

Universidad
Rey Juan Carlos

Escuela Técnica Superior
de Ingeniería Informática

Grado en Ingeniería del Software

Curso 2023 - 2024

Trabajo Fin de Grado

**AI MENTAL HEALTH: DESARROLLO DE SISTEMA
DE SOPORTE A LA DECISIÓN PARA LA
DETECCIÓN DE SEÑALES DE SUICIDIO EN
TEXTOS**

Autor | José Luis Toledano Díaz

Tutor | Antonio González Pardo

Agradecimientos

Empezando esta sección por las personas a las que más debo en todos los años que tengo, quiero agradecer a mis padres y a mi hermano el apoyo constante que me han demostrado siempre. Cada día me han inculcado valores que me definen y moldean para ser mejor persona. No hay espacio suficiente en esta sección ni existen las palabras que puedan comunicar lo que me han dado durante toda mi vida. Le debo todo a mi familia. **Gracias.**

También quiero dar créditos a varios profesionales que me han acompañado en mi educación, pasando muchos de ellos a ser amigos y compañeros. Al profesor José Luis quien me introdujo en el mundo de las matemáticas, un primer paso hacia la ingeniería desde el colegio. A Sol y Julio, dos profesores que supieron ver el potencial escondido de un chico tímido e introvertido, asumiendo papeles más allá de los deberes del profesor. A Fernando por revolucionar mi ser y cambiarme, dándome la oportunidad de abrirme la mente y descubrir un gran potencial. **Gracias.**

En mi etapa universitaria conocí a muchas personas con las que he compartido el camino y compartiré la senda que queda por recorrer. Amigos como Diego, Flavia, Debi, Ismael, Agudo y Marcos me han ayudado a superar esta etapa y juntos hemos compartido muchos momentos especiales. Muchas gracias por estar ahí, espero expresar en este corto espacio todo lo que os debo. **Gracias.**

A mi actual pareja, Andrea, le debo un gran agradecimiento especial por lo buena que es conmigo. Una persona que te escuche y te apoye con la confianza de decir lo que realmente piensa en cada momento es un suceso poco común, por eso, muypreciado. **Gracias.**

Por último, pero no menos importante, quiero expresar mi agradecimiento a mi tutor y guía de este trabajo, Antonio González Pardo, por apoyarme en el proyecto y darme las ideas necesarias para poder desarrollar e ingeniar la herramienta. Además, agradezco a la Universidad Rey Juan Carlos la oportunidad que me ha ofrecido para estudiar y desarrollarme en el campo de la ingeniería del software. **Gracias.**

Resumen

El presente documento detalla la composición y desarrollo de una aplicación **DSS** (*Decision Support System*) destinada a especialistas en materia de salud mental o a usuarios con inquietudes en el auto-descubrimiento del estado de su psique. Además, permite el desarrollo en investigación de Inteligencia Artificial (AI, por sus siglas en inglés).

Constando de una interfaz de texto mediante consola (p. ej., CMD), la aplicación denominada **AIMentalHealth**, proporciona soporte a usuarios no especializados en la informática, como personal sanitario y a usuarios interesados en el desarrollo; e investigación de Inteligencia Artificial.

El código fuente de este proyecto se distribuye bajo licencia de código abierto Apache 2.0 mediante un repositorio público de GitHub ¹. En dicho repositorio se encuentra documentación adicional respecto al desarrollo y uso de la aplicación. Además, en el mencionado repositorio se almacenan modelos pre-entrenados específicos ² para el uso de **AIMentalHealth** y el *dataset* original utilizado para el entrenamiento de estos.

Palabras clave:

- Inteligencia Artificial
- Transformers
- DSS
- Procesamiento Lenguaje Natural
- Análisis de Sentimientos
- Salud Mental

¹Enlace al repositorio: <https://github.com/JLToledano/AIMentalHealthProblems>.

²En el caso de los modelos entrenados que sobrepasen el límite de almacenamiento de la plataforma GitHub, se habilitará una carpeta con los ficheros en OneDrive.

Abstract

The present document details the composition and development of a **DSS** (*Decision Support System*) application aimed at mental health specialists or users interested in self-discovery of their psyche's state. Additionally, it allows for development in Artificial Intelligence (AI) research.

Featuring a text-based console interface (e.g., CMD), the application named **AIMentalHealth** provides support to non-specialized computer users, such as healthcare personnel, and users interested in the development and research of Artificial Intelligence.

The source code of this project is distributed under the Apache 2.0 open-source license through a public GitHub repository ³. In this repository, additional documentation regarding development and usage of application can be found. Additionally, pre-trained specific models ⁴ for use of **AIMentalHealth** and original dataset used for their training are stored in mentioned repository.

Keywords:

- Artificial Intelligence
- Transformers
- DSS
- Natural Language Processing
- Sentiment Analysis
- Mental Health

³Repository link: <https://github.com/JLToledano/AIMentalHealthProblems>.

⁴In case of trained models that exceed storage limit of GitHub platform, a folder with files will be provided on OneDrive.

Índice de contenidos

Índice de figuras	XI
Índice de tablas	XIII
Abreviaturas	XV
1. Introducción	1
1.1. Motivación	1
1.2. Estructura	3
2. Objetivos	5
2.1. Apoyo a especialista sanitario	5
2.2. Desarrollo del estudio en IA	6
3. Tecnologías, herramientas y metodología	7
3.1. Python	7
3.2. Metodología de trabajo	8
4. Descripción del sistema	13
4.1. Motivos del desarrollo	13
4.2. Situaciones de uso real	14
4.3. Elecciones de diseño	15
5. Desarrollo del sistema	17
5.1. Requerimientos	17
5.1.1. Requisitos funcionales	19
5.1.1.1. Requisito funcional 001	19
5.1.1.2. Requisito funcional 002	19
5.1.1.3. Requisito funcional 003	19
5.1.1.4. Requisito funcional 004	19
5.1.1.5. Requisito funcional 005	20
5.1.1.6. Requisito funcional 006	20
5.1.1.7. Requisito funcional 007	20
5.1.1.8. Requisito funcional 008	21

5.1.1.9.	Requisito funcional 010	21
5.1.1.10.	Requisito funcional 011	21
5.1.1.11.	Requisito funcional 012	22
5.1.1.12.	Requisito funcional 015	22
5.1.2.	Otros requisitos	23
5.1.2.1.	Requisito dominio 001	23
5.1.2.2.	Requisito dominio 002	23
5.1.2.3.	Requisito no funcional 004	23
5.1.2.4.	Requisito no funcional 005	23
5.1.2.5.	Requisito no funcional 006	24
5.1.2.6.	Requisito no funcional 007	24
5.2.	Historias de usuario	24
5.3.	Diseño y arquitectura de la aplicación	26
5.4.	Implementación	27
5.4.1.	Aspectos generales	27
5.4.2.	Historia de usuario 003	29
5.4.3.	Historia de usuario 004	29
5.4.4.	Historia de usuario 005	31
5.4.5.	Historia de usuario 007	33
5.4.6.	Historia de usuario 008	34
5.4.7.	Historia de usuario 009	35
5.4.8.	Historia de usuario 011	36
5.5.	Pruebas configuraciones de red neuronal	37
6.	Conclusiones y trabajos futuros	39
6.1.	Conclusiones	39
6.2.	Trabajos futuros	40
	Bibliografía	43
	Apéndices	47
A.	Estudio modelos neuronales	49
A.1.	Introducción	49
A.2.	Modelos	50
A.3.	Entrenamiento	50
A.4.	Evaluación	50

