810010,
    "message": "que buenos concejos yo no sabía eso ah 😊"
  }
]
```

Activar Windows
 Ver configuración para activar Windows.
[← Volver a la Lista de Archivos](#)

Ilustración 46 – Ver Colección (Después) 2

3.3 Tercera Iteración

3.3.1 Planificación

En esta tercera fase del proyecto se debe corregir los errores cometidos anteriormente y que se han hablado con el tutor al igual que en la fase anterior. Para esta última, tuvimos varios aspectos a corregir:

- Añadir un scroll en la pantalla de ver colección
- Añadir un buscador de archivos en la página de ver colección.
- Añadir iconos en todos los botones

Es por eso, que en primer lugar para esta iteración, debemos corregir los errores. Estas correcciones están visualizadas en el subapartado anterior para una mayor comprensión de los cambios realizados.

En esta tercera iteración, se creará la página del análisis descriptivo. A esta página se accederá al clicar en el botón de analizar de la pantalla principal. Tras esto, se debe realizar el análisis utilizando la herramienta de TextFlow. Posteriormente, deben aparecer los distintos metadatos pero todos sin analizar. A continuación, deberemos poder elegir los que queremos que se analicen y pasarán a la parte de analizados. Una vez, hecho todo lo anterior, debemos crear un botón o un enlace para cada uno de estos con el fin de poder ver el contenido y los detalles del análisis. Tras esto, se nos mandará a una página donde veremos lo comentado anteriormente. Finalmente, al igual que en todas también debemos tener la capacidad de volver a la página principal y tengo que añadir una opción para poder realizar un análisis contrastivo que se trabajará en la siguiente y última iteración.

3.3.2 Implementación

En esta tercera iteración se va a implementar el aspecto del análisis descriptivo como se ha comentado anteriormente. Esta página es la resultante de clicar sobre el botón de analizar. Una vez, pulsado este, se inicia el proceso del análisis propiamente dicho. Para esto, se ha creado un fichero auxiliar llamado *Archivo.py* el cual recogerá

este proceso. Este se encarga de añadir el proyecto de TextFlow a un directorio interno y desde este se importan todas las librerías disponibles. Tras esto, se añaden todos los análisis los cuales son funciones de la siguiente forma:

```
emotion = EmotionAnalyzer()
ironity = IronyAnalyzer()
isal = IsalAnalyzer()
lemma = LemmaAnalyzer()
lexicalDiversity = LexicalDiversityAnalyzer(lemmatizer=WordNetLemmatizer().lemmatize)
liwc = LiwcAnalyzer()
ncr = NcrAnalyzer()
ner = NERAnalyzer()
n1 = NGramsAnalyzer(tokenizer=RegexpTokenizer(r"[\w']+"), ngramsSize = 1)
n2 = NGramsAnalyzer(tokenizer=RegexpTokenizer(r"[\w']+"), ngramsSize = 2, stopwords=[])
n3 = NGramsAnalyzer(tokenizer=RegexpTokenizer(r"[\w']+"), ngramsSize = 3, stopwords=[])
n4 = NGramsAnalyzer(tokenizer=RegexpTokenizer(r"[\w']+"), ngramsSize = 4, stopwords=[])
perplexity = PerplexityAnalyzer(device="cpu")
polarity = PolarityAnalyzer()
```

Ilustración 47 – Análisis

Una vez realizado lo anterior, se ejecuta un `.analyze` para cada uno de estos y el resultado se guarda en un diccionario con una serie de etiquetas que serán muy útiles posteriormente. A continuación, el resultado de todo este diccionario se guarda en una Excel, donde las etiquetas serán las cabeceras de cada una de las columnas y los valores, los respectivos que se encuentran para cada uno de los análisis realizados.

Este archivo en Excel es el que se guarda en la base de datos como ruta del análisis descriptivo. Todo esto se realiza desde la parte del backend donde se llama a este archivo comentado de la siguiente manera, pasándole los parámetros que necesitan, los cuales son:

- Usuario activo: necesario para crear el nombre y saber la ruta donde se tiene que guardar el fichero resultante.
- Ruta de la carpeta: es necesario tener la ruta de la carpeta donde se encuentran todos los archivos para poder así realizar el análisis de los archivos existentes en su interior.

- Ruta del archivo csv: muy útil para saber los valores propiamente dichos sobre los cuales se tiene que realizar el análisis y sobrescribir este.

