



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**Aplicación de análisis de
reseñas basado en modelos de
lenguaje grandes mediante
prompt engineering.**



Presentado por Teodoro Ricardo García
Sánchez
en Universidad de Burgos — 23 de enero
de 2024

Tutores: Dra. Virginia Ahedo García y D. Jose
Ignacio Santos Martín

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	9
Apéndice B Especificación de Requisitos	13
B.1. Introducción	13
B.2. Objetivos generales	13
B.3. Catálogo de requisitos	13
B.4. Casos de uso	15
Apéndice C Especificación de diseño	23
C.1. Introducción	23
C.2. Diseño de datos	23
C.3. Diseño procedimental	24
C.4. Diseño arquitectónico	25
Apéndice D Documentación técnica de programación	31
D.1. Introducción	31
D.2. Estructura de directorios	31
D.3. Manual del programador	32

D.4. Compilación, instalación y ejecución del proyecto	33
D.5. Pruebas del sistema	34
Apéndice E Documentación de usuario	49
E.1. Introducción	49
E.2. Requisitos de usuario	49
E.3. Instalación	49
E.4. Manual del usuario	49
Apéndice F Anexo de sostenibilización curricular	63
F.1. Introducción	63
Bibliografía	65

Índice de figuras

A.1. Tareas sprint 0	2
A.2. Burndown chart sprint 0	2
A.3. Tareas sprint 1	3
A.4. Burndown chart sprint 1	3
A.5. Tareas sprint 2	3
A.6. Burndown chart sprint 2	4
A.7. Tareas sprint 3	4
A.8. Burndown chart sprint 3	5
A.9. Tareas sprint 4	5
A.10. Burndown chart sprint 4	6
A.11. Tareas sprint 5	6
A.12. Burndown chart sprint 5	7
A.13. Tareas sprint 6	7
A.14. Burndown chart sprint 6	8
A.15. Tareas sprint 7	8
A.16. Burndown chart sprint 7	9
 B.1. Diagrama de casos de uso	 15
 C.1. Arquitectura de la base de datos	 23
C.2. Login	24
C.3. Administración	25
C.4. Gestor	25
C.5. Arquitectura de la aplicación	26
C.6. Pantalla de login.	27
C.7. Pantalla usuario gestor.	28
C.8. Pantalla de administración.	29

D.1. Resultado pytest	34
---------------------------------	----

Índice de tablas

B.1. CU-1 Registro de usuarios.	16
B.2. CU-2 Login de usuarios.	17
B.3. CU-3 Información de la aplicación.	18
B.4. CU-04 Cargar reseñas de un archivo.	19
B.5. CU-5 Puntuar los textos.	20
B.6. CU-6 Almacenar resultados.	21
B.7. CU-07 Información estadística.	21
B.8. CU-08 Configurar el idioma de la aplicación.	22
D.1. CP-1 Registro de usuarios.	36
D.2. CP-2 Registro de usuarios incorrecto.	37
D.3. CP-3 Registro de usuarios: email incorrecto.	38
D.4. CP-4 Registro de usuarios: contraseña no es válida.	39
D.5. CP-5 Registro de usuarios(administrador).	40
D.6. CP-6 Registro de usuarios (administrador): usuario ya existe. . .	41
D.7. CP-7 Registro de usuarios (administrador): correo incorrecto. .	42
D.8. CP-8 Registro de usuarios.	43
D.9. CP-9 Login de usuarios.	44
D.10.CP-10 Login de usuario: usuario no existe.	45
D.11.CP-11 Login de usuario: contraseña incorrecta.	46
D.12.CP-1 Registro de usuarios.	47

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Este primer apartado de los anexos se ha dedicado a la explicación del desarrollo del proyecto, recogido en planificación temporal, y a la viabilidad del proyecto, dividida en viabilidad económica donde se hará una aproximación de los costes que conlleva el proyecto y viabilidad legal donde se debe tener en cuenta la legislación que puede afectar al proyecto.

A.2. Planificación temporal

Antes de comenzar el proyecto se decidió utilizar una metodología ágil para gestionarlo. Se han planificado sprints de dos semanas de duración. Al final de cada sprint se ha hecho revisión para comprobar si se habían conseguido los objetivos marcados al inicio y plantear las tareas a realizar para el nuevo sprint. Para la gestión del proyecto se han utilizado GitHub y Zube. El tablero que provee Zube ha permitido que visualmente la organización del proyecto haya sido más sencilla. Desde github se han vinculado los “commits” con cada tarea para hacer más fácil su seguimiento.

Sprint 0 - del 2 de Octubre de 2023 al 15 de Octubre de 2023

Este sprint 0 se ha dedicado a tareas de preparación. Se ha creado el repositorio en Github, un proyecto en Zube y se ha instalado un entorno

Latex para la documentación. Por otro lado me he dedicado a leer la documentación relacionada y realizar los cursos recomendados.

#12	Leer primeros pasos para TFG de Informática	🕒 1	📁 Archive	🔖	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 0	Documentación	Investigación
#4	Crear repositorio inicial en GitHub	🕒 #3	🕒 1	📁 Archive	🔖	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 0	Configuración
#5	Crear proyecto en Zúbe	🕒 #4	🕒 1	📁 Archive	🔖	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 0	Configuración
#3	Realizar los cursos recomendados	🕒 #2	🕒 3	📁 Archive	🔖	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 0	Investigación
#1	Examinar la documentación proporcionada para el proyecto.	🕒 #1	🕒 5	📁 Archive	🔖	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 0	Investigación
#8	Crear entorno inicial editor Latex	🕒 #6	🕒 1	📁 Archive	🔖	Crear la memoria del proyecto # Sprint 0	Configuración Documentación
#7	Configurar plantilla inicial Memoria (Latex)	🕒 #5	🕒 1	📁 Archive	🔖	Crear la memoria del proyecto # Sprint 0	Documentación

Figura A.1: Tareas sprint 0



Figura A.2: Burndown chart sprint 0

Sprint 1 - del 16 de Octubre de 2023 al 29 de Octubre de 2023

En el sprint 1 he empezado a definir el catálogo de especificaciones y empezar a pensar cómo implementar tanto el prototipo como la aplicación de usuario. También he buscado información sobre trabajos relacionados.

<input type="checkbox"/> #16	Investigar el uso de collab	#12	@ 1	Archive	🚩	✖ Crear la memoria del proyecto # Sprint 1 Investigación
<input type="checkbox"/> #13	Evaluar trabajos relacionados.	#11	@ 2	Archive	🚩	✖ Crear la memoria del proyecto # Sprint 1 Documentación Investigación
<input type="checkbox"/> #14	Definir catálogo de especificaciones	#10	@ 5	Archive	🚩	✖ Crear la memoria del proyecto # Sprint 1 Documentación
<input type="checkbox"/> #15	Evaluar tecnologías para implementar la aplicación.	#13	@ 5	Archive	🚩	✖ Crear la memoria del proyecto # Sprint 1 Investigación

Figura A.3: Tareas sprint 1



Figura A.4: Burndown chart sprint 1

Sprint 2 - del 30 de Octubre de 2023 al 12 de Noviembre de 2023

En este segundo sprint por un lado he empezado a definir la arquitectura de la aplicación y por otro he empezado con el prototipo en collab. Para ello primero he tenido que buscar un dataset que nos valga y un modelo que se pueda usar en collab. Después de mucho buscar decidí usar un modelo llama (con llama.cpp) y un dataset de hugging face para hacerlo más portable. Si la máquina asignada tiene tarjeta gráfica, se usará.

<input type="checkbox"/> #17	Implementar prototipo en collab	#14	@ 1	Archive	🚩	✖ Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 2 Desarrollo
<input type="checkbox"/> #10	Evaluar el uso de diferentes Modelos LLM	#8	@ 5	Archive	🚩	✖ Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 2 Investigación
<input type="checkbox"/> #9	Buscar datasets	#7	@ 3	Archive	🚩	✖ Preparación de conjuntos de datos # Sprint 2 Investigación
<input type="checkbox"/> #18	Definir arquitectura de la aplicación	#23	@ 5	Archive	🚩	✖ Desarrollar una aplicación de usuario # Sprint 2 Documentación

Figura A.5: Tareas sprint 2

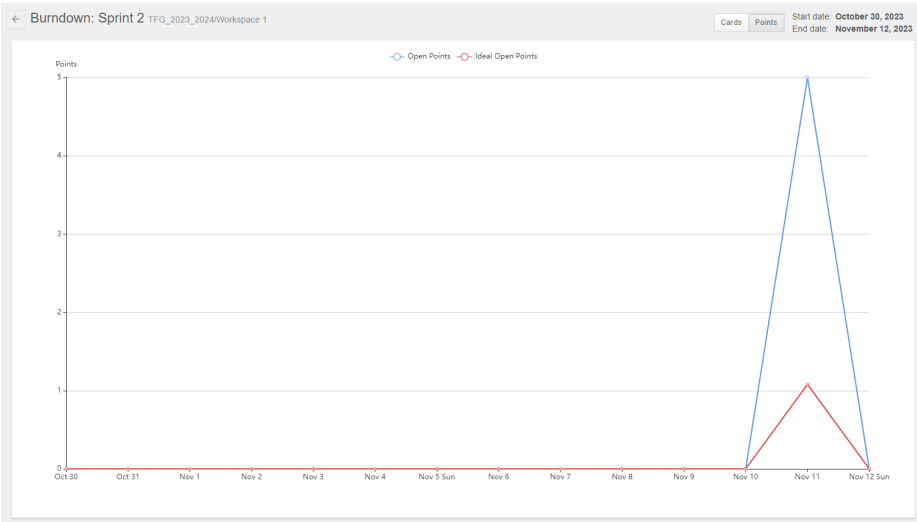


Figura A.6: Burndown chart sprint 2

Sprint 3 - del 13 de Noviembre de 2023 al 26 de Noviembre de 2023

En este sprint empecé a definir los casos de uso, el interfaz de usuario y empezar a pensar en el modelo de base de datos. En este momento decidí usar un ORM para hacer más flexible la aplicación e independiente del SGBD. Así sería más fácil cambiar de SGBD en caso de encontrar algún problema. En este momento también decidí empezar a usar postgres y docker por simplicidad.