Índice de figuras

1.1. Evolución de muertes por suicidio en la ciudad de Madrid.[3]	2
3.1. Posibilidades ofrecidas por la librería Rich [20]	9
3.2. Tablero Trello del proyecto.	11
5.1. Diagrama secuencia inicio de AIMentalHealth	27
5.2. Diagrama de flujo de AIMentalHealth	28
5.3. AIMentalHealth análisis de mensaje	29
5.4. AIMentalHealth resultado de un frase analizada	29
5.5. AIMentalHealth configuración de división de ficheros por directorios	30
5.6. AIMentalHealth división de ficheros por número indicado por pantalla	30
5.7. AIMentalHealth entrada incorrecta en selección de fichero o directorio	30
5.8. AIMentalHealth selección de despliegue de directorio	31
5.9. AIMentalHealth selección de fichero y carga del modelo	31
5.10. AIMentalHealth porcentaje de datos procesados	32
5.11. AIMentalHealth entrenamiento de red neuronal	32
5.12. AIMentalHealth evaluación de red neuronal	33
5.13. AIMentalHealth configuración de parámetros	34
5.14. AIMentalHealth confirmación de los cambios en la configuración .	34
5.15. AIMentalHealth panel de métricas	35
5.16. AIMentalHealth carga inicial del dataset predeterminado	36
A.1. Información de GPU	49
A.2. Información de memoria RAM	50
A.3. Información de memoria CPU	50

Índice de tablas

A.1. Configuraciones modelos de inteligencia artificial	51
A.2. Entrenamiento de modelos	52
A.3. Evaluación de modelos	53

Abreviaturas

- **AI:** Artificial Intelligence.
- **API:** Application Programming Interface.
- **ASCII:** American Standard Code for Information Interchange.
- **AUC:** Area Under the Curve.
- **BSD:** Berkeley Software Distribution.
- **CIAP:** Clasificación Internacional de Atención Primaria.
- **CMD:** Command Prompt.
- **COVID:** Coronavirus Disease 2019.
- **CPU:** Central Processing Unit.
- **DSS:** Decision Support System.
- **ERS:** Elicitation Requirements Specification.
- **FN:** False Negative.
- **FSME:** Fundación Española para la Prevención del Suicidio.
- **FP:** False Positive.
- **GPT:** Generative Pre-trained Transformer.
- **GPU:** Graphics Processing Unit.
- **HU:** Historias de Usuario.
- **IA:** Inteligencia Artificial.
- **IDE:** Integrated Development Environment.
- **KNN:** K-Nearest Neighbors.
- **LLM:** Large Language Models.

- **NLP**: Natural Language Processing.
- **RAM**: Random Access Memory.
- **ROC**: Receiver Operating Characteristic.
- **SO**: Sistema Operativo.
- **TFG**: Trabajo Fin de Grado.
- **TN**: True Negative.
- **TP**: True Positive.
- **TUI**: Text User Interface.
- **XOR**: Exclusive OR.
- **XP**: Extreme Programming.
- **YAML**: YAML Ain't Markup Language.

1

Introducción

En esta sección se abarcan las motivaciones iniciales que propulsaron el desarrollo del proyecto, las necesidades que se pretenden satisfacer y los problemas reales en los que se utilizaría el aplicativo. Además, se incluye la descripción de la estructura y división del presente documento, incluyendo explicaciones del contenido de cada uno de los apartados.

1.1. Motivación

En los últimos años, se ha subrayado en la sociedad un problema que afecta a gran parte de la población española. Siendo irrelevante la edad de la persona, desde la pandemia del COVID-19 se ha hecho notable el aumento de casos relacionados con enfermedades de la salud mental. Según la FSME [2], cerca de 4.000 personas se suicidaron en España en el año 2021. Además, más del 25 % de la población asignada a atención primaria presenta algunos de los problemas psicológicos o mentales recogidos en la CIAP-2 (bajo el epígrafe P-Problemas psicológicos).

La salud mental es un tema presente en la sociedad y muy investigado en los últimos años. Estos estudios han demostrado la gran necesidad de especialistas y herramientas dedicadas a este sector. La excesiva demanda de sesiones de atención privada y la poca oferta en el mercado actual provoca una situación de desequilibrio, teniendo como consecuencia listas de espera para el paciente de más de medio año.

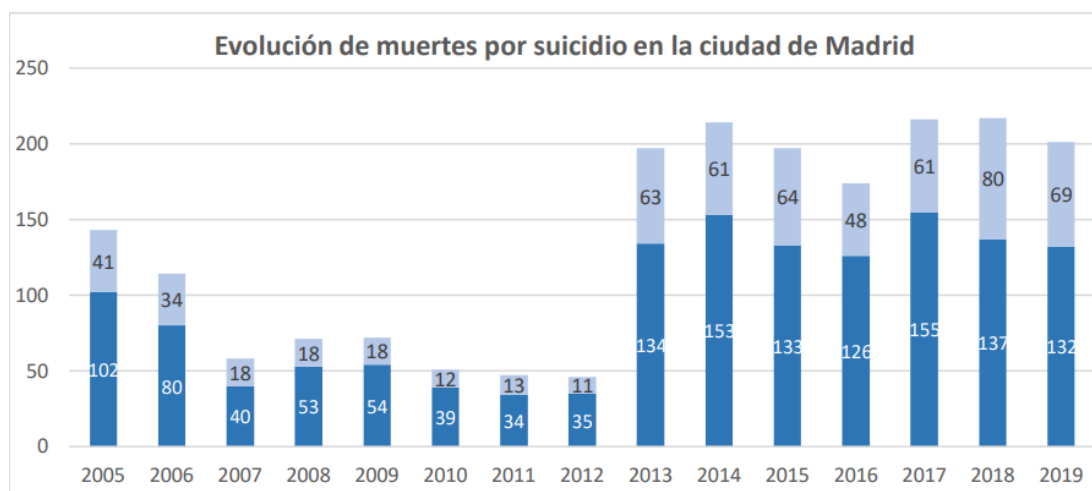


Figura 1.1: Evolución de muertes por suicidio en la ciudad de Madrid.[3]

En otro sector, la informática y el desarrollo de software, se investigan y desarrollan nuevos proyectos dedicados a la evolución de la Inteligencia Artificial. En la última década, han nacido nuevos modelos de redes neuronales, técnicas de entrenamiento y algoritmos de aprendizaje, dando como resultado herramientas dominantes y revolucionarias, teniendo el ejemplo reciente de Chat-GPT4. Consecuencia de esta popularidad; se han creado estudios superiores dedicados al área de Inteligencia Artificial, y el interés demostrado por estudiantes y profesores destaca la necesidad de plataformas y/o herramientas que faciliten el estudio.

Combinando las nuevas tecnologías emergentes de AI junto con las medidas tomadas por el gobierno para enfrentar la problemática del suicidio, es posible multiplicar la eficiencia y efectividad de las estrategias establecidas. Con la implementación de aplicaciones que faciliten o automaticen el diagnóstico, se agiliza el proceso de detección de señales de suicidio, mejorando, por ejemplo, el alcance los objetivos estratégicos 1 y 2 de las *Medidas de prevención del suicidio de la ciudad de Madrid*¹. En conjunto se consigue que las medidas tomadas lleguen a más población reduciendo así el número de suicidios cometidos y su aumento en los últimos años [ver Figura 1.1].

AIMentalHealth es un DSS destinado a dos posibles usos simultáneos. Su función principal consiste en el análisis de textos escritos para la detección de señales de suicidio, marcando finalmente dicho texto como suicidio o no suicidio. El segundo cometido de la aplicación es servir de herramienta de estudio en el desarrollo, entrenamiento, validación y uso de modelos neuronales basados en la tecnología Transformer.

¹Disponibles en el *Plan de actuación de prevención del suicidio del ayuntamiento de Madrid 2023-2024*. [4]

1.2. Estructura

La estructura del presente documento consiste en un total de seis secciones principales además de los apéndices añadidos y secciones propias de un documento de estas características: agradecimientos, resumen, abstract, índices, abreviaturas y bibliografía. Las secciones principales constituyen la mayor parte del contenido del documento y en ellas se refleja el trabajo completo del TFG.

