

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS
DE LA INFORMACIÓN



Biblioteca

"DESARROLLO DE UNA APLICACIÓN WEB
PARA EL ANÁLISIS DE SENTIMIENTOS EN
PUBLICACIONES EN TWITTER"

TRABAJO FIN DE GRADO

Julio - 2023

AUTOR: Sergio Muñoz Cano

DIRECTOR: Federico Botella Beviá



AGRADECIMIENTOS

Después de estos años de carrera, en la que he vivido muy buenos momentos, y conocido a gente estupenda, termina una de las mejores etapas de mi vida.

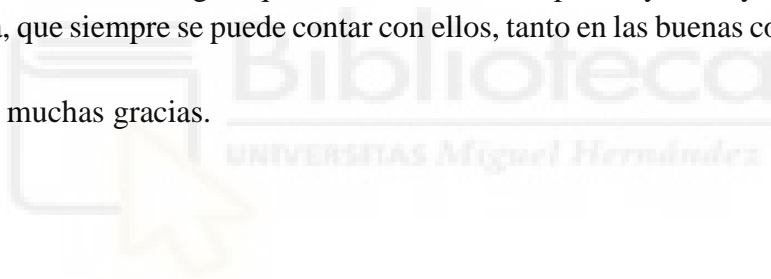
Por ello, me gustaría agradecer a todos mis compañeros, que han estado ahí en todo momento y han hecho que estos años sean más fáciles, y me han ayudado siempre que lo necesitara.

Agradecer también a todos los profesores, por enseñarnos todo lo necesario, y en especial a Federico Botella Beviá, por la asignatura que nos impartió en este último año de carrera, y por ser mi tutor de este trabajo de fin de grado.

Agradecer a mi familia, que siempre me han apoyado y ayudado cuando peor estaba, y me han ayudado a tomar decisiones difíciles, que a día de hoy, agradezco, especialmente a mi padre y a mi madre, que me han educado muy bien, y me han llevado por el buen camino, a mi hermano, que está estudiando fuera y aun así puedo contar con él, y a mis abuelas, que están siempre asegurándose que todo me vaya bien.

Por último, agradecer a mis amigos, que me han hecho siempre muy feliz, y hacen que la vida sea más divertida, que siempre se puede contar con ellos, tanto en las buenas como en las malas.

A todos, gracias, muchas gracias.



RESUMEN

Es un hecho que el uso de las redes sociales ha aumentado exponencialmente durante los últimos años. Cada vez más gente comparte sus opiniones, emociones y experiencias. Debido a esto, se ha vuelto una necesidad detectar los sentimientos que se desprenden en las redes, para muchos usos, como, por ejemplo, para analizar tendencias, toma de decisiones, investigación, estudios de mercado, etc.

Por ello, este trabajo se centrará en detectar sentimientos en tweets, puesto que Twitter es una de las redes sociales más utilizadas en todo el mundo, y la gente comparte sus opiniones ahí. Para conseguir este objetivo se va a desarrollar una aplicación web en Python, que permita a los usuarios ingresar un tweet o una palabra clave, y se obtendrán los sentimientos hallados en dichos tweets.

También será necesario, antes de desarrollar la aplicación, realizar un estudio teórico en el que se verán, los fundamentos del análisis de sentimientos, la recogida de datos mediante web scraping, el procesamiento de lenguaje natural (NLP), para procesar los datos recogidos con diversos métodos, y posteriormente la detección de emociones.

Una vez realizado dicho estudio teórico, se explicará la arquitectura de la aplicación, y posteriormente, se realizará su desarrollo, donde analizaremos los resultados obtenidos, y se sacarán las conclusiones necesarias.



Índice de contenido

1. INTRODUCCIÓN	11
1.1. Motivación del trabajo.....	11
1.2. Objetivos del trabajo.....	12
1.3. Motivación personal	12
2. ESTADO DEL ARTE.....	13
2.1. Emociones	13
2.2. Extracción de datos en Twitter	13
2.2.1. API de Twitter.....	14
2.2.2. Web Scraping.....	14
2.1.2.1. Beautiful Soup.....	15
2.1.2.2. SNScrape	15
2.3. Análisis de sentimientos	16
2.3.1. Definición de sentimiento	16
2.3.2. Concepto de análisis de sentimientos.....	16
2.3.3. Técnicas del análisis de sentimientos.....	16
2.4. Procesamiento del lenguaje natural	18
2.4.1. Preprocesamiento de texto para NLP	19
2.4.1.1. Limpieza del texto	19
2.4.1.2. Tokenización	19
2.4.1.3. Normalización de las palabras	19
2.4.1.4. Eliminación de stop words	20
2.4.1.5. N-gramas	20
2.4.1.6. Stemming.....	20
2.4.1.7. Lematización	20
2.4.2. Representación numérica del texto	21
2.4.2.1. Bolsa de palabras (Bag of words).....	21
2.4.2.2. TF-IDF.....	21
2.4.2.3. Word Embeddings	22
3. ARQUITECTURA DE LA APLICACIÓN WEB.....	24
3.1. Modelo-vista-controlador	24
3.2. Lenguajes y herramientas utilizados	25
3.2.1. Back-end	25
3.2.1.1. Python.....	25
3.2.1.1.1. Flask.....	26
3.2.1.1.2. Librerías utilizadas.....	27
3.2.1.2. XAMPP	27
3.2.1.2.1. MariaDB/MySQL	28
3.2.2. Front-end.....	29
3.2.2.1. HTML5.....	29
3.2.2.2. CSS.....	30

3.2.2.2.1 Bootstrap.....	31
3.2.2.3 JavaScript	32
3.2.2.3.1 JQuery.....	33
3.3 Disposición final de la aplicación web.....	34
4. METODOLOGÍA E IMPLEMENTACIÓN	35
4.1 Ciclo de vida del proyecto.....	35
4.1.1 Ventajas y desventajas de los diferentes ciclos de vida.....	37
4.2 Diagrama de Gantt.....	38
4.3 Especificación de requisitos.....	39
4.4 Actores y casos de uso.....	41
4.4.1 Actores	41
4.4.2 Casos de uso.....	43
4.5 Diagramas de secuencia y actividad.....	49
4.5.1 Diagramas de secuencia.....	49
4.5.2 Diagramas de actividad.....	55
4.6 Diseño de la aplicación.....	64
4.6.1 Interfaz de usuario.....	64
4.6.2 Base de datos.....	70
4.7 Implementación	71
4.7.1 Instalación y configuración del framework Flask.....	71
4.7.2 Configuración de la base de datos.....	72
4.7.3 Búsqueda de datos de Twitter	72
4.7.4 Análisis de sentimientos en los tweets.....	73
4.7.5 Generación de estadísticas	74
4.7.6 Aspectos importantes a tener en cuenta	75
4.7.7 Código fuente.....	76
5. CONCLUSIONES Y PROPUESTAS FUTURAS.....	77
5.1 Conclusiones	77
5.2 Propuestas futuras.....	78
6. BIBLIOGRAFÍA.....	79
7. ANEXO: GUÍA DE USUARIO	82
7.1 Landing Page:.....	82
7.2 Identificación de usuario	83
7.2.1 Registro de usuario	83
7.2.2 Inicio de sesión	84
7.3 Página principal.....	84
7.4 Buscar tweets.....	85
7.4.1 Página de búsqueda.....	85
7.4.2 Resultados de la búsqueda	86
7.5 Análisis de sentimientos	87
7.5.1 Página de análisis.....	87
7.5.2 Resultados del análisis de sentimientos	88
7.6 Estadísticas	89
7.6.1 Página de estadísticas.....	89

7.6.2 Resultados de las estadísticas.....	90
7.7 Perfil de usuario y cierre de sesión.....	90
7.7.1 Perfil de usuario	91



Índice de figuras

Figura 1. Versiones de la API de Twitter	14
Figura 2. Sentiment classification techniques Source: Medhat et al., 2014	17
Figura 3. NLP Imagen de Freepik	18
Figura 4. Ejemplo de tokenización.	19
Figura 5. Fórmula del TF-IDF	22
Figura 6. Modelo-Vista-Controlador(MVC)	24
Figura 7. Python - Hola Mundo	25
Figura 8. App con Flask 1	26
Figura 9. App con Flask 2	26
Figura 10. MySQL-Connector en Python	28
Figura 11. MariaDB en PHPMyAdmin	28
Figura 12. Hola mundo HTML	29
Figura 13. Ejemplo CSS	30
Figura 14. Ejemplo CSS en HTML	30
Figura 15. Unir HTML con CSS	30
Figura 16. Ejemplo Bootstrap	31
Figura 17. Unir bootstrap con HTML	31
Figura 18. Ejemplo JavaScript	32
Figura 19. Ejemplo JavaScript en HTML	32
Figura 20. Unir JavaScript con HTML	32
Figura 21. JQuery ejemplo 1	32
Figura 22. JQuery ejemplo 2	32
Figura 23. Unir JQuery con HTML	33
Figura 24. Ciclo de vida en cascada	34
Figura 25. Ciclo de vida iterativo	34
Figura 26. Ciclo de vida en espiral	35
Figura 27. Diagrama de Gantt	37
Figura 28. Requisitos funcionales	39
Figura 29. Requisitos no funcionales	39
Figura 30. Usuario no registrado	41
Figura 31. Usuario registrado	42
Figura 32. Diagrama de secuencia C01	51
Figura 33. Diagrama de secuencia C02	51
Figura 34. Diagrama de secuencia C03	52
Figura 35. Diagrama de secuencia C04	52
Figura 36. Diagrama de secuencia C05	53
Figura 37. Diagrama de secuencia C06	53
Figura 38. Diagrama de secuencia C07(Flujo normal)	54
Figura 39. Diagrama de secuencia de C08.	54
Figura 40. Diagrama de secuencia de C089	55
Figura 41. Diagrama de actividad C01	56
Figura 42. Diagrama de actividad C02	57
Figura 43. Diagrama de actividad C03	58
Figura 44. Diagrama de actividad C04	59
Figura 45. Diagrama de actividad C05	60

Figura 46. Diagrama de actividad C06	61
Figura 47. Diagrama de actividad C07(Flujo normal)	62
Figura 48. Diagrama de actividad C08	63
Figura 49. Diagrama de actividad C09	64
Figura 50. Boceto Landing Page	65
Figura 51. Boceto inicio de sesión	66
Figura 52. Boceto registro de usuario	66
Figura 53. Boceto Página principal	67
Figura 54. Boceto Perfil	67
Figura 55. Boceto Buscar tweets	68
Figura 56. Boceto resultados búsqueda	68
Figura 57. Boceto análisis de sentimientos	69
Figura 58. Boceto resultado análisis	69
Figura 59. Boceto estadísticas	70
Figura 60. Boceto resultados estadísticas.	70
Figura 61. Modelo relacional de la base de datos	71
Figura 62. Config.py	72
Figura 63. App.py	72
Figura 64. __Init__.py	73
Figura 65. Configuración de la base de datos	73
Figura 66. Módulo de Twitter de SNScrape	74
Figura 67. Uso de SNScrape	74
Figura 68. Importación de NLTK	74
Figura 69. Descargas necesarias de NLTK	75
Figura 70. Uso de matplotlib	75
Figura 71. Landing Page	82
Figura 72. Registro de usuario	83
Figura 73. Inicio de sesión	83
Figura 74. Página principal	84
Figura 75. Buscar tweets 1	84
Figura 76. Buscar tweets 2	85
Figura 77. Resultado de búsqueda 1	85
Figura 78. Resultado de búsqueda 2	86
Figura 79. Análisis de sentimientos	87
Figura 80. Resultados del análisis de sentimientos	88
Figura 81. Estadísticas	89
Figura 82. Resultados de estadísticas	90
Figura 83. Desplegable perfil y cerrar sesión	90
Figura 84. Perfil de usuario	91

Índice de tablas

Tabla 1. Ventajas y desventajas de los diferentes ciclos de vida	37
Tabla 2. Plantilla de actores	40
Tabla 3. Plantilla de casos de uso	40
Tabla 4. Actor, usuario no registrado	41
Tabla 5. Actor, usuario registrado	41
Tabla 6. C01 - Registro de usuario	43
Tabla 7. C02 - Inicio de sesión	43
Tabla 8. C03 - Recuperación de contraseña	44
Tabla 9. C04 - Editar perfil	45
Tabla 10. C05 - Cerrar sesión	45
Tabla 11. C06 - Buscar tweets	46
Tabla 12. C07 - Analizar tweets	47
Tabla 13. C08 - Ver estadísticas	48
Tabla 14. C09 - Eliminar consultas	49



1. INTRODUCCIÓN

1.1. Motivación del trabajo

Las redes sociales se han convertido en una parte esencial de nuestra vida cotidiana, y se han establecido como una herramienta importante para el intercambio de información y de opinión. Twitter es una de las redes sociales más populares, donde millones de usuarios comparten información, noticias, opiniones y sentimientos a través de tweets. A medida que el volumen de datos en Twitter continúa aumentando, se ha vuelto más difícil analizar la opinión pública de manera efectiva. Debido a esto, he decidido realizar este TFG sobre la clasificación de sentimientos en tweets, ya que es una técnica que puede ayudar a extraer información valiosa de la opinión pública sobre diferentes temas y eventos en tiempo real.

La clasificación de sentimientos en tweets se refiere al proceso de identificar la polaridad emocional de un tweet, es decir, si el tweet expresa un sentimiento positivo, negativo o neutral. Esto se hace mediante el análisis de texto y la identificación de palabras clave, frases y expresiones que sugieren un tono emocional. Por ejemplo, puede ayudar a identificar si los usuarios de Twitter tienen una opinión mayoritariamente positiva o negativa sobre un producto o servicio, una política gubernamental o una noticia de última hora.

Se pueden encontrar varios beneficios al realizar una aplicación web para clasificar los sentimientos. Algunos de estos beneficios son:

- Análisis de la opinión pública en tiempo real
- Identificación de tendencias y patrones de opinión
- Comprender la opinión de los clientes
- Toma de decisiones basada en datos
- Contribuir a la investigación

En consecuencia, una herramienta como esta, podría ser muy valiosa para empresas, organizaciones y/o partidos políticos.

1.2 Objetivos del trabajo

El objetivo principal del trabajo es crear una aplicación web que sea capaz de analizar los sentimientos positivos o negativos que transmiten usuarios de Twitter mediante sus publicaciones por medio de tweets.

Como objetivos secundarios podemos definir:

- a) La aplicación permitirá el registro de usuarios y el análisis de sentimientos de los tweets buscados, así como un resumen sencillo de estadísticas de los tweets analizados.
- b) Las búsquedas podrán filtrarse por palabra clave, por rango de fechas y por respuestas de los tweets. El resumen de estadísticas de tweets analizados podrá resumirse en una gráfica.
- c) Se pretende así obtener una aplicación sencilla que permita el análisis de sentimientos de tweets para realizar pequeños estudios de mercado o simplemente conocer la positividad de la imagen de marca de una empresa.

1.3 Motivación personal

La decisión de realizar el TFG sobre este tema se debe a mi interés por la inteligencia artificial, el machine learning (aprendizaje automático), y por supuesto el desarrollo de aplicaciones web. Poder juntar todos estos campos en un solo proyecto era una muy buena oportunidad. Todo esto sumado a mi afición por las redes sociales y la curiosidad de saber qué piensa la gente con respecto a cualquier tema, han sido los detonantes para que finalmente me decidiera por este tema. Además, como se explica en el punto 1.1, es importante conocer la opinión de las personas sobre ciertos temas como, por ejemplo, estudios de mercado o análisis de datos.

2. ESTADO DEL ARTE

2.1 Emociones

A lo largo de la historia, han surgido muchas versiones de lo que podría significar una emoción, sin llegar todos a un consenso sobre la definición de esta. Según la RAE, una emoción es una alteración del ánimo intensa y pasajera, agradable o penosa, que va acompañada de cierta conmoción somática [1]. Sin embargo, se puede llegar a definir en términos generales, como un fenómeno complejo que involucra una serie de cambios fisiológicos, cognitivos, subjetivos y conductuales en respuesta a un estímulo o situación relevante para el individuo. La emoción se caracteriza por la activación de diversos sistemas del organismo, como el sistema nervioso o el endocrino, que producen cambios en la frecuencia cardíaca, o la presión arterial entre otros, así como están acompañados por experiencias subjetivas de intensidad, por ejemplo.

Además, existen varias teorías sobre la emoción, las más conocidas son:

- Teoría de las emociones básicas de Paul Ekman (1992).[2]
- Teoría relacional emocional, propuesta por James-Lange (James,1890-Lange,1885).[3][4]
- Teoría de la evaluación cognitiva propuesta por Lazarus (1991).[5]
- Teoría de los sistemas de procesamiento emocional de Ledoux (1996).[6]

Para la elaboración de este trabajo, se va a tener en cuenta la teoría de las emociones básicas de Paul Ekman, donde se encuentran las seis emociones básicas de Ekman, que son:

- Alegría
- Tristeza
- Miedo
- Ira
- Sorpresa
- Asco

Una vez se conocen lo que son las emociones, el proyecto se centrará en detectar los sentimientos, que son la interpretación que se hacen sobre las emociones.

2.2 Extracción de datos en Twitter

El primer paso para realizar la detección de sentimientos será la extracción de datos de Twitter, para así poder procesarlos y posteriormente detectar sus sentimientos. Para ello existen varias técnicas que van a ser explicadas a continuación.

2.2.1 API de Twitter

La forma clásica de extraer datos de Twitter, es mediante la API [7] REST de Twitter [8]. Para acceder a ella es necesario registrarse en su portal de desarrolladores: <https://developer.twitter.com/en>.

Una vez registrado, es necesario crear un proyecto y dentro de ese proyecto, una app. Una vez hecho esto, se nos proporcionará unas claves de acceso que habrá que utilizar en el código de Python, y a partir de ahí se obtiene acceso a utilizar todas las funcionalidades disponibles que ofrece la API para manipular datos de Twitter.

Según la funcionalidad que se necesite, existen diferentes planes sobre el uso de la API.

Free	Basic	Pro	Enterprise
For write-only use cases and testing the Twitter API	For hobbyists or prototypes	For startups scaling their business	For businesses and scaled commercial projects
<ul style="list-style-type: none">Rate limited access to v2 tweet posting and media upload endpoints1,500 Tweets per month - posting limit at the app level1 app IDLogin with TwitterFree	<ul style="list-style-type: none">Rate limited access to suite of v2 endpoints3,000 Tweets per month - posting limit at the user level50,000 Tweets per month - posting limit at the app level10,000 Tweets per month - read-limit rate cap2 app IDsLogin with Twitter\$100 per month	<ul style="list-style-type: none">Rate-limited access to suite of v2 endpoints, including search and filtered stream1,000,000 Tweets per month - GET at the app level300,000 Tweets per month - posting limit at the app level3 app IDsLogin with Twitter\$5,000 per month	<ul style="list-style-type: none">Commercial-level access that meets your and your customer's specific needsManaged services by a dedicated account teamComplete streams: replay, engagement metrics, backfill, and more featuresMonthly subscription tiers
Get started	Subscribe now	Subscribe now	Apply now

Figura 1. Versiones de la API de Twitter

2.2.2 Web Scraping

El web scraping [9], también conocido como extracción de datos web, es una técnica utilizada para obtener información estructurada de páginas web de manera automatizada. Básicamente consiste en escribir código o utilizar herramientas específicas para recopilar datos de sitios web de manera sistemática, generalmente a través del análisis del código HTML de las páginas web.

El proceso para realizar la extracción de datos es:

1. Identificar el objetivo: Primero se selecciona el sitio web y se buscan los elementos específicos de los que queremos la información.

2. Inspeccionar el sitio web: Posteriormente hay que examinar el código fuente HTML de las páginas para entender su estructura y donde se encuentran los datos a extraer. Para ello se puede utilizar la opción de inspeccionar elementos del navegador.
3. Seleccionar la técnica de extracción: Para elegir que técnica utilizar, puede ser mediante análisis directo del HTML, uso de librerías como BeautifulSoup, o SNScrape en Python, o uso de APIs.
4. Desarrollo del código: Lo próximo sería desarrollar el código en el lenguaje elegido, mediante el cual se extraen los datos, y se almacenan en la base de datos o en archivos como el csv.
5. Problemas asociados: A la hora de realizar la extracción, pueden surgir problemas como, por ejemplo, los CAPTCHAs u otras medidas de seguridad que tenga la página, y habrá que buscar la manera de solventarlos.

El web scraping es muy utilizado en temas como investigación o análisis de sentimientos entre otras opciones.

2.1.2.1 BeautifulSoup

Existen diversos métodos de web scraping, pero este es uno de los más utilizados. BeautifulSoup [10] es una librería de Python creada en 2004 para realizar este propósito, el web scraping, de una manera sencilla, y que cuenta con numerosas funciones que te permiten realizarlo.

Algunas de sus características más importantes son:

- BeautifulSoup provee varios métodos para navegar, buscar y modificar el árbol de elementos mediante herramientas que extraen los datos que necesites, y no se necesita mucho código para realizarlo.
- BeautifulSoup convierte automáticamente los datos recibidos a Unicode [11], y los datos salientes a UTF-8 [12], por lo que no hay que preocuparse por eso a no ser que el documento analizado no especifique el método de codificación, que entonces se deberá especificar manualmente.
- BeautifulSoup cuenta con importantes analizadores de Python como “lxml” o “html5lib”, que permiten utilizar diferentes estrategias de parseo o cambiar velocidad por flexibilidad.

Sin embargo, uno de los grandes inconvenientes que tiene BeautifulSoup es que no tiene compatibilidad con JavaScript, por lo que no detecta el contenido dinámico generado por JavaScript, y Twitter funciona de esta manera, cargando su contenido dinámicamente mediante este lenguaje.

2.1.2.2 SNScrape

SNScrape [13] es una herramienta de scraping web para Python especializada en redes sociales, con módulos independientes para cada red social. Es bastante sencilla de usar y tiene diversas opciones como buscar usuarios, tweets, hashtags, entre otras opciones. SNScrape y no es necesario registrarse en ninguna API ni utilizar credenciales.

Debido a su facilidad de uso, su flexibilidad, y su adaptación a las nuevas modificaciones de Twitter, que dificultan e incluso bloquean la opción de recopilar datos en Twitter, se convierte en una muy buena opción a tener en cuenta.