<input type="checkbox"/>	#24	Definir casos de uso	👤 #33	🕒 3	📁 Archive	🚩	🔗 Crear la memoria del proyecto # Sprint 3 Documentación
<input type="checkbox"/>	#22	Diagrama de casos de uso	👤 #32	🕒 3	📁 Archive	🚩	🔗 Crear la memoria del proyecto # Sprint 3 Documentación
<input type="checkbox"/>	#25	Definir modelo de base de datos	👤 #27	🕒 3	📁 Archive	🚩	🔗 Crear la memoria del proyecto # Sprint 3 Documentación
<input type="checkbox"/>	#21	Definir interfaz de usuario	👤 #29	🕒 3	📁 Archive	🚩	🔗 Desarrollar una aplicación de usuario # Sprint 3 Documentación
<input type="checkbox"/>	#11	Investigar el uso de LangChange en el proyecto	👤 #9	🕒 2	📁 Archive	🚩	🔗 Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 3 Investigación

Figura A.7: Tareas sprint 3

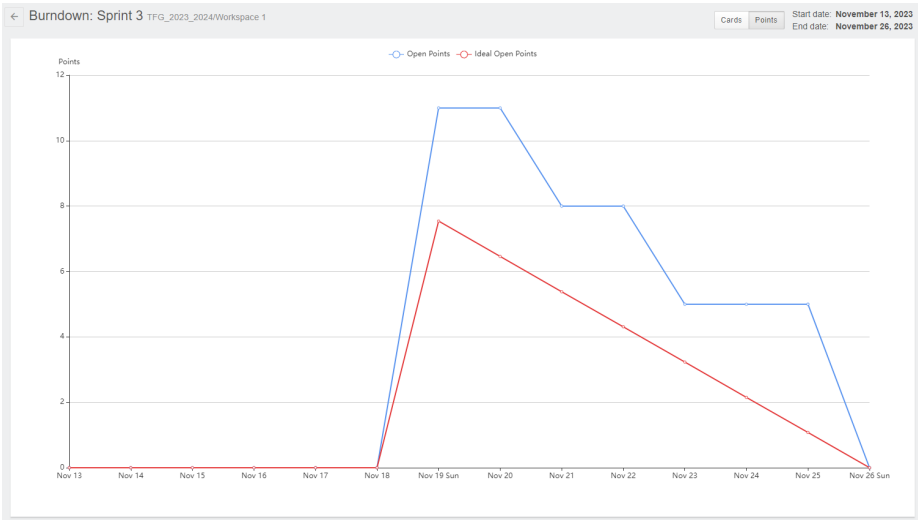


Figura A.8: Burndown chart sprint 3

Sprint 4 - del 27 de Noviembre de 2023 al 10 de Diciembre de 2023

En este sprint ya empecé más en serio con el desarrollo de la aplicación de usuario con la implementación de la pantalla de login, la pantalla de administración y la pantalla de ejecución del proceso. Por otro lado también empecé a con la implmentación de la base de datos, el cliente OpenAI y la elección de un hosting.

<input type="checkbox"/> #27	Crear login de la aplicación	#25	1	Archive	15	-: Desarrollar una aplicación de usuario # Sprint 4 Desarrollo
<input type="checkbox"/> #20	Implementar gestión de base de datos	#30	3	Archive	15	-: Desarrollar una aplicación de usuario # Sprint 4 Desarrollo
<input type="checkbox"/> #23	Diseño procedimental	#28	2	Archive	15	-: Crear la memoria del proyecto # Sprint 4 Documentación
<input type="checkbox"/> #28	Crear pantalla de administración	#24	1	Archive	15	-: Desarrollar una aplicación de usuario # Sprint 4 Desarrollo
<input type="checkbox"/> #36	Crear proceso de importación de datos.	#22	2	Archive	15	-: Preparación de conjuntos de datos # Sprint 4 Desarrollo
<input type="checkbox"/> #32	Elegir hosting	#15	1	Archive	15	-: Desarrollar una aplicación de usuario # Sprint 4 Investigación
<input type="checkbox"/> #19	Implementar cliente API OpenAI.	#31	1	Archive	15	-: Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 4 Desarrollo
<input type="checkbox"/> #29	Crear pantalla de ejecución del proceso	#21	2	Archive	15	-: Preparación de conjuntos de datos # Sprint 4 Desarrollo
<input type="checkbox"/> #39	Configurar docker	#36	2	Archive	15	-: Desarrollar una aplicación de usuario # Sprint 4 Configuración

Figura A.9: Tareas sprint 4

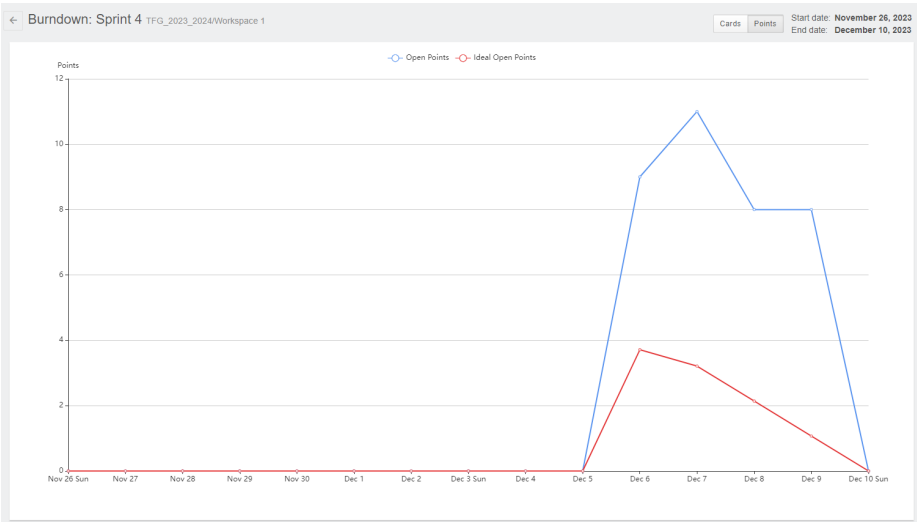


Figura A.10: Burndown chart sprint 4

Sprint 5 - del 11 de Diciembre de 2023 al 24 de Diciembre de 2023

En este sprint se empieza a probar la aplicación, se resuelven los errores encontrados y se empieza con la documentación técnica.

<input type="checkbox"/>	#42	Al editar un usuario nos devuelve un error de acceso.	▼ #39	0.5	Archive	❌	❌	Desarrollar una aplicación de usuario # Sprint 5 Error
<input type="checkbox"/>	#41	La pantalla de administración sigue haciendo consultas.	▼ #36	0.5	Archive	❌	❌	Desarrollar una aplicación de usuario # Sprint 5 Error
<input type="checkbox"/>	#40	Falta la funcionalidad de registrar nuevos usuarios	▼ #37	0.5	Archive	❌	❌	Desarrollar una aplicación de usuario # Sprint 5 Error
<input type="checkbox"/>	#26	Ejecutar proceso para obtener datos de prueba	▼ #26	3	Archive	❌	❌	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM # Sprint 5 Desarrollo
<input type="checkbox"/>	#34	Documentación técnica	▼ #17	3	Archive	❌	❌	Crear la memoria del proyecto # Sprint 5 Documentación
<input type="checkbox"/>	#30	Crear pantalla de estudio de datos	▼ #20	5	Archive	❌	❌	Desarrollar una aplicación de usuario # Sprint 5 Desarrollo

Figura A.11: Tareas sprint 5

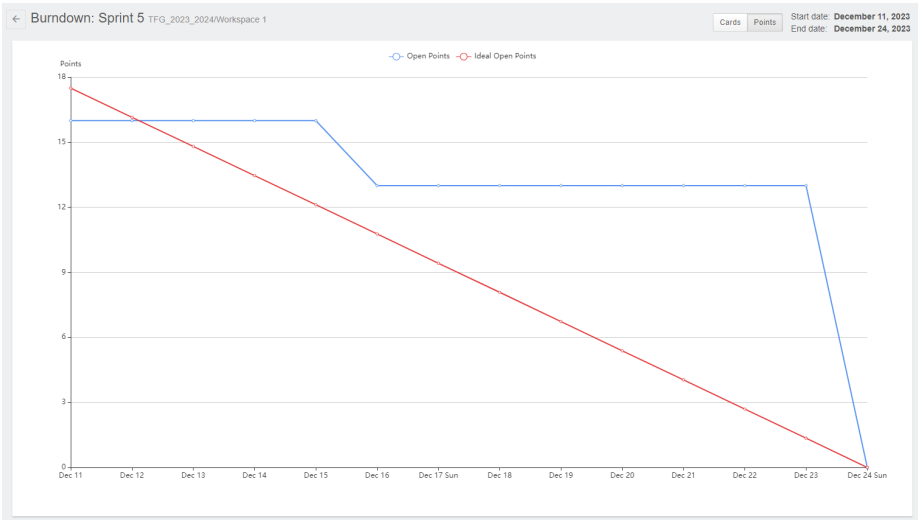


Figura A.12: Burndown chart sprint 5

Sprint 6 - del 25 de Diciembre de 2023 al 7 de Enero de 2024

A raíz de una reunión con el tutor se propone un cambio en el enfoque de la aplicación y se crea una nueva pantalla de análisis de reseñas. También se empieza con el estudio de viabilidad.

<input type="checkbox"/> #44	Actualizar documentación con la nueva pantalla de análisis de reseñas.	#41	5	Archive	1/1	- Desarrollar una aplicación de usuario # Sprint 6 Desarrollo
<input type="checkbox"/> #43	Nueva pantalla de análisis de reseñas.	#40	5	Archive	1/1	- Desarrollar una aplicación de usuario # Sprint 6 Desarrollo
<input type="checkbox"/> #45	Preparar ejemplos de uso	#42	3	Archive	1/1	- Preparación de conjuntos de datos # Sprint 6 Tarea
<input type="checkbox"/> #46	Pantalla de presentación de resultados.	#43	3	Archive	1/1	- Desarrollar una aplicación de usuario # Sprint 6 Desarrollo
<input type="checkbox"/> #33	Estudio de viabilidad	#16	3	Archive	1/1	- Crear la memoria del proyecto # Sprint 6 Documentación

Figura A.13: Tareas sprint 6

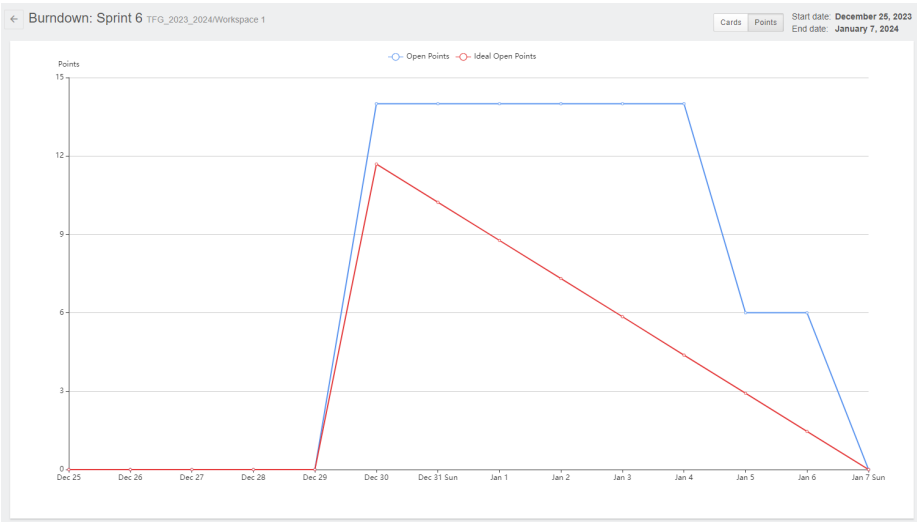


Figura A.14: Burndown chart sprint 6

Sprint 7 - del 7 de Enero de 2024 al 21 de Enero de 2024

Se continua con las pruebas, y se corrige la memoria con las sugerencias de los tutores. Se avanza con la memoria y se implementa la traducción de la aplicación. Se configura Codacy para las pruebas de calidad del código.