Para llamar al archivo se realiza de la siguiente forma:

```
ejecutar_Archivo = f"python archivo.py {usuario_activo} {rutaArchivoAux[0][0]} {archivo_csv}"  
subprocess.run(ejecutar_Archivo, shell=True, check=True)
```

Ilustración 48 – Ejecución archivo

Una vez hecho todo lo anterior, el resultado es un archivo en Excel, el cual se guardará en la base de datos mediante un Update en la tabla colecciones, sustituyendo el null existente al añadir una colección por la ruta del archivo Excel resultante.

Como ya se ha realizado el análisis, se muestra la pantalla con los metadatos que se pueden visualizar y que han sido analizados. Una vez, aquí, en primer lugar, están todas las opciones en sin analizar como es obvio, lo que se traduce con un 0 en la base de datos. Al clickar sobre analizar en cada una de ellas, se van a pasar a la parte superior de la pantalla, lo que significa que en la base de datos, este valor habrá cambiado a 1 y por lo tanto se podrá elegir para ver su contenido.

Una vez aquí, cada nombre de las características se pondrá subrayado, con un aspecto para poder seleccionarlo ya que es un enlace. Al pulsar sobre este, se nos manda a la opción de poder ver su contenido.

Para la parte del contenido hay que tener en cuenta que en la base de datos los valores se guardan como un string de la siguiente manera:

Min/max/moda/desTip/media

- Min: hace referencia al valor mínimo de todos los existentes
- Max: hace referencia al valor máximo de todos los existentes

- Moda: hace referencia al valor más repetido de todos.
- desTip: se refiere a la desviación típica de todos los existentes.
- Media: hace referencia a la media entre todos los valores.

Es por esto, que a la hora de leerlos en la parte del frontend, se realizará de la siguiente forma:

```
<table style={{ width: '100%', borderCollapse: 'collapse' }}>
  <tbody>
    <tr>
      <td style={{ padding: '10px', fontWeight: 'bold' }}>Minimo</td>
      <td style={{ padding: '10px' }}>{getValor(1)}</td>
    </tr>
    <tr>
      <td style={{ padding: '10px', fontWeight: 'bold' }}>Maximo</td>
      <td style={{ padding: '10px' }}>{getValor(2)}</td>
    </tr>
    <tr>
      <td style={{ padding: '10px', fontWeight: 'bold' }}>Varianza</td>
      <td style={{ padding: '10px' }}>{getValor(3)}</td>
    </tr>
    <tr>
      <td style={{ padding: '10px', fontWeight: 'bold' }}>Moda</td>
      <td style={{ padding: '10px' }}>{getValor(4)}</td>
    </tr>
    <tr>
      <td style={{ padding: '10px', fontWeight: 'bold' }}>Media</td>
      <td style={{ padding: '10px' }}>{getValor(5)}</td>
    </tr>
  </tbody>
</table>
```

Ilustración 49 – Tabla de metadatos

Esta es una forma bastante evidente, ya que en primer lugar se separan los valores con el carácter “/” y tras esto, simplemente hay que acceder a cada uno de estos.

Finalmente, debe existir un botón en esta página el cual nos permita descargarnos la tabla para poder trabajar fuera de esta aplicación. Esto se consigue con una función denominada *toCSV*. En esta, se separan los valores con el carácter anterior, lo que indicará que cada uno de los valores entre “/”, formará parte de una celda distinta dentro de cada fila. Para acabar, se define el nombre de las cabeceras de las columnas y se añaden los valores, acabando con *link.setAttribute('download',*

'emofinder.csv') para descargar el archivo. El siguiente ejemplo para Emofinder muestra el funcionamiento:

```
const rows = [  
  ["Característica", "Mínimo", "Máximo", "Varianza", "Moda", "Media"],  
  ["Emofinder", ...data.Emofinder]  
];
```

Ilustración 50 – Descargar datos

3.3.3 Pruebas y Resultado final

Tras esta tercera fase de implementación, donde se ha realizado la página del análisis descriptivo, junto con las correspondientes visualizaciones de los metadatos y la descarga de las mismas, se han realizado una serie de pruebas para ver si todo ha ido correctamente.

Una vez probado el funcionamiento de estas páginas, se llegó a la conclusión de que funcionaba correctamente. Tras esto, fue revisada por el tutor, el cual indicó que no era del todo correcto lo que estaba realizando y debía hacer algunos cambios tanto de funcionalidad como de diseño.