2.3 Análisis de sentimientos

2.3.1 Definición de sentimiento

Antes de hablar de la detección de sentimientos, hay que definir que es un sentimiento.[14]
Un sentimiento se podría definir como un estado de ánimo que se produce ante una estimulación externa. Esto ocurre cuando se procesa una emoción, que entonces da lugar al sentimiento. Dicho de otra manera, los sentimientos son emociones conceptualizadas que determinan el estado de ánimo. Además, cabe recalcar que los sentimientos pueden ser positivos o negativos. De ahí que se haya definido previamente el concepto de emoción, ya que a partir de las emociones, se extraen los sentimientos.

2.3.2 Concepto de análisis de sentimientos

El análisis de sentimientos [15] (o minería de opinión) es una técnica del procesamiento del lenguaje natural que tiene como objetivo identificar y clasificar las emociones y opiniones expresadas en los textos de manera automática. Consiste en determinar la polaridad de un texto, es decir, clasificar los sentimientos en positivos, negativos o neutros, o determinar sus emociones, como las 6 emociones básicas de Ekman, que son: alegría, tristeza, miedo, ira, sorpresa y asco.

2.3.3 Técnicas del análisis de sentimientos

Para realizar el análisis de sentimientos, existen diferentes técnicas [16]:

- Técnicas basadas en aprendizaje automático supervisado: Con esta técnica se pretende entrenar un modelo utilizando un conjunto de datos etiquetados, donde cada ejemplo de texto está asociado con una etiqueta de sentimiento conocida. Se extraen características del texto, como frecuencias de palabras, n-gramas o características lingüísticas, y se utilizan como entradas para el modelo.
Dentro de este tipo de técnica encontramos varios algoritmos de clasificación como árboles de decisión, SVM (Support Vector Machines), Redes neuronales, o Naive Bayes entre otras.

Mediante estas técnicas, el rendimiento del modelo depende en gran parte de la calidad del conjunto de datos etiquetados y de las características seleccionadas.

- Técnicas basadas en aprendizaje automático no supervisado: Esta técnica se utiliza principalmente para descubrir patrones y estructuras en los datos de texto no etiquetados.
- Técnicas basadas en léxico: Este tipo de técnicas es muy utilizado en el análisis de sentimientos y consiste en utilizar diccionarios o léxicos de palabras previamente etiquetadas con su polaridad (positiva, negativa o neutra). Cada palabra en el texto se compara con el léxico y se asigna una puntuación en función de su polaridad. Una vez comparadas todas las palabras, se hace una media o se suman las puntuaciones y así se obtiene la polaridad, es decir, el sentimiento del texto.

Hay varios ejemplos de léxicos bastante utilizados, como puede ser SentiWordNet, Opinion Lexicon, AFINN o VADER entre otros.

Sin embargo, este tipo de métodos puede llegar a tener limitaciones, como la falta de contexto, sarcasmo o ironía, entre varios problemas.

En la siguiente imagen se muestra los diferentes métodos de análisis de sentimientos:

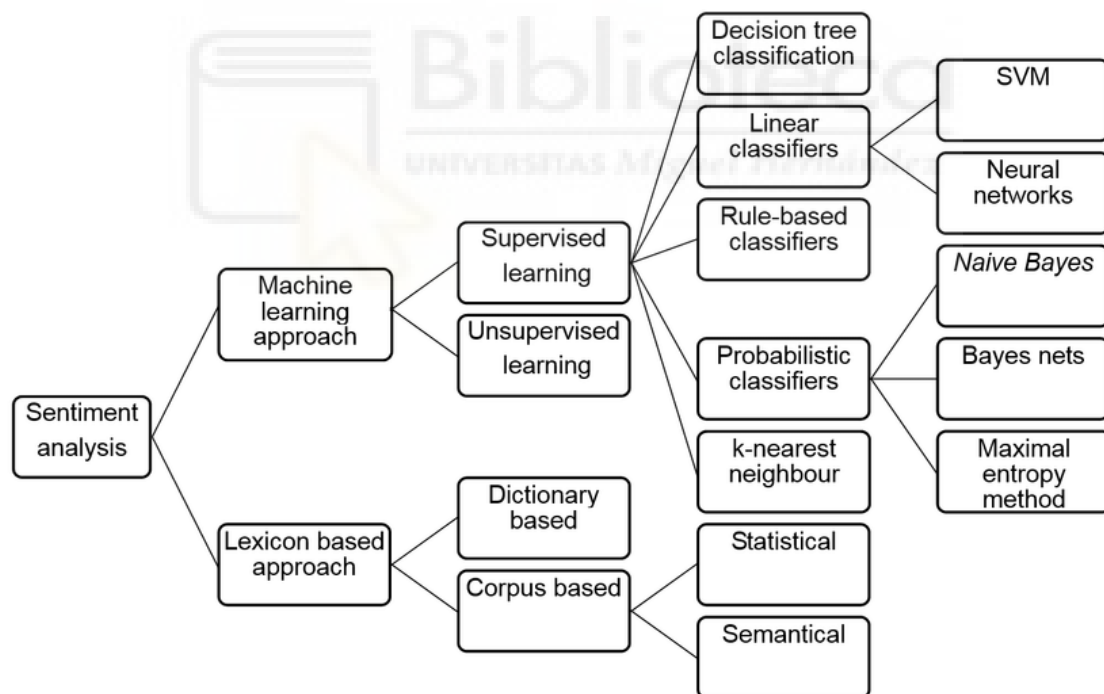


Figura 2. Sentiment classification techniques Source: Medhat et al., 2014

2.4 Procesamiento del lenguaje natural

Uno de los pasos más importantes a la hora de detectar sentimientos, es el procesamiento del lenguaje natural (PLN o NLP)[17].

El procesamiento del lenguaje natural es una rama de la informática, concretamente de la inteligencia artificial [18] que se encarga de brindar a las computadoras la capacidad de interpretar, manipular y comprender el lenguaje humano.

El NLP combina la lingüística computacional con modelos estadísticos, de machine learning y deep learning. Mediante estas tecnologías, es como el ordenador procesa el lenguaje humano. Actualmente, el PLN es muy utilizado, y se puede ver en el día a día, por ejemplo, en los GPS, para la síntesis de voz, los chatbots, asistentes digitales, etc. Y cada vez más, se ve en el ámbito empresarial.

Nuestro lenguaje, está lleno de ambigüedades, sarcasmos, homónimos, metáforas y diversos recursos lingüísticos que un ordenador no puede detectar, por eso nace el NLP, para poder detectar todo esto, y a raíz de eso, han surgido muchas aplicaciones para el NLP, como:

- Reconocimiento de voz
- Traducción automática
- Resumen y clasificación de textos
- Análisis de tendencias
- Desambiguación del sentido de la palabra
- Reconocimiento de entidades nombradas
- Análisis de sentimientos.

Como ya hemos mencionado, en este trabajo nos centraremos en el análisis de sentimientos.



Figura 3. NLP Imagen de Freepik

2.4.1 Preprocesamiento de texto para NLP

A la hora de procesar el texto para posteriormente realizar el análisis de sentimientos, lo que nos interesa es que haya el menor número de palabras posibles que analizar, es decir, realizar diferentes técnicas que eliminen el contenido que no aporte nada, y quedarnos solo con el contenido útil, para eso tenemos varias técnicas, así que a continuación vamos a explicar las más comunes.[19][20]

2.4.1.1 Limpieza del texto

Como ya sabemos, para obtener el texto de una web, en este caso Twitter, se utiliza web scraping, y así eliminamos las etiquetas HTML, sin embargo, eso no es todo lo que hay que hacer para limpiar el texto, puesto que debemos eliminar primero los caracteres especiales o números. Para ello tenemos la limpieza del texto, que sería un método muy simple por el que podemos eliminar estos símbolos sustituyéndolos por espacios, por ejemplo.

2.4.1.2 Tokenización

Una de las técnicas de PLN más utilizadas es la tokenización, que consiste en segmentar el texto en palabras o grupos de palabras, llamadas tokens. Todo esto se utiliza para luego facilitar su procesamiento posterior.

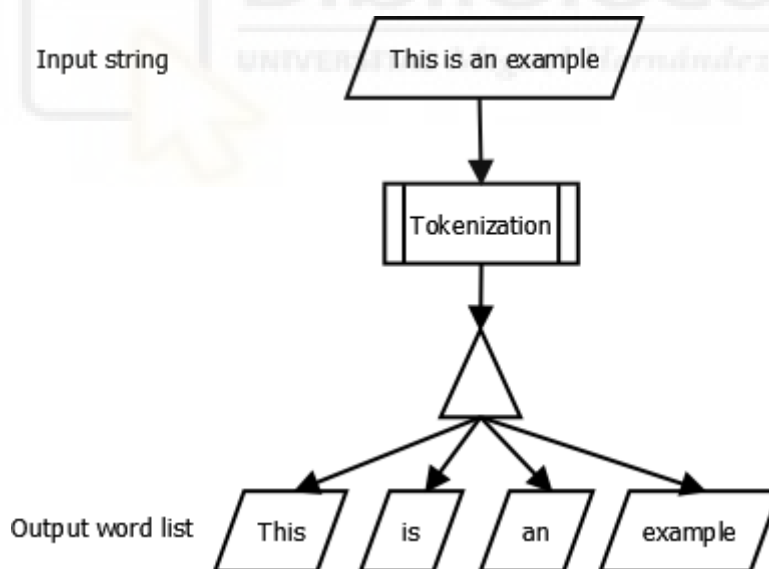


Figura 4. Ejemplo de tokenización.

2.4.1.3 Normalización de las palabras

El siguiente paso es realizar la normalización, es decir, simplificar todavía más el texto. Para ello tenemos diferentes técnicas. Una de ellas es convertir todo a minúsculas o a mayúsculas, generalmente a minúsculas. Otra técnica sería la corrección ortográfica, es decir, eliminar tildes.

2.4.1.4 Eliminación de stop words

Una vez hecha la normalización, una técnica muy importante que eliminaría aún más palabras es la eliminación de stopwords, que consiste en eliminar palabras comunes pero que no aportan nada, como artículos, o preposiciones, como “el” o “de”. Normalmente se suelen utilizar listas predefinidas que contienen estas stopwords, pero también se puede realizar una lista personalizada según las necesidades.

2.4.1.5 N-gramas

No solo existen los tokens de una palabra, que es lo que habíamos visto hasta ahora, sino que también tenemos tokens formados por varias palabras. Como sabemos, en nuestro lenguaje existen frases que pueden contener nombres de personas, lugares, o nombres compuestos como “sala de estar” o “mesita de noche”.

Estas expresiones tienen un significado diferente cuando se usan juntas, es decir, si se usaran por separado significarían otra cosa totalmente distinta y al preprocesarlas se perdería el significado.

Por eso, se utilizan estos tokens que guardan la expresión entera, y así su significado. Entonces la técnica utilizada es la de agrupar estos términos en n-gramas. Aquí encontramos unigramas, para una sola palabra, bigrama, para dos palabras, o trigramas para 3 palabras. Realmente existen n-gramas más grandes, pero en NLP se suelen utilizar solo hasta trigramas.

2.4.1.6 Stemming

El Stemming es otro de los métodos más utilizados en NLP, y consiste en reducir las palabras a su raíz, como indica su nombre en inglés (stem), así acotamos todavía más el lenguaje y será más fácil su posterior detección de sentimientos.

Por ejemplo, tenemos las palabras: caminando, caminar, camino. Estas palabras comparten la raíz: camin.

Un detalle a tener en cuenta es que la raíz no tiene por qué ser una palabra en sí.

Mediante esta técnica, encontramos problemas como el over-stemming y el under-stemming. El primero sucede cuando se han encontrado palabras con la misma raíz que deberían tener raíces diferentes, puesto que no tienen nada que ver una con la otra. El segundo problema sucede cuando dos palabras que deberían tener la misma raíz tienen raíces diferentes.

2.4.1.7 Lematización

Es un método muy similar al stemming, pero en este caso, se reducen sus palabras a su forma base, a su lema, considerando el contexto gramatical en el que se encuentra dicha palabra. Teniendo en cuenta esto, hay que saber que una misma palabra tenga un lema distinto según el contexto en el que se encuentre.

Para entender mejor este concepto, aquí tenemos un ejemplo:

- Texto original: Los perros están ladrando en el parque.
- Lematización: El perro ladrar en el parque.

2.4.2 Representación numérica del texto

Una vez realizadas todas las técnicas de preprocesamiento del texto, el siguiente paso es convertir el texto a números o vectores, para que pueda ser interpretado y utilizado por el algoritmo de aprendizaje automático. Sin embargo, para realizar esto se deben utilizar técnicas específicas que al convertir ese texto en números no se pierda información. A continuación, hablaremos de algunas de las técnicas más utilizadas en este sector.

2.4.2.1 Bolsa de palabras (Bag of words)

En esta técnica, se crea un vector que representa la frecuencia de cada palabra en un corpus completo. Cada coordenada del vector corresponde a una palabra del corpus, y el valor en el vector es la frecuencia. Cabe destacar que esta técnica sólo aporta información sintáctica, y no semántica.

En resumen, esta técnica simplemente sirve para contar las palabras.

Sin embargo, esta técnica está bastante limitada debido a que, como se ha mencionado, solo se basa en el recuento de palabras y no tiene en cuenta su significado o contexto:

- Este modelo reduce la importancia de los tokens que aparecen pocas veces, y que un token este pocas veces no significa que no sea importante.
- Mediante esta técnica no se tiene en cuenta la semántica, es decir, su significado.
- Por último, si tenemos demasiados tokens, este modelo puede fallar.

2.4.2.2 TF-IDF

El TF-IDF (Term Frequency - Times Inverse Document) es una técnica utilizada para medir la relevancia de una palabra en un documento de una colección de documentos o corpus. Básicamente mide la frecuencia con la que aparece un término dentro de un documento, y lo compara con la frecuencia con la que aparece en el corpus, y así obtiene la relevancia de dicho término en ese documento.

Por ejemplo, si vemos que la palabra “el” aparece con mucha frecuencia en un documento, no significa nada, puesto que, en el resto del corpus, también aparece con mucha frecuencia, sin embargo, si encontramos que la palabra “pelota” aparece frecuentemente en un documento, pero no en el resto del corpus, significa que esa palabra es relevante.

El valor del TF-IDF se obtiene multiplicando las medias del valor TF y el valor IDF. Esto se obtiene de las siguientes fórmulas

$$TF = \frac{\text{Number of times a word "X" appears in a Document}}{\text{Number of words present in a Document}}$$

$$IDF = \log \left(\frac{\text{Number of Documents present in a Corpus}}{\text{Number of Documents where word "X" has appeared}} \right)$$

$$TF\ IDF = TF * IDF$$

Figura 5. Fórmula del TF-IDF

Por tanto, un valor alto en esta medida indica una frecuencia alta de un término en el documento y la frecuencia de ese término en el corpus es baja.

Sin embargo, aunque este modelo mejore al anterior, sigue teniendo problemas similares como, por ejemplo, que no tiene en cuenta la semántica o el contexto, y que sigue dando problemas cuando encontramos cantidades de datos muy grandes.

2.4.2.3 Word Embeddings

Hasta ahora, hemos visto modelos que se enfocan en el aspecto sintáctico, al simplemente contar palabras, lo que conlleva ciertas limitaciones en los modelos, ya que es importante fijarse también en el contexto de una palabra o su significado. Por eso tenemos este método, que se encarga de la parte semántica de las palabras.

El word embedding es una técnica que consiste en asignar un vector numérico a cada palabra en un cierto espacio vectorial. La idea es que palabras que tengan significados parecidos tengan asociado un vector cercano en el espacio vectorial. Esto se hace para capturar la semántica y las relaciones entre palabras basándose en su posición en el espacio. Esta técnica se realiza mediante algoritmos de aprendizaje automático o mediante redes neuronales.

Este método presenta varias características bastante interesantes:

- Semántica de palabras: Los word embeddings obtienen la semántica de las palabras en función de cómo se relacionan con otras palabras en el espacio.
- Relación entre palabras: Los word embeddings pueden capturar relaciones lingüísticas entre palabras, como por ejemplo sinonimia o antonimia, o relaciones en el contexto. Un ejemplo sería la diferencia entre los vectores “rey” y “reina”, que es similar a la diferencia entre los vectores “hombre” y “mujer”.

- Operaciones algebraicas: Los vectores pueden someterse a operaciones algebraicas como sumas o restas, para descubrir relaciones entre palabras. Por ejemplo: “Rey” - “Hombre” + “Mujer” \approx “Reina”.
- Generalización: Este método también tiene la capacidad de generalizar, para palabras que no han sido observadas durante el entrenamiento. Por ejemplo, si el modelo ha aprendido la relación entre “perro” y “animal”, podrá determinar la relación “gato” y “animal”, aunque esa relación no se haya dado en el entrenamiento.

Los word embeddings se han vuelto muy populares en NLP debido a su capacidad para capturar el significado de las palabras y mejorar el rendimiento de tareas como el análisis de sentimientos.

Existen varios modelos preentrenados de word embeddings, como por ejemplo, uno de los más famosos, por no decir el más famoso, “Word2Vec”, publicado en 2013 por Mikolov en su paper “Efficient Estimation of Word Representations in Vector Space”[21].



3. ARQUITECTURA DE LA APLICACIÓN WEB

A la hora de desarrollar una aplicación web hay que tener en cuenta muchos factores, ya que el diseño de la aplicación es algo fundamental en cualquier proyecto, ya que se determina la estructura, el comportamiento y la usabilidad de la aplicación. En este caso, una aplicación web bien diseñada puede ayudar a los usuarios a comprender de una mejor manera los resultados del análisis de sentimientos.

Por eso mismo, ahora se va a describir el diseño de esta aplicación web para la detección de sentimientos en tweets.

En primer lugar, necesitamos saber qué arquitectura se va a utilizar. En este caso, se utilizará el patrón modelo-vista-controlador (MVC) [22].

3.1 Modelo-vista-controlador

Este patrón de arquitectura se distingue porque se encarga de separar en 3 componentes muy claros y definidos, un proyecto software, principalmente aplicaciones web. Básicamente está dividido en:

- **Modelo:** El modelo es la representación de la información y los datos que utiliza la aplicación, es decir, el modelo es el responsable de la lógica de negocio y el acceso a datos. Aquí es donde se maneja la base de datos.
- **Vista:** La vista es la representación visual de la información y los datos del modelo, es decir, la vista es la responsable de la presentación de la información del usuario y de la interacción con el usuario. Aquí es donde se utiliza el HTML y relacionados.
- **Controlador:** Por último, el controlador básicamente es un intermediario entre el modelo y la vista. En general, el controlador es responsable de manejar las solicitudes de usuario y actualizar el modelo y la vista correspondiente.

Para el funcionamiento de esta arquitectura, el usuario interactúa con la vista, y ésta envía solicitudes al controlador. Entonces el controlador procesa las solicitudes y actualiza el modelo y la vista.

La ventaja de este patrón es que separa las funcionalidades en 3 partes bien definidas, y eso facilita su modificación y mantenimiento, ya que se puede modificar una parte sin tener que cambiar el resto.

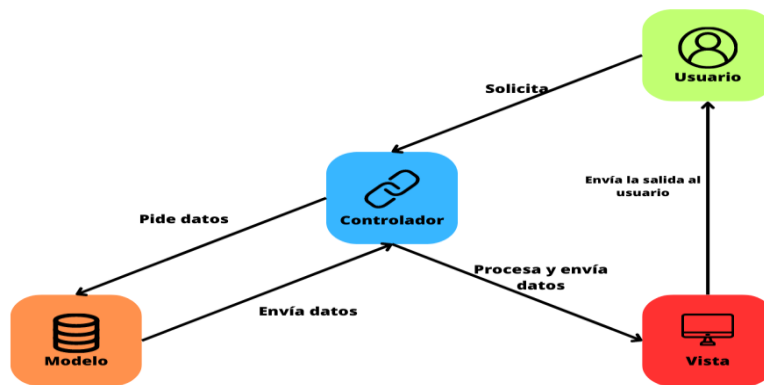


Figura 6. Modelo-Vista-Controlador(MVC)

3.2 Lenguajes y herramientas utilizados

A la hora de hacer una aplicación web, uno de los factores más importantes, es seleccionar correctamente en qué lenguaje de programación se va a realizar. En este caso, como lenguaje principal, se va a utilizar Python. Como se ha mencionado anteriormente, la aplicación se ha realizado con la arquitectura modelo-vista-controlador (MVC), sin embargo, para la distinción en todos los lenguajes, herramientas, frameworks y librerías utilizadas, se va a dividir todo entre front-end y back-end.

3.2.1 Back-end

El back-end es la parte lógica de la página web. Donde se gestiona toda la información, se encarga de que todo funcione correctamente, de la seguridad y de la optimización de los recursos. Esta es la parte que el usuario final no puede ver.[23]

Concretamente, el back-end contiene:

- La conexión a la base de datos
- Lenguajes de programación
- Lógica de la web
- Seguridad de la web
- Servidor web

3.2.1.1 Python

Python [24] es un lenguaje de programación interpretado de alto nivel. Es conocido por su facilidad de uso, su simpleza en la sintaxis, y su legibilidad. Y se suele utilizar para muchos tipos de proyectos, como aplicaciones web, aplicaciones científicas, inteligencia artificial, automatización de tareas, etc. Python es también muy extensible, lo cual significa que se puede agregar funcionalidades adicionales a través de módulos y paquetes de terceros.

Además, Python se caracteriza por ser de código abierto y poseer una gran comunidad de desarrolladores que mejoran continuamente el lenguaje.

```
>>> print("hola mundo")
hola mundo
>>>
```

Figura 7. Python - Hola Mundo

3.2.1.1.1 Flask

Python, es un lenguaje mediante el cual se pueden realizar aplicaciones web, sin embargo, para realizarlas, es necesario utilizar un framework que permita realizar dicha aplicación. Existen varios frameworks compatibles con python, siendo los más populares Django o Flask.

Para este proyecto se ha optado por utilizar Flask [25].

Flask es un framework de Python, que permite crear aplicaciones web de manera rápida y sencilla. Está considerado como un micro framework porque no requiere herramientas ni librerías particulares y permite construir aplicaciones con una cantidad mínima de código, pero a su vez, es muy flexible y escalable.