<input type="checkbox"/> #38 L10n	#35 3 Done 1	Desarrollar una aplicación de usuario Sprint 7 Desarrollo
<input type="checkbox"/> #31 Crear manual de usuario	#19 3 In Progress 1	Desarrollar una aplicación de usuario Sprint 7 Documentación
<input type="checkbox"/> #47 Crear presentación para el trabajo	#44 2 Ready 1	Crear la memoria del proyecto Sprint 7 Documentación
<input type="checkbox"/> #37 Calcular coste de analizar un texto en OpenAI	#34 2 Ready	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM Sprint 7 Desarrollo
<input type="checkbox"/> #50 Implementar cliente de llama_cpp	#47 1 Ready 1	Desarrollar un sistema de análisis de sentimiento mediante modelos LLM Sprint 7 Desarrollo
<input type="checkbox"/> #35 Sostenibilidad curricular	#18 3 Ready 1	Crear la memoria del proyecto Sprint 7 Documentación

Figura A.15: Tareas sprint 7



Figura A.16: Burndown chart sprint 7

A.3. Estudio de viabilidad

Viabilidad Técnica

Las tecnologías utilizadas en este trabajo se emplean ampliamente en muchos proyectos de software, y tienen un amplio soporte por parte de su comunidad de desarrollo. Son conocidas por su eficiencia y flexibilidad en el desarrollo web.

Basándonos en los requisitos del proyecto las tecnologías elegidas para el desarrollo del trabajo resultan adecuadas para su correcto funcionamiento. La mayor complejidad del trabajo se encuentra en el modelo LLM que se gestiona desde una instancia ajena al resto del proyecto de desarrollo lo que nos facilita su gestión.

Para el caso concreto de OpenAI resulta muy útil su uso pero hay que tener en cuenta los costes asociados así como sus políticas de uso.

Viabilidad económica

La viabilidad económica de un proyecto de software construido con las tecnologías mencionadas dependerá de varios factores, incluyendo el alcance y los objetivos del proyecto, el mercado al que se dirige y la estrategia de monetización.

Desarrollo

En este apartado vamos a calcular el coste de desarrollo del proyecto. Por un lado tenemos el coste del desarrollador.

Se han empleado alrededor de 4 meses para completar el trabajo, vamos a considerar un 50. Por lo tanto tenemos unas 20 horas semanales durante 16 semanas. El sueldo medio por hora de un desarrollador en España es de unos **15 euros**. Así el salario total ha sido: $20 \times 16 \times 15 = 4800$ euros. A esto hay que agregar el coste de la cotización a la seguridad social. En estos momentos la cotización por parte de la empresa se distribuye en ¹:

- Contingencias: 23,6
- Desempleo: 5,5
- FOGASA: 0,20
- Formación profesional (FP): 0,60
- Mecanismo de equidad intergeneracional (MEI): 0,5

Así el coste total será:

$$4800(\text{salario}) + 1132,8(\text{contingencias}) + 264(\text{desempleo}) + 9,6(\text{FOGASA}) + 28,8(\text{FP}) + 24(\text{MEI}) = 6259,2$$

De la misma forma se valora el coste de asesoría (tutores), contando 2 horas a la semana con un coste de 40 euros la hora. En este caso el salario total sería: $40 \times 2 \times 16 \times 2 = 2560$

Y el coste total:

$$2560(\text{salario}) + 604,16(\text{contingencias}) + 140(\text{desempleo}) + 5,12(\text{FOGASA}) + 15,36(\text{FP}) + 12,8(\text{MEI}) = 3337,44$$

Software

Todo el software utilizado en este proyecto es gratuito para aplicaciones no comerciales pero tienen un coste para aplicaciones comerciales excepto OpenAI que siempre tiene un coste por uso.

¹<https://www.seg-social.es/wps/portal/wss/internet/Trabajadores/CotizacionRecaudacionTrabajadores/3>

Software	Coste
Python	Gratuito
node.js	Gratuito
Docker	Gratuito
Vue	Gratuito
Postgres	Gratuito
OpenAI	Por uso

Para calcular el coste de OpenAI tenemos que tener en cuenta que éste se calcula según el número de **tokens** enviados/recibidos.²

	Model	Input	Output
	gpt-3.5-turbo-1106	\$0.0010 / 1K tokens	\$0.0020 / 1K tokens
	gpt-3.5-turbo-instruct	\$0.0015 / 1K tokens	\$0.0020 / 1K tokens

1000 tokens son alrededor de 750 palabras.

Hardware

Acerca del hardware utilizado en el proyecto, se ha utilizado un alojamiento llamado **Contabo**.

Este alojamiento tiene un coste de 4,5 euros al mes (+IVA = 5,45€). Ese precio incluye una máquina virtual linux (Ubuntu), con 8Gb de Memoria RAM y 200Gb de disco duro.

Esta configuración es suficiente para el proyecto realizado.

El resto de hardware utilizado se considera ya amortizado en proyectos pasados y por tanto no se va a contar para este.

Otros

Otros costes asociados a este desarrollo serían: Conexión a internet (40€ al mes) El alquiler del dominio, que ha costado 1€ al año en **arsys**. El certificado SSL se ha obtenido gratuitamente de **letsencrypt**.

Total

Concepto	Coste
Desarrollo	9596,54
Software	10€
Hardware	20€
Otros	161€
Total	9787,54

²fuelle: <https://openai.com/pricing>

Viabilidad legal

Para estudiar la viabilidad legal del proyecto vamos a analizar las licencias de las herramientas utilizadas.

Software	Licencia
Python	PSF (compatible con GPL)
Flask	BSD
SQLAlchemy	MIT
Github	GNU
node.js	MIT
Docker	Apache 2.0
Vue	MIT
Postgres	Licencia Postgres
OpenAI	Propietaria
Visual Studio Code	MIT

Como vemos todas las herramientas utilizadas durante el desarrollo son Open Source, excepto OpenAI.

Apéndice B

Especificación de Requisitos

B.1. Introducción

En este apartado se enumerarán y desarrollarán los objetivos y requisitos que debe tener la aplicación según han sido marcados al comienzo del proyecto. Además de estos, se han añadido algunos a lo largo del desarrollo.

B.2. Objetivos generales

Este trabajo tiene los siguientes objetivos:

- Crear una aplicación que permita realizar análisis de sentimiento a textos extraídos de reseñas de Amazon.
- Almacenar dicha información en una base de datos para que sea fácilmente accesible.
- Mostrar dicha información con gráficos y tablas para poder analizarlos.
- Crear un manual de usuario.

B.3. Catálogo de requisitos

Requisitos funcionales

- **RF1 - Registro de usuarios:** la aplicación debe permitir dar de alta usuarios y asignarles un rol.

- **RF2 - Login de usuarios:** la aplicación debe ser capaz de dadas las credenciales de un usuario permitirle acceder a las funcionalidades.
- **RF3 - Información de la aplicación:** el usuario debe poder obtener información sobre el funcionamiento de la aplicación.
- **RF4 - Analizar textos:** la aplicación debe ser capaz de analizar textos.
 - **RF4.1 - Cargar reseñas de un archivo:** la aplicación debe poder cargar las reseñas de un archivo de texto
 - **RF4.2 - Puntuar los textos:** la aplicación debe realizar el análisis del texto obtenido y devolver un valor entero entre 1 y 5 dependiendo del sentimiento del mismo, siendo 1 un sentimiento negativo y 5 un sentimiento positivo.
 - **RF4.3 - Almacenar resultados:** la aplicación debe almacenar las puntuaciones junto con el texto analizado, fecha en la que se escribió y un identificador.
- **RF5 - Datos estadísticos:** debe mostrar datos estadísticos sobre el análisis realizado.
- **RF6 - Configurar idioma:** el usuario debe poder cambiar el idioma de la aplicación.

Requisitos no funcionales

- **RNF1 - Usabilidad:** la aplicación debe ser amigable para que la experiencia del usuario sea lo más positiva posible. Además debe poder adaptarse a diferentes formatos de pantalla.
- **RNF2 - Eficiencia:** el almacenamiento de resultados en la base de datos y la respuesta de la aplicación al navegar entre ventanas, sobre todo, al mostrar los gráficos debe ser lo más rápido posible.
- **RNF3 - Disponibilidad:** la aplicación debe estar disponible durante el mayor tiempo posible y en la mayoría de lugares.
- **RNF4 - Escalabilidad:** la aplicación debe estar preparada para soportar nuevos desarrollos que generen mayor cantidad de trabajo.
- **RNF5 - Confiabilidad:** la aplicación cumplirá la función para la que se ha creado.
- **RNF6 - Mantenibilidad:** la aplicación debe permitir cambios y correcciones de forma eficaz.

- **RNF7 - Internacionalización:** la aplicación debe soportar varios idiomas.
- **RNF8 - Seguridad:** la aplicación debe soportar usuarios con diferentes roles que tendrán acceso a diferentes funcionalidades.