En primer lugar, la parte de sin analizar y analizados, estaba realizado con un checkbox que te permitía marcar varios metadatos y tras clickar en analizar, estos se subían a analizados. Sin embargo, lo que el tutor me pidió fue que debía tener tanto un botón para añadir a analizados como otro para eliminar en los que ya lo estaban y quitar el checkbox.

Dentro de los valores de cada metadato, en mi caso solamente añadí la media de estos. Pero esto no era así, ya que lo que el tutor quería era añadir tanto el mínimo, máximo, moda, desviación típica como la media en cada una de estas características.

Al igual que en ver colección, en este caso también era conveniente añadir un scroll que cuadrara más lo anterior. Finalmente, también se comentó que sería necesario añadir el botón de descargar la tabla para trabajar con ella posteriormente.

A continuación, se muestra una visión del resultado final comparado con la visión anterior para que se comprenda mejor al igual que en la anterior iteración:

Análisis descriptivo (Antes)

The screenshot shows a web interface for corpus analysis. At the top, there is a blue header with the text 'Analiza tu corpus'. Below this, there are two buttons: 'Análisis Descriptivo' (highlighted in white) and 'Análisis Contrastivo' (highlighted in grey). Under the 'Análisis Descriptivo' button, the text 'Colección "ED"' is displayed. Below this, there is a section titled 'Sin analizar' with a list of analysis options. Each option has a checkbox, and some are checked. The checked options are: Emociones, Ironía, Polaridad, and Volumetría. At the bottom right of the list, there is a blue button labeled 'Analizar'. At the bottom center, there is a blue button labeled 'Volver a Página Principal'.

Analiza tu corpus

Análisis Descriptivo Análisis Contrastivo

Colección "ED"

Sin analizar

- BDLASE
- Complejidad
- Emofinder
- Emojis
- Polaridad Emojis
- Polaridad Emoticonos
- Emociones
- Ironía
- ISAL
- Lemas
- Diversidad Lexica
- Polaridad Emoticonos
- Emociones
- Ironía
- ISAL
- Lemas
- Diversidad Lexica
- LIWC
- NCR
- NER
- N-Grams
- Perplejidad
- Polaridad
- POS
- SEL
- Estilometría
- Volumetría

Analizar

Volver a Página Principal

Ilustración 51 – Análisis Descriptivo Antes

Análisis descriptivo (Después)

Analiza tu corpus

Colección "ED"

Análisis Descriptivo
Análisis Contrastivo

Analizados

BDLASE	- Eliminar análisis
Complejidad	- Eliminar análisis
Emofinder	- Eliminar análisis
Emojis	- Eliminar análisis
SEL	- Eliminar análisis
Estilometría	- Eliminar análisis
Volumetría	- Eliminar análisis

Sin analizar

Emofinder	+ Analizar
Emojis	+ Analizar
Emociones	+ Analizar
ISAL	+ Analizar
NER	+ Analizar

← Volver a Página Principal

Activar Windows
Ve a Configuración para activar Windows.

Ilustración 52 – Análisis Descriptivo Después

Ver Metadato (Antes)



Colección "ED"

Volumetría

Resumen para la Característica "VOLUMETRÍA"

Número de palabras: **10 palabras**

Número de caracteres: **49 caracteres**

Número de palabras únicas: **10 palabras**

Media longitud de palabras: **5 caracteres**

[Volver Página Principal](#)

Ilustración 53 – Ver Metadato Antes

Ver Metadato (Después)

Volumetría

Resumen para la Característica "VOLUMETRÍA"

Característica	Valor
Número de Palabras	Minimo 4.00
	Maximo 18.00
	Varianza 17.00
	Moda 11.00
	Media 10.00
Número de caracteres	Minimo 16.00
	Maximo 83.00
	Varianza 459.69
	Moda 16.00
	Media 48.90

Activar Windows
Vé a Configuración para activar Windows.

[Descargar tabla en formato CSV](#)
[← Volver Página Principal](#)

Ilustración 54 – Ver Metadato Después

3.4 Cuarta Iteración

3.4.1 Planificación

En esta cuarta y última fase del proyecto se debe corregir los errores cometidos anteriormente y que se han hablado con el tutor al igual que en la fase anterior. Para esta última, tuvimos varios aspectos a corregir:

- Cambiar aspectos de sin analizar y analizar, añadiendo botón de eliminar análisis
- Añadir mínimo, máximo, moda, desviación típica y media en lugar de solo la media.