Con este framework, se puede manejar la creación de rutas, las conexiones a bases de datos, las cookies y sesiones, la autenticación y la autorización de usuario, el envío de mails, mensajes de error, uso de APIs, entre muchas otras opciones.

```
1  from flask import Flask
2
3  app = Flask(__name__)
4
5  @app.route("/")
6  def hello():
7      return "¡Hola, mundo!"
8
9  if __name__ == "__main__":
10     app.run()
```

Figura 8. App con Flask 1

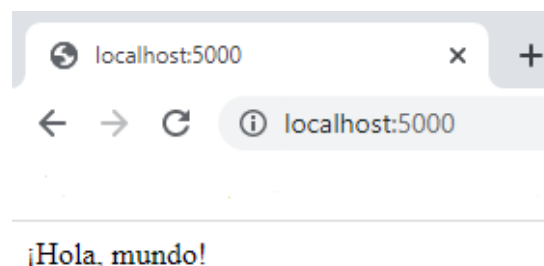


Figura 9. App con Flask 2

3.2.1.1.2 Librerías utilizadas

Para desarrollar una buena aplicación web, muchas veces no basta simplemente con utilizar Python, con su correspondiente framework. Normalmente es necesario utilizar otras herramientas, como en este caso las librerías, para poder ayudarnos de estas, y simplificar mucho el proyecto, obteniendo mejores resultados.

Por tanto, estas son las librerías más relevantes que se han utilizado para este proyecto:

- Flask-login: Es una extensión de Flask, que proporciona funcionalidades para gestionar la autenticación y la sesión de los usuarios en una aplicación web.
- Flask-Paginate: Es otra extensión de Flask, que ofrece herramientas para implementar la paginación de manera muy sencilla en una aplicación web.
- MySQL-Connector: Es una herramienta de Python que permite conectarse y comunicarse con una base de datos MYSQL. Permite todas las funcionalidades que ofrecen las bases de datos de una manera muy sencilla.
- SnScrape: Es una biblioteca de Python que permite extraer datos públicos de las redes sociales. Además, está especialmente diseñada para extraer datos de las redes sociales, y contiene módulos de cada red social, incluida Twitter.
- NLTK: Es una biblioteca de Python que facilita el preprocesamiento del lenguaje natural, y que incluye un módulo de análisis de sentimientos de VADER.
- Matplotlib: Es una herramienta de Python que permite crear diferentes tipos de gráficos a partir de datos. Es la más conocida y utilizada para este fin.

Esas son las principales librerías utilizadas para este proyecto, evidentemente se han utilizado más, pero con fines menores y menor relevancia.

3.2.1.2 XAMPP

Como toda aplicación web, esta debe tener un servidor donde alojarse, para poder utilizarse. En este caso, se ha optado por XAMPP [26].

XAMPP es un paquete de software libre que proporciona un entorno de servidor web completo y fácil de instalar y de utilizar. Se puede utilizar tanto en Windows, como en Linux y en Mac OS X.

Como su nombre indica, XAMPP está compuesto por un servidor apache, mariaDB, PHP y Perl:

- X: Multiplataforma (cross-platform)
- A: Apache
- M: MariaDB
- P: PHP
- P: Perl

Con esto, XAMPP permite a los desarrolladores probar su aplicación de manera local antes de publicarlas en un servidor web en línea.

Además, esta herramienta tiene un panel de control con el que iniciar o detener fácilmente los servicios de Apache y la base de datos, y ajustar la configuración del servidor web.

3.2.1.2.1 MariaDB/MySQL

Toda aplicación necesita un servidor, y por supuesto una base de datos, y en este caso, se ha elegido MariaDB [27], que viene integrada con XAMPP, y se puede manejar muy fácilmente desde el panel de PHPMyAdmin.

MariaDB es un sistema de gestión de base de datos relacional, de código abierto que proviene de MySQL. Fue desarrollado por la comunidad de MySQL cuando este fue adquirido por Oracle Corporation. Debido a su relación con MySQL, ambos son altamente compatibles, sin embargo, MariaDB proporciona mejoras, con respecto a MySQL, en el rendimiento, la escalabilidad y mayor seguridad.

MariaDB utiliza el lenguaje SQL para utilizar la base de datos, y como es evidente, es compatible con Python. Sin embargo, se va a utilizar el conector en Python de MySQL puesto que es más sencillo, sin embargo, la base de datos que es MariaDB, que es la que viene en XAMPP. No hay ningún problema con esto, pues como ya se ha mencionado, son compatibles entre sí.

```
13 import mysql.connector
14
15 db = mysql.connector.connect(
16     host="localhost",
17     user="usuario",
18     password="password",
19     database="basededatos"
20 )
21
22 cursor = db.cursor()
23
24 cursor.execute("SELECT * FROM tabla")
25
26 result = cursor.fetchall()
27
28 for x in result:
29     print(x)
```

Figura 10. MySQL-Connector en Python

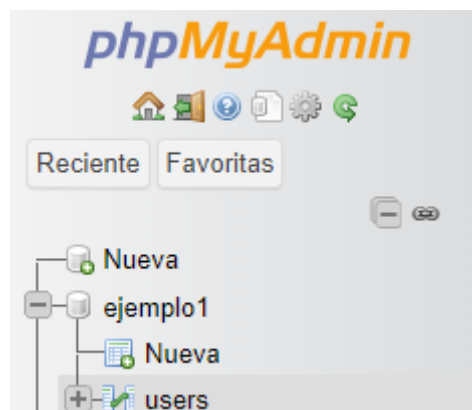


Figura 11. MariaDB en PHPMyAdmin

3.2.2 Front-end

El front-end es la parte de la aplicación que se ejecuta en el navegador del usuario, es decir, la parte visual, la interfaz, la parte donde el usuario interactúa y se comunica con el servidor [28]. Principalmente, el front-end se desarrolla con lenguajes como HTML, CSS y Javascript.

3.2.2.1 HTML5

Lo primero que hay que tener en cuenta a la hora de diseñar un front-end, es su estructura, es decir, el texto, las imágenes, u otros elementos, para ello, tenemos un lenguaje, HTML[29]. HTML (Hipertext Markup Language/Lenguaje de Marcas de hipertexto), es el lenguaje estándar para crear páginas web. Como ya se ha mencionado, proporciona la estructura del contenido, y define cómo se muestran estos elementos al usuario en el navegador web. Todo esto se hace mediante las etiquetas que tiene este lenguaje. Algunas de estas son: <head>, <body>, <article>, <p>, <div>, <footer>, etc. Con esto es posible dividir el contenido en diferentes secciones y tenerlo todo organizado.

Para explicar el funcionamiento de este lenguaje podríamos decir que las etiquetas HTML, como las previamente mencionadas, son interpretadas por el navegador web, que utilizan la información proporcionada por el código HTML para mostrar la página en la pantalla.

En este caso, estamos utilizando HTML5, que es la quinta versión de este lenguaje, que fue introducida en 2014, y trajo varias mejoras importantes con respecto a las otras versiones de HTML, como nuevas etiquetas, para mejorar la organización, soporte para video, para audio y capacidad de crear aplicaciones web más interactivas y dinámicas.

A screenshot of a code editor showing the HTML5 code for a simple web page. The code is displayed on a dark background with light-colored text. The code includes a DOCTYPE declaration, an HTML tag with a language attribute, a head section with meta tags for charset, http-equiv, and viewport, and a title tag. The body section contains a single paragraph with the text "Hola mundo".

```
<> prueba.html > html
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10  <p>Hola mundo</p>
11 </body>
12 </html>
```

Figura 12. Hola mundo HTML

3.2.2.2 CSS

Una vez se tiene la estructura del documento, el HTML, es el momento de darle formato al documento, es decir, hacerlo visualmente bonito, para ello usaremos CSS. CSS (Cascading Style Sheets) es un lenguaje diseñado para darle formato al HTML y describir su presentación

visual [30]. Este proporciona reglas y estilos que se aplican a los elementos HTML, y mediante esto podemos cambiar colores, tipografía, ubicación de los elementos, espaciado, etc.

```
1 #titulo {
2     color: blue;
3     font-size: 24px;
4 }
5
6 .parrafo {
7     color: green;
8     font-family: Arial, sans-serif;
9 }
10
```

Figura 13. Ejemplo CSS

Para utilizar estos estilos, hay que incluirlos en el HTML mediante atributos que se añaden a los elementos. Estos atributos pueden ser clases, para un uso más genérico, o IDs, para un uso específico en dicho elemento.

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4     <title>Ejemplo de CSS con Clases y IDs</title>
5     <link rel="stylesheet" type="text/css" href="styles.css">
6 </head>
7 <body>
8     <h1 id="titulo">Hola mundo!</h1>
9     <p class="parrafo">Este es un ejemplo de CSS con clases y IDs.</p>
10 </body>
11 </html>
12
```

Figura 14. Ejemplo CSS en HTML

Además, se debe tener en cuenta que para utilizar estos estilos, no basta con tener el documento CSS y el documento HTML, es necesario que estén unidos. Para ello, en el documento HTML habrá que incluir lo siguiente:

```
<link rel="stylesheet" type="text/css" href="styles.css">
```

Figura 15. Unir HTML con CSS

3.2.2.2.1 Bootstrap

CSS no es difícil de aprender, sin embargo, construir una web bien detallada y estructurada puede llegar a ser algo más complicado, aunque no imposible. Por tanto, existen diversos frameworks, que facilitan mucho esta tarea de diseñar una página web bien hecha, como puede ser Bootstrap [31]. Bootstrap es un framework de diseño front-end de código abierto utilizado

para crear interfaces de usuario responsivas y modernas. Mediante bootstrap se tienen muchos componentes, estilos y utilidades predefinidos que simplemente hay que añadir, no diseñar, con lo cual el trabajo se facilita mucho.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de Bootstrap</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/b
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/boo
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <div class="container">
      <a class="navbar-brand" href="#">Mi Sitio</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="#">Inicio</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="#">Acerca de</a>
          </li>
        </ul>
      </div>
    </div>
  </nav>
```

Figura 16. Ejemplo Bootstrap

Al igual que ocurre con el CSS, para utilizar bootstrap es necesario unirlo al HTML, para ello, son necesarias las siguientes líneas de código:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/css/bootstrap.min.css" rel="
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha3/dist/js/bootstrap.bundle.min.js"
```

Figura 17. Unir bootstrap con HTML

Como se observa en la imagen, no está el código entero, para obtener estas líneas simplemente hay que acceder a la web de Bootstrap: <https://getbootstrap.com/>, y ahí se encuentran, junto con toda la información necesaria para utilizarlo.

3.2.2.3 JavaScript

Normalmente es necesario agregar algo de interactividad y funcionalidad a la web, para ello, es necesario utilizar JavaScript [32]. JavaScript es un lenguaje de programación de alto nivel que se utiliza principalmente para el desarrollo web, al añadir interactividad y funcionalidad al HTML. Al ser un lenguaje interpretado, se ejecuta en tiempo real en el navegador sin necesidad de compilarlo.

Algunas de las características más destacables de JavaScript, son:

- Interactividad en el cliente
- Manipulación del DOM (Modelo de objeto del documento)
- Programación orientada a objetos
- Bibliotecas y frameworks
- Comunicación con el servidor mediante tecnologías como Ajax.

```
function holaMundo() {
    alert('Hola Mundo');
}
```

Figura 18. Ejemplo JavaScript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <script src="prueba.js"></script>
  <title>Hola mundo</title>
</head>
<body>
  <h1>Hola mundo</h1>
  <button onclick="holaMundo()">Haz clic aquí</button>
</body>
</html>
```

Figura 19. Ejemplo JavaScript en HTML

Al igual que con el CSS, también hay que unir el JavaScript al HTML, esto se hace de la siguiente manera:

```
<script src="prueba.js"></script>
```

Figura 20. Unir JavaScript con HTML

3.2.2.3.1 JQuery

A la hora de utilizar javascript, hay herramientas que facilitan mucho su uso, una de ellas es JQuery [33]. JQuery es una biblioteca de JavaScript que simplifica la manipulación del HTML. Básicamente fue creado para simplificar las complejidades e inconsistencias de JavaScript, ya que abstrae muchos detalles de bajo nivel que facilitan su uso con respecto a JavaScript.

```
$(document).ready(function() {
    // Seleccionar el elemento con el id "elemento" y aplicarle una animación
    $("#elemento").animate({left: '250px'}, 2000);
});
```

Figura 21. JQuery ejemplo 1

```
<div id="elemento"></div>
```

Figura 22. JQuery ejemplo 2

Además, al igual que ocurre con Bootstrap, debemos unirlo al HTML con un enlace:

```
<script src="https://code.jquery.com/jquery-3.7.0.js"
  integrity="sha256-JlqSTELeR4TLqP00G9dxM7yDPqX1ox/HfgiSLBj8+kM=" crossorigin="anonymous">
</script>
```

Figura 23. Unir JQuery con HTML

3.3 Disposición final de la aplicación web.

En este apartado, simplemente se va a ver cómo se distribuirá la aplicación web después de haber visto todos los lenguajes de programación que se van a utilizar, y el patrón modelo-vista-controlador.

Modelo

- Python
- MariaDB/MySQL
- Flask

Vista

- HTML
- CSS
- Bootstrap
- JS

- Jquery

Controlador

- Python
- Flask



4. METODOLOGÍA E IMPLEMENTACIÓN

En esta sección se va a desarrollar la metodología utilizada para este proyecto, así como por qué se han utilizado estas técnicas y no otras. Se hablará sobre el ciclo de vida, la especificación de requisitos, los casos de uso, diagramas, el diseño de la aplicación, y su implementación.

4.1 Ciclo de vida del proyecto

El primer paso a la hora de realizar un proyecto es definir su ciclo de vida [34].

El ciclo de vida de un proyecto es la serie de fases o etapas por las que pasa un proyecto desde el inicio hasta el final. Estas etapas están diseñadas para guiar el desarrollo y ejecución del proyecto y pueden variar según el enfoque utilizado. Existen diferentes tipos de ciclo de vida según las características y necesidades del proyecto.

Estas son los más comunes y utilizados:

- Ciclo de vida en cascada: Es el más común y simple, tiene un enfoque lineal y secuencial, donde cada fase se completa antes de pasar a la siguiente etapa. Este tipo de ciclo de vida es útil para proyectos bien definidos desde el comienzo, lo que significa que es poco flexible para realizar cambios una vez comenzado.

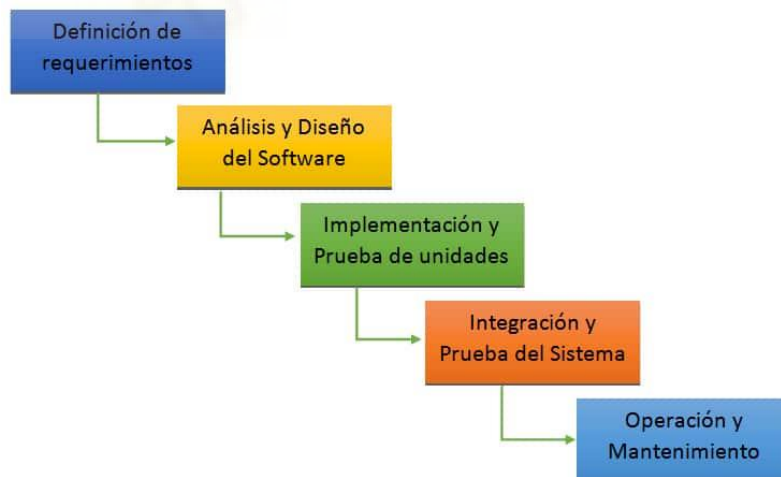


Figura 24. Ciclo de vida en cascada

- Ciclo de vida iterativo: En este enfoque, el proyecto se divide en iteraciones, cada una de las cuales sigue un ciclo de vida en miniatura. Cada iteración incluye la planificación, análisis, diseño, implementación y pruebas. A medida que se completa cada iteración, se obtiene retroalimentación y se realizan ajustes para las siguientes iteraciones. Este

modelo es útil cuando los requisitos son difíciles de definir al principio y es posible que cambien con el tiempo.

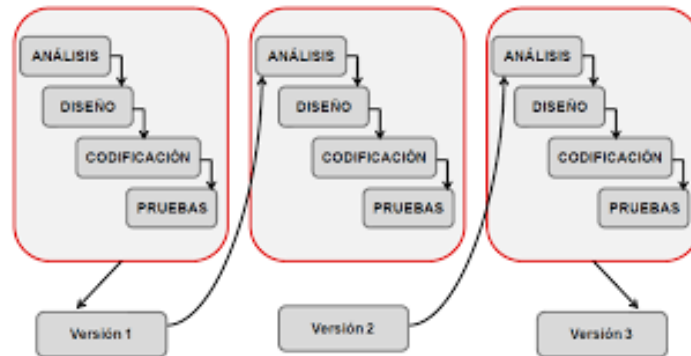


Figura 25. Ciclo de vida iterativo

- Ciclo de vida en espiral: Este modelo incluye elementos del enfoque en cascada y del enfoque iterativo, en él, se divide el proyecto en ciclos más pequeños conocidos como espirales, donde en cada espiral incluye la planificación, el análisis de riesgo, el desarrollo y la evaluación. Cada ciclo amplía y mejora el trabajo realizado en los ciclos anteriores. Es adecuado para proyectos grandes y complejos donde el riesgo es alto.

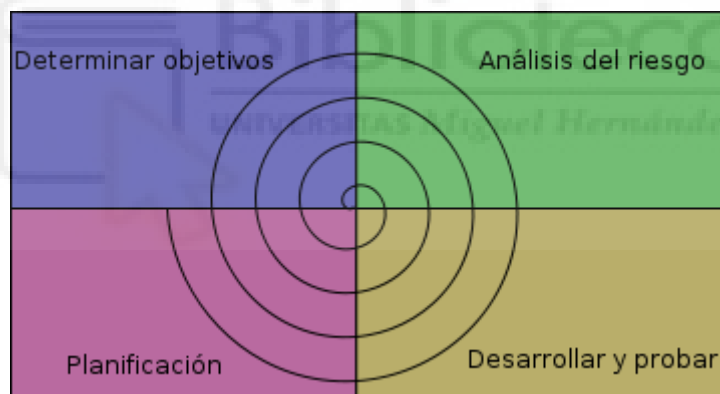


Figura 26. Ciclo de vida en espiral

- Ciclo de vida ágil: El enfoque ágil se basa en principios como la revisión con el cliente cada poco tiempo, y la capacidad de respuesta al cambio. Es decir, promueve un desarrollo fácil y rápido. Se basa en iteraciones cortas y frecuentes llamadas "sprints". Algunas de las técnicas más populares son Scrum, Kanban o Extreme Programming

4.1.1 Ventajas y desventajas de los diferentes ciclos de vida.

Como ocurre cada vez que toca hacer una elección entre varias opciones, hay que evaluar las ventajas y desventajas de todas las opciones, así que se va a realizar una evaluación para seleccionar el tipo de ciclo de vida según las necesidades y características del proyecto.

Tipo de ciclo de vida	Ventajas	Desventajas
En cascada	<ul style="list-style-type: none"> - Estructura y secuencia clara. - Fácil de entender y seguir. - Documentación completa en cada fase. - Útil para desarrollos pequeños y de pocas personas. 	<ul style="list-style-type: none"> - Poca flexibilidad. - Dificultad para corregir errores en fases anteriores. - Riesgo de entregar un producto final que no sea el deseado por el cliente. - El cliente no ve resultados hasta el final.
Iterativo	<ul style="list-style-type: none"> - Mayor flexibilidad para cambios de requisitos. - Mayor participación del cliente. - Las iteraciones tempranas dan oportunidades de aprendizaje y mejora. 	<ul style="list-style-type: none"> - El riesgo puede ser indefinido si no se gestiona adecuadamente. - Requiere más tiempo y recursos. - Inadecuado para equipos pequeños.
En espiral	<ul style="list-style-type: none"> - Enfoque iterativo con énfasis en la gestión de riesgos. - Adaptabilidad a los cambios de requisitos. - Evaluación continua durante todo el proyecto. 	<ul style="list-style-type: none"> - Requiere experiencia en la gestión de riesgos. - Mayor complejidad y costo. - Difícil para equipos pequeños.

Ágil	<ul style="list-style-type: none"> - Adaptabilidad a los cambios de requisitos. - Alta interacción con el cliente. - Enfoque en la mejora y la retroalimentación. 	<ul style="list-style-type: none"> - Requiere mucha colaboración y comunicación continua. - Puede faltar documentación. - El cliente debe participar mucho. - Es difícil estimar el tiempo de entrega.
-------------	--	--

Tabla 1. Ventajas y desventajas de los diferentes ciclos de vida

Después de ver las ventajas y desventajas, hay que elegir, según las necesidades de este proyecto, cuál sería el más adecuado. En este caso, se ha optado por el ciclo de vida en cascada, pero en combinación con el desarrollo ágil puesto que el proyecto puede ser cambiante y aparecer nuevos requisitos con el tiempo, o se pueden corregir errores en el mismo proceso. Además, se ha optado también por esta combinación, debido a que la documentación puede no ser clara y completa desde el principio, y puede adaptarse con el paso del tiempo.

4.2 Diagrama de Gantt

Es importante planificar correctamente todo el proceso seguido a la hora de realizar el proyecto. Por eso, una opción muy útil es realizar un diagrama de Gantt [35].

En este proyecto se ha creado un diagrama de Gantt enfocado en dividir las principales tareas de este proyecto, que son la investigación acerca del tema, el desarrollo de la memoria, el desarrollo del proyecto, las revisiones finales y la preparación del PowerPoint, además de sus subniveles correspondientes.

El diagrama queda de la siguiente manera:

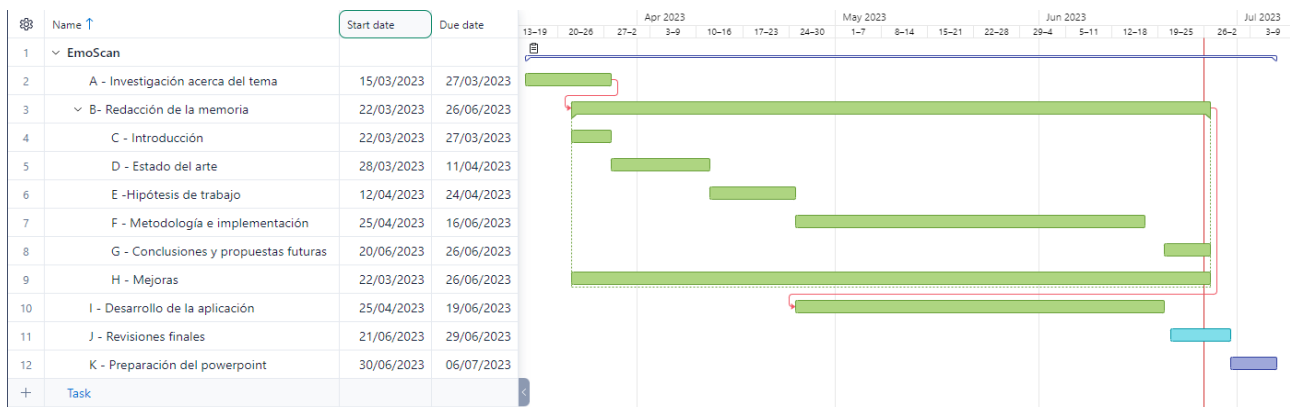


Figura 27. Diagrama de Gantt

Como se aprecia, se encuentran todas las tareas, desde la fecha de inicio hasta su final.