B.4. Casos de uso

Diagrama de casos de uso

Como resumen de los casos de uso anteriores obtenemos el siguiente diagrama de casos de uso:

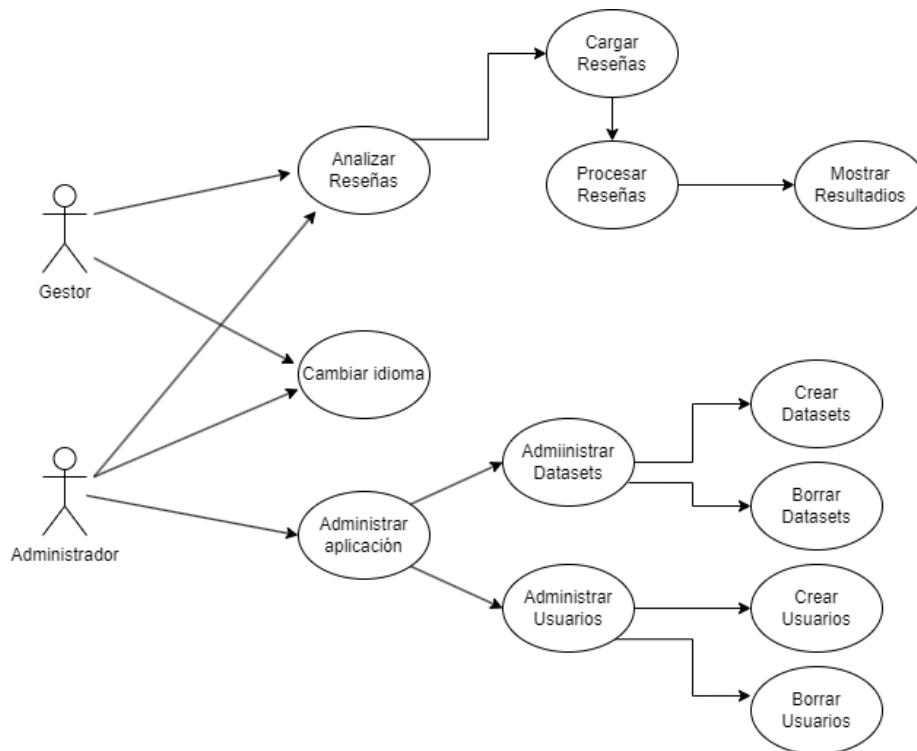


Figura B.1: Diagrama de casos de uso

CU-1	Registro de usuarios
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador podrá crear cuentas de usuario y asignarles un rol.
Precondición	El usuario tiene que tener el rol de administrador
Acciones	<ol style="list-style-type: none"> 1. El usuario administrador accede a la interfaz de la aplicación. 2. Accede a la opción Crear usuarios 3. Introduce los datos del usuarios 4. Confirma los cambios.
Postcondición	La cuenta de usuario no debe existir con antelación
Excepciones	<ol style="list-style-type: none"> 1. El usuario ya existe 2. Complejidad de la contraseña no válida 3. Datos incorrectos
Importancia	Alta

Tabla B.1: CU-1 Registro de usuarios.

CU-2	Login de usuarios
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-2
Descripción	El usuario accede a la aplicación tendiendo acceso a la parte que tenga acceso según su rol.
Precondición	Usuario registrado
Acciones	<ol style="list-style-type: none">1. Accede a la pantalla de login2. Introduce nombre y contraseña3. Pulsa en Aceptar
Postcondición	La contraseña debe coincidir
Excepciones	<ol style="list-style-type: none">1. El usuario no existe2. La contraseña no es válida
Importancia	Alta

Tabla B.2: CU-2 Login de usuarios.

CU-3	Información de la aplicación.
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-3
Descripción	Permite al usuario obtener información sobre la configuración de la aplicación.
Precondición	Usuario logado.
Acciones	<ol style="list-style-type: none">1. El usuario accede a la opción información.
Postcondición	
Excepciones	
Importancia	Baja

Tabla B.3: CU-3 Información de la aplicación.

CU-4	Cargar reseñas de un archivo
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF4.1
Descripción	Un usuario administrador puede cargar reseñas desde un archivo
Precondición	Usuario logado
Acciones	<ol style="list-style-type: none"> 1. Acceder a la aplicación. 2. Seleccionar archivo. 3. Seleccionar opciones. 4. Pulsar en cargar.
Postcondición	El fichero tiene el formato correcto.
Excepciones	<p>El fichero no tiene el formato correcto.</p> <p>El fichero no se puede abrir.</p> <p>El fichero no se ha podido cargar.</p>
Importancia	Alta

Tabla B.4: CU-04 Cargar reseñas de un archivo.

CU-5	Puntuar los textos
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF4.2
Descripción	La aplicación debe realizar el análisis del texto obtenido y devolver un valor entero entre 1 y 5 dependiendo del sentimiento del mismo, siendo 1 un sentimiento negativo y 5 un sentimiento positivo.
Precondición	Los textos están cargados
Acciones	<ol style="list-style-type: none"> 1. Se accede a la aplicación. 2. Fichero de reseñas se ha cargado. 3. Se selecciona la opción procesar.
Postcondición	Se recibe una respuesta del servicio de análisis.
Excepciones	No hay respuesta
Importancia	Alta

Tabla B.5: CU-5 Puntuar los textos.

CU-6	Almacenar resultados
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF4.3
Descripción	La aplicación debe almacenar las puntuaciones junto con el texto analizado, fecha en la que se escribió y un identificador.
Precondición	Se han procesado las reseñas
Acciones	<ol style="list-style-type: none"> 1. Se procesan los elementos 2. El resultado se guarda
Postcondición	La base de datos tiene que estar disponible.
Excepciones	La base de datos no está disponible
Importancia	Alta

Tabla B.6: CU-6 Almacenar resultados.

CU-07	Información estadística.
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF5
Descripción	Debe mostrar la información estadística para los datos analizados.
Precondición	Análisis de sentimiento ejecutado.
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción de visualizar resultados.
Postcondición	
Excepciones	Datos no generados
Importancia	Media

Tabla B.7: CU-07 Información estadística.

CU-08	Configurar idioma de la aplicación.
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF6
Descripción	El usuario debe poder cambiar el idioma de la aplicación.
Precondición	El usuario debe de estar logado
Acciones	<ol style="list-style-type: none"> 1. Seleccionar la opción configuración 2. Elegir el idioma deseado, según los idiomas disponibles
Postcondición	
Excepciones	
Importancia	Media

Tabla B.8: CU-08 Configurar el idioma de la aplicación.

Apéndice C

Especificación de diseño

C.1. Introducción

C.2. Diseño de datos

Para almacenar la información necesaria para la aplicación he diseñado una base de datos que sirva para facilitar el procesado de las reseñas y su posterior análisis por parte de los usuarios de la aplicación:

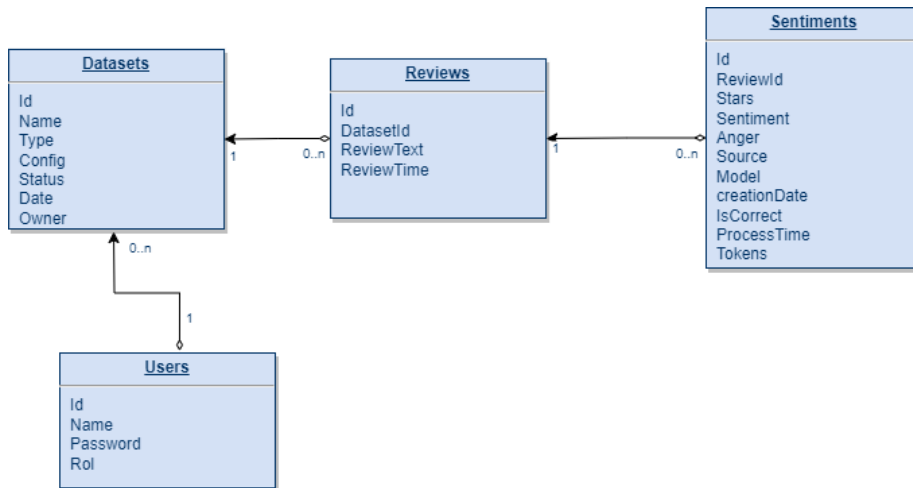


Figura C.1: Arquitectura de la base de datos

Como vemos almacenamos la información relativa a los conjuntos de datos en la tabla "datasets". En esta tabla almacenamos la información

necesaria para cargarlo como es el tipo (type) y el payload. Según el tipo de conjunto de datos el método para cargarlo será diferente. Una vez iniciado el proceso de carga, la información relativa a las reseñas se almacenan en la tabla Review", "Products", "ReviewUsers".

Por último, se almacena en la tabla "sentiments" la información relativa al análisis de sentimiento. Se incluye la información obtenida del modelo así como el tiempo que ha tardado y el número de tokens procesados. Este último campo (tokens) se puede utilizar para calcular el coste en el caso de OpenAI. También se indica si la respuesta del modelo se considera errónea por tener un formato incorrecto. En este caso la respuesta del modelo se almacena en el campo "source".