- Añadir scroll para que esté todo más compacto.
- Añadir un botón para descargar la tabla y poder trabajar con ella fuera de la aplicación

Es por eso, que en primer lugar para esta iteración, debemos corregir los errores. Estas correcciones están visualizadas en el subapartado anterior para una mayor comprensión de los cambios realizados.

En esta última iteración, se creará la página del análisis contrastivo. A esta página se accederá al clickar en el botón de análisis contrastivo situado en la parte superior derecha del análisis descriptivo junto a este mismo nombre. En primer lugar, se deberá elegir la colección con la que se quiere comparar. Si esta no está analizada, deberá salir un mensaje de error. Tras esto, se debe realizar este análisis utilizando la herramienta de TextFlow. Posteriormente, deben aparecer los metadatos analizados por ambas colecciones descriptivamente. A continuación, deberemos poder el que queremos que se analice. Una vez hecho todo lo anterior, debemos crear un botón con el fin de poder ver el contenido y los detalles del análisis. Tras esto, se nos mandará a una página donde veremos lo comentado anteriormente. Finalmente, al igual que en todas también debemos tener la capacidad de volver a la página principal y tengo que añadir una opción para poder realizar un análisis contrastivo que se trabajará en la siguiente y última iteración.

3.4.2 Implementación

En esta cuarta iteración se va a implementar el aspecto del análisis contrastivo como se ha comentado anteriormente. A esta página se accede después de clicar sobre la pestaña de Análisis contrastivo. Una vez, pulsado sobre esta, se mostrará un desplegable donde podremos seleccionar entre todas las colecciones que este usuario tiene disponible. Tras esto, si este corpus no tiene la ruta del archivo del análisis descriptivo en su tupla, se mostrará un mensaje de error. En caso contrario, se realizará el análisis propiamente dicho. Para esto, se ha creado un fichero auxiliar llamado *pruebaContrastivo.py* el cual recogerá este proceso. Este se encarga de coger ambas colecciones y guardarlas en un diccionario y tras esto, realizar los test correspondientes mediante la función *Test()*. Tras esto, se realiza el análisis contrastivo utilizando la función *.report* de la siguiente forma:

```
testResults = t.report(df1_comun, df2_comun, nombre_coleccion_uno, nombre_coleccion_dos, v, 'collection', [None, 'collection'], [None, 'collection'])
```

Ilustración 55 – Realizar análisis contrastivo

El resultado de todo esto se guarda, como vemos en la imagen en una variable denominada *testResults*. Tras esto, sabemos que existen tres tipos de test, los de **normalidad**, los **paramétricos** y los **no paramétricos**.

Una vez realizado lo anterior, añadimos cada test en un Excel y lo guardamos internamente en una carpeta que recogerá todos los análisis contrastivos realizados.

Este archivo en Excel que tendrá cada uno de los test es el que se guarda en la base de datos como ruta del análisis contrastivo donde cada colección tendrá varias asociadas en función de las colecciones con las que se compare. Todo esto se realiza desde la parte del backend donde se llama a este archivo comentado de igual forma que anteriormente, pasándole los parámetros que necesitan, los cuales son:

- Usuario activo: necesario para crear el nombre y saber la ruta donde se tiene que guardar el fichero resultante.

- Ruta análisis descriptivo colección: es necesario tener la ruta del análisis descriptivo de la colección que tenemos activa en ese momento.
- Ruta análisis descriptivo colección a comparar: es necesario tener la ruta del análisis descriptivo de la colección con la que queremos comparar la que tenemos activa en ese momento.

Una vez hecho todo lo anterior, el resultado son varios archivos Excel, el cual se guardará en la base de datos mediante realizando un Insert y creando una tupla nueva para cada uno de los análisis contrastivos que se han realizado. La ruta que se crea tiene el nombre análisis{coleccion1}{coleccion2}.

Cuando ya se ha hecho esto, ya se ha realizado el análisis, por lo que aparece otro desplegable. Este, contendrá los metadatos que han sido analizados comúnmente por las dos colecciones, esto es, que el mismo metadato tenga un 1 en la base de datos, para esa característica. Cabe destacar, que las únicas características que se pueden comparar son las numéricas ya que TextFlow está diseñado para esto únicamente.