Cabe destacar que, como se aprecia, el ciclo de vida del proyecto se realiza en cascada como se explicó anteriormente, y con ciertos toques de desarrollo ágil para volver revisar y corregir, o mejorar ciertas partes ya hechas.

Lo primero de todo es investigar acerca del tema, y como se aprecia, aunque se haya terminado esta tarea, con los conocimientos ya adquiridos durante esos primeros días ya se puede comenzar a desarrollar la memoria.

Para seguir, se observa que durante todo el desarrollo de la memoria se van haciendo revisiones y mejoras para comprobar que esté todo bien, y además, una vez terminada la hipótesis de trabajo, no solo comienza la metodología e implementación, sino que ya comienza el desarrollo de la aplicación puesto que la mejor manera de optimizarlo era combinar ambos desarrollos para llevar un seguimiento exacto de lo que se va haciendo en cada momento.

Poco antes de terminar el proyecto y la memoria, ya se empieza a revisar que todo esté hecho correctamente, y una vez terminado ya se revisa más exhaustivamente.

Para finalizar, una vez que se ha comprobado que esté todo correcto, se empieza a preparar el PowerPoint final con el que se realizará la presentación.

4.3 Especificación de requisitos.

La especificación de requisitos [36] es la parte de la ingeniería de requisitos donde se documenta todos los requisitos del sistema y del usuario. Estos requisitos se negocian previamente con el cliente y como su nombre indica, determinará cómo será el producto software. Además, estos requisitos deben ser claros, completos y coherentes.

Existen 2 tipos de requisitos, los funcionales y los no funcionales:

- **Requisitos funcionales:** Estos requisitos describen las funciones del sistema que se va a diseñar. Es una descripción de lo que será el sistema y cómo funcionará para satisfacer las necesidades del usuario.
- **Requisitos no funcionales:** Estos requisitos explican las limitaciones y restricciones del sistema a diseñar, y no tienen ningún impacto en la funcionalidad de la aplicación.

Para realizar esta especificación se ha realizado una tabla en Excel con todos los requisitos, indicando su tipo, una descripción, y la prioridad que merece:

ID	Requisito	Descripción	Tipo	Prioridad
RF1	Autenticación de usuarios	El sistema debe permitir a los usuarios autenticarse creando una cuenta o logeando en ella para acceder a las funcionalidades de la aplicación.	Requisito funcional	Alta
RF2	Cierre de sesión	El sistema debe proporcionar la funcionalidad de cerrar sesión para los usuarios.	Requisito funcional	Alta
RF3	Editar perfil	El sistema debe permitir a los usuarios editar sus datos personales.	Requisito funcional	Media
RF5	Búsqueda de publicaciones de Twitter	El sistema debe permitir la búsqueda de publicaciones de Twitter basadas en parámetros predefinidos, como usuario, palabra clave, idioma y/o intervalo de fechas.	Requisito funcional	Alta
RF6	Procesamiento de lenguaje natural	El sistema debe aplicar algoritmos de procesamiento de lenguaje natural para el posterior análisis de sentimientos en las publicaciones de Twitter.	Requisito funcional	Alta
RF7	Análisis de sentimientos	El sistema debe aplicar los algoritmos de análisis de sentimientos.	Requisito funcional	Alta
RF8	Generación de informes estadísticos	El sistema debe generar informes estadísticos que muestren el sentimiento (amor u odio) en función del momento, lugar y fechas específicas de las publicaciones analizadas.	Requisito funcional	Alta
RF9	Interfaz de usuario intuitiva y fácil de usar	El sistema debe contar con una interfaz de usuario intuitiva y fácil de usar que permita a los usuarios interactuar con la aplicación de manera eficiente y sin dificultades.	Requisito funcional	Alta
RF10	Eliminar cuenta de usuario	El sistema debe brindar la oportunidad al usuario de eliminar su cuenta y borrar todos sus datos.	Requisito funcional	Media
RF11	Eliminar consultas	El sistema debe dar la oportunidad al usuario de eliminar consultas realizadas previamente	Requisito funcional	Media

Figura 28. Requisitos funcionales

NF1	Rendimiento	El sistema debe ser capaz de procesar grandes volúmenes de datos de Twitter de manera eficiente y responder a las solicitudes de los usuarios dentro de un tiempo razonable (por ejemplo, menos de 5 segundos).	Requisito no funcional	Alta
NF2	Seguridad	El sistema debe garantizar la confidencialidad de los datos del usuario y las publicaciones descargadas, así como implementar medidas de seguridad para proteger las cuentas de usuario.	Requisito no funcional	Alta
NF3	Escalabilidad	El sistema debe ser escalable y capaz de manejar un aumento en el número de usuarios y volúmenes de datos sin degradar el rendimiento.	Requisito no funcional	Media
NF4	Compatibilidad	El sistema debe ser compatible con los principales navegadores web (por ejemplo, Chrome, Firefox, Safari) para garantizar una experiencia de usuario consistente en diferentes entornos de navegación.	Requisito no funcional	Media
NF5	Usabilidad	La aplicación debe ser intuitiva y fácil de usar para los usuarios.	Requisito no funcional	Alta
NF6	Disponibilidad	La aplicación debe estar disponible para su acceso en todo momento.	Requisito no funcional	Alta
NF7	Mantenibilidad	El código de la aplicación debe ser modular y fácil de mantener.	Requisito no funcional	Media
NF8	Cumplimiento de normativas	La aplicación debe cumplir con las normativas y regulaciones de protección de datos.	Requisito no funcional	Alta

Figura 29. Requisitos no funcionales

4.4 Actores y casos de uso

Los casos de uso [37] son técnicas utilizadas para describir las interacciones entre los actores y el sistema. Un caso de uso describe una funcionalidad específica del sistema desde el punto de vista del usuario.

En los casos de uso, encontramos dos partes bien diferenciadas, los actores, y los casos de uso, para definirlos, se necesitan diferentes tablas, estas son las plantillas de ambos elementos:

Actor		
Descripción		
Comentarios		
Casos de uso en los que interviene		

Tabla 2. Plantilla de actores

Caso de uso			
Actor principal			
Descripción			
Precondición			
Secuencia normal	Paso	Acción	
Postcondición			
Excepciones			
Comentarios			

Tabla 3. Plantilla de casos de uso

4.4.1 Actores

Actor	A01	Usuario no registrado
Descripción	Usuarios que acceden a la web por primera vez y no poseen una cuenta en ella.	
Comentarios	No podrá realizar nada más que registrarse.	
Casos de uso en los que interviene	- C01: Registro de usuario	

Tabla 4. Actor, usuario no registrado

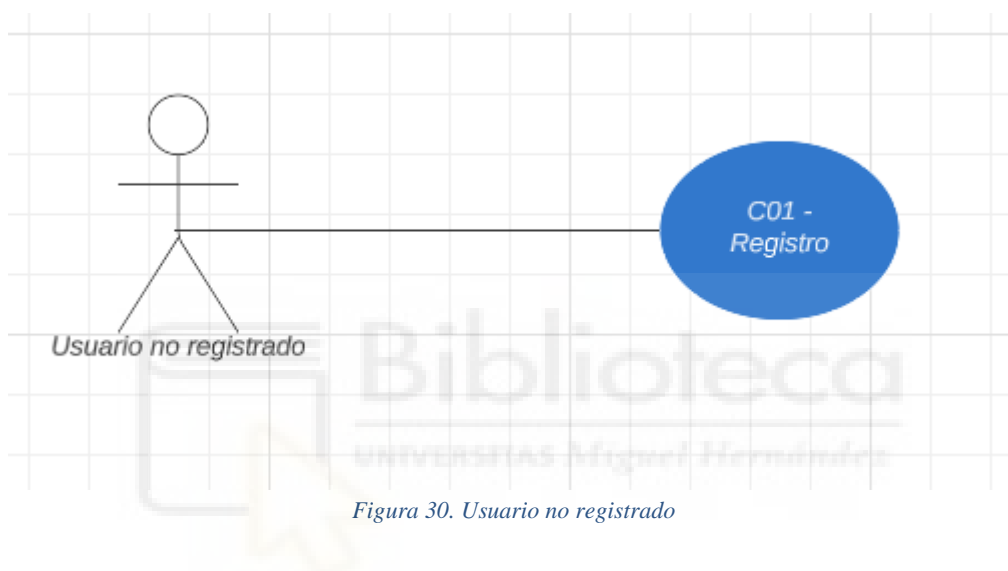


Figura 30. Usuario no registrado

Actor	A02	Usuario registrado
Descripción	Usuarios que ya poseen una cuenta en la aplicación	
Comentarios	Primero deberán iniciar sesión, para un posterior uso del resto de funcionalidades.	
Casos de uso en los que interviene	<ul style="list-style-type: none"> - C02: Iniciar sesión - C03: Recuperación de contraseña - C04: Editar perfil - C05: Cerrar sesión - C06: Buscar tweets - C07: Analizar tweets - C08: Ver estadísticas - 	

Tabla 5. Actor, usuario registrado

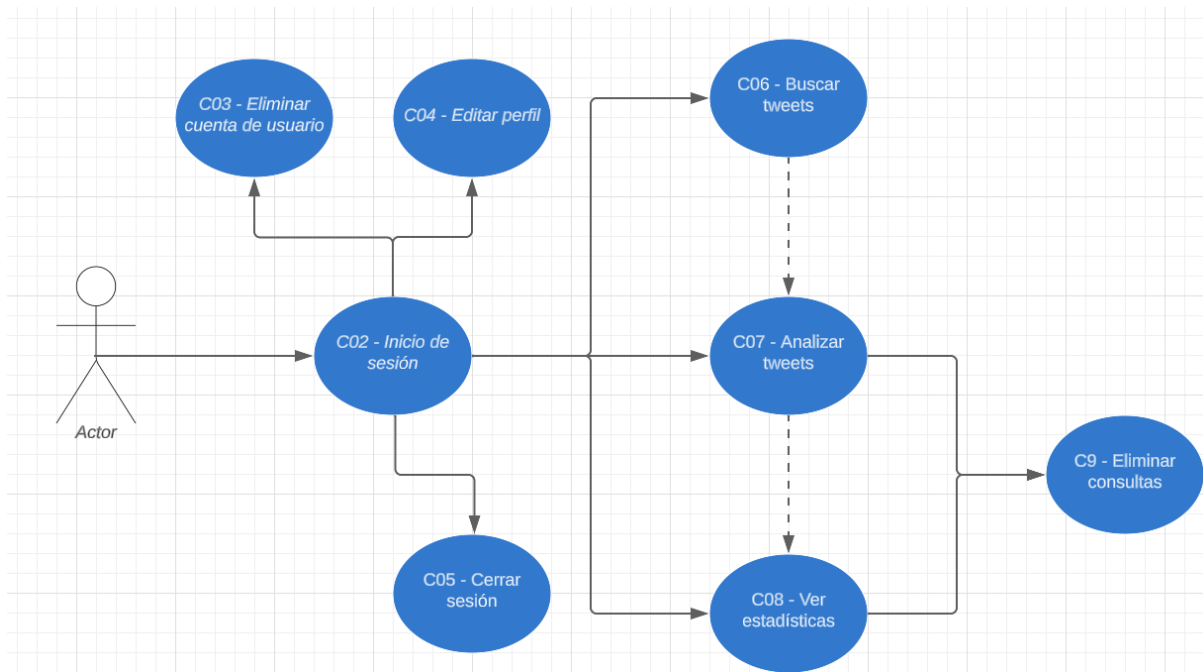


Figura 31. Usuario registrado

4.4.2 Casos de uso

Caso de uso	C01	Registro de usuario
Actor principal	A01: Usuario no registrado	
Descripción	Registro de usuario en la aplicación web introduciendo los datos que sean solicitados.	
Precondición		
Secuencia normal	Paso	Acción
	1	Acceder a la zona de registro
	2	Introducir los datos necesarios
	3	El sistema comprueba que los datos son correctos
	4	En caso de que sean correctos los datos, el sistema comprueba que no existan en la base de datos
	5	En caso de que no existan, el sistema creará el usuario y se almacenará en la base de datos

	6	El sistema automáticamente se redirigirá hacia la zona de usuarios identificados.
Postcondición	El usuario quedará guardado y listo para iniciar sesión en cualquier momento.	
Excepciones	<ul style="list-style-type: none"> - Paso 3, si los datos no son correctos se volverán a solicitar - Paso 4, si el usuario ya está registrado, se le informará al usuario de esto mismo, y se volverán a solicitar los datos. 	
Comentarios		

Tabla 6. C01 - Registro de usuario

Caso de uso	C02	Inicio de sesión	
Actor principal	A02: Usuario registrado		
Descripción	Inicio de sesión en la aplicación web introduciendo los datos que sean solicitados.		
Precondición	Estar registrado en la aplicación.		
Secuencia normal	Paso	Acción	
	1	Acceder a la zona de inicio de sesión	
	2	Introducir los datos necesarios	
	3	El sistema comprueba que los datos introducidos se encuentran en la base de datos	
	4	En caso de que existan, el sistema iniciará la sesión del usuario.	
Postcondición	El usuario habrá iniciado sesión y podrá utilizar todas sus funcionalidades.		
Excepciones	<ul style="list-style-type: none"> - Paso 3, si los datos no se encuentran en la base de datos, se mostrará un mensaje de error y se volverán a solicitar. 		
Comentarios			

Tabla 7. C02 - Inicio de sesión

Caso de uso	C03	Eliminar cuenta de usuario	
Actor principal	A02: Usuario registrado		
Descripción	Eliminar la cuenta del usuario		
Precondición	Haber iniciado sesión		
Secuencia normal	Paso	Acción	
	1	Acceder al perfil, situado en la barra de navegación.	
	2	Seleccionar la opción: “eliminar cuenta”	
	3	Introducir la contraseña para confirmar.	
	4	En caso de que la contraseña sea correcta, el sistema eliminará la cuenta.	
Postcondición	El sistema cerrará la sesión y la cuenta estará eliminada con todos sus datos.		
Excepciones	- Paso 4, si la contraseña es incorrecta, no se eliminará la cuenta.		
Comentarios			

Tabla 8. C03 - Recuperación de contraseña

Caso de uso	C04	Editar perfil	
Actor principal	A02: Usuario registrado		
Descripción	Editar tus datos personales.		
Precondición	Iniciar sesión		
Secuencia normal	Paso	Acción	
	1	Acceder a la zona del perfil, situado en la barra de navegación	
	2	Seleccionar la opción editar perfil	

	3	Introducir los datos a cambiar
	4	El sistema comprueba que son datos válidos
	5	Se guardan los datos
Postcondición	Los datos se habrán actualizado correctamente	
Excepciones	- Paso 4, si los datos no son correctos se volverán a solicitar después de un mensaje de error.	
Comentarios		

Tabla 9. C04 - Editar perfil

Caso de uso	C05	Cerrar sesión	
Actor principal	A02: Usuario registrado		
Descripción	Cerrar sesión en la aplicación		
Precondición	Iniciar sesión		
Secuencia normal	Paso	Acción	
	1	Acceder a la opción cerrar sesión que se encuentra en el mismo sitio del perfil, en la barra de navegación superior.	
	2	Seleccionar la opción de cerrar sesión	
	3	Confirmar que quieres cerrar sesión	
	4	El sistema procesará la solicitud y se cerrará sesión.	
Postcondición	El usuario ya no tendrá su sesión iniciada y no podrá realizar ninguna función hasta que vuelva a iniciar sesión.		
Excepciones			
Comentarios			

Tabla 10. C05 - Cerrar sesión

Caso de uso	C06	Buscar tweets	
Actor principal	A02: Usuario registrado		
Descripción	Realizar consultas de tweets a través de una palabra clave, o usuario, o diferentes filtros como rangos de fechas.		
Precondición	Iniciar sesión		
Secuencia normal	Paso	Acción	
	1	Acceder a la zona de buscar tweets	
	2	Seleccionar la opción de palabra clave, usuario, o hashtag	
	3	Escribir su consulta	
	4	Introducir el número de tweets a buscar (No obligatorio, por defecto 100)	
	5	Añadir los filtros que se quieran (no obligatorio)	
	6	Pulsar el botón de buscar	
	7	El sistema muestra una tabla con los tweets, mediante paginación de 10 en 10	
Postcondición	Los tweets se habrán guardado en la base de datos y ya se podrá proceder a su análisis.		
Excepciones			
Comentarios			

Tabla 11. C06 - Buscar tweets

Caso de uso	C07	Analizar tweets	
Actor principal	A02: Usuario registrado		
Descripción	Analizar los sentimientos de los tweets previamente buscados y guardados.		

Precondición	Haber buscado y guardado tweets previamente.	
Secuencia normal	Paso	Acción
	1	Acceder a la zona de analizar tweets
	2	Seleccionar una de las consultas que has buscado previamente
	3	El sistema mostrará una tabla similar a la de buscar tweets, añadiendo el sentimiento que contiene cada tweet
Secuencia alternativa	Paso	Acción
	1	Acceder a la zona de buscar tweets y realizar una consulta como la explicada en C06
	2	Pulsar el botón de analizar tweets que aparecerá.
	3	El sistema accederá a la página de analizar tweets y mostrará en una tabla los tweets buscados previamente, de manera similar a la de buscar tweets, pero añadiendo el sentimiento de cada tweet
4	El sistema mostrará una tabla similar a la de buscar tweets, añadiendo el sentimiento que contiene cada tweet.	
Postcondición	El sentimiento se habrá añadido a la base de datos y estará listo para su posterior muestra de estadísticas.	
Excepciones	<ul style="list-style-type: none"> - Paso 2(Secuencia normal): Si nunca has realizado ninguna búsqueda de tweets, no habrá tweets a analizar. 	
Comentarios		

Tabla 12. C07 - Analizar tweets

Caso de uso	C08	Ver estadísticas
Actor principal	A02: Usuario registrado	
Descripción	Ver las estadísticas de los tweets analizados.	

Precondición	Haber analizado tweets previamente.	
Secuencia normal	Paso	Acción
	1	Acceder a la zona de estadísticas
	2	Seleccionar la consulta analizada sobre la que quieres ver las estadísticas
	3	El sistema te redirigirá a una página donde se mostrarán las estadísticas en forma de gráfica.
Secuencia alternativa	Paso	Acción
	1	Realizar un análisis de sentimientos en la sección de analizar tweets como se explica en C07
	2	Pulsar el botón de ver estadísticas que aparecerá.
	3	El sistema accederá a la página de estadísticas y se mostrarán las estadísticas en forma de gráficas.
Postcondición	Se habrá realizado un estudio estadístico de los datos.	
Excepciones	<ul style="list-style-type: none"> - Paso 2(Secuencia normal): Si no has realizado ningún análisis de sentimientos previo, no habrá estadísticas que mostrar. 	
Comentarios		

Tabla 13. C08 - Ver estadísticas

Caso de uso	C09	Eliminar consultas
Actor principal	A02: Usuario registrado	
Descripción	Eliminar consultas realizadas previamente.	
Precondición	Haber realizado alguna búsqueda	
Secuencia normal	Paso	Acción
	1	Acceder a la zona de análisis de sentimientos/estadísticas.

	2	Pulsar el botón de eliminar consulta
	3	Pulsar el botón de confirmar la eliminación.
	4	El sistema eliminará la consulta de la base de datos.
Secuencia alternativa	Paso	Acción
	1	Acceder a la zona de análisis de sentimientos/estadísticas.
	2	Seleccionar las opciones a eliminar mediante los checkbox.
	3	Pulsar el botón de eliminar consultas seleccionadas que aparecerá abajo.
	4	Pulsar el botón de confirmar la eliminación.
	5	El sistema eliminará la consulta de la base de datos.
Postcondición	Se habrá eliminado una o varias consultas de la base de datos.	
Excepciones		
Comentarios		

Tabla 14. C09 - Eliminar consultas

4.5 Diagramas de secuencia y actividad

Para entender mejor la aplicación, se han realizado diversos diagramas, cada uno de ellos relacionado con un caso de uso, en el que muestran el proceso de llevar a cabo dichos casos de uso, de forma gráfica.

4.5.1 Diagramas de secuencia.

Los diagramas de secuencia son representaciones visuales que muestran la interacción entre los diferentes componentes que interactúan en un proceso. En este caso se encuentra el usuario, el sistema y la base de datos. A continuación, se presentan los diagramas de secuencia para cada caso de uso definido en este proyecto.