En la sección 1 se detallan aspectos generales del proyecto desarrollado, explicando las motivaciones que condujeron a la idea desarrollada [subsección 1.1] y la estructura del documento [subsección 1.2]. Posteriormente, en la sección 2 se detallan en profundidad los objetivos concretos que se pretenden cumplir con la aplicación: el apoyo al especialista sanitario [subsección 2.1] y la implementación de una herramienta para impulsar y facilitar el estudio de las AI [subsección 2.2]. Seguidamente, en la sección 3 se detalla el lenguaje utilizado en la implementación del aplicativo [subsección 3.1] y la metodología de trabajo seguida [subsección 3.2].

La siguiente sección constituye uno de los pilares del documento, debido a que detallan los motivos por los cuales se desarrolla esta aplicación y las necesidades que suple [sección 4].

Las dos últimas secciones constituyen la explicación técnica de la propia aplicación [sección 5] y las conclusiones y trabajos futuros para esta [sección 6]. La descripción técnica se divide en subsecciones correspondientes a la definición técnica de requisitos [subsección 5.1] y de las historias de usuario [subsección 5.2], arquitectura de la aplicación [subsección 5.3] y la implementación de las principales historias de usuario [subsección 5.4], además de una última subsección destinada a la explicación de las configuraciones probadas para entrenar y evaluar modelos [subsección 5.5]. Por otro lado, la última sección del documento se divide en dos apartados correspondientes a las conclusiones [sección 6.1] y los trabajos futuros [sección 6.2].

El apartado de apéndices lo constituye un documento independiente: el estudio de modelos neuronales [apéndice A]. Consiste en un resumen de los resultados obtenidos por cada entrenamiento y evaluación de modelos, especificando los parámetros de configuración del modelo y el valor obtenido en cada medida.

Existe una versión extendida de este documento en un repositorio GitHub [<https://github.com/JLToledano/AIMentalHealthProblems>] junto con el proyecto desarrollado. Esta versión es autocontenida y se desarrollan en profundidad las herramientas utilizadas para el desarrollo además de las tecnologías y la explicación de todas las historias de usuario. Por último, se incluyen dos apéndices adicionales correspondiendo a una especificación técnica de requisitos completa y a la explicación detallada de la evolución del Machine Learning desde la neurona hasta Transformers.

2

Objetivos

Esta sección está dedicada a la explicación de los objetivos principales que se han marcado como metas en el desarrollo de la aplicación AIMentalHealth.

AIMentalHealth abarca dos metas principales: apoyo al especialista sanitario [subsección 2.1] y ser una herramienta de investigación para el campo de la Inteligencia Artificial [subsección 2.2].

2.1. Apoyo a especialista sanitario

AIMentalHealth surgió de la idea de querer ayudar a los especialistas sanitarios en área de psiquiatría e inteligencia emocional en sus análisis diarios ofreciendo una herramienta adicional a los *tests* ya existentes. Es interesante el concepto de una aplicación que ofrezca la posibilidad de agilizar el filtrado y análisis de textos escritos por un paciente para reducir los costes en tiempo y esfuerzo que supone un análisis.

Por otra parte, la aplicación está diseñada de tal forma que un paciente puede autoanalizar sus textos sin un especialista presente para después comunicar los resultados. Sin embargo, tal como se refleja en la propia documentación de AIMentalHealth, los resultados mostrados por la aplicación no determinan una sentencia final sin el posterior apoyo de un especialista.

La combinación de ambas opciones da lugar a un abanico de posibilidades para el desarrollo del diagnóstico psiquiátrico, permitiendo al doctor elegir cómo

utilizar la herramienta para adaptarse a las necesidades de cada paciente y al flujo y saturación de trabajo.

2.2. Desarrollo del estudio en IA

Debido al estudio de las redes neuronales complejo por la rápida evolución y el incremento exponencial en la complejidad de los modelos se dificulta la investigación para nuevos desarrolladores, la falta de herramientas simples con las que investigar el entrenamiento y evaluación de modelos simples supone una barrera importante a los nuevos ingenieros interesados en el área de las redes neuronales.

Reutilizando la plataforma creada con el fin de cumplir el primer objetivo, se han expandido los usos de AIMentalHealth para posibilitar la creación de nuevas redes neuronales con distintas configuraciones. De esta forma, se tiene una herramienta útil para entrenar, validar y probar redes neuronales con casos reales de forma unificada y sencilla.

Se estableció este segundo objetivo en la herramienta para ampliar el rango de usuarios a los que les interesaría utilizar la aplicación. Estudiantes de Inteligencia Artificial o iniciados en áreas similares podrían apreciar el valor de la aplicación gracias a su uso amplio e intuitivo para el desarrollo de redes neuronales. De esta forma, se abre una puerta a los iniciados y se reduce la curva de aprendizaje existente para las personas con pocos recursos tecnológicos debido a la oferta de pequeños modelos por parte del aplicativo.

3

Tecnologías, herramientas y metodología

En esta sección, el objetivo es explicar de las tecnologías implicadas en el desarrollo del aplicativo, además de las metodologías utilizadas.

3.1. Python

Python es un lenguaje de programación de alto nivel superando en uso a otros famosos lenguajes de programación como Java [14]. A diferencia de este último, es un lenguaje de programación interpretado; es decir, no necesita de compilación previa a la ejecución del código, además, es un lenguaje de tipado dinámico.

Una propiedad adicional de este lenguaje de programación es el soporte que ofrece a diversos paradigmas de programación, permitiendo al desarrollador decidir con qué paradigma programar, pudiendo usar varios en un mismo programa. Los paradigmas de programación admitidos por Python son el imperativo, orientado a objetos y el funcional. Gracias a este soporte, se pueden utilizar las características más ventajosas de cada paradigma para solucionar un mismo problema.

Por otro lado, la disponibilidad de una extensa biblioteca estándar, añadiendo un gran número de librerías implementadas con el fin de utilizarlas simultáneamente, permite a los desarrolladores experimentados desarrollar aplicaciones con más sencillez gracias a los recursos existentes y especializados.

Algunas de las características de programación de Python son:

- Los tipos de datos son dinámicos.
- Mezclar tipos incompatibles genera una excepción, por lo que los errores se detectan con facilidad. Esto se puede producir debido al intento de una suma entre una variable *string* y una variable numérica.
- Python contiene características avanzadas de programación, como generadores y comprensiones de listas.

Al tratarse de una aplicación utilizada por consola en el propio equipo del usuario, las tecnologías utilizadas para su desarrollo consisten en el lenguaje de programación Python y librerías compatibles.

- **Pandas:** Utilizada para la creación y manipulación de datos estructurados en tablas (*dataframes*).
- **Numpy:** Proyecto de código abierto que permite la computación numérica con Python.
- **Transformers:** Proporciona herramientas y API para entrenar de forma sencilla modelos preentrenados. Los modelos disponibles son compatibles y admiten la interoperabilidad con las principales tecnologías de desarrollo de inteligencia artificial utilizadas en Python: PyTorch, TensorFlow y JAX.
- **ScikitLearn:** Proporciona diversas herramientas y recursos que facilitan el entrenamiento y evaluación de un modelo. Entre las herramientas que ofrece, destacan funciones para clasificación, regresión, agrupamiento, reducción de dimensionalidad en el aspecto del modelo, además de funciones de preprocesamiento de datos.
- **Rich:** Posibilita al desarrollador implementar una TUI compatible con sistemas operativos Linux, macOS y Windows.

3.2. Metodología de trabajo

En los proyectos relacionados con la ingeniería del software, existen múltiples metodologías de trabajo que se pueden seguir en la situación de dirigir y coordinar un equipo o el propio trabajo. Entre las múltiples opciones como la tradicional con la documentación exhaustiva del trabajo o las más vanguardistas denominadas metodologías *agile*, se ha escogido estas últimas para el desarrollo de este trabajo de fin de grado.

La motivación principal de escoger una metodología *agile* para el desarrollo ha sido por su enfoque al factor humano y principalmente por su capacidad de respuesta al cambio. Además, estos métodos de trabajo son el futuro y presente de los proyectos software en empresas, por lo que su uso en este trabajo sirve como demostración de la adaptación hecha al futuro laboral.