Una vez aquí, podremos seleccionar la que queramos y al clickar sobre analizar se nos muestra la pantalla que contiene el contenido con las tablas necesarias.

En este caso, la forma de mostrar los valores en la parte de frontend es un poco distinta. Ahora, los valores no se guardan de la misma forma que antes, si no que se guarda la ruta directamente, por lo que desde el backend se mandan los valores de cada uno de estos separados por "/" y por "," dentro de cada test. En primer lugar para mostrar los datos, se accede a cada uno de los test por separado y a la hora de ponerlos por pantalla separamos por la "," para saber el valor de cada uno de los test de cada tipo de la siguiente forma:


```
<table style={{ borderCollapse: 'collapse', width: '100%', height: '100%' }}>
  <thead>
    <tr style={{ backgroundColor: 'lightgray' }}>
      <th style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>Característica</th>
      <th style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>Test de Normalidad</th>
      <th style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>Test Paramétricos</th>
      <th style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>Test No Paramétricos</th>
    </tr>
  </thead>
  <tbody>
    {DatosTabla.map((row, index) => (
      <tr key={index}>
        <td style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>{row.name}</td>
        <td style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>{row.values[0]}</td>
        <td style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>{row.values[1]}</td>
        <td style={{ border: '1px solid black', padding: '5px', fontSize: '12px' }}>{row.values[2]}</td>
      </tr>
    ))}
  </tbody>
</table>
```

Ilustración 56 – Tabla análisis contrastivo

En la anterior fotografía se ve explicado lo comentado anteriormente. Sin embargo, para llegar a este punto es necesario primero haber separado por “/”. También se comprueba y se pasa a string el valor en caso de que no lo estuviera.

```
return data.split('/').map(item => {
  try {
    const ItemCambiado = JSON.parse(item.replace(/'/g, ''));

    if (Array.isArray(ItemCambiado)) {
      return ItemCambiado.map(value => Array.isArray(value) ? value : [value]);
    }

    if (typeof ItemCambiado === 'string' && ItemCambiado.startsWith '[' && ItemCambiado.endsWith (']')) {
      return convertidorArrayString(ItemCambiado).map(value => [value]);
    }

    return [[ItemCambiado]];
  } catch (e) {
    console.error('Error cambiando el atributo:', e);
    return [[item]];
  }
});
```

Ilustración 57 – Tabla análisis contrastivo 2

3.4.3 Pruebas y Resultado final

Tras esta última fase de implementación, donde se ha realizado la página del análisis contrastivo, junto con la comprobación de los metadatos analizados y la visualización de la característica elegida, se han realizado una serie de pruebas para ver si todo ha ido correctamente.

Una vez probado el funcionamiento de estas páginas, se llegó a la conclusión de que quizás no funcionaba de la manera correcta. Tras esto, fue revisada por el tutor, el cual indicó que no era lo que se pedía con este tipo de análisis y debía hacer un cambio tanto de funcionalidad como de diseño. Esto, era debido a que lo que se pedía era un análisis contrastivo y yo lo entendí como un análisis comparativo entre dos análisis descriptivo de dos colecciones diferentes.

En primer lugar, como he comentado debía cambiar la funcionalidad de realizar el análisis de uno comparativo a uno contrastivo.

Además, no debían verse todos los metadatos, ya que para este tipo de comparación solo se tienen en cuenta las características numéricas puesto que con las otras TextFlow no puede trabajar.

Posteriormente, dentro de la visualización de estas, había que ajustar las tablas y alinearlas horizontal y verticalmente de forma que todo quede cuadrado, además de añadir un scroll igual que para todas las pantallas comentadas anteriormente.

Para finalizar, era conveniente añadir iconos a los botones de este apartado también, al igual que era necesario comprobar si la colección había sido analizada descriptivamente antes o no.

A continuación, se muestra una visión del resultado final comparado con la visión anterior para que se comprenda mejor al igual que en la anterior iteración. No se mostrarán más iteraciones puesto que serían solo de corregir estos fallos.