C.3. Diseño procedimental

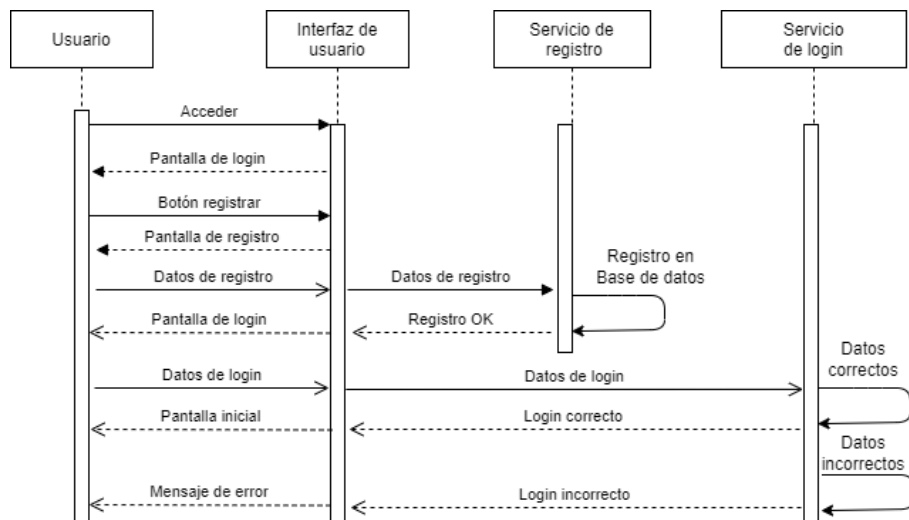


Figura C.2: Login

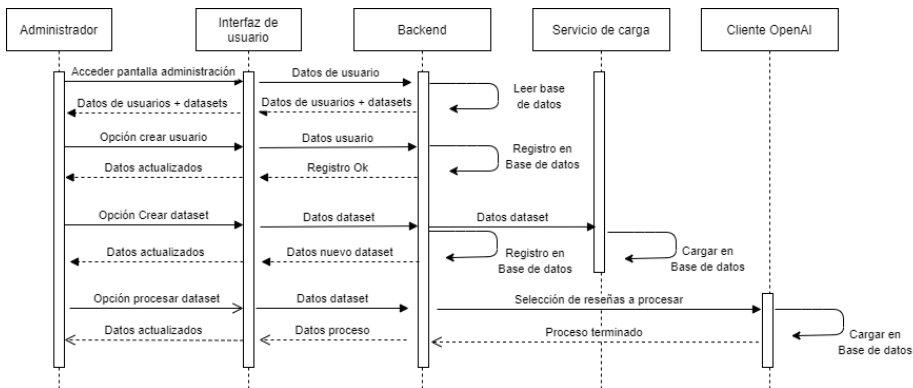


Figura C.3: Administración

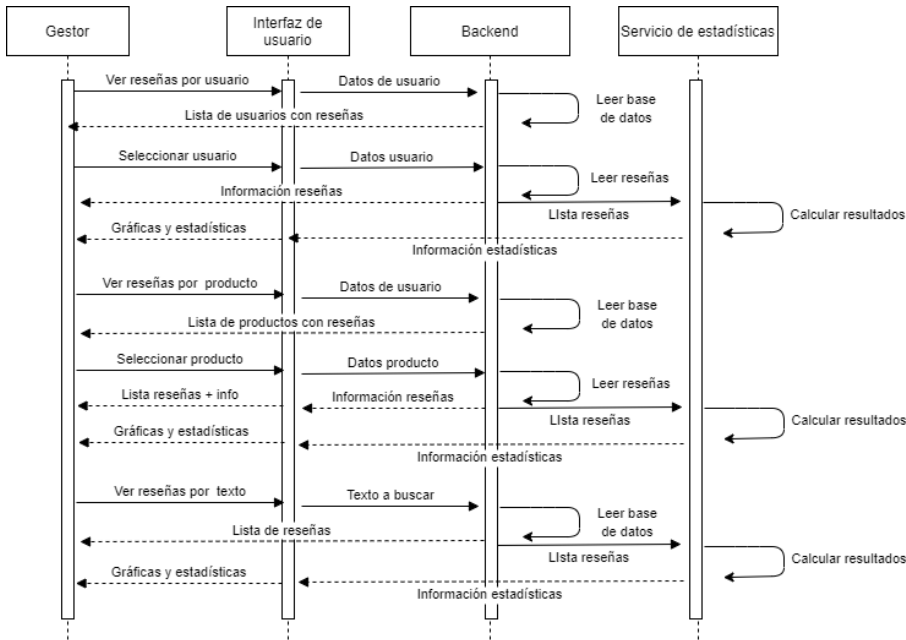


Figura C.4: Gestor

C.4. Diseño arquitectónico

Arquitectura de la aplicación

En el caso de la arquitectura de la aplicación, vemos que está centrada en la base de datos. Desde el módulo de administración se pueden gestionar los usuarios y las reseñas. Desde el módulo de servicios se cargan las reseñas

y se gestionan los diferentes procesos de análisis. A través de un servicio web se expone al usuario los diferentes informes de resultados.

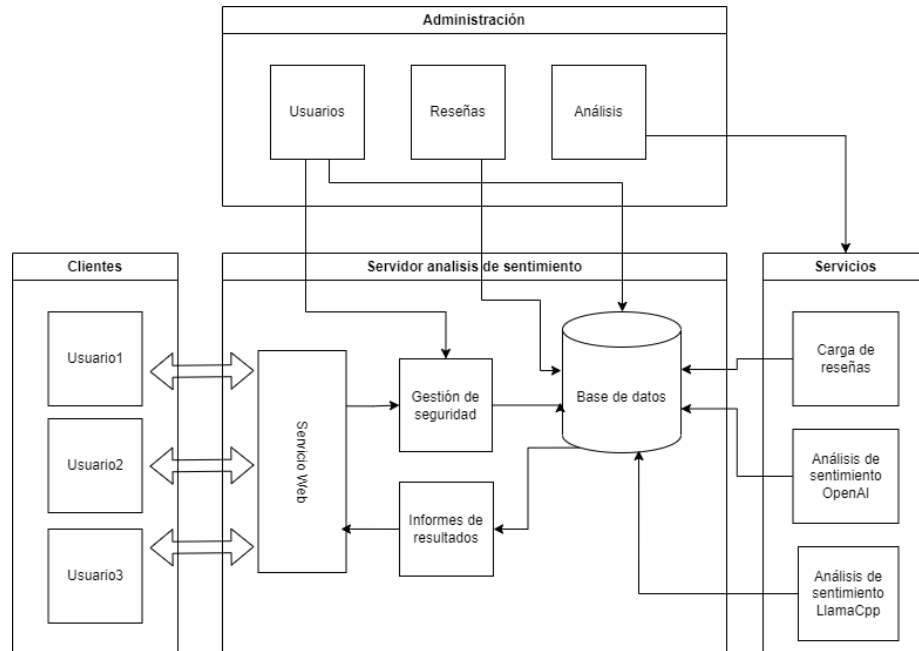
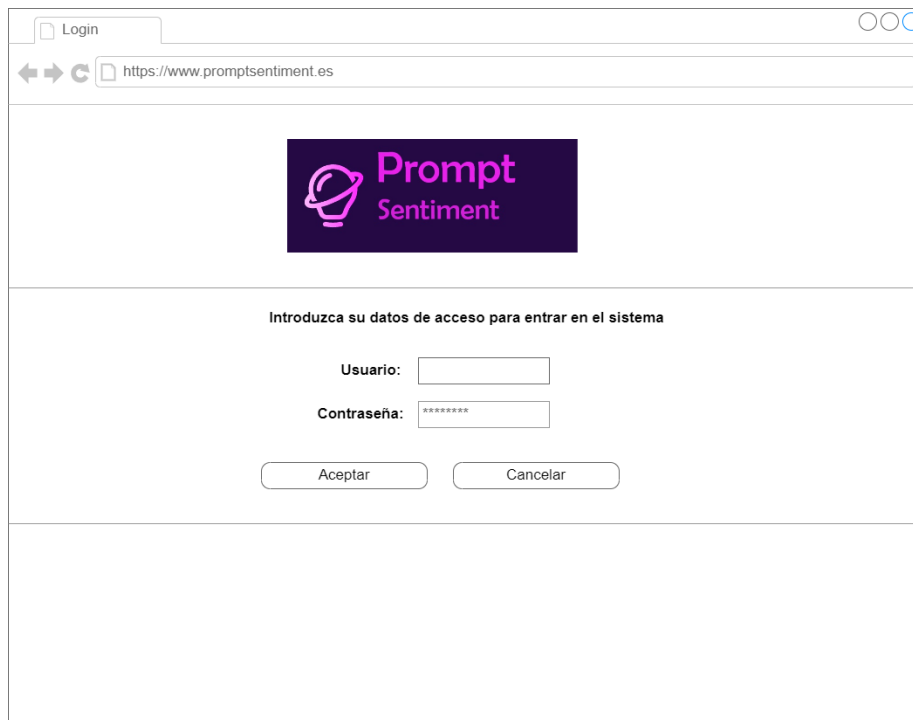


Figura C.5: Arquitectura de la aplicación

Interfaz de usuario

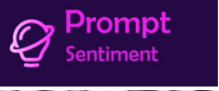
En la pantalla de login debe aparecer el logo de la aplicación junto con la información necesaria para poder acceder a la aplicación. Se puede mostrar también una pequeña introducción sobre el proposito de la misma y un pequeño vínculo para acceder al registro como usuario.



The image shows a web browser window with a single tab titled "Login". The address bar displays "https://www.promptsentiment.es". The main content area features the "Prompt Sentiment" logo, which consists of a purple square containing a white icon of a speech bubble with a heart inside, followed by the text "Prompt Sentiment" in white. Below the logo, the text "Introduzca su datos de acceso para entrar en el sistema" is centered. Underneath this text are two input fields: "Usuario:" followed by a text box, and "Contraseña:" followed by a password box with masked characters (*****). At the bottom of the form are two buttons: "Aceptar" and "Cancelar".

Figura C.6: Pantalla de login.

En esta pantalla se mostrará la información relativa a las reseñas y al análisis de sentimiento realizado hasta el momento.



Usuario1

Search

Por usuario

Por producto

Por texto

Estadísticas

Buscar usuario

Usuario 1

Usuario 2

Usuario 3

Usuario 4

Usuario 5

Usuario 6

Usuario 7

Usuario 8

Usuario 9

Usuario 10

Usuario 11

Usuario 12

Usuario 13

Reseña	Producto	Estrellas	Estimación	Diferencia
Reseña1	Producto1	1	2	
Reseña2	Producto2	2	2	
Reseña3	Producto3	2	3	

API: OpenAI

Modelo: ChatGpt 3.5 Turbo

Fecha: 1 nov. 2023

Tiempo (seg.): 4

Coste: 0,01 euros

Conjunto de datos: Amazon Reviews

Origen: Hugging face

Url Modelo

Url conjunto de datos

Texto de la reseña

Figura C.7: Pantalla usuario gestor.

En la pantalla de administración se podrá gestionar los usuarios y las reseñas.

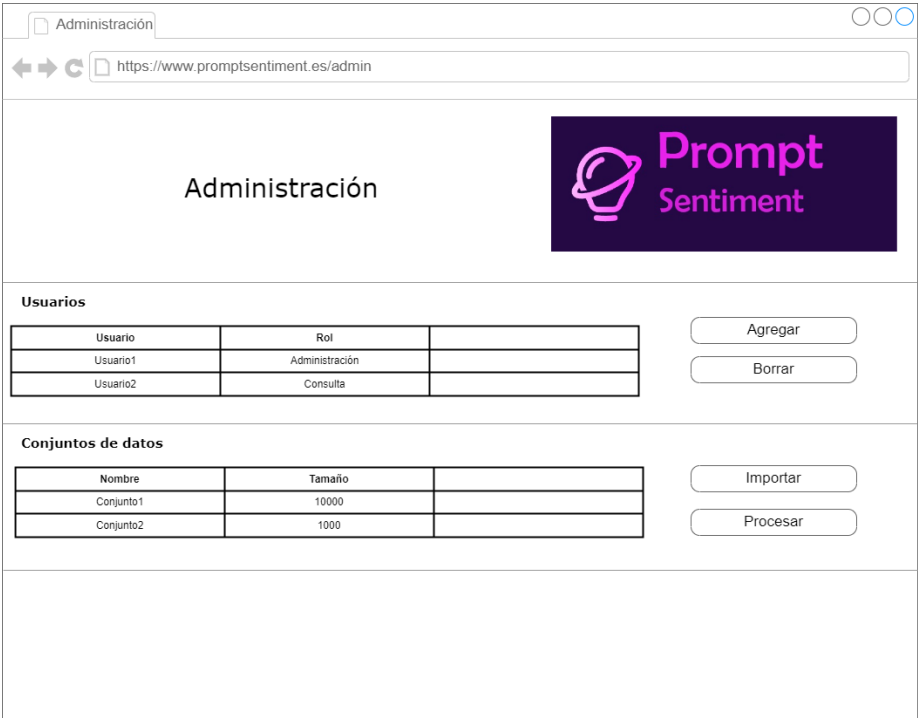


Figura C.8: Pantalla de administración.