Concretamente, se ha adaptado el *framework* ágil Scrum, respetando los valores detallados en el Manifiesto Ágil [24]. La adaptación ha consistido en la re-



Figura 3.1: Posibilidades ofrecidas por la librería Rich [20]

ducción o eliminación de las dinámicas de roles y trabajos en equipo propuestas debido a que el desarrollo ha sido realizado por una única persona. Sin embargo, se ha utilizado la organización propuesta para las tareas surgidas del proyecto y el trabajo mediante iteraciones incrementales.

Scrum [9] es una metodología ágil desarrollada en la década de 1990 por Jeff Sutherland y Ken Schwaber, pero no es hasta el año 2001 cuando se instaure como componente fundamental en el desarrollo software, debido a la firma del Manifiesto Ágil (en el que participaron ambos autores). La principal característica de esta metodología es la división por roles de los miembros de un equipo (no aplicable en este proyecto) y la organización del trabajo en dos «zonas» principales: *Product Backlog* (lista priorizada de todas las características y tareas del producto) y *Sprint Backlog* (lista de elementos en los que se trabajará en la iteración). El *Sprint Backlog* se divide en tres secciones por las que las tareas a realizar tienen que pasar:

- **TO DO:** Tareas estudiadas y preparadas para ser desarrolladas.
- **DOING:** Tareas en desarrollo.
- **DONE:** Tareas finalizadas.

En el desarrollo del proyecto también se ha utilizado otro *framework agile* denominado Extreme Programming, presentando así los requisitos de la aplicación de una forma cercana al usuario, simple y comprensible.

La pizarra de trabajo se ha implementado con la herramienta Trello [7], como se puede apreciar en la imagen 3.2, se ha estructurado siguiendo la metodología Scrum. Es apreciable en la organización de la pizarra en dos partes: *Product Backlog* y *Sprint Backlog* (este último dividido ya en sus columnas TO DO, DOING y DONE). Como añadido, se tiene una columna de información en la que se detallan aspectos generales del proyecto, como el nombre de este y estándares a seguir.

Por último, se ha utilizado un sistema de estimación para las historias de usuario creadas en el proyecto denominado *planning poker* consistente en asignar, en este caso, una talla de ropa (XS, S, M, L, XL) a cada historia para así apreciar la cantidad de tiempo y trabajo necesarios para cada una de ellas. El resultado es apreciable en el tablero Trello (imagen 3.2) en la columna de *Product Backlog*, en el centro de cada HU.

Como añadido, el sistema de control de versiones seleccionado y utilizado en el desarrollo de la aplicación ha sido Git, la plataforma de almacenamiento GitHub y el flujo de trabajo GitFlow.

Git [22] es un sistema de control de versiones que permite a los desarrolladores construir un seguimiento de los cambios en un proyecto software. Utilizando

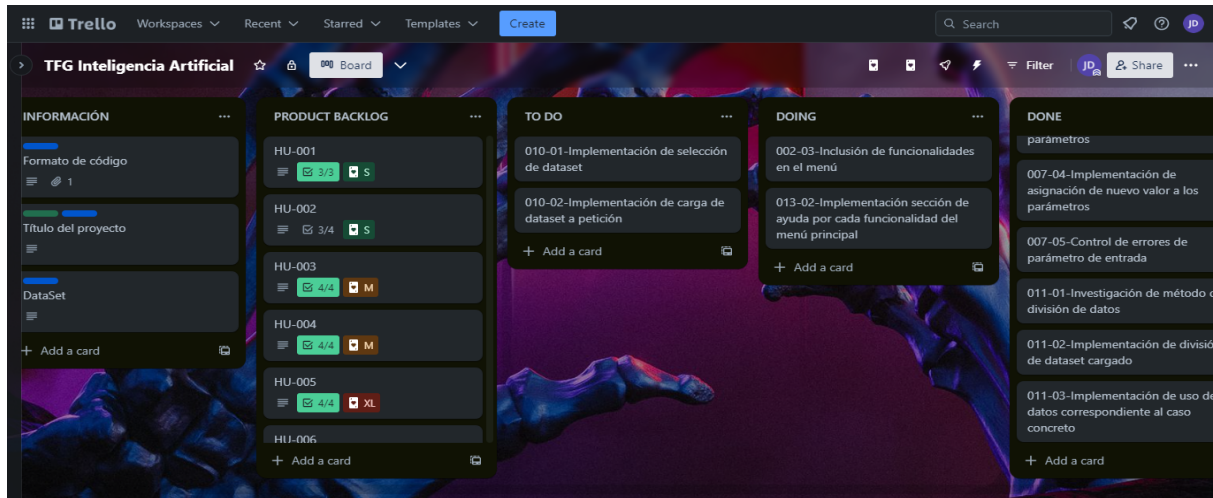


Figura 3.2: Tablero Trello del proyecto.

una analogía de ramificaciones con una raíz común, este sistema de control de versiones permite el desarrollo en paralelo a varios programadores en múltiples ramas, y los conflictos generados serán resueltos al momento de fusionar dichas ramas. El proyecto software se almacena en la plataforma GitHub, que utiliza Git como sistema de control de versiones, permitiendo el acceso al repositorio en remoto mediante web.

Por último, en la gestión de *commits* y ramas de Git, se ha utilizado la metodología GitFlow [23]. Esta metodología de ramificación se basa en el uso de diferentes tipos de ramas con objetivos específicos. Las ramas básicas existentes son:

- **Master**
- **Develop**
- **Feature**
- **Release**
- **Hotfix**

4

Descripción del sistema

Esta sección incluye información respecto a los fundamentos seguidos, estudiados y desarrollados para la creación del aplicativo. El estudio de concepto consiste en el por qué de la aplicación, casos reales de posible uso y elecciones de diseño.

4.1. Motivos del desarrollo

Reflexionando sobre las principales razones por las que se ha desarrollado el sistema presentado se reflejan las necesidades de la sociedad actual que satisface. Destaca el aumento de las listas de espera para tratamiento psicológico, el aumento de casos de suicidio y la integración mayor de la IA.

El uso de la aplicación por la comunidad especialista en tratamiento psicológico permite el ahorro del tiempo dedicado al análisis del paciente debido a que se pueden filtrar los mensajes escritos por el sujeto en un corto periodo de tiempo. El proceso de lectura por parte del especialista consumiría un tiempo excesivo provocando una demora en la atención, causa del aumento de las listas de espera.

Otra causa que desemboca en la primera motivación del desarrollo es el aumento de casos de suicidio en la sociedad moderna. Los casos no tratados de depresión y enfermedades mentales con síntomas similares pueden desembocar en un alto porcentaje en suicidio o intento de suicidio. La aplicación desarrollada permite a los propios usuarios autodiagnosticarse (en cierto grado debido

a la falta de formación en la mayoría de los casos). Sin embargo, el aplicativo puede ofrecer las primeras señales de un posible trastorno depresivo existente en los mensajes escritos por el usuario en diversas plataformas e introducidos en la herramienta para su análisis. De esta forma el usuario puede observar de forma aproximada si realmente está sufriendo el episodio de depresión.

Por último, una herramienta sencilla que permita la manipulación de parámetros de configuración (hiperparámetros) de los modelos neuronales es llamativa debido al área en el que se enfoca su funcionamiento, la IA. Los ingenieros y en general la sociedad tienen interés por los nuevos sistemas que utilizan Inteligencia Artificial en su desarrollo y la implementación de una herramienta que permita de forma sencilla su uso y estudio es aceptada con deseo.

4.2. Situaciones de uso real

La aplicación desarrollada ofrece a sus múltiples perfiles de usuario una extensa variedad de funcionalidades que son posibles de utilizar en diversas situaciones. Estas variaciones difieren en la temática de uso y los objetivos con los que se utiliza la aplicación.