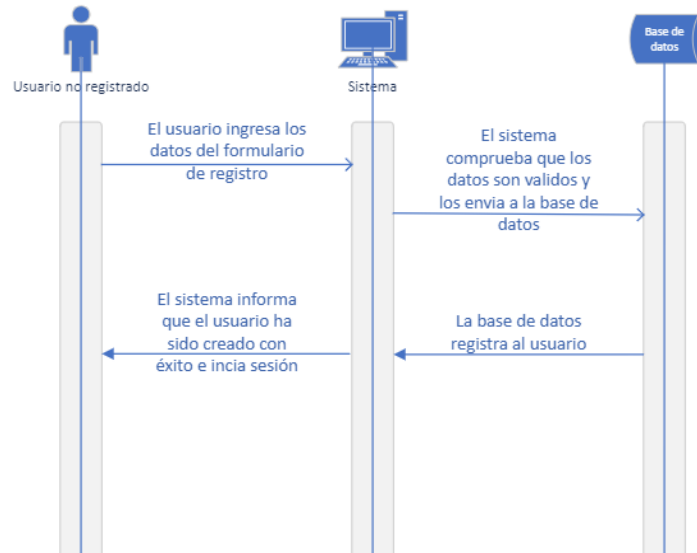


Figura 32. Diagrama de secuencia C01

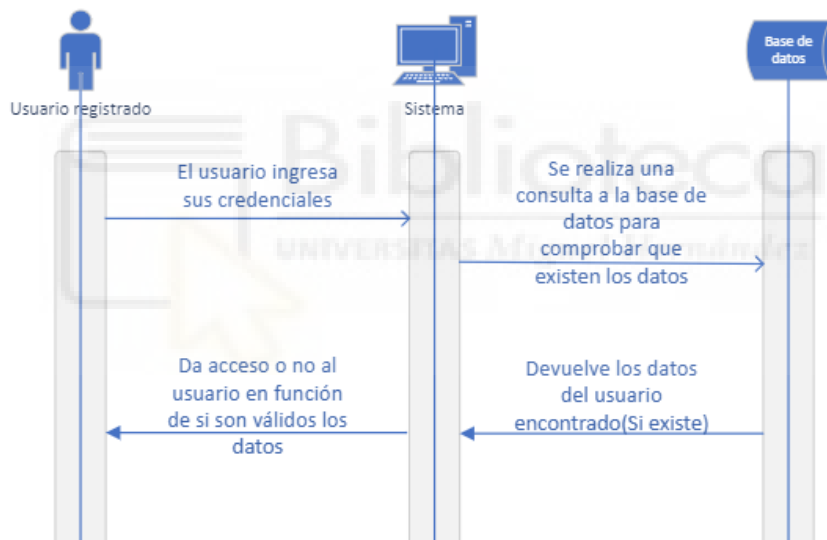


Figura 33. Diagrama de secuencia C02

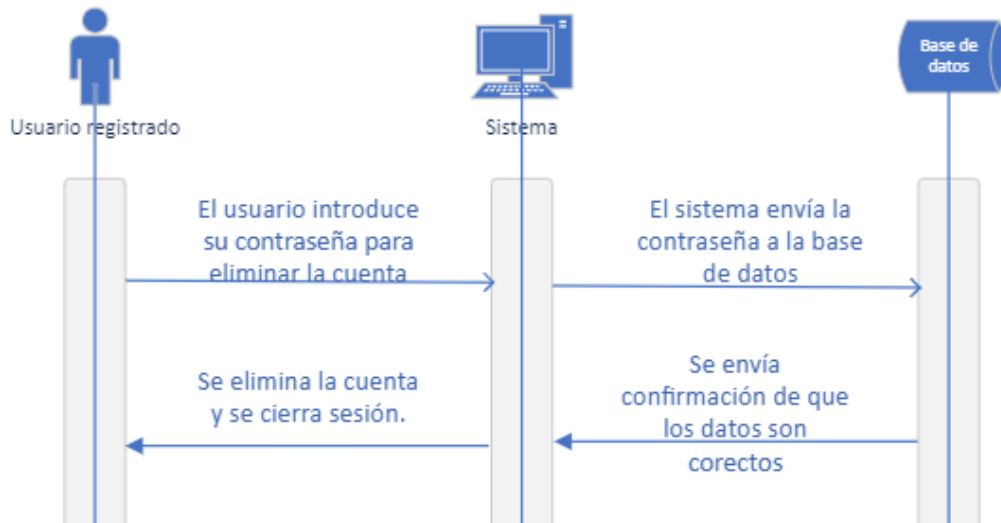


Figura 34. Diagrama de secuencia C03

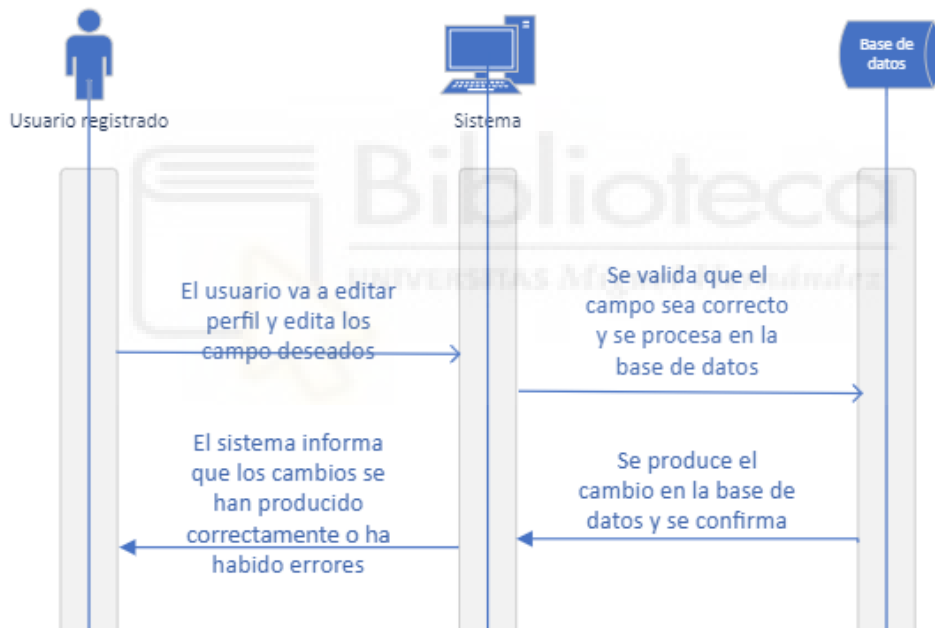


Figura 35. Diagrama de secuencia C04

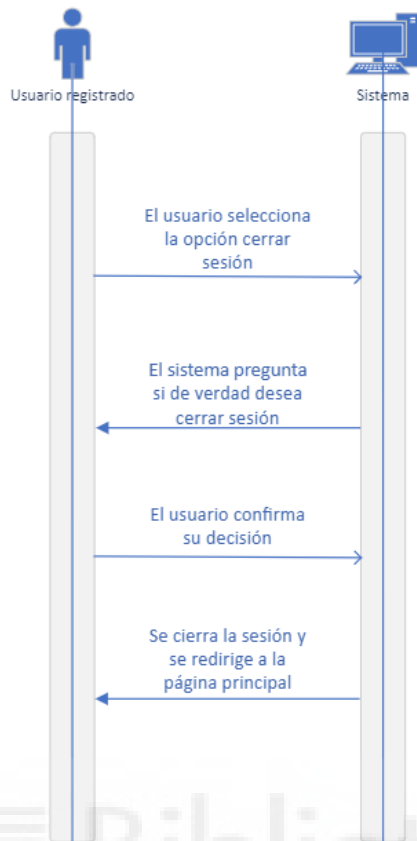


Figura 36. Diagrama de secuencia C05

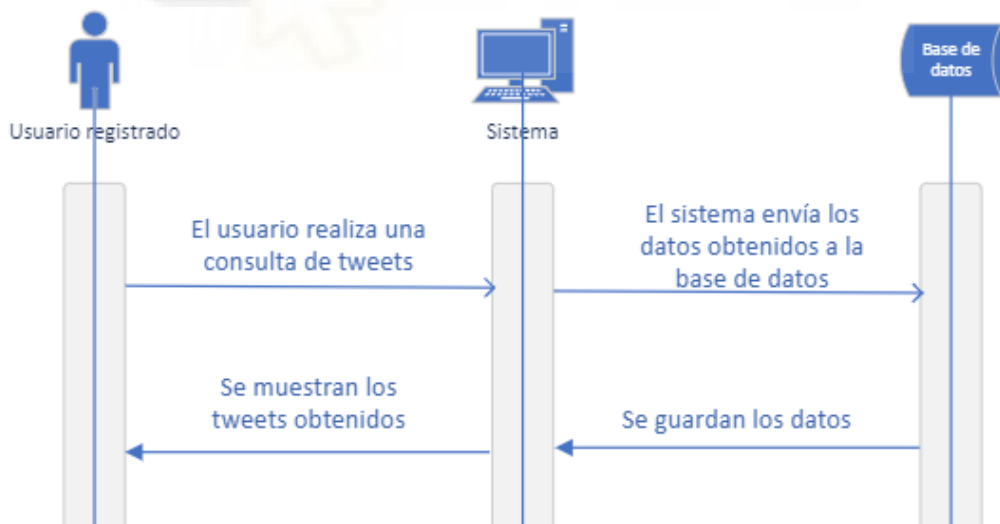


Figura 37. Diagrama de secuencia C06

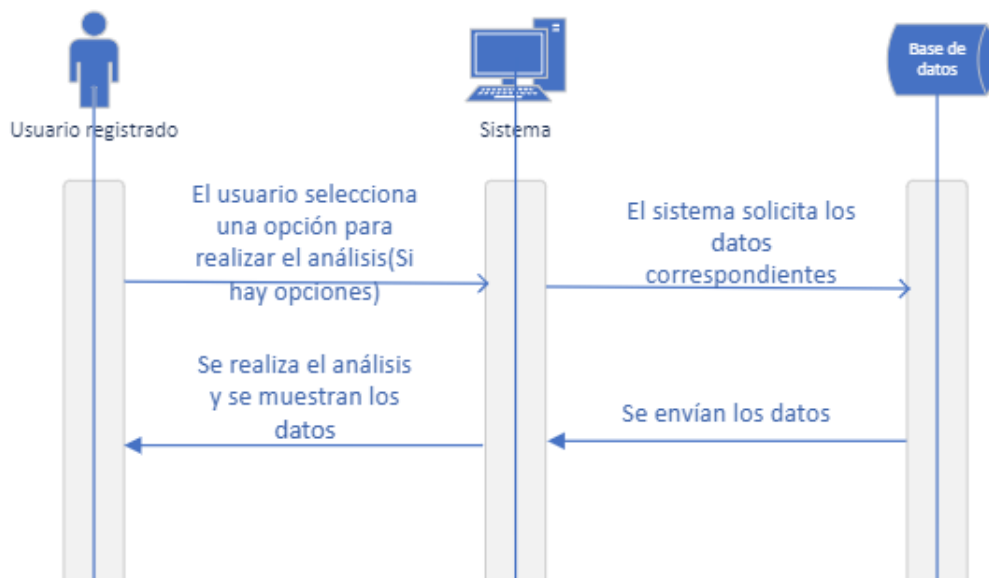


Figura 38. Diagrama de secuencia C07(Flujo normal)

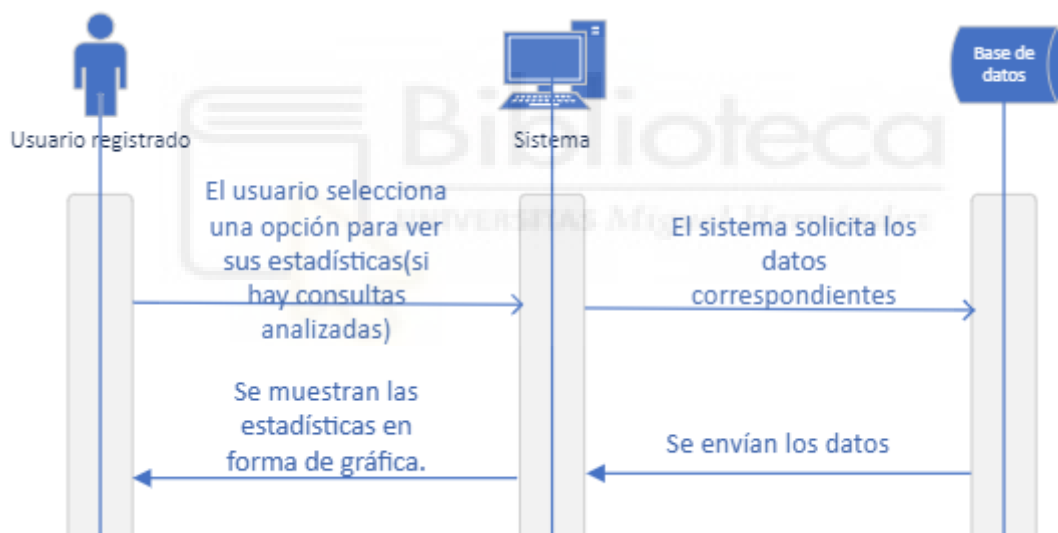


Figura 39. Diagrama de secuencia de C08.

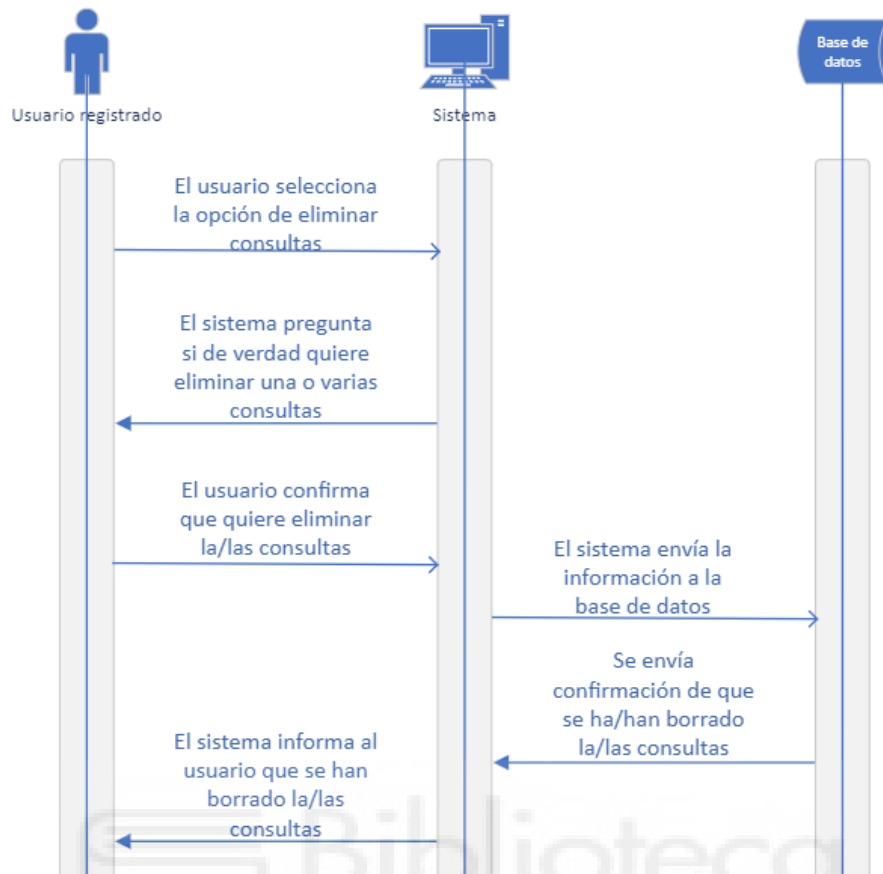


Figura 40. Diagrama de secuencia de C089

UNIVERSITAS Miguel Hernández

4.5.2 Diagramas de actividad.

Los diagramas de actividad son una representación visual de las actividades o acciones en un proceso. A continuación, se presentan los diagramas de actividad para cada caso de uso definido en este proyecto.