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este apéndice se va a explicar cómo está organizado el proyecto, el manual del programador y los requisitos necesarios para poder ejecutar el proyecto.

D.2. Estructura de directorios

En el proyecto hay una estructura de directorios que intenta organizar el trabajo de manera lógica: Por un lado tenemos el código y por otra la documentación. Dentro del código podemos apreciar una carpeta llamada colab con un cuaderno de Jupyter para poder ejecutarlo en Colab. Hay otra carpeta con el código del proyecto es sí. La estructura de directorios quedaría así:

- Código
 - colab: Contiene un cuaderno de Jupyter preparado para ejecutar un ejemplo en Colab.
 - prompt_sentiment: Contiene el código del trabajo.
 - frontend: Contiene el código de la interfaz de usuario. Está hecha con Vue, usando el plugin Vuetify.

- backend: Contiene el código del servicio web de la aplicación. Está hecha en Flask (python) usando Flask-restx y SQLAlchemy para el acceso a las base de datos.
- nginx: En este directorio está la configuración del servidor nginx que se usa en la aplicación como proxy inverso y para gestionar la configuración https.

- Documentación

- Manual de usuario: El manual de uso de la aplicación
- Memoria: Es la memoria del proyecto

D.3. Manual del programador

Instalación local

Para la instalación local del proyecto tenemos realizar las siguientes instalaciones:

Instalación de Python 3 [1]

Para ejecutar nuestro proyecto es necesario instalar Python, para ello podemos descargar la versión aquí: <https://www.python.org/downloads/> Cuando la descarga haya finalizado ejecutamos e instalamos. También se recomienda usar conda para crear un entorno virtual para python. <https://docs.conda.io/projects/conda/en/latest/index.html>

Instalación de node.js

Para la parte de angular es necesario instalar Nodejs, por tanto accedemos a la página oficial, <https://nodejs.org/es/download/>, y descargaremos la versión que necesitemos y llevaremos a cabo su instalación

Instalación de PostgreSQL

Para la instalación de PostgreSQL tenemos que bajarnos la versión adecuada a nuestro sistema operativo e instalarla: <https://www.postgresql.org/download/>

Docker

Para la instalación de docker sólo hay que instalar Docker-Desktop.

Instalación de Docker Desktop

<https://docs.docker.com/engine/install/>

D.4. Compilación, instalación y ejecución del proyecto

Ejecución local

El proyecto tiene 2 partes claramente diferenciadas: *frontend* y *backend*. Para la ejecución del backend tenemos que abrir una consola de comandos, movernos al directorio backend y ejecutar:

```
pip install -r requirements.txt
```

Creamos la base de datos (sólo en el caso de que no exista)

```
python manage.py crear  
python manage.py rellenar (en caso de que esté vacía)  
python manage.py run
```

Para el frontend, nos cambiamos la carpeta *frontend* y ejecutamos:

```
npm install  
npm run serve
```

Ejecución en docker

Para la ejecución en docker sólo hay que ejecutar en el directorio raíz del código:

```
docker-compose up
```

D.5. Pruebas del sistema

A partir de los casos de uso del anexo b, diseñamos unos casos de prueba para validar la aplicación desde el punto de vista de un usuario.

Pruebas unitarias

Para las pruebas unitarias del backend se ha utilizado la librería `pytest`. Se han realizado tanto pruebas de caja blanca como pruebas de caja negra. Las pruebas unitarias se encuentran en el directorio `backend/app/test` del repositorio. Para la ejecución de las pruebas unitarias del *backend* hay que ejecutar:

`pytest`

El resultado de las pruebas unitarias es:

```
===== test session starts =====
platform linux -- Python 3.8.10, pytest-7.4.3, pluggy-1.3.0
rootdir: /home/tgs1003/Documents/GitHub/TFG_2023_2024/Código/prompt_sentiment/backend
plugins: anyio-3.7.1
collected 97 items

app/tests/test_auth.py ..... [ 12%]
app/tests/test_config.py ... [ 15%]
app/tests/test_datasets.py ..... [ 25%]
app/tests/test_datasets_model.py . [ 26%]
app/tests/test_datasets_unit.py ..... [ 32%]
app/tests/test_ping.py . [ 34%]
app/tests/test_reviews.py ..... [ 44%]
app/tests/test_reviews_model.py . [ 45%]
app/tests/test_reviews_unit.py ..... [ 51%]
app/tests/test_sentiments.py ..... [ 61%]
app/tests/test_sentiments_model.py . [ 62%]
app/tests/test_sentiments_unit.py ..... [ 69%]
app/tests/test_users.py ..... [ 83%]
app/tests/test_users_model.py ... [ 86%]
app/tests/test_users_unit.py ..... [100%]
```

Figura D.1: Resultado `pytest`

Plan de pruebas

CP-1	Registro de usuarios
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario podrá crearse una cuenta en la aplicación.
Precondiciones	<ol style="list-style-type: none">1. No existe un usuario con la misma cuenta de correo.2. El formato de correo es correcto.3. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Accede a la opción Registrar3. Introduce los datos del usuario4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario creado correctamente.

Tabla D.1: CP-1 Registro de usuarios.

CP-2	Registro de usuarios incorrecto: el usuario ya existe
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario no puede crearse una cuenta con email existente.
Precondiciones	<ol style="list-style-type: none">1. Existe un usuario con la misma cuenta de correo.2. El formato de correo es correcto.3. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Accede a la opción Registrar3. Introduce los datos del usuario4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario no creado.2. La aplicación devuelve un error: el usuario ya existe.

Tabla D.2: CP-2 Registro de usuarios incorrecto.

CP-3	Registro de usuarios: email incorrecto
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	El email tiene que tener un formato correcto
Precondiciones	<ol style="list-style-type: none">1. No existe un usuario con la misma cuenta de correo.2. El formato de correo es incorrecto.3. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Accede a la opción Registrar.3. Introduce los datos del usuario.4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario no creado.2. La aplicación devuelve un error: el formato de correo no es correcto.

Tabla D.3: CP-3 Registro de usuarios: email incorrecto.

CP-4	Registro de usuarios: contraseña no válida
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	La contraseña tiene que ser válida.
Precondiciones	<ol style="list-style-type: none">1. No existe un usuario con la misma cuenta de correo.2. El formato de correo es correcto.3. La contraseña no es válida.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Accede a la opción Registrar.3. Introduce los datos del usuario.4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario no creado.2. La aplicación devuelve un error: la contraseña no es válida.

Tabla D.4: CP-4 Registro de usuarios:contraseña no es válida.

CP-5	Registro de usuarios (administrador).
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador podrá crear cuentas de usuario y asignarles un rol.
Precondiciones	<ol style="list-style-type: none">1. El usuario creador tiene que tener el rol de administrador.2. No existe un usuario con la misma cuenta de correo.3. El formato de correo es correcto.4. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario administrador accede a la interfaz de la aplicación.2. Accede a la opción Crear usuarios.3. Introduce los datos del usuarios.4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario creado correctamente.

Tabla D.5: CP-5 Registro de usuarios(administrador).

CP-6	Registro de usuarios (administrador): usuario ya existe
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador no puede crear un usuario duplicado.
Precondiciones	<ol style="list-style-type: none">1. El usuario creador tiene que tener el rol de administrador.2. Existe un usuario con la misma cuenta de correo.3. El formato de correo es correcto.4. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario administrador accede a la interfaz de la aplicación.2. Accede a la opción Crear usuarios.3. Introduce los datos del usuarios.4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. Usuario no creado.2. La aplicación muestra el mensaje: el usuario ya existe.

Tabla D.6: CP-6 Registro de usuarios (administrador): usuario ya existe.

CP-7	Registro de usuarios (administrador):correo incorrecto
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador no puede crear un usuario con un correo con formato incorrecto.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario creador tiene que tener el rol de administrador. 2. No existe un usuario con la misma cuenta de correo. 3. El formato de correo es incorrecto. 4. La contraseña es válida.
Pasos	<ol style="list-style-type: none"> 1. El usuario administrador accede a la interfaz de la aplicación. 2. Accede a la opción Crear usuarios. 3. Introduce los datos del usuarios. 4. Confirma los cambios.
Resultado	<ol style="list-style-type: none"> 1. Usuario no creado 2. La aplicación muestra el mensaje: el correo electrónico no tiene el formato correcto.

Tabla D.7: CP-7 Registro de usuarios (administrador): correo incorrecto.

CP-8	Registro de usuarios (administrador): contraseña no válida
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador no podrá crear cuentas de usuario con una contraseña no válida.
Precondiciones	<ol style="list-style-type: none"> 1. El usuario creador tiene que tener el rol de administrador. 2. No existe un usuario con la misma cuenta de correo. 3. El formato de correo es correcto. 4. La contraseña no es válida.
Pasos	<ol style="list-style-type: none"> 1. El usuario administrador accede a la interfaz de la aplicación. 2. Accede a la opción Crear usuarios. 3. Introduce los datos del usuarios. 4. Confirma los cambios.
Resultado	<ol style="list-style-type: none"> 1. Usuario no creado. 2. La aplicación muestra el mensaje: contraseña no válida.

 Tabla D.8: CP-8 Registro de usuarios.

CP-9	Login de usuarios
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario podrá logarse en la aplicación.
Precondición	<ol style="list-style-type: none">1. Usuario no logado.2. El usuario existe.3. La contraseña es válida.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Introduce los datos de usuario.3. Confirma los datos.
Resultado	<ol style="list-style-type: none">1. Usuario logado correctamente.

Tabla D.9: CP-9 Login de usuarios.

CP-10	Login de usuario: usuario no existe
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario no podrá logarse en la aplicación.
Precondición	<ol style="list-style-type: none">1. Usuario no logado.2. El usuario no existe.3. La contraseña es indiferente.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Introduce los datos de usuario.3. Confirma los datos.
Resultado	<ol style="list-style-type: none">1. Usuario no logado.2. La aplicación muestra el error: el usuario no existe.

Tabla D.10: CP-10 Login de usuario: usuario no existe.

CP-11	Login de usuario: contraseña incorrecta.
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario no podrá logarse en la aplicación.
Precondición	<ol style="list-style-type: none">1. Usuario no logado.2. El usuario existe.3. La contraseña es incorrecta.
Pasos	<ol style="list-style-type: none">1. El usuario accede a la interfaz de la aplicación.2. Introduce los datos de usuario.3. Confirma los datos.
Resultado	<ol style="list-style-type: none">1. Usuario no logado.2. La aplicación muestra el error: contraseña incorrecta.