La aplicación se puede emplear de diversas formas en función de los objetivos que persiga el usuario. En la siguiente lista se recogen algunos de los más relevantes:

- **Autoanálisis:** Un usuario en su privacidad puede analizar los mensajes que escribe para saber si el sentimiento demostrado en los escritos corresponde a un posible suicidio. Con este objetivo se aportan modelos preentrenados para evitar al usuario la necesidad de entrenar uno.
- **Diagnóstico:** Un usuario no especializado en informática dedicado al sector del análisis psicológico puede usar la aplicación como herramienta de diagnóstico con sus pacientes. De esta manera se filtran masivamente los mensajes que no muestren señales de suicidio pudiendo centrar el esfuerzo y tiempo en los mensajes relevantes. Además, puede entrenar nuevos modelos para personalizarlos a cada paciente si así lo desea.
- **Estudio de NLP:** Un estudiante de ingeniería del software o de alguna rama del conocimiento similar interesado en la AI, concretamente en el análisis del lenguaje natural, tiene la posibilidad de utilizar la aplicación para estudiar cómo funciona esta técnica y en qué consiste en todas sus fases.
- **Estudio de configuraciones:** La aplicación dispone de la opción de personalizar los hiperparámetros que influyen en el entrenamiento y evaluación

de modelos para posibilitar el estudio de la mejor configuración con un modelo y dataset determinados.

- **Desarrollo de modelos:** Estudiantes e investigadores tienen la opción de investigar con datasets de diferentes temáticas el desarrollo de modelos de análisis de lenguaje natural y realizar clasificación binaria.

4.3. Elecciones de diseño

El objetivo de la presente sección es justificar de forma concisa y breve las principales decisiones de diseño relevantes en el producto final desarrollado.

En primer lugar, el "entorno gráfico" escogido para la herramienta ha sido consola de comandos debido a la sensación de seguridad que aporta al usuario. Un usuario con conocimientos en ingeniería informática es capaz de reconocer los riesgos que puede sufrir un dispositivo mediante la manipulación con comandos por parte de terceros. Sin embargo, un usuario no informado tiene la sensación de seguridad al no estar publicando datos fuera del entorno de su ordenador como podría ocurrir en una página web. Sin las adecuadas medidas de seguridad ambos casos son igual de vulnerables pero utilizando consola los usuarios sin conocimiento informático tienen sensación de seguridad beneficiosa para que se empiece a utilizar la aplicación.

Por otro lado, las opciones disponibles en la aplicación están colocadas de tal forma que las opciones más complejas de entender estén en la parte central del listado debido a que un usuario lee en una lista corta el principio y el final primero, donde se sitúan las opciones de análisis de mensaje y ayuda/salir respectivamente.

5

Desarrollo del sistema

La presente sección contiene información detallada del desarrollo informático realizado, demostrando el alcance de los objetivos establecidos. En primer lugar se describen las historias de usuario creadas y seguidas para el desarrollo de la aplicación. Posteriormente se detalla la implementación para cada HU y en último lugar la verificación del software y su despliegue.

El código fuente resultado del desarrollo se encuentra disponible de forma pública en un repositorio GitHub.

5.1. Requerimientos

Esta sección refleja el resultado del proceso de análisis, extracción y redacción de los requisitos exigidos para el desarrollo del aplicativo desarrollando los principales requisitos divididos en tres secciones: requisitos funcionales [subsección 5.1.1], requisitos de dominio y requisitos no funcionales [subsección 5.1.2].

La codificación de los requerimientos consiste en un número identificativo de tres dígitos precedido de la partícula RF/RD/RNF: **RF-XXX** según corresponda (**R**equisito **F**uncional, **R**equisito de **D**ominio, **R**equisito **N**o **F**uncional respectivamente). Además, en el desarrollo de cada uno de los requisitos se especifica el código, su título, el procesamiento y el output.

El listado completo de los requerimientos es el siguiente:

- **RF-001**: Análisis de texto introducido por teclado.
- **RF-002**: Implementación diversas tecnologías Transformers.
- **RF-003**: Entrenamiento de red neuronal.
- **RF-004**: Almacenamiento modelo entrenado en fichero.
- **RF-005**: Selección nombre fichero modelo entrenado.
- **RF-006**: Evaluación de modelo de red neuronal existente.
- **RF-007**: Métricas de inteligencia artificial.
- **RF-008**: Carga de dataset.
- **RF-009**: Elección de nuevo dataset.
- **RF-010**: División de dataset.
- **RF-011**: Selección de modelo entrenado.
- **RF-012**: Personalización parámetros de configuración.
- **RF-013**: Menú de ayuda.
- **RF-014**: Opción fin de ejecución.
- **RF-015**: Control de input incorrecto.
- **RD-001**: Sistemas operativos admitidos.
- **RD-002**: Gestor de dependencias.
- **RNF-001**: Mensaje de bienvenida.
- **RNF-002**: Menú minimalista y secuencial.
- **RNF-003**: Estructuración de menú de ayuda.
- **RNF-004**: Muestra de datos cargados.
- **RNF-005**: Panel de métricas de entrenamiento/evaluación.
- **RNF-006**: Muestra valor parámetro.
- **RNF-007**: Muestra ficheros de modelos entrenados.
- **RNF-008**: Mensaje de advertencia.

5.1.1. Requisitos funcionales

5.1.1.1. Requisito funcional 001

- **Código:** RF-001
- **Título:** Análisis de texto introducido por teclado
- **Input:** Mensaje de texto en inglés.
- **Procesamiento**
Es necesario la existencia de una funcionalidad que pida al usuario un mensaje de texto para ser analizado por la inteligencia artificial devolviendo la clasificación deducida.
- **Output:** La aplicación muestra por pantalla el resultado del análisis siendo Suicide o Non-Suicide.

5.1.1.2. Requisito funcional 002

- **Código:** RF-002
- **Título:** Implementación diversas tecnologías Transformers
- **Input:** —
- **Procesamiento**
Implementación de modelo de red neuronal integrando una tecnología Transformers existente relacionada con el análisis de texto. Se debe posibilitar diversas tecnologías para su posible elección por parte del usuario.
- **Output:** Posibilidad de entrenar/evaluar un modelo con diversas tecnologías relacionadas con el análisis de texto y basadas en la tecnología Transformers

5.1.1.3. Requisito funcional 003

- **Código:** RF-003
- **Título:** Entrenamiento de red neuronal
- **Input:** Tecnología Transformers y dataset.
- **Procesamiento**
Dada una tecnología Transformers seleccionada por el usuario y un dataset válido cargado, se entrena un modelo según ciertos parámetros de configuración (número de etapas, tamaño de bloque, ...).
- **Output:** Modelo de red neuronal entrenado.

5.1.1.4. Requisito funcional 004

- **Código:** RF-004

- **Título:** Almacenamiento modelo entrenado en fichero
- **Input:** Modelo de red neuronal entrenado.
- **Procesamiento**
Dado un modelo de red neuronal entrenado por la aplicación se debe almacenar en un fichero con el formato estándar .pt para permitir su posterior uso.
- **Output:** Fichero con extensión .pt.

5.1.1.5. Requisito funcional 005

- **Código:** RF-005
- **Título:** Selección nombre fichero modelo entrenado
- **Input:** Fichero con extensión .pt.
- **Procesamiento**
El usuario tiene que indicar el nombre del fichero que almacena el modelo de red neuronal entrenado y lo especifica utilizando el teclado de la computadora dónde se ejecuta la aplicación.
- **Output:** Fichero con extensión .pt teniendo como nombre el especificado por el usuario.

5.1.1.6. Requisito funcional 006

- **Código:** RF-006
- **Título:** Evaluación de modelo de red neuronal existente
- **Input:** Fichero con extensión .pt.
- **Procesamiento**
Dado un fichero con extensión .pt seleccionado por el usuario se evalúa el modelo almacenado en el fichero según ciertos parámetros de configuración (número de etapas, tamaño de bloque, ...).
- **Output:** —

5.1.1.7. Requisito funcional 007

- **Código:** RF-007
- **Título:** Métricas de inteligencia artificial
- **Input:** Proceso de entrenamiento/evaluación de modelo.
- **Procesamiento**
Por cada etapa completada en un entrenamiento/evaluación de un modelo se debe calcular y mostrar las principales métricas del área de la inteligencia artificial además de otras complementarias como el tiempo de entrenamiento/evaluación, el porcentaje, en tiempo real; de datos analizados en la

duración de cada etapa y número de etapa. Las métricas necesarias son las siguientes:

- Matriz de confusión
 - Exactitud
 - Sensibilidad
 - Precisión
 - F1
- **Output:** Resultado de métricas impresas por consola de comandos.