Análisis contrastivo (Antes)



Ilustración 58 – Análisis Descriptivo Antes

Análisis contrastivo (Después)



Ilustración 59 – Análisis Descriptivo Después

Ver metadato Contrastivo (Antes)

Analiza tu corpus

Ironía

Colección "ED" VS Colección "Anxiety"

Resumen para la Característica "IRONIA" en ED

Valor No Ironía: **0.059286900842562315**

Valor Sí Ironía: **0.9407131016254425**

VS

Resumen para la Característica "IRONIA" en Anxiety

Valor No Ironía: **0.5551220105960966**

Valor Sí Ironía: **0.4448779986705631**

[Volver Página Principal](#)

Ilustración 60 – Ver metadato contrastivo Antes

Ver metadato Contrastivo (Después)

Analiza tu corpus

Ironía

Característica	Test de Normalidad														Test Paramétricos							
	Shapiro stat	Shapiro p-value	D'Agostino stat	D'Agostino p-value	Anderson-Darling stat	Anderson-Darling crit_val	Anderson-Darling sig_level	Chi-Square stat	Chi-Square p-value	Lilliefors stat	Lilliefors p-value	Jarque-Bera stat	Jarque-Bera p-value	Kolmogorov-Smirnov stat	Kolmogorov-Smirnov p-value	Criteria_1	Criteria_2	Students t-test stat	Students t-test p-value	Paired Students t-Test stat	Paired Students t-Test p-value	ANO' sta
No Ironía	0.64	0.00	10.17	0.01	2.10	[0.497 0.566 0.679 0.792 0.942]	[15. 10. 5. 2.5 1.]	5.89	0.92	0.31	0.00	6.14	0.05	0.50	0.00	ED	Anxiety	1.00	0.01	-2.35	0.14	19.3
Ironía	0.64	0.00	10.17	0.01	2.10	[0.497 0.566 0.679 0.792 0.942]	[15. 10. 5. 2.5 1.]	1.35	1.00	0.31	0.00	6.14	0.05	0.60	0.00	ED	Anxiety	29.00	0.01	2.35	0.14	19.3

[← Volver Página Principal](#)

Activar Windows
Ve a Configuración para activar Windows.

Ilustración 61 – Ver metadato contrastivo Después

4 CONCLUSIONES Y TRABAJOS FUTUROS

Para finalizar con la memoria de este trabajo fin de grado, se van a realizar una serie de conclusiones y mejoras de cara al futuro para que esta aplicación tenga éxito y pueda ser usada por todos los públicos y no solo por expertos en la materia.

En el apartado 1.5 definimos una serie de requisitos iniciales que debía tener la aplicación web que el tutor nos había pedido. Tras analizarlos, llegamos a la conclusión de que todos se han cumplido satisfactoriamente. Todo esto, se ha logrado a través de un enfoque basado en la metodología ágil SCRUM, se han logrado estos requisitos. Pero no solo esto, también se han añadido nuevas funcionalidades para que la página sea mucho más intuitiva y pueda ser usada para usuarios sin experiencia en esta asignatura.

Mediante el uso de React.js en el frontend se ha permitido desarrollar una interfaz de usuario dinámica, fácil e intuitiva a pesar de su complicada instalación. Por otra parte, Flask me ha proporcionado una base sólida y escalable, además de modular en el backend. Finalmente, cabe destacar que la elección de MySQL para la gestión de bases de datos ha sido crucial para garantizar un almacenamiento eficiente y seguro de la información, permitiendo consultas rápidas y una gestión eficaz de los corpus, además de ser una interfaz gráfica lo que ayuda en no tener que preocuparte por ejecutar comandos literalmente y una fácil extensión del proyecto para nuevas actualizaciones.

La fácil comunicación entre frontend, backend y base de datos han facilitado no solo el desarrollo inicial sino también la posibilidad de futuras extensiones y mantenimiento del sistema. La implementación de análisis descriptivos y contrastivos ha sido desarrollada de forma accesible, permitiendo a los usuarios obtener resultados significativos y utilizables a partir de los corpus gestionados. Además, con la utilización de un botón de descargas, permitimos que los usuarios puedan trabajar con estos resultados en muchos otros ámbitos.

Otra de las actuaciones que han ayudado a conseguir los objetivos marcados ha sido las constantes comprobaciones en cada una de las reuniones realizadas.

Cada iteración del proceso de desarrollo ha permitido refinar las funcionalidades, mejorar el rendimiento y garantizar que el producto final cumple con las expectativas tanto en términos de usabilidad como de eficiencia.

Finalmente, aunque el proyecto ha cumplido sus objetivos iniciales, se han identificado varias áreas de mejora y expansión que podrían considerarse en futuras iteraciones y trabajos con este proyecto.