Tabla D.11: CP-11 Login de usuario: contraseña incorrecta.

CP-3	Registro de usuarios
Versión	1.0
Autor	Teodoro Ricardo García Sánchez
Requisitos asociados	RF-1
Descripción	Un usuario con el rol de administrador podrá crear cuentas de usuario y asignarles un rol.
Precondición	El usuario tiene que tener el rol de administrador
Pasos	<ol style="list-style-type: none">1. El usuario administrador accede a la interfaz de la aplicación.2. Accede a la opción Crear usuarios3. Introduce los datos del usuarios4. Confirma los cambios.
Resultado	<ol style="list-style-type: none">1. El usuario ya existe2. Complejidad de la contraseña no válida3. Datos incorrectos

Tabla D.12: CP-1 Registro de usuarios.

Apéndice E

Documentación de usuario

E.1. Introducción

En este apéndice vamos a explicar cómo funciona la aplicación desde el punto de vista de un usuario.

E.2. Requisitos de usuario

La aplicación es accesible desde la url <https://promptsentiment.es>. Para acceder a la aplicación sólo hace falta un navegador moderno (Chrome, Opera, Safari, etc) Con respecto a los ficheros con las reseñas de usuario, hace falta que tengan un formato de texto delimitado (coma o tabulador) con cabecera o formato Json. También hay una limitación en cuanto al tamaño del archivo (1 MB).

E.3. Instalación

No es necesaria ninguna instalación para usar la aplicación.

E.4. Manual del usuario



Manual de usuario

Version 1.0

Teodoro Ricardo García Sánchez
14 de febrero de 2024

Índice

1. Introducción	2
2. Registro	3
3. Login	4
4. Inicio	5
5. Análisis de reseñas	6
6. Histórico de reseñas	8
7. Administración	9
7.1. Usuarios	9
7.2. Conjuntos de datos	10
8. Información	12

1. Introducción

“Prompt sentiment” es una aplicación que realiza un análisis de sentimientos sobre un fichero que contiene una serie de reseñas de usuario relacionadas con un producto o servicio.

Esta aplicación utiliza OpenAI para analizar las reseñas mediante técnicas de “prompt engineering”. En un futuro próximo se ampliará el soporte con otros modelos LLM como Llama o Mistral.

2. Registro

Para empezar a utilizar la aplicación lo primero que nos hace falta es crear un usuario. Empiece accediendo a la página de registro: <https://promptsentiment.es/registro>.



Registro

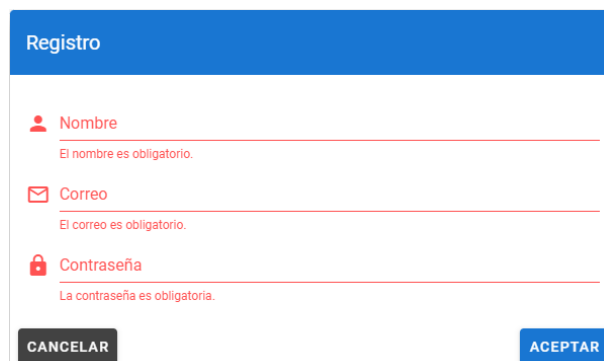
Nombre

Correo

Contraseña

CANCELAR ACEPTAR

Si falta algún dato o es incorrecto (por ejemplo, el formato de correo no es correcto), se mostrará un mensaje de error.



Registro

Nombre

El nombre es obligatorio.

Correo

El correo es obligatorio.

Contraseña

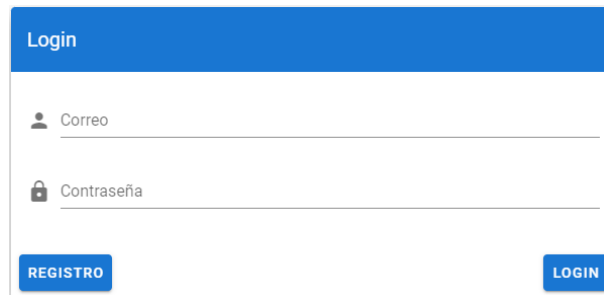
La contraseña es obligatoria.

CANCELAR ACEPTAR

NOTA: El perfil asignado al nuevo usuario creado es el de Gestor. Éste es un perfil limitado. Contacte con el administrador de la aplicación para actualizar su perfil a otro más avanzado.

3. Login

Un vez se ha registrado correctamente en la aplicación aparecerá la pantalla de login.



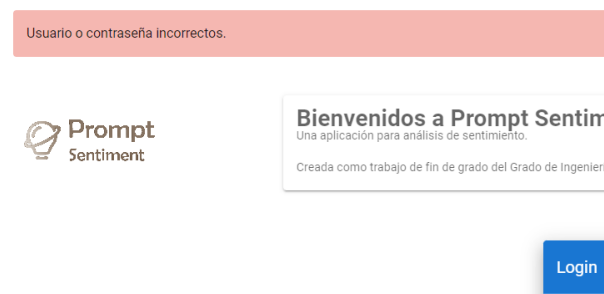
A screenshot of the login form. It has a blue header with the word "Login". Below the header, there are two input fields: "Correo" (Email) and "Contraseña" (Password). At the bottom, there are two buttons: "REGISTRO" (Registration) and "LOGIN".

Si falta algún dato o es incorrecto (por ejemplo, el formato de correo no es correcto), se mostrará un mensaje de error.



A screenshot of the login form showing validation errors. The "Correo" field has a red error message "El correo es obligatorio." (Email is required). The "Contraseña" field has a red error message "Contraseña obligatoria." (Password is required). The "REGISTRO" and "LOGIN" buttons are still visible at the bottom.

Si el usuario no existe o la contraseña es incorrecta se mostrará un mensaje de error.



A screenshot of the login page. At the top, there is a red error message: "Usuario o contraseña incorrectos." (User or password incorrect). Below the error message, there is a logo for "Prompt Sentiment" and a welcome message: "Bienvenidos a Prompt Sentiment. Una aplicación para análisis de sentimiento. Creada como trabajo de fin de grado del Grado de Ingeniería." (Welcome to Prompt Sentiment. An application for sentiment analysis. Created as a final degree project of the Degree in Engineering). At the bottom right, there is a blue "Login" button.

4. Inicio

Una vez hecho login correctamente en la aplicación aparecerá la pantalla de inicio.

En la pantalla de inicio hay 2 opciones que se pueden seleccionar:

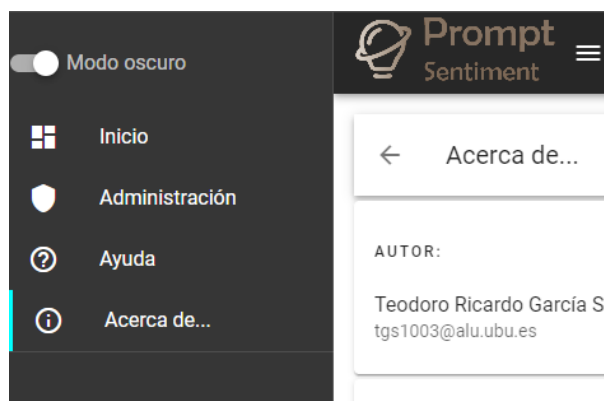
- Análisis de reseñas
- Histórico de reseñas

Análisis de reseñas.	Seleccione esta opción para analizar un fichero de reseñas.
ANALIZAR RESEÑAS	
Histórico de análisis.	Seleccione esta opción para ver los análisis anteriores.
VER ANÁLISIS	

También se puede cambiar el idioma del interfaz.



Y acceder al menú de la aplicación y activar el modo oscuro.



5. Análisis de reseñas

El análisis de reseñas se realiza en 3 pasos:

- Paso 1: selección del fichero de reseñas

The screenshot shows the 'Análisis de sentimiento' (Sentiment Analysis) interface. At the top, there's a back arrow and the title 'Análisis de sentimiento'. Below it, a progress bar indicates three steps: 1. 'Seleccione el fichero con las reseñas' (highlighted with a blue circle), 2. 'Configuración del fichero', and 3. 'Resultados'. Step 1 includes a sub-header 'Seleccione el fichero con las reseñas' and a description 'Seleccione el fichero con las reseñas que quiere analizar'. There are two input fields: 'Nombre del producto' and 'Fichero de reseñas'. Below these fields are two buttons: 'SIGUIENTE' (highlighted in blue) and 'REINICIAR'.

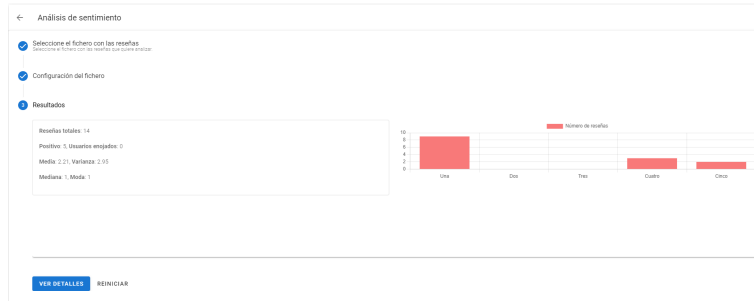
- Paso 2: configuración del formato del fichero

The screenshot shows the 'Análisis de sentimiento' interface at Step 2: 'Configuración del fichero'. The progress bar now shows Step 1 as completed (with a blue checkmark) and Step 2 as the current step (highlighted with a blue circle). Step 2 includes a sub-header 'Configuración del fichero' and a description 'Configure el fichero con las reseñas que quiere analizar'. It displays 'Nombre del fichero: 80032F0SF2.json' and 'Número de reseñas: 17'. Below this, 'Formato: JSON' is shown. There's a label 'Campo que contiene las reseñas:' followed by a dropdown menu currently set to 'text'. At the bottom, there are 'PROCESAR' (highlighted in blue) and 'REINICIAR' buttons. Step 3 'Resultados' is shown as a future step in the progress bar.

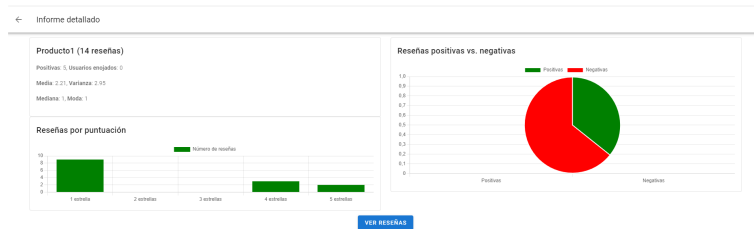
- Paso 3: importar y procesar las reseñas

The screenshot shows the 'Análisis de sentimiento' interface at Step 3: 'Resultados'. The progress bar shows Steps 1 and 2 as completed (with blue checkmarks) and Step 3 as the current step (highlighted with a blue circle). Step 3 includes a sub-header 'Resultados' and a description 'Visualice los resultados del análisis de sentimiento'. It displays the same file information as Step 2: 'Nombre del fichero: 80032F0SF2.json' and 'Número de reseñas: 17'. Below this, there's a 'Formato: JSON' label and a 'Campo que contiene las reseñas:' dropdown set to 'text'. At the bottom, there are 'PROCESAR' and 'REINICIAR' buttons. A large blue button labeled 'DESCARGAR' is prominently displayed in the center of the interface.