5.1.1.8. Requisito funcional 008

- **Código:** RF-008
- **Título:** Carga de dataset
- **Input:** Fichero con datos y resultados.
- **Procesamiento**

Al inicio de la aplicación se debe cargar y estructurar los datos disponibles en un fichero .csv para su utilización en los modelos. La estructura debe contener por cada elemento un campo texto con el mensaje original y un campo con la clasificación predeterminada del texto (suicide/non-suicide).
- **Output:** Estructura de datos cuyo contenido corresponde a los datos de entrenamiento/evaluación.

5.1.1.9. Requisito funcional 010

- **Código:** RF-010
- **Título:** División de dataset
- **Input:** Estructura de datos con la información del dataset.
- **Procesamiento**

La estructura de datos que almacena la información del dataset se divide en dos secciones representando los datos para entrenamiento y los datos para evaluación. Es necesario dividir los datos de manera aleatoria y además siguiendo el porcentaje estándar por cada sección. El porcentaje estándar en el entrenamiento y evaluación de los modelos de red neuronal indica que se debe destinar el 80 % de los datos a entrenamiento y el 20 % de los datos a evaluación.
- **Output:** Dos estructuras de datos con los respectivos datos destinados a entrenamiento y a evaluación.

5.1.1.10. Requisito funcional 011

- **Código:** RF-011
- **Título:** Selección de modelo entrenado

- **Input:** Opción elegida por el usuario (dependiendo del sistema elegido puede ser el nombre del modelo, su posición, ...).
- **Procesamiento**
El usuario debe elegir el modelo a utilizar entre las opciones disponibles para su uso o la evaluación del modelo. El modelo es seleccionado utilizando el teclado.
- **Output:** Modelo seleccionado.

5.1.1.11. Requisito funcional 012

- **Código:** RF-012
- **Título:** Personalización parámetros de configuración
- **Input:** —
- **Procesamiento**
El usuario debe de tener la opción de personalizar los parámetros de configuración predefinidos en la aplicación. Estos contarán con un valor predefinido al iniciar la aplicación pero el nuevo valor dado por el usuario debe perdurar hasta la finalización de la ejecución de la aplicación. Los parámetros configurables son:
 - Máxima longitud input
 - Tamaño de lote
 - Número de épocas
 - Número de procesos paralelos
 - Drop out
- **Output:** Parámetros de configuración actualizados.

5.1.1.12. Requisito funcional 015

- **Código:** RF-015
- **Título:** Control de input incorrecto
- **Input:** Valor introducido por teclado.
- **Procesamiento**
Dada una petición de datos por teclado hecha al usuario se debe comprobar que el valor introducido es correcto acorde a la situación presente. En caso contrario se debe informar por pantalla que el valor introducido no es correcto y requerir un nuevo valor.
- **Output:** —

5.1.2. Otros requisitos

5.1.2.1. Requisito dominio 001

- **Código:** RD-001
- **Título:** Sistemas operativos admitidos
- **Input:** —
- **Procesamiento**
La aplicación debe de ejecutarse sin inconvenientes en las consolas de comandos de los SO Windows y Linux.
- **Output:** —

5.1.2.2. Requisito dominio 002

- **Código:** RD-002
- **Título:** Gestor de dependencias
- **Input:** —
- **Procesamiento**
La aplicación debe ejecutarse en un entorno virtual Conda para evitar conflictos con las versiones de las librerías de las que depende.
- **Output:** —

5.1.2.3. Requisito no funcional 004

- **Código:** RNF-004
- **Título:** Muestra de datos cargados
- **Input:** —
- **Procesamiento**
Al cargar el dataset al iniciar la aplicación o cuando se cambie el dataset se debe imprimir el número de datos destinados a entrenamiento y a evaluación respectivamente en dos recuadros.
- **Output:** Número de datos cargados para entrenamiento y evaluación.

5.1.2.4. Requisito no funcional 005

- **Código:** RNF-005
- **Título:** Panel de métricas de entrenamiento/evaluación
- **Input:** —
- **Procesamiento**
Dada una situación de entrenamiento o de evaluación de un modelo se debe

mostrar por CMD un panel dividido en secciones y formando tablas para que se pueda saber el valor de una métrica en un golpe visual.

Se debe mostrar en la parte de arriba el tiempo de duración de la etapa y debajo dos secciones en horizontal. La sección de la izquierda debe corresponder a la matriz de confusión en formato tabla y la sección derecha las métricas restantes.

- **Output:** Panel de métricas estructurado y modularizado.

5.1.2.5. Requisito no funcional 006

- **Código:** RNF-006
- **Título:** Muestra valor parámetro
- **Input:** —
- **Procesamiento**

En la situación en la que se pida al usuario cambiar el valor de un parámetro se deberá mostrar junto a la petición correspondiente el valor actual de dicho parámetro entre paréntesis.

- **Output:** Valor actual entre paréntesis del parámetro.

5.1.2.6. Requisito no funcional 007

- **Código:** RNF-007
- **Título:** Muestra ficheros de modelos entrenados
- **Input:** Lista de ficheros entrenados.
- **Procesamiento**

En la situación en la que se requiera la elección de un modelo entrenado se deben mostrar los ficheros existentes simulando una estructura de directorios. Para ello se necesita que el usuario especifique cuántos ficheros quiere por directorio y posteriormente que indique si quiere desplegar un directorio para ver su contenido o seleccionar un fichero (por su posición). Además, los ficheros se deben ordenar por orden alfabético.

- **Output:** Simulación de estructura de directorios por pantalla.

5.2. Historias de usuario

Partiendo de los objetivos establecidos en la sección 2 y junto con la especificación de requisitos disponible en apéndices de la versión extendida de este documento se procede a definir y detallar las historias de usuario que han desembocado en la creación y desarrollo del aplicativo.

Siguiendo **eXtreme Programming** (XP) y su definición de **historia de**

usuario por Kent Beck en la que se indica que una HU consiste en una descripción breve y simple de una característica, funcionalidad o requisito del sistema desde la perspectiva del usuario o cliente; se han estructurado siguiendo el esquema propuesto:

Como [*rol del usuario*], quiero [*hacer algo*] para [*lograr algún objetivo o beneficio*].

La codificación de las historias de usuario consiste en un número identificativo de tres dígitos precedido de la partícula HU: **HU-XXX**. Además, en el desarrollo de cada una de las historias de usuario se especifica qué requisitos abarca y qué se ha implementado para cumplirlos.

- **HU-001**: Como usuario, quiero un mensaje o señal para saber que se ha iniciado la ejecución de la aplicación.
- **HU-002**: Como usuario, quiero un menú principal para acceder rápidamente a las funcionalidades.
- **HU-003**: Como usuario no especializado, quiero analizar textos evitando configuraciones para comprender lo que realizo y sus resultados.
- **HU-004**: Como usuario, quiero poder utilizar modelos ya existentes para clasificar los textos.
- **HU-005**: Como usuario especializado, quiero entrenar un modelo y evaluarlo cuando requiera para poder determinar la factibilidad de un modelo.
- **HU-006**: Como usuario especializado, quiero elegir entre varias tecnologías para construir mi modelo.
- **HU-007**: Como usuario especializado, quiero configurar los parámetros del modelo para personalizar los entrenamientos y evaluaciones.
- **HU-008**: Como usuario especializado, quiero saber los resultados de las métricas estándar en AI para estudiar un modelo.
- **HU-009**: Como usuario especializado, quiero que se cargue un dataset pre-determinado para poder entrenar un modelo.
- **HU-010**: Como usuario especializado, quiero cargar un dataset propio para entrenar modelos con mis ejemplos.
- **HU-011**: Como usuario especializado, quiero que los datos se dividan en entrenamiento y evaluación para poder estudiar un modelo.