Mejorar la Velocidad de Carga es uno de los aspectos a tener en cuenta. Ya que, aunque la aplicación actual funciona de manera fluida, es posible optimizar aún más los tiempos de carga y respuesta, especialmente cuando se manejan corpus de gran tamaño. Esto podría implicar la optimización del código de frontend o la implementación de técnicas avanzadas en el backend.

Implementar procesamiento asíncrono para análisis más complejos podría liberar recursos del servidor y mejorar la experiencia del usuario al permitir que ciertas tareas se ejecuten en segundo plano sin bloquear la interacción del usuario y que este pueda seguir interactuando con la página web.

Sería conveniente añadir una opción para que los usuarios puedan personalizar su experiencia en la aplicación, como elegir temas de interfaz o configurar paneles personalizados, algo que podría mejorar la satisfacción del usuario y la adopción del sistema.

Para terminar, sería conveniente añadir una infraestructura en la Nube con la cual se consiguen varios aspectos:

- Si el número de usuarios creciera rápidamente, al igual que el tamaño de los corpus número de usuarios y en el tamaño de los corpus gestionados, sería recomendable subir el proyecto a Google Cloud o. Esto garantizaría la escalabilidad del sistema y la capacidad de manejar una mayor carga sin comprometer el rendimiento. Además, podrías monetizarlo con lo que obtendríamos un gran beneficio.

Y para casos más extremos:

- Balanceo de Carga y Tolerancia a Fallos: Implementar técnicas de balanceo de carga y sistemas redundantes para asegurar la disponibilidad del servicio, incluso en situaciones de alta demanda o fallos de hardware.
- Seguridad: Para mejorar aún más la seguridad, se podría implementar un sistema de autenticación mejor, además de protegernos ante posibles ataques que pongan en aprieto a nuestro sistema.

Al terminar este Trabajo Fin de Grado, he llegado a la conclusión de que la integración de las tecnologías utilizadas, como React.js para frontend, Flask para backend y MySQL para la gestión de la base de datos, ha sido la elección correcta, ya que me ha permitido desarrollar un proyecto de manera fácil y muy personalizada. Durante este, han aparecido diferentes dificultades, pero gracias a los foros de las distintas tecnologías he podido solucionarlas.

Además, gracias a la planificación realizada al inicio del trabajo, he aprendido a gestionar mejor mi tiempo y a darle la importancia que tiene al hecho de diseñar antes de implementar. De esta manera, la implementación es mucho más rápida y no requiere un gran coste solucionar los problemas ocurridos en el desarrollo.

Finalmente, al realizar el presupuesto, también he entendido el precio que requiere realizar un trabajo como el mío si se realizara de manera profesional y no a nivel educativo. Esto me lleva a darle aún más valor al proyecto realizado.

5 DEFINICIONES Y ABREVIATURAS

- **Análisis de corpus:** Un corpus es un conjunto muy grande de textos ya sean científicos o literarios que se utilizan para su posterior análisis. Este consiste en examinar patrones y tendencias lingüísticas o extraer características, que pueden servir para comparar dos o más documentos.
- **Backend:** parte del sistema o componentes que trabajan “detrás” de la interfaz visible para los usuarios finales. Este conjunto de elementos se encarga de procesar los datos y la lógica que permiten a las funcionalidades ser accesibles a través de la UI.
- **Frontend:** parte del sistema o componentes que los usuarios finales utilizan directamente. Es la capa visible y accesible del sistema, donde los usuarios pueden interactuar y acceder a los servicios que la aplicación ofrece.
- **UI:** Interfaz de Usuario
- **PLN:** Procesamiento del Lenguaje Natural
- **DL:** Deep Learning
- **WWW:** World Wide Web
- **Framework:** es un conjunto de herramientas, guías y estructuras prediseñadas que se emplean para desarrollar y organizar software de forma eficiente. Es como una plataforma base sobre la cual los programadores y desarrolladores pueden construir y adaptar sus aplicaciones.
- **CSS:** lenguaje encargado de describir la presentación de documentos escritos en HTML o XML. Este, permite a los desarrolladores separar el contenido del documento de su presentación visual.
- **DOM:** es una interfaz de programación para documentos HTML y XML. Este, representa la estructura del documento y permite que sus elementos sean manipulados mediante JavaScript.