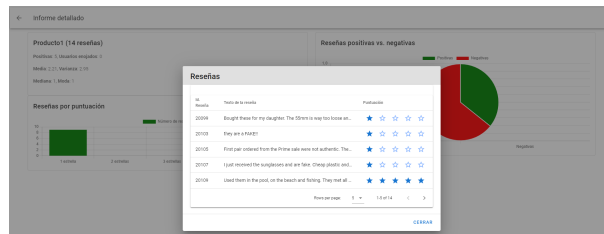
Una vez terminado el proceso obtenemos los resultados:



También podemos ver los resultados detallados:



Y ver las reseñas desde esa pantalla:

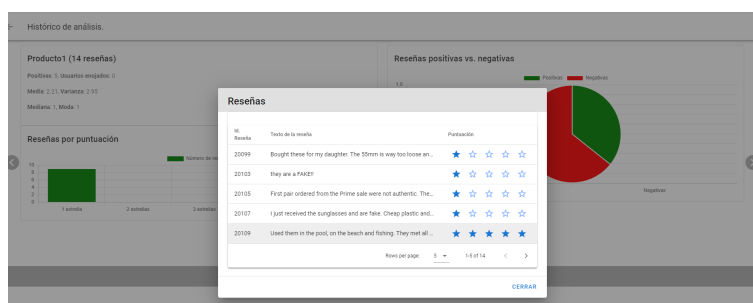


6. Histórico de reseñas

Desde la página de inicio también podemos acceder al histórico de reseñas. En el histórico podemos ver los ficheros de reseñas analizados en la aplicación.



También podemos ver las reseñas desde esta pantalla:



7. Administración

La pantalla de administración es sólo accesible para usuarios con perfil de administración (Admin).

Desde esta pantalla podemos administrar usuarios y conjuntos de datos:

← Administración

Lista de usuarios

CREAR USUARIO

admin.users.name	Correo ↕	Rol	Acciones
Admin	admin@promptsentiment.es	Admin	✎ 🗑
Ricardo García	rg1003@alu.ubu.es	Gestor	✎ 🗑

Rows per page: 5 1-2 of 2 < >

Lista de conjunto de datos

CREAR DATASET

Nombre	Tipo	Reviews totales	Reviews procesadas	Acciones
3 campos	json	10	10	✎ 🗑
Amazon Shoe Reviews (test)	Hugging face	10000	10000	✎ 🗑
Burton RFID Wizard Wallet	json	21	21	✎ 🗑
E-Dream Women Handbag	json	28	28	✎ 🗑
Product1	JSON	11	11	✎ 🗑

Rows per page: 5 1-5 of 9 < >

7.1. Usuarios

Para crear un usuario pulsamos en el botón “Nuevo usuario”

Nuevo usuario

Nombre de usuario

Correo

Contraseña

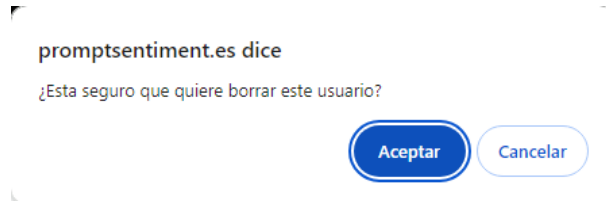
Rol ▼

CANCELAR

GUARDAR

Hay que rellenar los datos y pulsar en Aceptar.

Para borrar un usuario hay que pulsar en el icono de la papelera correspondiente al usuario que queremos borrar. La aplicación pedirá confirmación:




Para modificar los datos de un usuario hay que pulsar en el icono del lápiz.

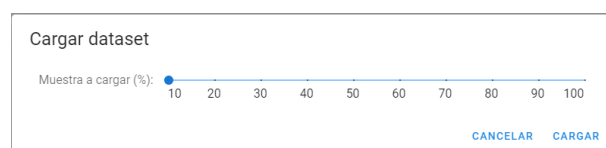
7.2. Conjuntos de datos



Al igual que con los usuarios, desde la pantalla de administración se pueden administrar los conjuntos de datos. En la versión actual de la aplicación sólo se pueden dar de alta desde de la pantalla de administración conjuntos de datos de tipo "Hugging face".

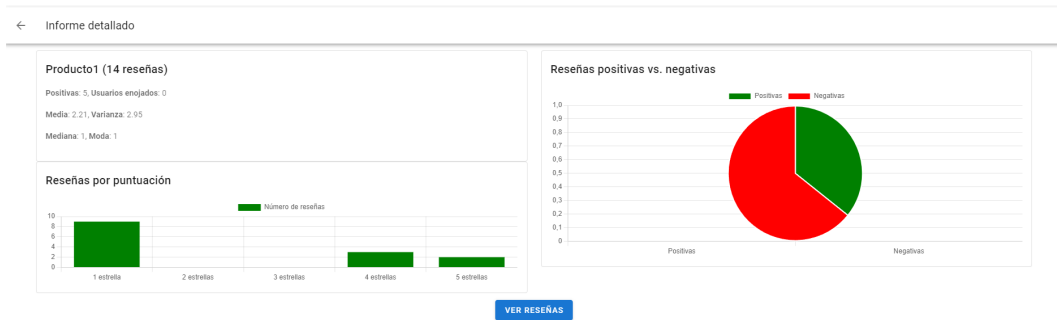
Estos datasets están alojados en la página huggingface.com y necesitan un JSON de configuración. Ejemplo:

```
{
  "name": "Amazon Shoe Reviews (test)",
  "type": "Hugging face",
  "path": "mesmalif/amazon-shoe-reviews",
  "subset": "test",
  "mapping": {
    "review_text": "text",
    "review_headline": "review_headline",
    "stars": "labels",
    "review_id": "review_id",
    "review_date": "review_date"
  },
  "correct_stars": 0,
  "date_format": "%Y-%m-%d"
}
```

Una vez configurado podemos cargarlo pulsado el icono  (cargar). Nos preguntará el porcentaje a cargar:



Cuando esté cargado se puede procesar pulsando el icono  (procesar). Al finalizar se pueden ver los resultados pulsando en el icono  (ver resultados)



8. Información

En la pantalla de información podemos ver los datos de los autores, información de las herramientas utilizadas para su desarrollo así cómo una acceso al código de la aplicación en github.

← Acerca de...	
AUTOR: Teodoro Ricardo García Sánchez tgs1003@alu.ubu.es	TUTORES: Virginia Ahedo García (vahedo@ubu.es) y Jose Ignacio Santos Martín (jisantos@ubu.es) Departamento de Ingeniería de Organización
Software:	
BACKEND: Python , Flask , Flask-restx , pytest , SqlAlchemy , PostgreSql , OpenAI	FRONTEND: Node.js , Vue.js , Vuetify , Chart.js , Vue-i18n
Otros:	
ABOUT INFRASTRUCTURE Ubuntu , Docker , Contabo , Nginx , Portainer , Let's Encrypt	EDITOR: Visual Studio Code
Desarrollado durante el curso 2023-2024 como trabajo de fin de grado en la Universidad de Burgos	
VER REPOSITORIO	

Anexo de sostenibilización curricular

F.1. Introducción

Durante la realización de este trabajo he tenido presente la importancia que tiene que un desarrollo sea sostenible.

Para hacer la aplicación he seguido una serie de técnicas reconocidas de forma general como buenas prácticas y que implican un enfoque orientado a la sostenibilidad. Entre otras he seguido las siguientes pautas:

- Uso herramientas de código abierto: Al usar herramientas de código abierto no solo ahorramos el coste de las licencias sino que contribuimos a su desarrollo como herramientas universales.
- Uso de patrones de desarrollo: los patrones de desarrollo son soluciones simples a problemas complejos. Haciendo uso de éstas conseguimos reducir la complejidad del desarrollo y lo hace más sostenible y accesible al resto de la comunidad de desarrolladores.
- Código autocomentado: código fácil de leer y entender para facilitar el apredidaje de futuros desarrolladores y su mantenimiento.
- Código libre y disponible en Github: haciendo el código abierto a todo el mundo facilitamos el desarrollo de futuros proyectos reduciendo su coste y complejidad (en muchos casos).

- Uso de marcos de desarrollo de uso generalizado: usar marcos de desarrollo de uso común hace que nuestra aplicación sea más mantenible y extensible para otros desarrolladores.
- Gestión de proyectos ágil: al seguir una metodología ágil se consigue reducir la complejidad del proyecto y adaptarse mejor a los cambios. Si en medio de un sprint surgía algún impedimento podía cambiar su planificación sin tener efecto en la velocidad del equipo.
- Principios KISS: los principios KISS se basan en mantener los desarrollos simples y no hacer desarrollos que no sean necesarios.
- Enfoque Devops: al hacer el despliegue de la aplicación por configuración y hacerse desde el primer momento éste presenta menos problemas al final del proyecto y la aplicación está publicada desde el primer momento. Esto también está en consonancia con los principios de la gestión ágil.
- Escalabilidad: al usar docker y nginx la aplicación está preparada para la escalabilidad desde el primer momento.
- Interfaz en múltiples idiomas: el tener una interfaz preparada para varios idiomas hace que sea más accesible a un grupo más amplio de usuarios.

Desde el punto de vista de costes de producción, debido a los bajos requisitos que tiene para poder funcionar en un entorno de producción, se puede alojar en multitud de servidores en función de su utilización.

Al estar configurada para funcionar en docker puede funcionar en multitud de entornos y configuraciones. Es fácil de escalar y se puede configurar en modo pago por uso (en muchos alojamientos). De esta forma el retorno de la inversión será muy rápido.

Como hemos visto en la sección de Viabilidad económica el único gasto fijo es el alojamiento (5€ al mes). Los gastos variables son el uso de OpenAI que depende exclusivamente del uso.

Desde un punto de vista social tener una herramienta que ayude a las empresas a identificar los puntos de mejora de un producto o servicio es bueno para todos. Por un lado la empresa mejora sus resultados al poder identificar los puntos de mejora demandados. Y por otro el cliente se siente escuchado y recibe un mejor servicio.

Bibliografía

- [1] Mark Lutz. *Programming python*. O'Reilly Media, Inc., 2001.