- **HU-012:** Como usuario, quiero poder apagar la aplicación para finalizar su ejecución sin errores.
- **HU-013:** Como usuario, quiero un menú de ayuda para saber en que consiste cada opción disponible.
- **HU-014:** Como dueño de la aplicación, quiero una advertencia a los usuarios para evitar consecuencias legales derivadas de los resultados de clasificación.

5.3. Diseño y arquitectura de la aplicación

La arquitectura de la aplicación AIMentalHealth está dividida en dos grandes secciones correspondientes al componente principal de esta y al conjunto de dependencias utilizadas en la aplicación.

Definiendo en primer lugar la arquitectura de las dependencias de la aplicación, esta arquitectura se rige por el modelo modular; debiéndose a la independencia de los diversos códigos fuente correspondientes a cada librería utilizada (especificadas en la sección 3.1). Mediante un entorno virtual para evitar conflictos con lo existente en el entorno local se importan las librerías y se almacenan en el entorno para, posteriormente, utilizar funciones de estas en la implementación de la aplicación.

La arquitectura de la aplicación principal corresponde a una arquitectura monolítica debido a la facilidad de la implementación en esta primera versión de la aplicación. Además, permite un despliegue sencillo facilitando el uso de los primeros usuarios y simplificando el proceso de creación y diseño.

Con referente al diseño de la aplicación se ha optado por un sistema que cargue las configuraciones iniciales y el dataset por defecto mientras se refleja el menú de bienvenida siendo así un proceso transparente por el usuario [imagen 5.1]. Posteriormente, se accede a un menú principal del cuál inician todas las posibles funcionalidades disponibles y que una vez concluidas regresan al mismo menú exceptuando la finalización de la aplicación. Este flujo se ha diseñado con la motivación de facilitar a un usuario poco experimentado en el uso de la CMD o PowerShell la utilización de la aplicación.

La imagen 5.2 refleja el diagrama de flujo de AIMentalHealth.

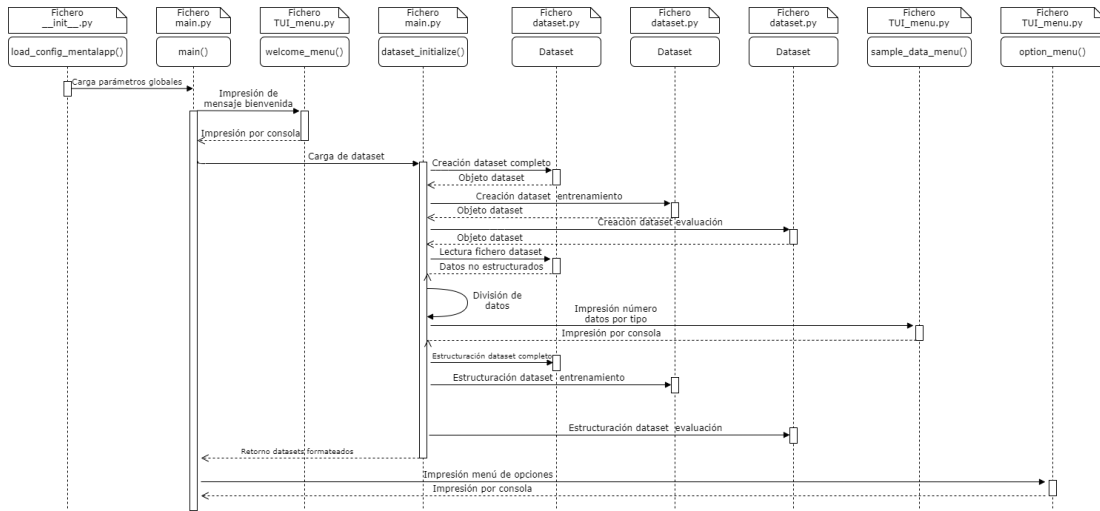


Figura 5.1: Diagrama secuencia inicio de AIMentalHealth

5.4. Implementación

El desarrollo de la presente sección es estructurada por algunas de las historias de usuario definidas en la sección 5.2. El primer apartado recoge los cimientos generales que permiten el funcionamiento de la aplicación y los siguientes apartados con HU asignada relacionarán el objetivo de esta con los requisitos que abarque además de los detalles técnicos consecuentes.

5.4.1. Aspectos generales

La aplicación AIMentalHealth consume para la carga de los parámetros utilizados un fichero `.yaml`¹ permitiendo así la utilización de la configuración como constantes globales. Sin embargo, aún definiendo lo disponible en el fichero como constante; solamente se mantendrá el valor predefinido (inicial) de estos hasta que el usuario desee expresamente realizar un cambio de valor. Este nuevo valor se mantendrá hasta un nuevo cambio o hasta el fin de ejecución de la aplicación ya que en un nuevo inicio los valores cargados serán los predefinidos.

La carga de los parámetros se realiza en formato diccionario, es decir, pares clave-valor en los que invocando la variables con su nombre (clave) se obtiene su magnitud (valor).

¹Formato de texto plano utilizado para representar datos estructurados de forma legible para el ser humano.