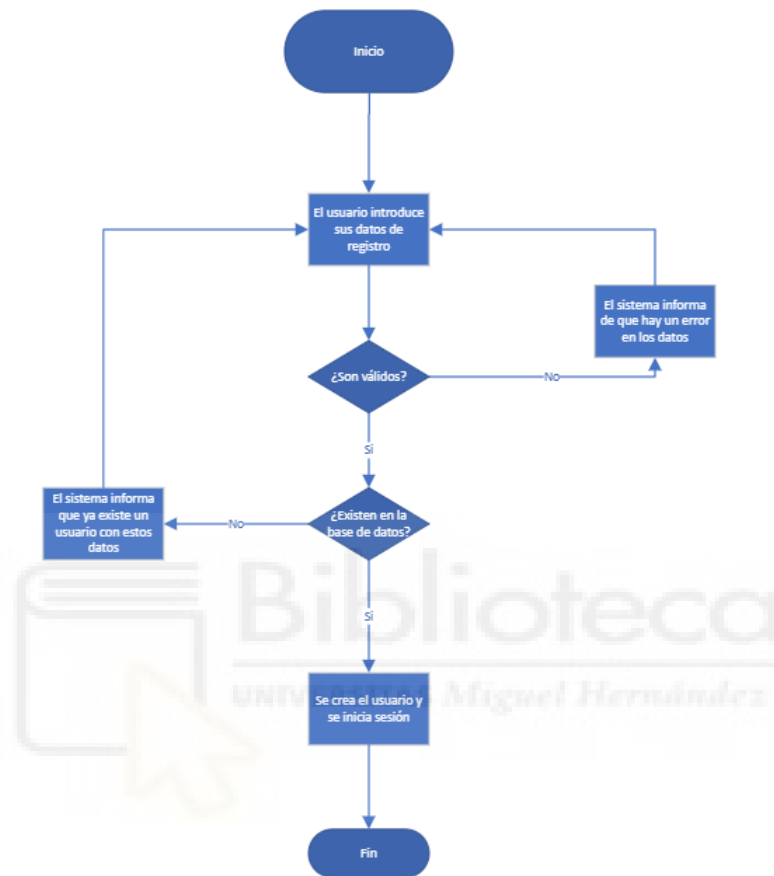


Figura 41. Diagrama de actividad C01

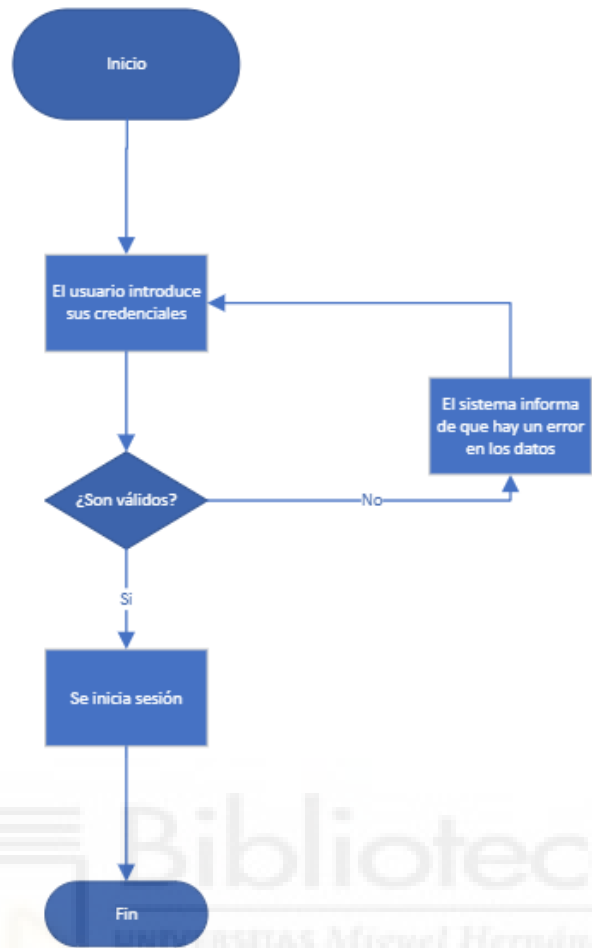


Figura 42. Diagrama de actividad C02

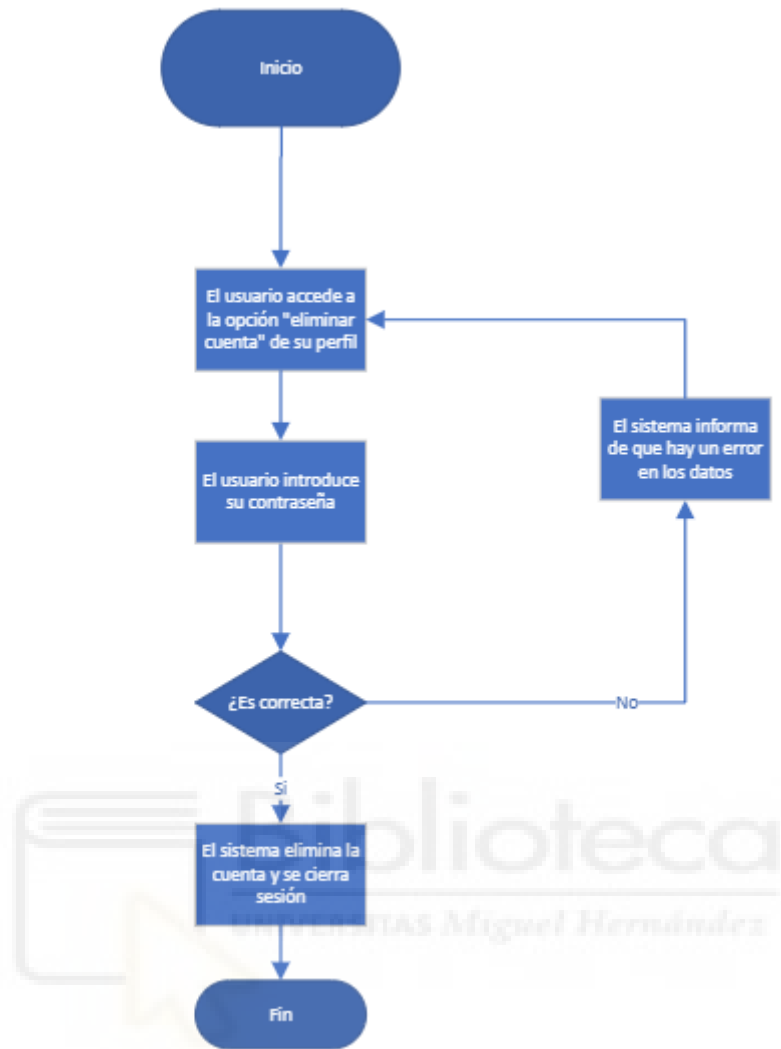


Figura 43. Diagrama de actividad C03

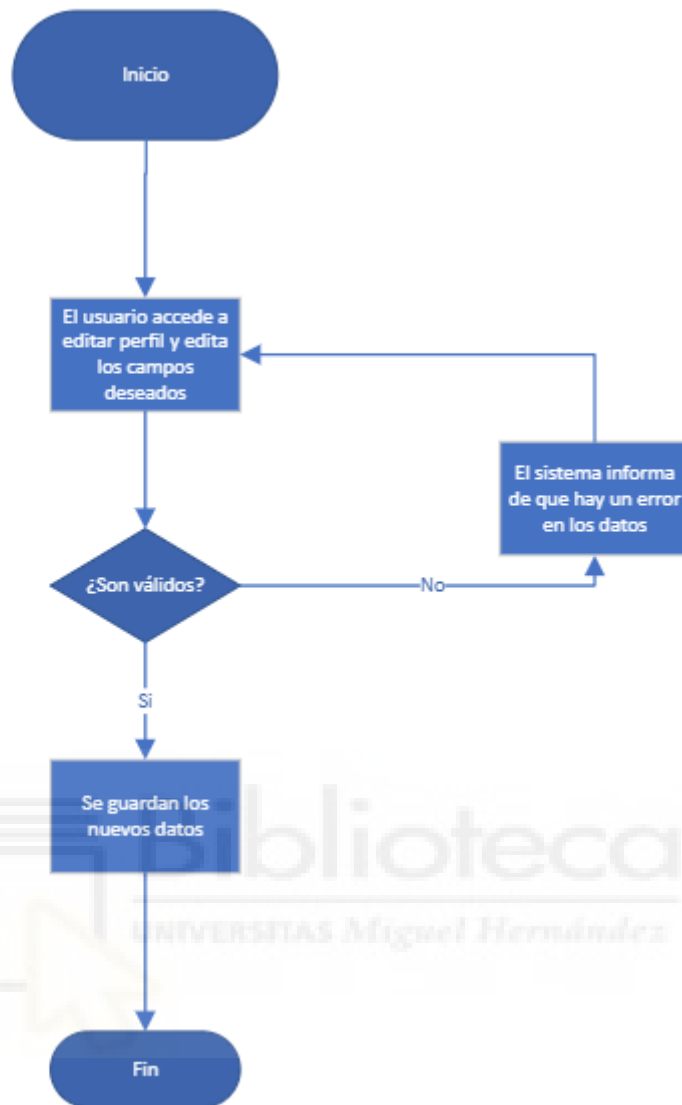


Figura 44. Diagrama de actividad C04

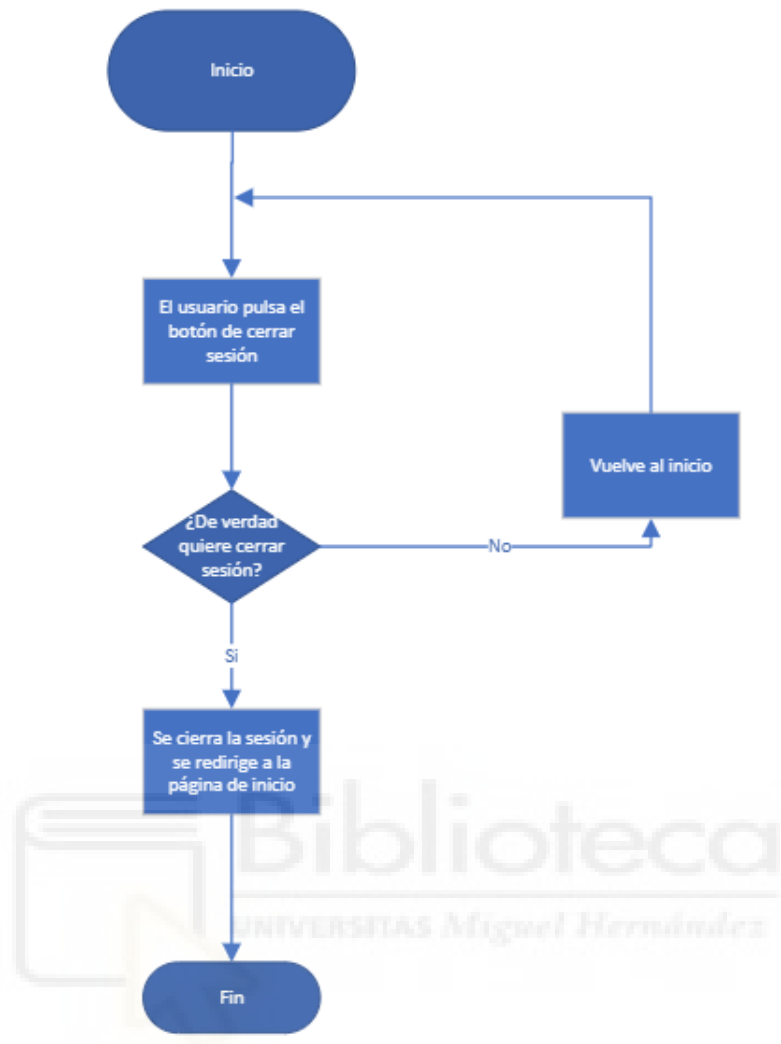


Figura 45. Diagrama de actividad C05



Figura 46. Diagrama de actividad C06

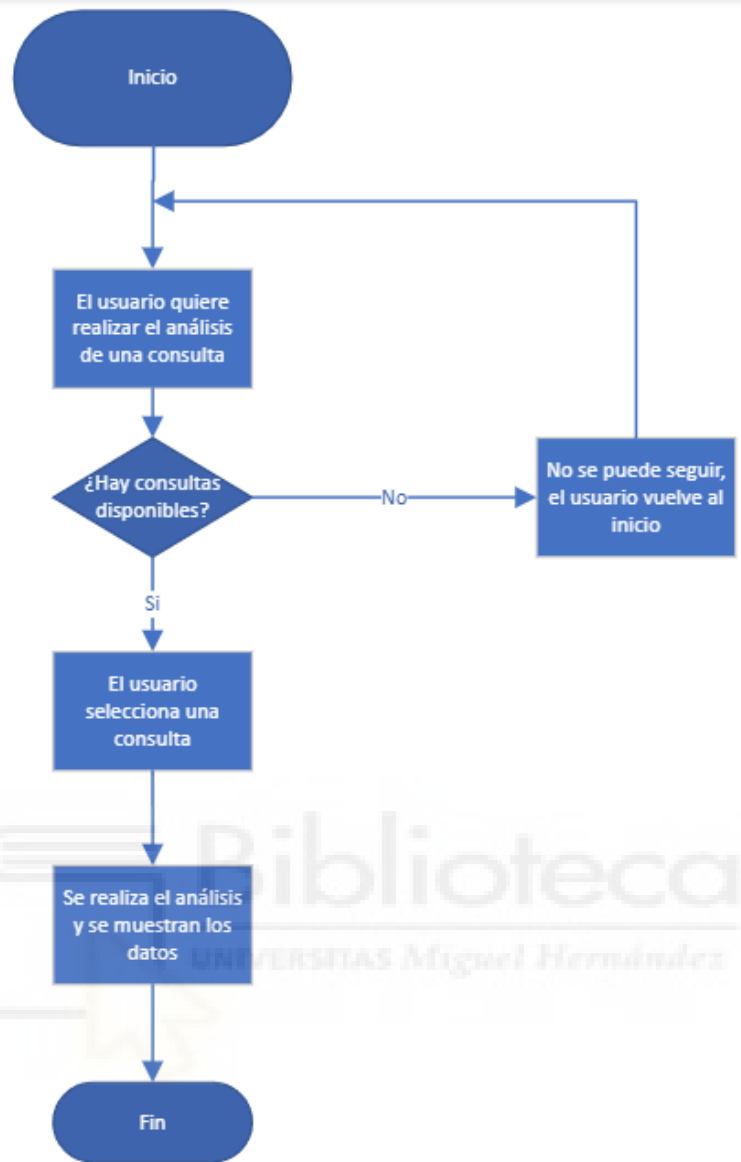


Figura 47. Diagrama de actividad C07(Flujo normal)

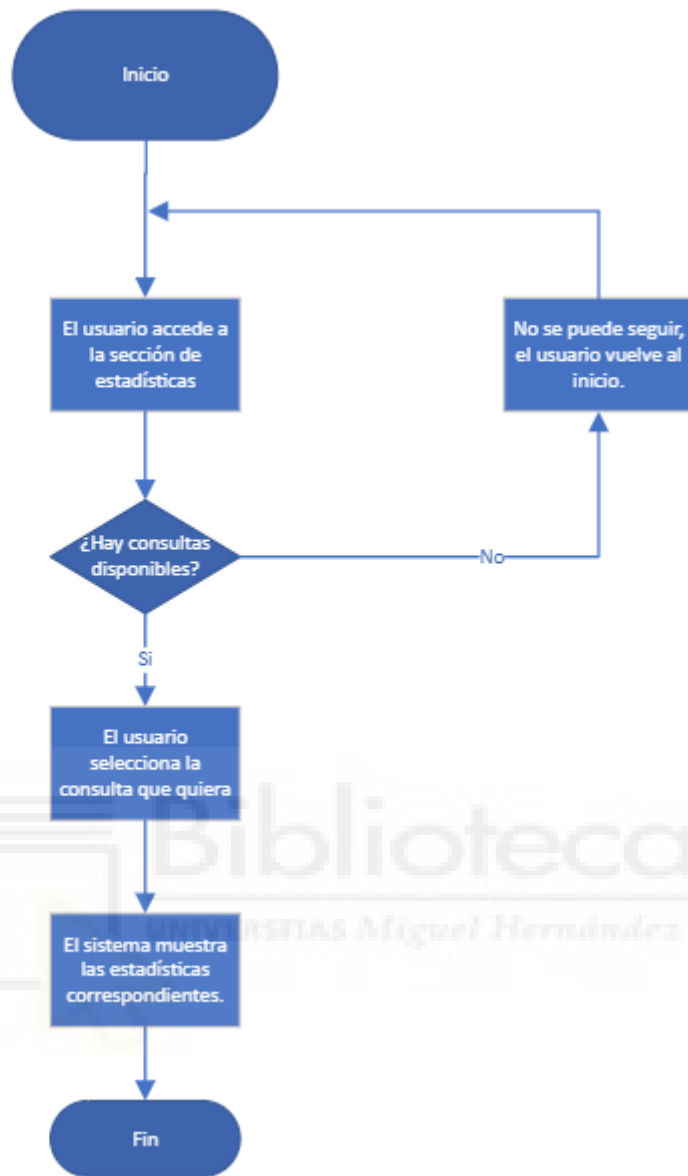


Figura 48. Diagrama de actividad C08



Figura 49. Diagrama de actividad C09

4.6 Diseño de la aplicación

4.6.1 Interfaz de usuario

El estar ante una aplicación web, significa que usuarios externos van a interactuar con la aplicación, para ello la parte más importante, es tener una interfaz de usuario, que es la manera de que se interactúe con el sistema de una forma fácil y sencilla. Para realizar una interfaz de usuario, hay que tener en cuenta varios aspectos, ya que la interfaz debe ser agradable a la vista, accesible para todo el mundo, fácil de usar para cualquier usuario sin experiencia, o que sea responsive, es decir, se pueda utilizar tanto en pantallas grandes, como medianas, como pantallas más pequeñas y va a seguir siendo visualmente agradable e igual de intuitiva y funcional. Para conseguir esto se ha diseñado una interfaz mediante HTML, Bootstrap 5, css, js, y jquery.

Pero antes de todo eso, el primer paso es crear un boceto con el que puedas expresar tus ideas de una manera rápida y sencilla, y cambiarla sin que haya ningún problema antes de empezar a desarrollar la aplicación.

A continuación, se muestran imágenes de los bocetos realizados. Todos los bocetos se han realizado con la herramienta Canva [38].

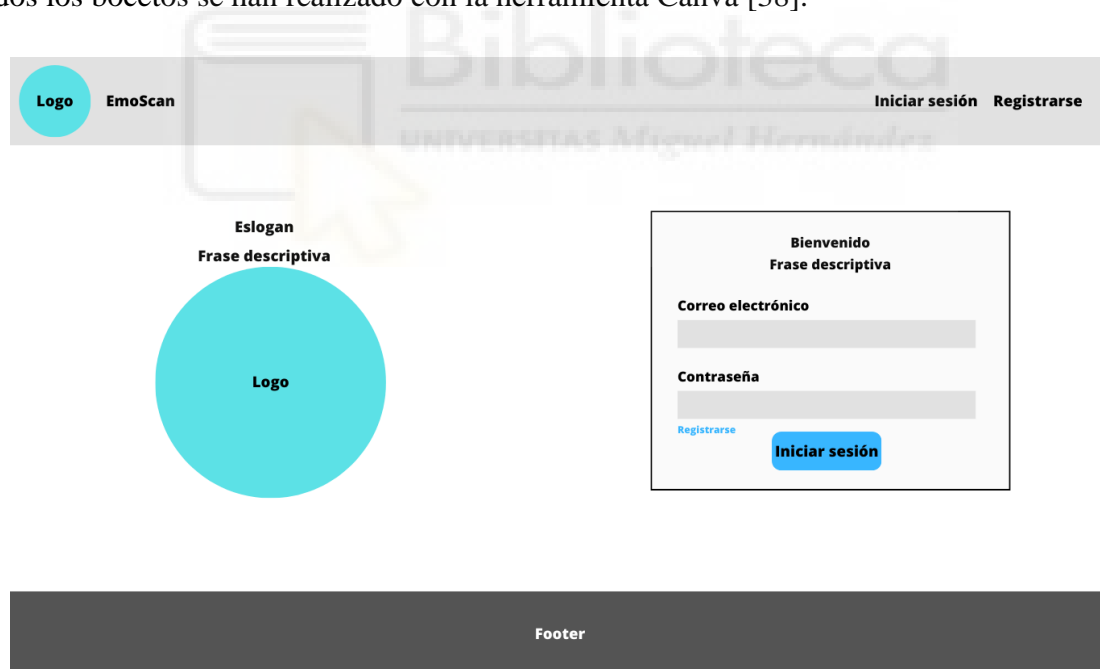


Figura 50. Boceto Landing Page

Inicio de sesión

Correo electrónico
[Input field]

Contraseña
[Input field]

[Registrarse](#) **Iniciar sesión**

Figura 51. Boceto inicio de sesión

Registro de usuario

Nombre * [Input field] **Apellidos *** [Input field]

Nombre de usuario * [Input field]

Correo electrónico * [Input field]

Contraseña * [Input field] **Confirmar contraseña *** [Input field]

Aceptar términos y condiciones *

Registrarse

*Requerido

Figura 52. Boceto registro de usuario

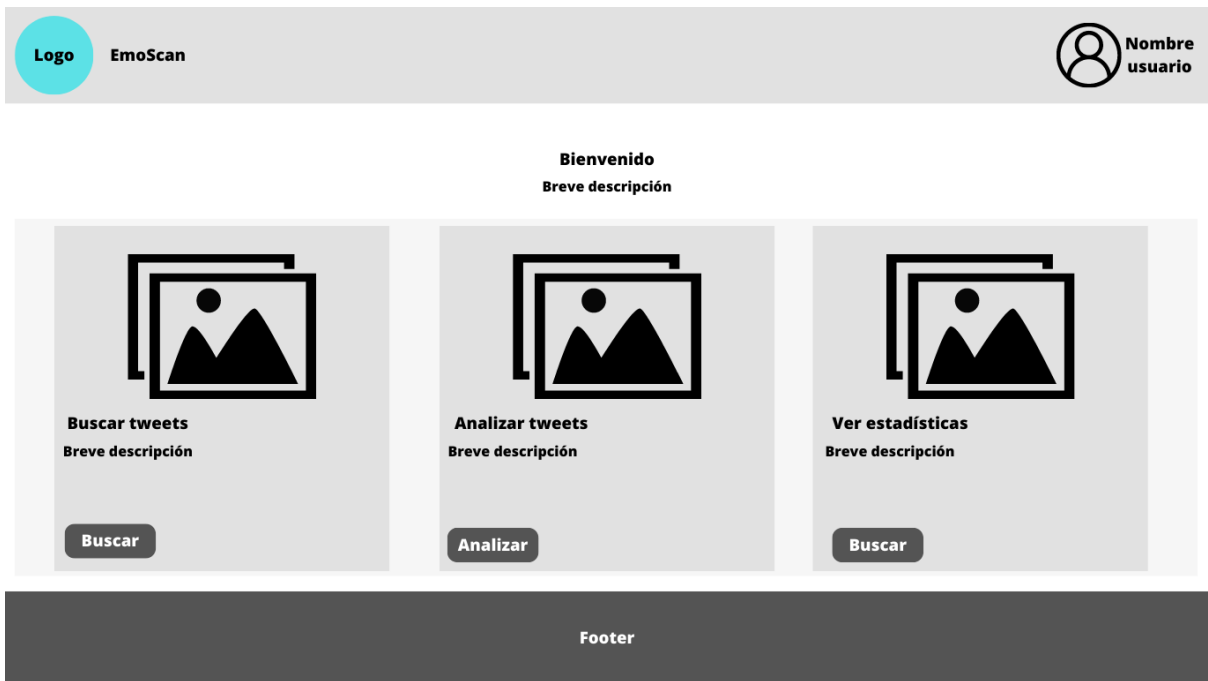


Figura 53. Boceto Página principal

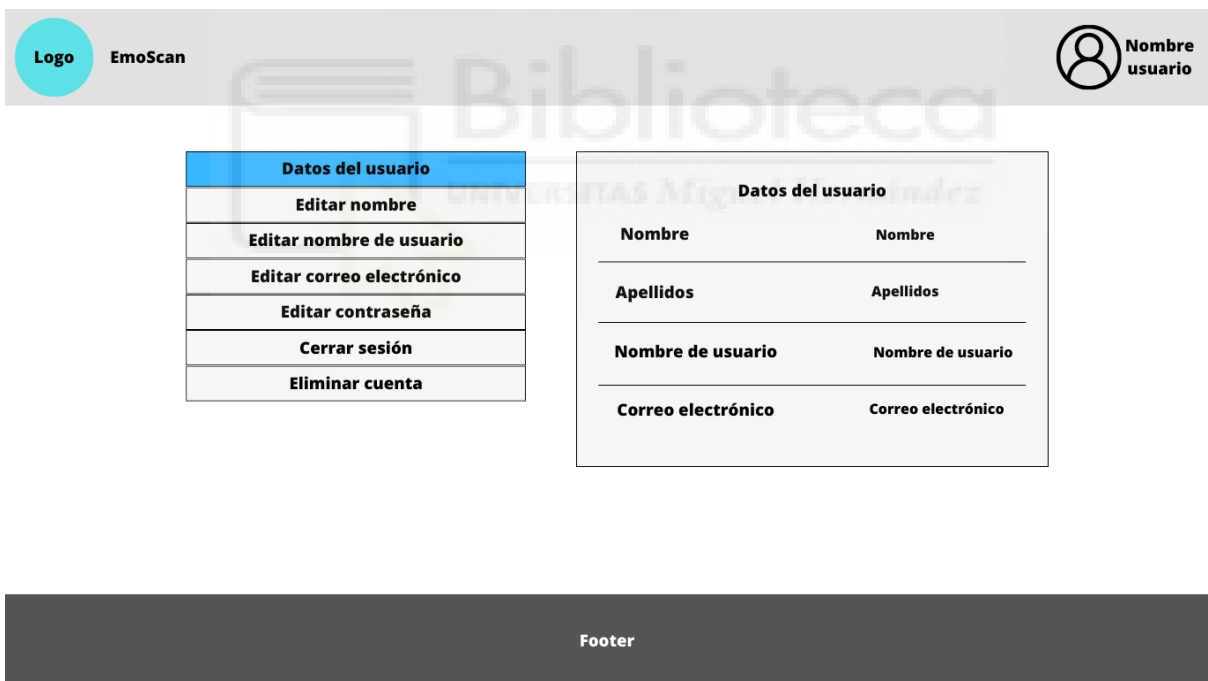


Figura 54. Boceto Perfil

Logo EmoScan Nombre usuario

Selecciona un método de búsqueda

Selecciona una opción ▼

Realizar una consulta

Buscar usuario

Buscar hashtag

Introduce una consulta/usuario/hashtag Nº tweets Filtros

Mostrar respuestas Idioma Inglés ▼ Desde dd/mm/aaaa Hasta dd/mm/aaaa

Buscar

Footer

Figura 55. Boceto Buscar tweets

Logo EmoScan Nombre usuario

Resultados de la búsqueda
Parámetros utilizados

Lista con los datos de la consulta

Fecha	Contenido del tweet	Usuario
Tabla con los tweets		

Paginación

Volver a buscar sin guardar
Analizar tweets
Guardar y volver al menú principal

Footer

Figura 56. Boceto resultados búsqueda

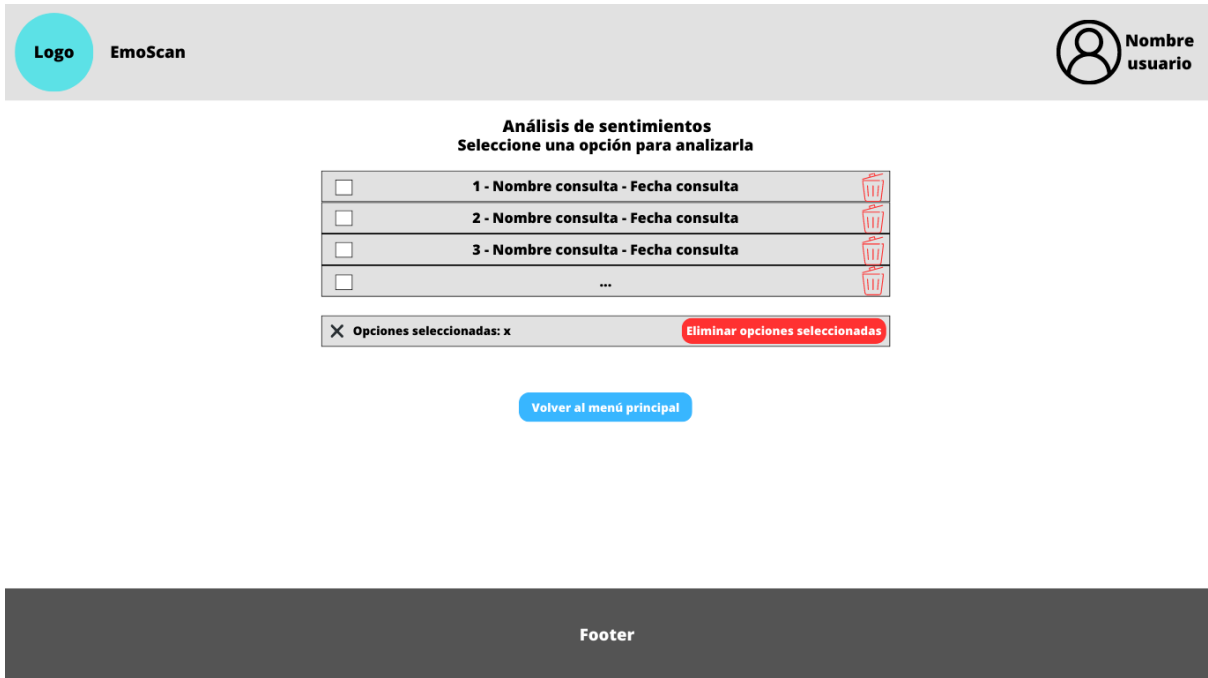


Figura 57. Boceto análisis de sentimientos



Figura 58. Boceto resultado análisis



Figura 59. Boceto estadísticas



Figura 60. Boceto resultados estadísticas.

4.6.2 Base de datos

Otro punto muy importante es la base de datos, como ya se mencionó anteriormente, estará desarrollada en MariaDB/MySQL.

Mediante la base de datos se almacenarán todos los datos, tanto del usuario como de los tweets, y para una mayor optimización, algo fundamental en las bases de datos, se ha creado una tabla consultas, donde se almacenará las consultas realizadas por el usuario, y conectará los tweets con las consultas y con los usuarios.