6 MANUAL DE DESPLIEGUE

En este apartado se mostrará una serie de pasos a seguir para poner en marcha la página web. Como ya se ha comentado anteriormente la página web está realizada con React.js para la parte del frontend, Flask de Python para la parte del backend y MySQL para la gestión de la base de datos.

En primer lugar, la carpeta que contiene el trabajo estará en github por lo que el primer paso será clonarlo:

1- git clone (enlace del github encontrado en el archivo EnlaceTFG)

2- cd ProyectoTFG

Tras esto, pasaremos a la parte del **frontend**. El proyecto contiene un archivo denominado **package.json**. Este, se encuentra dentro de la carpeta *client* por lo que para instalar esta parte hay que seguir los siguientes pasos:

3- cd client

4- npm install

De esta manera, se iniciarán todas las dependencias. Cabe destacar que hay que solucionar los problemas que existan de dependencias durante este proceso, los cuales se muestran fácilmente por la línea de comandos. Posteriormente, para ponerlo en marcha, simplemente hay que realizar lo siguiente:

5- npm start

Esto ejecutará la aplicación en el puerto predeterminado (usualmente `http://localhost:3000`) y abrirá una nueva ventana del navegador. Es importante destacar que previamente debe estar instalado npm para poder ejecutar todo lo anterior y haberlo añadido a las variables de entorno. Además previamente a esto es necesario haber instalado las siguientes bibliotecas:

- **react-router-dom**: Biblioteca utilizada para comunicar las páginas del frontend. Se instala con `npm install react-router-dom`.
- **react-icons**: Biblioteca utilizada para añadir iconos. Se instala con `npm install react-icons`.

Para la parte del **backend** tenemos un archivo denominado *app.py*. Para ejecutar esto, en primer lugar, debemos abrir una nueva terminal. Tras esto, hay que meterse de nuevo en el proyecto y tener instalado python y pip, y haberlo añadido a las variables de entorno. Tras esto, simplemente hay que hacer lo siguiente:

1- `python app.py`

Esto iniciará el servidor backend en un puerto especificado (usualmente `http://localhost:5000` o `http://127.0.0.1:5000`). Es importante destacar que previamente a esto, debemos instalar unas cuantas bibliotecas que serán necesarias para realizar distintas funcionalidades:

- **flask_cors**: Biblioteca utilizada para la comunicación con el frontend. Se instala con `pip install flask_cors`.
- **flask-mysqldb**: Biblioteca utilizada para comunicarse con la base de datos. Se instala con `pip install flask-mysqldb`.
- **flask_bcrypt**: Biblioteca utilizada para cifrar y descifrar la contraseña. Se instala con `pip install flask_bcrypt`.

Finalmente, para la parte de la **base de datos**, simplemente hay que instalar Xampp, la cual se instala desde la página oficial y posteriormente se clicka sobre el ejecutable, asegurando que tenemos elegida MySQL en los pasos. Tras esto lo iniciamos y le damos a start en la parte de Apache y MySQL. Para verla gráficamente, simplemente hay que escribir localhost en el navegador o clickar en Admin. Para terminar, dentro de esta aplicación, hay que buscar la opción de importar y añadir el archivo `.sql` que se encuentra en el repositorio.

7 BIBLIOGRAFÍA

- [1] Moreno, Antonio. <<Instituto de Ingeniería del conocimiento - Procesamiento del lenguaje natural ¿qué es?>> <https://www.iic.uam.es/inteligencia/que-es-procesamiento-del-lenguaje-natural/>
- [2] <<Deep Talk - Historia y actualidad del Procesamiento de Lenguaje Natural>>. Actualizado a 6 de noviembre de 2021. <https://blog.deep-talk.ai/historia-y-actualidad-del-procesamiento-de-lenguaje-natural-8de41a357ca9>
- [3] E. A. Puertas Del Castillo: Análisis de elementos fonéticos y elementos emocionales para predecir la polaridad en fuentes de microblogging, 34-35
- [4] Tutorialspoint. (s.f.). Introduction to Natural Language Processing. Recuperado de https://www.tutorialspoint.com/natural_language_processing/index.htm
- [5] Molla, Victor. <<Corpus Lingüísticos en la Inteligencia Artificial>>. Actualizado a 21 de Junio de 2024. <https://www.victormolla.com/corpus-linguisticos>
- [6] Saavedra, José Ángel.<<¿Qué es la visualización de datos?>>. Actualizado a 1 de Junio de 2023.<https://ebac.mx/blog/visualizacion-de-datos>