Para empezar, es necesario la tabla usuarios (users), donde se almacenarán los datos del usuario cuando se registra: id de usuario, nombre, apellidos, nombre de usuario(único), correo electrónico (único), y contraseña (se guarda un hash de la contraseña generado mediante una función criptográfica, para añadirle la seguridad necesaria).

Posteriormente cuando se realice la búsqueda de tweets, cada búsqueda será guardada en la tabla consultas (queries), donde se guardará el id de la consulta, el id del usuario que la ha buscado, el nombre de la consulta, el tipo de consulta, la fecha y hora de la consulta y luego los filtros aplicados, si se desea mostrar respuestas o no, el idioma, desde que fecha hasta que fecha quieres los tweets, y el número de tweets a buscar en la consulta.

Por último, al realizar dicha búsqueda, también se guardarán en otra tabla, los tweets encontrados en esta búsqueda, almacenándose de ellos: el id del tweet, el id de la consulta, para relacionarlos entre sí, el id del usuario que ha realizado la consulta, el contenido del tweet, la fecha en la que se ha escrito, y el usuario de Twitter que la ha realizado, también se almacenará el sentimiento encontrado, que primeramente será un campo vacío, y posteriormente se añadirá una vez hecho su análisis.

Para entender todo esto de una manera más sencilla, se ha realizado el modelo relacional de la base de datos:

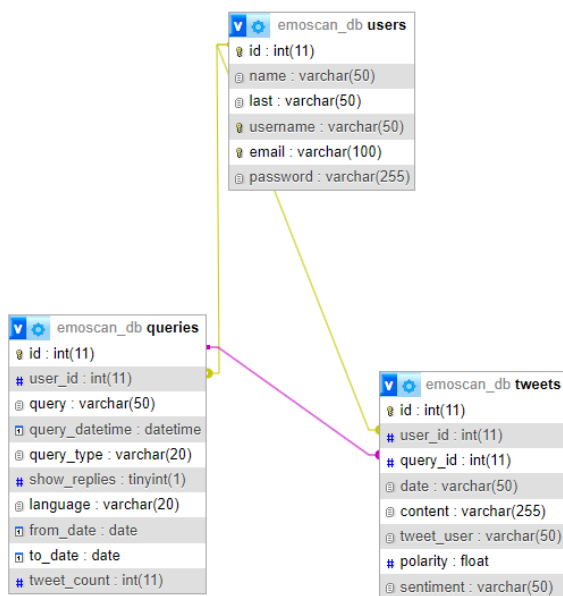


Figura 61. Modelo relacional de la base de datos

4.7 Implementación

4.7.1 Instalación y configuración del framework Flask

El primer paso para empezar a desarrollar la aplicación es instalar el framework elegido, que entre las múltiples opciones que existen, es Flask. Como cualquier librería de python, su instalación es tan simple como usar pip, el instalador de paquetes de Python se hace mediante este código: “pip install Flask”.

Una vez hecho esto, Flask está instalado y listo para su uso y configuración.

Para configurar Flask, lo primero es crear un archivo “config.py” de la siguiente manera:

```
1 class Config:
2     DEBUG = True
3     TESTING = True
4
5
6 class ProductionConfig(Config):
7     DEBUG = False
8
9 class DevelopmentConfig(Config):
10    SECRET_KEY = 'dev'
11    DEBUG = True
12    TESTING = True
```

Figura 62. Config.py

Este archivo básicamente contiene la configuración para usar la aplicación en modo desarrollo o en modo producción.

Posteriormente hay que crear el archivo principal “app.py”, que es donde se ejecutará la aplicación:

```
1 from EmoScan import app
2
3
4 if __name__ == '__main__':
5
6     app.run()
```

Figura 63. App.py

Para continuar hay que crear el archivo donde se encuentra el código principal: “__init__.py”. Además del código principal, debe incluir la creación del objeto Flask, y utilizar una de las configuraciones que hemos hecho.

```
app = Flask(__name__)

app.config.from_object('config.DevelopmentConfig')
app.secret_key = 'dev'
```

Figura 64. __Init__.py

Ahora una vez hecho esto, ya se puede comenzar a desarrollar la aplicación.

4.7.2 Configuración de la base de datos

Para realizar una aplicación web, es indispensable tener una base de datos donde almacenar todos los datos de la app. En este caso, se utiliza MySQL, alojado en XAMPP.

El primer paso, evidentemente, es descargar, instalar y configurar XAMPP, y una vez que tenemos la base de datos, hay que configurarla para usarla en nuestra aplicación.

Lo primero de todo, dentro de “__init__.py”, es crear la configuración de esta, que se ha realizado mediante mysql-connector:

```
#Configuración de la base de datos
db = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="emoscan_db"
```

Figura 65. Configuración de la base de datos

Una vez hecho esto, está lista para ser usada y crear las tablas y consultas SQL necesarias.

4.7.3 Búsqueda de datos de Twitter

Uno de los principales puntos de esta aplicación, es la búsqueda de tweets, es decir, la recopilación de estos, para guardarlos en la base de datos y analizarlos posteriormente.

Para ello, se ha utilizado la librería SNScrape, que cuenta con un módulo específico para la búsqueda de Tweets.

Lo primero de todo, es instalar la librería, la cual está contenida en git, y se instala con el siguiente comando: “ pip install git+https://github.com/JustAnotherArchivist/snsrape.git@786815dd05681e2421cd03aa9acf5ab5c773bce9”.

Una vez instalada, se va a importar su módulo de Twitter:

```
import snsrape.modules.twitter as sntwitter
```

Figura 66. Módulo de Twitter de SNSrape

Su uso es tan simple que en una línea podemos realizar la búsqueda:

```
scraper = sntwitter.TwitterSearchScrapper(consulta)
```

Figura 67. Uso de SNSrape

La consulta, previamente se ha obtenido mediante un formulario en HTML, que ha recogido la información necesaria para la consulta.

Posteriormente, para acceder a cada tweet obtenido, se debe realizar un bucle for que nos guarde cada tweet en la base de datos y se mostrará al usuario los datos más relevantes de cada tweet: la fecha, el contenido del tweet y el usuario que lo ha escrito.

4.7.4 Análisis de sentimientos en los tweets.

Una vez tenemos los tweets guardados, se le brindará al usuario la opción de analizarlos. Para ello, el contenido de los tweets tiene que pasar por un proceso de preprocesamiento del texto, que ya se explicó anteriormente, y una vez hecho este preprocesamiento se procederá a su análisis de sentimiento, donde se obtendrán unos valores entre -1 y 1, siendo cuanto más negativo el valor, más negativo el sentimiento, y cuanto más positivo, más positivo el sentimiento, siendo el 0 el neutro, es decir, no expresa sentimientos negativos ni positivos.

Para realizar este preprocesamiento y posterior análisis se ha utilizado la librería NLTK, después de hacer pruebas con esta y otras librerías como TextBlob, se ha llegado a la conclusión de que NLTK era la mejor opción y que mejores resultados proporciona. Para usarla, basta con instalarla como cualquier otro paquete, mediante “pip install NLTK”.

Además, debemos importarla de la siguiente manera:

```
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.sentiment import SentimentIntensityAnalyzer
```

Figura 68. Importación de NLTK

También hay que añadir las siguientes descargas:

```
# Descargar los recursos necesarios de NLTK
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('vader_lexicon')
```

Figura 69. Descargas necesarias de NLTK

Una vez hecho esto, simplemente hay que crear una función que realice el preprocesamiento de datos con los complementos de NLTK que se han importado y descargado, y con el texto preprocesado, aplicarla otra función que analice el texto y devuelva la polaridad, que son los valores de -1 a 1, y su sentimiento. Estos valores se almacenan en la base de datos, y se muestran al usuario en una tabla junto con el resto de los tweets.

4.7.5 Generación de estadísticas

Para la generación de estadísticas, se ha utilizado la librería Matplotlib, que, a la hora de generar gráficas en Python, es la mejor opción.

Una vez instalada la librería con “pip install matplotlib”, y haberla importado al proyecto, su uso es tan simple como esto:

```
# Crear la gráfica
labels = sentiment_counts.keys()
values = sentiment_counts.values()

plt.figure(figsize=(5, 4))
plt.bar(labels, values)
plt.xlabel('Sentimiento')
plt.ylabel('Cantidad de Tweets')
plt.title('Distribución de Sentimientos')
```

Figura 70. Uso de matplotlib

Primero se seleccionan los datos con los que se quiere generar la gráfica, que son en este caso, los diferentes valores que toman los sentimientos, que son negativos, neutros o positivos, y el número de sentimientos de cada tipo.

A continuación, se crea la gráfica indicando el tamaño, el tipo, en este caso de barras, que contiene el eje x y el eje y, y el título.

Para mostrar estas gráficas en HTML, se guardan en un PNG temporal, y se muestran en pantalla.

4.7.6 Aspectos importantes a tener en cuenta

Como en casi cualquier proyecto, a medida que se va avanzando, van surgiendo problemas con los cuales hay que lidiar, y solventar de la mejor manera posible. Durante la realización de este proyecto han surgido varios problemas los cuales han sido solucionados, así que este apartado va a tratar sobre los principales problemas que se han encontrado durante el desarrollo del proyecto.

El primero de todos, y de los más importantes, fue a la hora de recolectar información de Twitter, la idea inicial era utilizar la API de Twitter, la cual es sin duda la mejor herramienta para conseguir información de Twitter, ya que dicha API es oficial y está bien optimizada. Sin embargo, el problema radica en que, durante el mes de febrero de 2023, Twitter hizo un cambio en su API, y limitó casi por completo su uso de manera gratuita.

La versión gratuita solamente deja postear tweets o eliminarlos, lo cual no es suficiente para este proyecto, por lo que era necesario buscar una alternativa.

Cabe recalcar que debido a que la actualización de la API es tan reciente, al buscar información sobre cómo usarla, dicha información no estaba actualizada y fue todavía más difícil de solventar.

Para solucionarlo, se buscaron herramientas de web scraping, como la que se utiliza en este proyecto, SNScrape, la cual, lógicamente no es tan potente como la API de Twitter.

Como segundo problema, ocurre algo similar al primer problema, pero esta vez con el envío de correos electrónicos a través de la aplicación, ya que Google prohibió esta funcionalidad el 30 de mayo de 2022 en aplicaciones de terceros poco seguras como lo es esta.

En este caso, no se ha solucionado el problema, ya que actualmente no es algo prioritario, y se dejará como propuesta futura.

El tercer problema fue el idioma de los tweets, en un principio el objetivo era analizar tweets en español o en inglés, sin embargo, los modelos preentrenados para la clasificación de sentimientos no funcionan correctamente en español por lo que se tuvo que descartar esta opción, ya que desarrollar un modelo de este estilo no es viable para este proyecto. Por tanto, solo se realiza el análisis en inglés.

El último problema surge durante el primer fin de semana de Julio. Elon Musk ha prohibido temporalmente el scraping en Twitter, por lo que de momento, la aplicación no va a funcionar correctamente ya que no se va a poder buscar ningún tweet. Actualmente esto no tiene solución y la única forma de solucionar esto será esperar que Twitter vuelva a activar esta funcionalidad.

4.7.7 Código fuente

El código de la aplicación se encuentra alojado en github y se puede acceder mediante este enlace:

<https://github.com/sergielx/EmoScan>



5. CONCLUSIONES Y PROPUESTAS FUTURAS

5.1 Conclusiones

En este trabajo se ha creado una aplicación web capaz de detectar sentimientos en Twitter.

Para el desarrollo de la aplicación, primero se realizó un estudio previo en el que se analizaron los métodos para realizar todo el proceso de análisis de sentimiento, quedando dividido en 3 grandes puntos: la extracción de datos, el procesamiento del lenguaje natural, y el análisis de sentimientos. Posteriormente se realizó otro estudio sobre las tecnologías que se utilizan para desarrollar la aplicación: Python, con el uso del framework web Flask, base de datos en MySQL para el backend, y HTML, CSS, bootstrap y JS y JQuery para el front-end.

Una vez realizado los estudios necesarios, y el desarrollo del proyecto mediante ingeniería del software, se comenzó a desarrollar la aplicación.

Durante el desarrollo, como en todos los proyectos software, surgieron varios problemas que se fueron solucionando.

Se ha conseguido el objetivo de crear una aplicación web capaz de detectar sentimientos en tweets, sin embargo, hay que tener en cuenta las limitaciones que tiene el análisis de sentimientos actualmente, ya que los tweets suelen ser muy subjetivos y suelen contener un lenguaje informal que puede llegar a ser difícil de entender para los modelos de análisis, por tanto, el porcentaje de acierto puede no ser el deseado.

También se cuentan con limitaciones económicas, ya que, al no disponer de financiación para este proyecto, podemos encontrarnos con problemas como el mencionado anteriormente con la API de Twitter, el no disponer de un modelo más avanzado al comprar herramientas de pago, o la creación de la interfaz de usuario, desde cero, en vez de tener una plantilla comprada.

Sin embargo, a pesar de todas estas limitaciones, se ha cumplido el objetivo propuesto, y personalmente, me encuentro satisfecho de haber desarrollado este proyecto. Quiero destacar en este punto que, para realizar este proyecto contaba con ciertas carencias de conocimientos que no se explican durante mis estudios de Grado, como el lenguaje Python, conocimientos básicos sobre Inteligencia Artificial, sobre Procesamiento de Lenguaje Natural, sobre análisis de sentimientos, o sobre Machine Learning. Pero con la democratización del conocimiento que nos permite Internet hoy en día, considero que con este proyecto se demuestra que un Graduado en Ingeniería Informática en Tecnologías de la Información dispone de la formación de base suficiente para lograr desarrollar un proyecto de estas características, y adquirir en un breve espacio de tiempo los conocimientos y la experiencia necesaria en este tipo de aplicaciones que podrá utilizar en su futuro profesional.

5.2 Propuestas futuras

Al realizar este proyecto, el desarrollo de la aplicación se ha hecho de tal manera que es muy fácilmente escalable, es decir, las funcionalidades hechas se pueden mejorar de una manera muy simple, y se pueden añadir más funcionalidades, estas son algunas de las que se podrían añadir:

- Posibilidad de contactar con la empresa mediante un formulario de contacto, el cual está hecho, pero no tiene funcionalidad.
- Implementar el análisis de sentimientos en español, un punto que actualmente es complicado, pero posible.
- Añadir la opción de descargar en un pdf o csv las estadísticas.
- En caso de contar con más presupuesto, mejorar tecnologías, como el uso de la API de Twitter en lugar de SNScrape.
- Añadir la opción de recuperación de contraseña, para la cual también se necesita poder enviar correos electrónicos.
- Implementar el análisis en otras plataformas como Facebook, Instagram, o reseñas de Google.



6. BIBLIOGRAFÍA

[1]Asale, R.-. (s. f.). *emoción* / *Diccionario de la lengua española*. «Diccionario de la lengua española» - Edición del Tricentenario. <https://dle.rae.es/emoci%C3%B3n>

[2]Ekman, P. (1992). An argument for basic emotions. *Cognition & Emotion*, 6(3-4), 169-200. <https://doi.org/10.1080/02699939208411068>

[3]James, W. (1890b). *The Principles of Psychology*. https://pure.mpg.de/pubman/item/item_2350624_2/component/file_2350623/James_1890_the_principles.pdf

[4]Ueber Gemüthsbewegungen, eine Psycho-Physiologische Studie; von Dr. C. Lange Autorisirte Uebersetzung, von Dr. H. Kurella Leipzig: T. Thomas, 1887. (1888). *The journal of mental science*, 34(147), 426-430. <https://doi.org/10.1192/bjp.34.147.426-a>

[5]Lazarus, R. S. (1991). *Emotion and Adaptation*. <http://ci.nii.ac.jp/ncid/BA13146090>

[6]LeDoux, J. E. (2004). *The Emotional Brain: The Mysterious Underpinnings of Emotional Life*. <http://ci.nii.ac.jp/ncid/BA70677860>

[7]J, P. P., & Gardey, A. (2017). API - Qué es, definición y concepto. *Definición.de*. <https://definicion.de/api/>

[8]*Use Cases, Tutorials, & Documentation*. (s. f.). Twitter Developer Platform. <https://developer.twitter.com/en>

[9]Zhao, B. (2017). Web scraping. *Encyclopedia of big data*, 1-3. https://www.researchgate.net/profile/Bo-Zhao-3/publication/317177787_Web_Scraping/links/5c293f85a6fdccfc7073192f/Web-Scraping.pdf

[10]Richardson, L. (s. f.). *Beautiful Soup*. <https://www.crummy.com/software/BeautifulSoup/>

[11]Diana. (s. f.). *Home*. Unicode. <https://home.unicode.org/>

[12]colaboradores de Wikipedia. (2023). UTF-8. *Wikipedia, la enciclopedia libre*. <https://es.wikipedia.org/wiki/UTF-8>

[13] JustAnotherArchivist. (s. f.). *GitHub - JustAnotherArchivist/snsrape: A social networking service scraper in Python*. GitHub. <https://github.com/JustAnotherArchivist/snsrape>

- [14] J, P. P., & Gardey, A. (2021). Sentimiento - Qué es, definición y concepto. *Definición.de*. <https://definicion.de/sentimiento/>
- [15] Pang, B., & Lee, L. (2008). *Opinion Mining and Sentiment Analysis*. Now Publishers Inc.
- [16] Medhat, W., Hassan, A. E., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113. <https://doi.org/10.1016/j.asej.2014.04.011>
- [17] ¿Qué es el procesamiento del lenguaje natural? | IBM. (s. f.). <https://www.ibm.com/es-es/topics/natural-language-processing>
- [18] ¿Qué es la inteligencia artificial o IA? | Google Cloud | Google Cloud. (s. f.). Google Cloud. <https://cloud.google.com/learn/what-is-artificial-intelligence?hl=es-419>
- [19] Alejandra, N. C. (2022, 23 junio). *Técnicas de machine learning para procesamiento del lenguaje natural*. Universidad de Oviedo. <http://hdl.handle.net/10651/63977>
- [20] Battocchia, M. (2020, 23 julio). Preprocesamiento de texto para NLP (parte 1). *datitos*. <https://matiasbattocchia.github.io/datitos/Preprocesamiento-de-texto-para-NLP-parte-1.html#El-pre-procesamiento-hasta-ahora>
- [21] Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space*. ArXiv Preprint arXiv:arXiv:1301.3781v3 [cs.CL]. <https://arxiv.org/pdf/1301.3781.pdf>
- [22] colaboradores de Wikipedia. (2021). Modelo–vista–controlador. *Wikipedia, la enciclopedia libre*. <https://es.wikipedia.org/wiki/Modelo%20%80%93vista%20%80%93controlador>
- [23] Nestrategia, Agencia de posicionamiento web en Madrid. (2022). ¿Qué es Back End, Front End y Back Office y por qué es importante para tu web? *Agencia Inbound Marketing Madrid*. <https://nestrategia.com/desarrollo-web-back-end-front-end/#que-es-el-back-end>
- [24] *Welcome to Python.org*. (2023, 1 julio). Python.org. <https://www.python.org/>
- [25] *Welcome to Flask — Flask Documentation (2.3.x)*. (s. f.). <https://flask.palletsprojects.com/en/2.3.x/>
- [26] *XAMPP Installers and Downloads for Apache Friends*. (s. f.). <https://www.apachefriends.org/es/index.html>
- [27] MariaDB.org. (2019, 13 noviembre). *MariaDB Foundation - MariaDB.org*. <https://mariadb.org/>

- [28] ▷ *¿Qué es el Front-end y para qué sirve? [Diccionario]* . (s. f.). Agencia de marketing online | APPYWEB. <https://www.appyweb.es/diccionario/front-end/>
- [29] *HTML: Lenguaje de etiquetas de hipertexto* | MDN. (2023, 11 junio). <https://developer.mozilla.org/es/docs/Web/HTML>
- [30] *CSS* / MDN. (2023, 13 marzo). <https://developer.mozilla.org/es/docs/Web/CSS>
- [31] Contributors, M. O. J. T. A. B. (s. f.). *Bootstrap*. <https://getbootstrap.com/>
- [32] *JavaScript Guide - JavaScript* | MDN. (2023, 1 mayo). <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>
- [33] JS Foundation - js.foundation. (s. f.). *jQuery*. <https://jquery.com/>
- [34] Riveros, A. (2023). ¿En qué consiste el ciclo de vida de un proyecto? *EALDE Business School*. <https://www.ealde.es/definicion-ciclo-vida-proyecto/>
- [35] Rodó, P. (2022). Diagrama de Gantt. *Economipedia*. <https://economipedia.com/definiciones/diagrama-de-gantt.html>
- [36] Solutions, V., & Jain, A. (2023). Qué es la especificación de requisitos: definición, mejores herramientas y técnicas | Guía. *Visure Solutions*. <https://visuresolutions.com/es/blog/requirements-specification/#:~:text=Una%20especificaci%C3%B3n%20de%20requisitos%20es, trabajo%20futuro%20en%20el%20proyecto>
- [37] Vega, M. (2010). Casos de uso. *Recuperado el, 12*. <https://lsi2.ugr.es/~mvega/docis/casos%20de%20uso.pdf>
- [38] *Inicio* - Canva. (n.d.). Canva. <https://www.canva.com/>

7. ANEXO: GUÍA DE USUARIO

Esta guía muestra paso a paso y con capturas de pantalla como debe interactuar el usuario con la aplicación web:

7.1 Landing Page

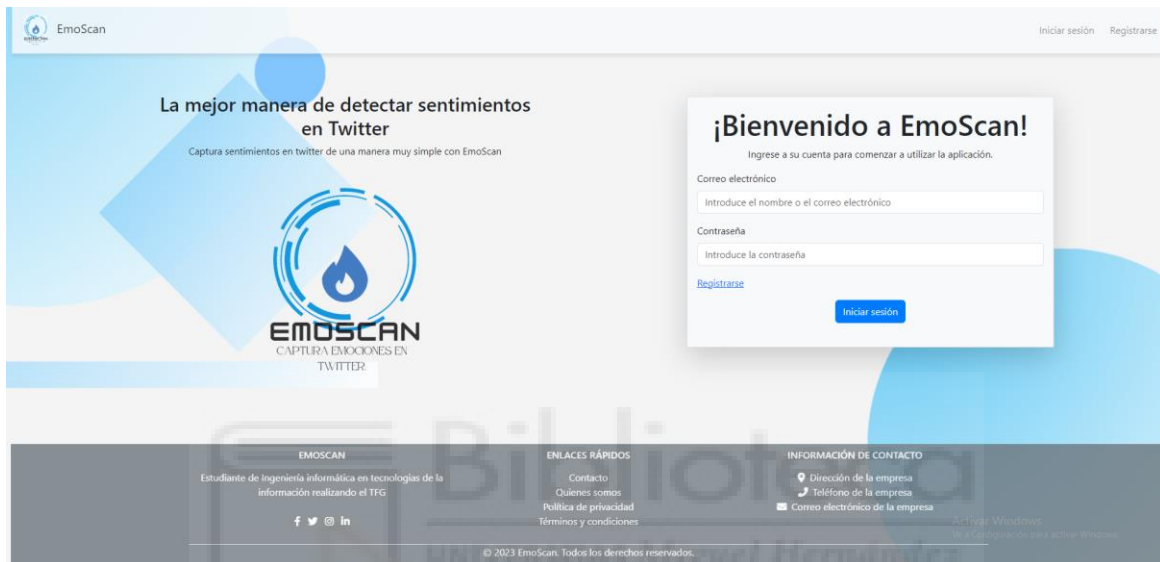


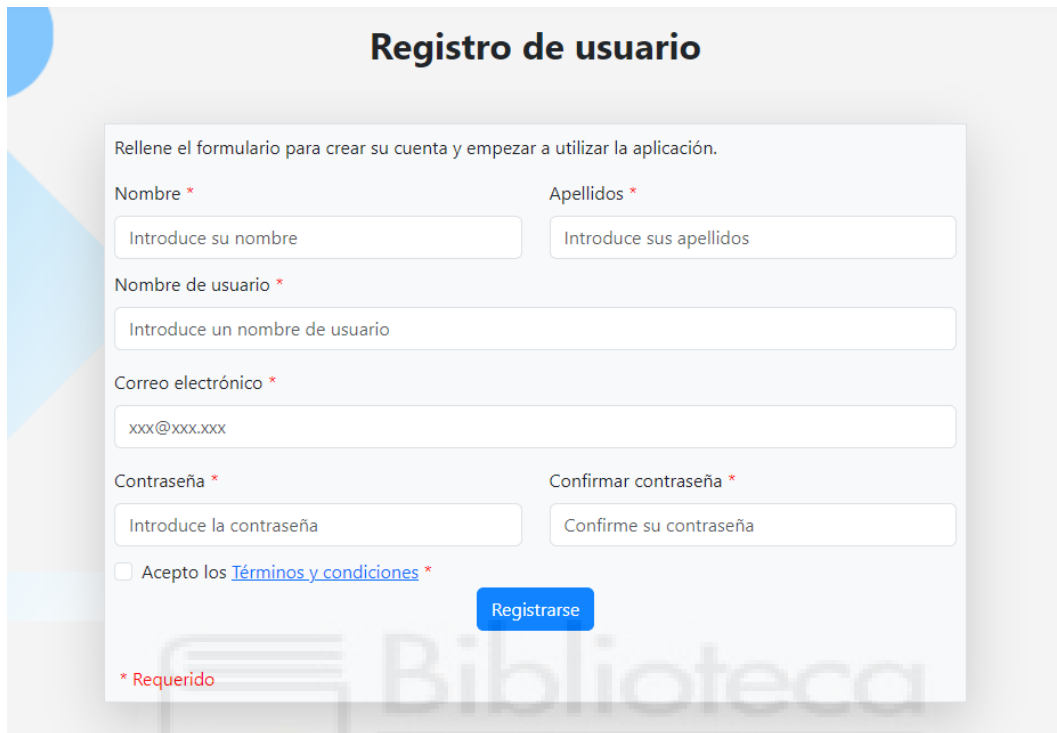
Figura 71. Landing Page

El usuario cuando vaya a visitar la web accederá a esta página.

Aquí el usuario podrá identificarse, si ya posee una cuenta en la aplicación, o en su defecto, podrá registrarse en caso de no tener una cuenta.

7.2 Identificación de usuario

7.2.1 Registro de usuario



Registro de usuario

Rellene el formulario para crear su cuenta y empezar a utilizar la aplicación.

Nombre * Apellidos *

Nombre de usuario *

Correo electrónico *

Contraseña * Confirmar contraseña *

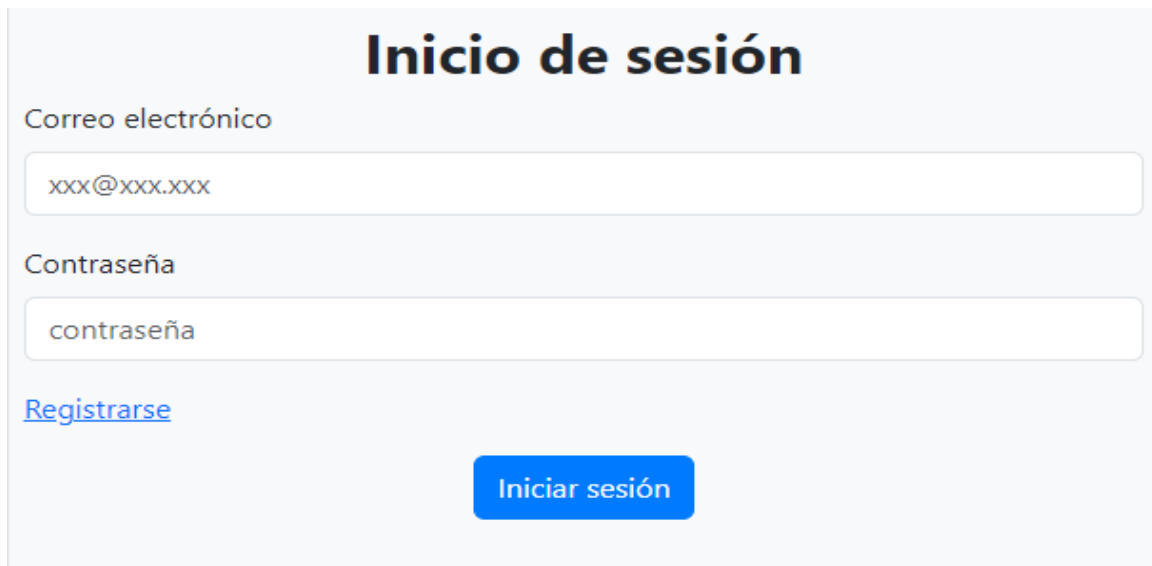
Acepto los [Términos y condiciones](#) *

* Requerido

Figura 72. Registro de usuario

El usuario, en caso de no tener una cuenta, podrá registrarse completando los campos necesarios para crear una cuenta.

7.2.2 Inicio de sesión



Inicio de sesión

Correo electrónico

xxx@xxx.xxx

Contraseña

contraseña

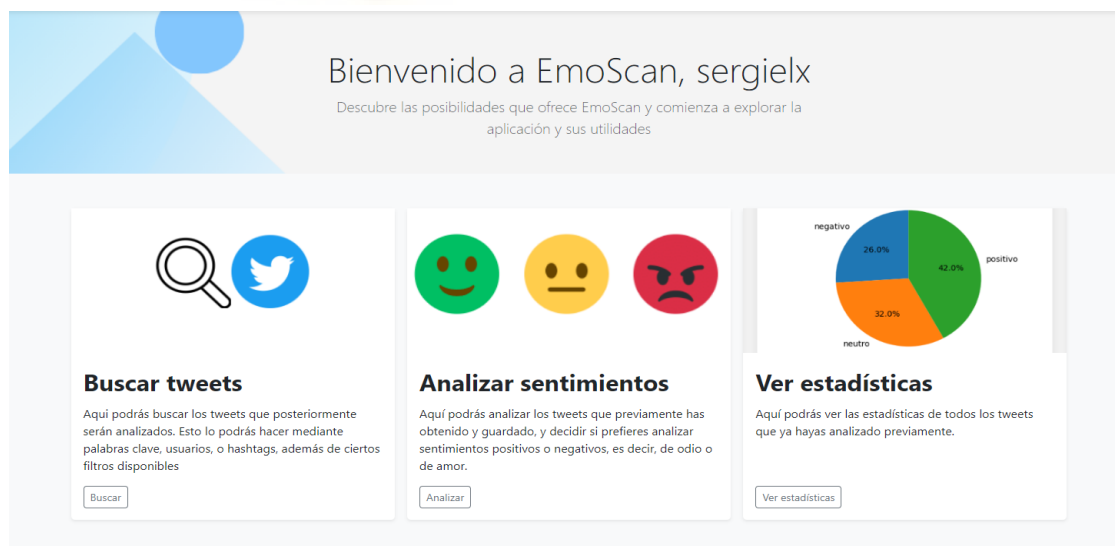
[Registrarse](#)

Iniciar sesión

Figura 73. Inicio de sesión

En caso de ya tener una cuenta, el usuario podrá ingresar a la aplicación utilizando su correo electrónico y su contraseña.

7.3 Página principal



Bienvenido a EmoScan, sergielx

Descubre las posibilidades que ofrece EmoScan y comienza a explorar la aplicación y sus utilidades

Buscar tweets

Aquí podrás buscar los tweets que posteriormente serán analizados. Esto lo podrás hacer mediante palabras clave, usuarios, o hashtags, además de ciertos filtros disponibles

Analizar sentimientos

Aquí podrás analizar los tweets que previamente has obtenido y guardado, y decidir si prefieres analizar sentimientos positivos o negativos, es decir, de odio o de amor.

Ver estadísticas

Aquí podrás ver las estadísticas de todos los tweets que ya hayas analizado previamente.

Sentimiento	Porcentaje
negativo	26.0%
positivo	42.0%
neutro	32.0%

Figura 74. Página principal

Una vez se ha realizado el registro o se ha iniciado sesión, se accede a esta página donde el usuario tendrá las funcionalidades principales de la aplicación, que son buscar tweets, analizar sentimientos, o ver estadísticas.

7.4 Buscar tweets

7.4.1 Página de búsqueda

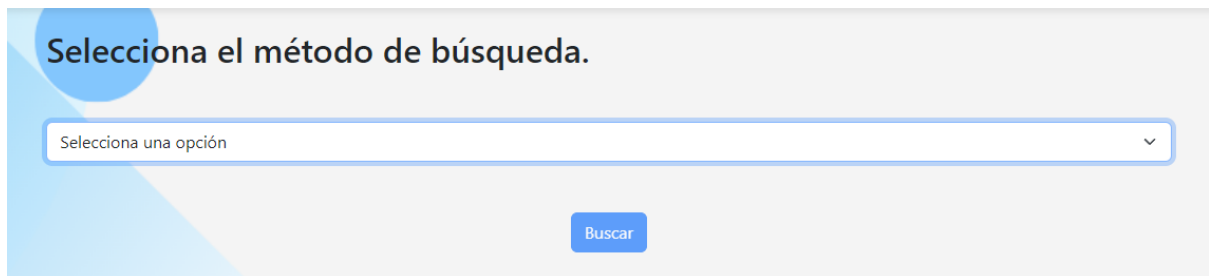


Figura 75. Buscar tweets 1

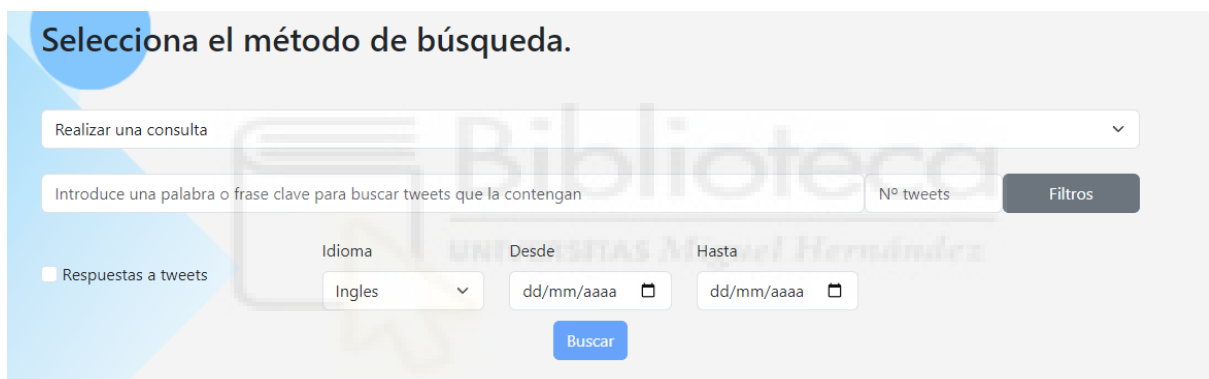


Figura 76. Buscar tweets 2

La primera funcionalidad principal de la aplicación es buscar tweets. Al acceder a la sección de buscar tweets, se encuentra esta página en la que se pedirá al usuario elegir una opción, entre realizar una consulta, buscar un usuario, o buscar un hashtag.

Una vez seleccionado el método de búsqueda, se le pedirá al usuario introducir la consulta deseada, el número de tweets (por defecto es 100), y aplicar ciertos filtros si así lo desea el usuario.

7.4.2 Resultados de la búsqueda

Resultados de la búsqueda

Parámetros utilizados en la búsqueda

Tipo de consulta: Palabra clave
Consulta: elon
Número de tweets: 100
Mostrar respuestas: No
Idioma: Inglés
Desde: No especificado
Hasta: No especificado

Figura 77. Resultado de búsqueda 1

fecha	tweet	usuario
30/06/2023	Elon got served ??????????????	Emmbee_brown
30/06/2023	DOX Coin Thailand's Securities and Exchange ??? around meme coins affects their value. or sheer profit, and hype builds quickly. as a joke coin based on Elon Musk's dog but has #DOXcoin https://t.co/35yklcFDXc	SarahBaker27739
30/06/2023	DOX Coin Thailand's Securities and Exchange ??? around meme coins affects their value. or sheer profit, and hype builds quickly. as a joke coin based on Elon Musk's dog but has #DOXcoin https://t.co/NOFa5nuIU5	JackieStev74577
30/06/2023	DOX Coin Thailand's Securities and Exchange ??? around meme coins affects their value. or sheer profit, and hype builds quickly. as a joke coin based on Elon Musk's dog but has #DOXcoin https://t.co/0UN20tGDKL	NicoleThom47253
30/06/2023	Elon looks so cool in this ???art https://t.co/HT8Wdop0GH	ladooyar
30/06/2023	DOX Coin and has only been successfully achieved ??? They're often created as a joke, and Are they a viable venture? And how should one as a joke coin based on Elon Musk's dog but has #DOXcoin https://t.co/IRR6DNAXV0	Christi56554940
30/06/2023	Dear Elon Musk...What about that "Craft"...? See Apollo 20...? Want to know about the City....?	francescowdrey
30/06/2023	Who told Elon to hire the WEF CEO? That's the question. No one in their right mind would pick a leftist, WEF, hand puppet if they were concerned about free speech. My guess: The regime is calling the shots. They threaten him with pulling his subsidies u	nbbb2385
30/06/2023	It's the last day of PRIDE month, 2023, and I have a couple things to say: 1. I will never forgive ANYONE that did not vote for Hillary Clinton, no matter the excuse. 2. Motherfuck Elon Musk That is all.	MotherfuckMusk
30/06/2023	!! Italy has offered to host Elon Musk vs Mark Zuckerberg in Rome's Colosseum ?????????????? https://t.co/XFyL1jTKf1	lfnBoxing

< 1 2 3 4 5 ... 9 10 >

Volver a buscar
Guardar y analizar
Guardar y volver al menú principal

Figura 78. Resultado de búsqueda 2

Una vez realizada la búsqueda, se mostrarán los resultados, y los parámetros utilizados. Los parámetros son los establecidos previamente, mientras que la tabla de resultados muestra la fecha del tweet, el contenido del tweet, y el usuario del tweet.

Además, los resultados se muestran paginados de diez en diez para favorecer su visualización y organización.

Posteriormente se encuentran tres botones, uno para volver a la búsqueda de tweets y no guardar los resultados, otro para guardar los resultados y analizarlos, y otro para guardar los resultados y volver al menú principal.

7.5 Análisis de sentimientos

7.5.1 Página de análisis



Figura 79. Análisis de sentimientos

La segunda funcionalidad principal de esta aplicación es analizar sentimientos.

Una vez se accede a la sección de analizar sentimientos se mostrarán todas las opciones con las que se ha realizado una búsqueda, y su fecha.

Junto a estas opciones hay un botón para eliminar, es decir, se puede eliminar dicha consulta.

Para seleccionar una opción y realizar su análisis hay que pulsar sobre dicha opción y se accederá a la página con los resultados del análisis.

7.5.2 Resultados del análisis de sentimientos

24/06/2023	Conservatives, especially #MAGA have claimed for years that #Putin is a strong man with a better military than ours. https://t.co/11FJpk8Ocl	BlueContrino	0.7351	positivo
24/06/2023	Would love for Rick Astley to release a Putin special Wagner gonna give you up Wagner gonna let you down Wagner gonna run around and desert you	StuntcockV	0.7845	positivo
24/06/2023	The Turkish readout of the phone call between President Recep Tayyip Erdogan and Russian President Vladimir Putin. During the meeting, the latest developments in Russia were discussed. President Erdogan underlined the importance of acting in a calm and	YusufErim34	0.5859	positivo
24/06/2023	Putin is losing leverage and power, who thinks he will give up his nukes? If Prigo gets nukes and power over Russia who knows what he will do? He's reactive, and a business man, not a military man.	Librarian6301	-0.4515	negativo
24/06/2023	What would it be like to see Ukraine and Wagner join forces to defeat Putin? And where would a republican leaders be after that? If sold their soul to the devil, may all these evil people rest in hell.	ER_RN_TNCCI	-0.9332	negativo
24/06/2023	What Russian President Vladimir Putin told Cyril Ramaphosa in their behind-the-scenes meeting, in the #SundayTimesZA tomorrow.	SundayTimesZA	0.0	neutro
24/06/2023	Wagner mutiny: Putin vows swift retribution for 'treason' I suppoPrigozhin concerning this matter. Russia & Putin had NO justification to Invade Ukraine. https://t.co/gvT4wgxbjz via @YouTube	DJx1555	-0.2519	negativo

« 1 2 3 4 5 ... 9 10 »

[Ver estadísticas](#) [Guardar y volver al menú principal](#)

Figura 80. Resultados del análisis de sentimientos

Esta página es exactamente igual que la de resultados de la búsqueda, pero se ha añadido a la tabla, los campos de polaridad, y sentimiento.

La polaridad va entre -1 y 1, cuanto más próximo a -1 es el valor, más negativo es el comentario y cuanto más próximo al 1, más positivo es el comentario.

Esta vez aparecen dos botones, uno sirve para mostrar estadísticas y el otro para volver al menú principal.

7.6 Estadísticas

7.6.1 Página de estadísticas



Figura 81. Estadísticas

La última de las funcionalidades principales es ver las estadísticas. Esta sección es igual que la de análisis de sentimientos, con la diferencia de que aquí aparecen solamente las opciones a las que ya se ha realizado un análisis.

7.6.2 Resultados de las estadísticas



Figura 82. Resultados de estadísticas

Esta página muestra las estadísticas de la consulta seleccionada. Se encuentra un gráfico que muestra el valor absoluto de los sentimientos encontrados, otro gráfico que muestra el porcentaje de los sentimientos encontrados, y otro que muestra el número de sentimientos encontrados según la fecha.

7.7 Perfil de usuario y cierre de sesión

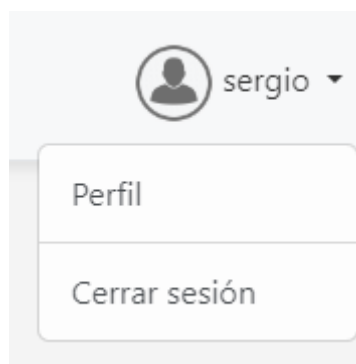


Figura 83. Desplegable perfil y cerrar sesión

En el menú superior de la web, al pulsar en el nombre de usuario, se muestra un desplegable que nos permite acceder al perfil, o cerrar sesión.

Para cerrar sesión simplemente hay que pulsar dicha opción y confirmarla.

7.7.1 Perfil de usuario

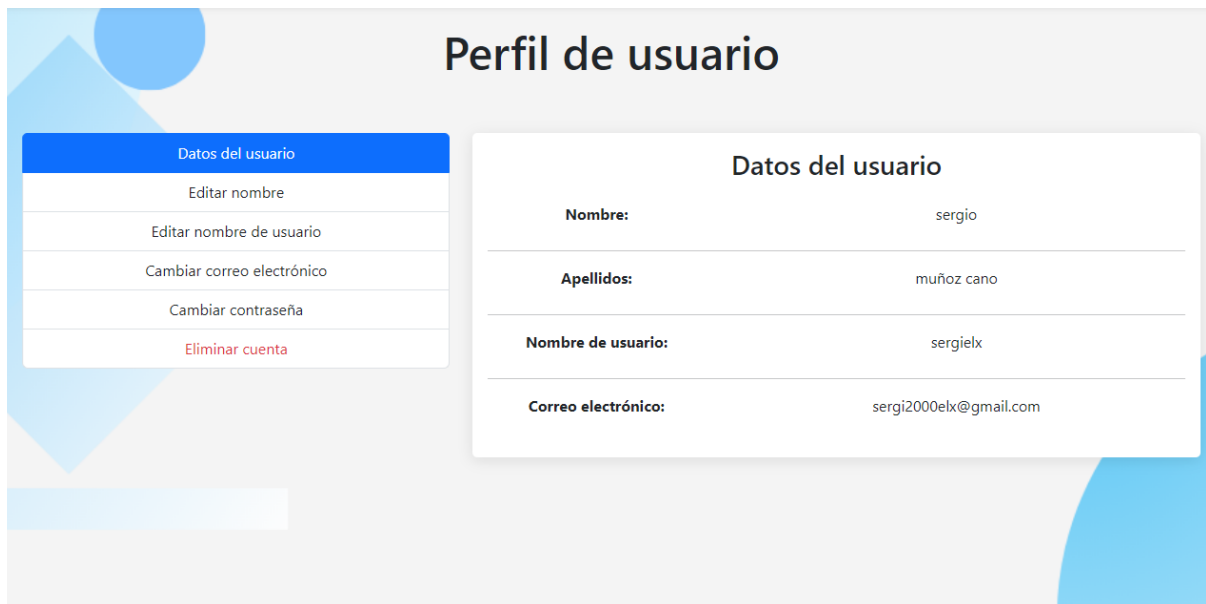


Figura 84. Perfil de usuario

Dentro del perfil de usuario, se encuentran diferentes secciones.

Primeramente, se encuentran los datos del usuario, posteriormente se pueden modificar los datos de la cuenta, y por último se puede eliminar la cuenta.