5.4. Implementación

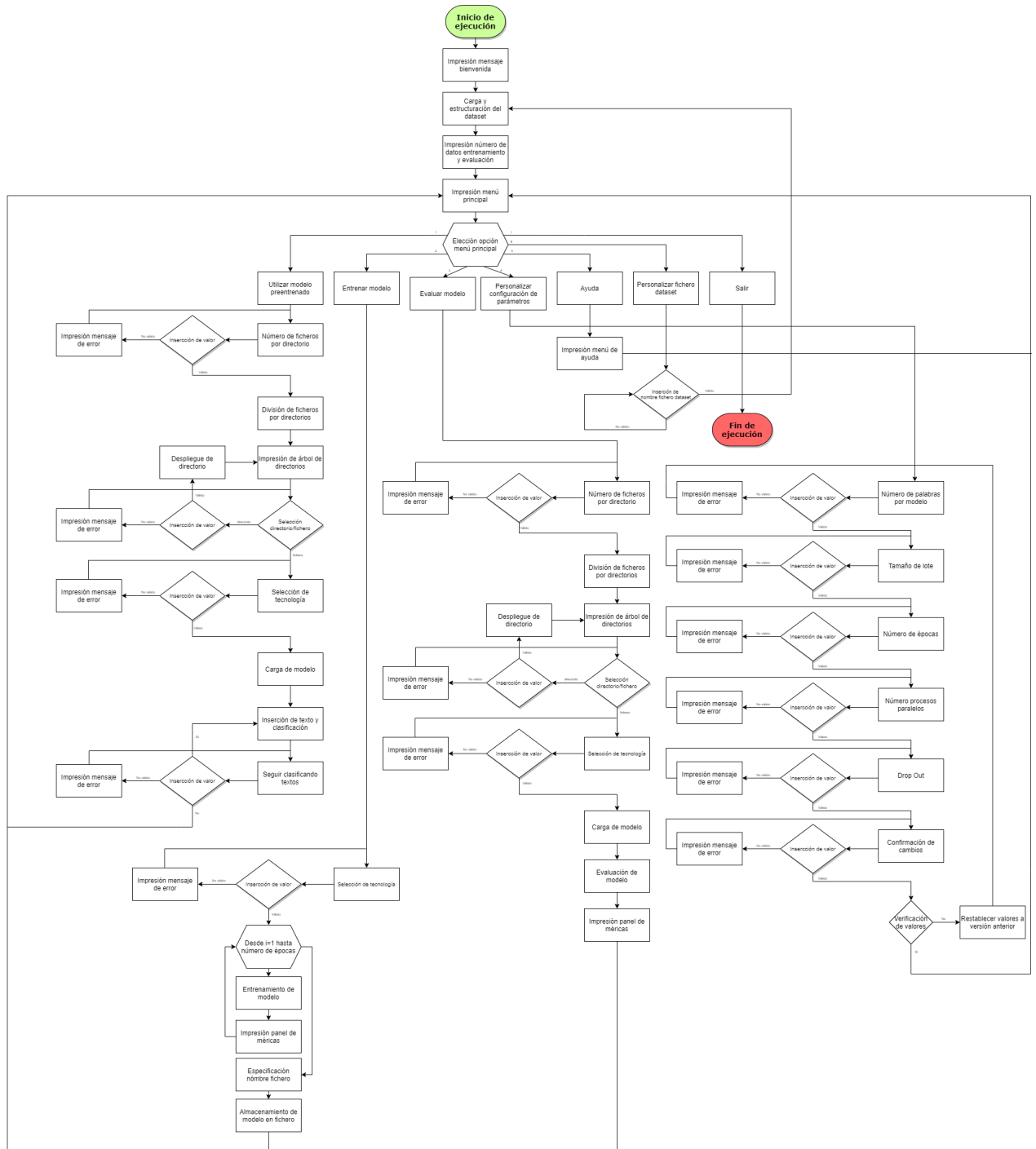


Figura 5.2: Diagrama de flujo de AIMentalHealth

```
Inserte el texto que quiere clasificar en inglés:  
I love my life_
```

Figura 5.3: AIMentalHealth análisis de mensaje

```
Inserte el texto que quiere clasificar en inglés:  
I love my life  
  
Clasificación: Non-Suicide
```

Figura 5.4: AIMentalHealth resultado de un frase analizada

5.4.2. Historia de usuario 003

«Como usuario no especializado, quiero analizar textos evitando configuraciones para comprender lo que realizo y sus resultados.»

Historia de usuario relacionada con los requisitos: [RF-001](#), [RF-002](#), [RF-015](#)

Incluyendo un acceso para esta funcionalidad en el menú principal [opción 1 del menú], una vez seleccionada por el usuario se escoge un modelo de red neuronal para el análisis. El modelo es cargado desde un fichero .pt cuyo contenido corresponde a las diversas capas de la red con los pesos de las conexiones y sesgos de cada neurona.

Con el modelo ajustado y desplegado con su tecnología asociada se indica al usuario que inserte un texto para procesar detallando que debe ser en inglés. Introduciendo el texto mediante teclado o pegado sobre la CMD se procede a analizar con la red neuronal y se devuelve el resultado [imagen 5.3].

El proceso de análisis es transparente para el usuario y se muestra en una línea contigua la clasificación resultante siendo una de las dos disponibles: Suicide o Non-Suicide. Un ejemplo práctico es apreciable en la imagen 5.4.

5.4.3. Historia de usuario 004

«Como usuario, quiero poder utilizar modelos ya existentes para clasificar los textos.»

Historia de usuario relacionada con los requisitos: [RF-011](#), [RNF-006](#), [RNF-007](#)

En los diversos puntos de la aplicación en la que se requiere la selección de un modelo de red neuronal pre-entrenado se utiliza la implementación de la división y estructuración de ficheros.

```

Por favor indique el número de ficheros por directorio (25): input no válido
Please enter a valid integer number
Por favor indique el número de ficheros por directorio (25):
Directorios
└─ 1. Ficheros del 1 al 11

```

Figura 5.5: AIMentalHealth configuración de división de ficheros por directorios

```

Por favor indique el número de ficheros por directorio (25): 3
Directorios
└─ 1. Ficheros del 1 al 3
└─ 2. Ficheros del 4 al 6
└─ 3. Ficheros del 7 al 9
└─ 4. Ficheros del 10 al 11

```

Figura 5.6: AIMentalHealth división de ficheros por número indicado por pantalla

En primer lugar se pide al usuario el número de ficheros que quiere por “directorio” y mostrando el valor predefinido si el input introducido es vacío. En caso de no ser una entrada válida se indica al usuario que inserte un valor válido [imagen 5.5].

Seleccionado un número válido, la aplicación busca en el directorio de modelos `ruta_aplicacion/mentalapp/models/` incluido en el repositorio de código del aplicativo y, ordenando por orden alfabético, se dividen los ficheros en grupos de tantos elementos como haya indicado el usuario. Por cada grupo formado se muestra una línea con origen en la “raíz” que indica qué conjunto numérico de ficheros abarca [imagen 5.6].

Posteriormente, se pide al usuario si desea un directorio o un fichero teniendo que introducir unos de estos dos valores por teclado. En caso de insertar un valor distinto o vacío se muestra un mensaje de error y se exige nuevamente [imagen 5.7].

En el primer caso mostrado, selección de directorio, se pide un número de directorio entre los existentes (formados y mostrados con anterioridad). Una vez seleccionado uno de ellos se volverá a mostrar las líneas anteriores pero en el caso del directorio seleccionado se muestra en un subnivel los ficheros que contiene junto con su número correspondiente (en distinto color). Se vuelve a pedir nueva-

```

¿Qué desea seleccionar? [directorio/fichero]:
Please select one of the available options
¿Qué desea seleccionar? [directorio/fichero]: entrada no válida
Please select one of the available options
¿Qué desea seleccionar? [directorio/fichero]:

```

Figura 5.7: AIMentalHealth entrada incorrecta en selección de fichero o directorio

```

¿Qué desea seleccionar? [directorio/fichero]: directorio
Por favor indique el número de directorio a desplegar (1): 2
Directorios
├─ 1. Ficheros del 1 al 3
├─ 2. Ficheros del 4 al 6
│   ├── 4. 0004_BERT_200_16_1_1_03_25_80.pt
│   ├── 5. 0005_ALBERT_200_16_1_1_03_25_80.pt
│   └── 6. 0006_ROBERTA_200_16_1_1_03_25_80.pt
├─ 3. Ficheros del 7 al 9
└─ 4. Ficheros del 10 al 11

¿Qué desea seleccionar? [directorio/fichero]: _
    
```

Figura 5.8: AIMentalHealth selección de despliegue de directorio

```

Por favor indique el número de ficheros por directorio (25): 3
Directorios
├─ 1. Ficheros del 1 al 3
├─ 2. Ficheros del 4 al 6
├─ 3. Ficheros del 7 al 9
└─ 4. Ficheros del 10 al 11
    
```

Figura 5.9: AIMentalHealth selección de fichero y carga del modelo

mente directorio o fichero repitiendo el proceso en caso de seleccionarse la opción directorio [imagen 5.8].

En el camino alterno se pide al usuario el número de fichero que desea, mostrando a continuación un mensaje de error si se ofrece una entrada no válida y mostrando la petición de nuevo. Si se selecciona un fichero existente se carga el modelo junto con su tecnología asociada [imagen 5.9].

5.4.4. Historia de usuario 005

«Como usuario especializado, quiero entrenar un modelo y evaluarlo cuando requiera para poder determinar la factibilidad de un modelo.»

Historia de usuario relacionada con los requisitos: [RF-003](#), [RF-004](#), [RF-005](#), [RF-006](#), [RF-011](#)

Al poder evaluar un modelo existente múltiples veces y no solamente cuando se entrene, la ejecución de estas dos funcionalidades debe de ser independiente. Por tanto, en el menú principal de la aplicación se añaden dos funcionalidades: Entrenar modelo y Evaluar modelo [opciones 2 y 3 del menú principal respectivamente].

Si se selecciona entrenar modelo se pide al usuario la elección de una tecnología



Figura 5.10: AIMentalHealth porcentaje de datos procesados



Figura 5.11: AIMentalHealth entrenamiento de red neuronal

y comienza el entrenamiento del modelo básico, utilizando los datos cargados al iniciar la aplicación destinados a este propósito. Por cada época de entrenamiento se ajustan los pesos de las conexiones entre neuronas y sesgos por cada lote de datos hasta abarcar el conjunto completo.

En cada época se muestra en tiempo real el porcentaje de datos analizados mediante una barra de progreso junto con el valor en tanto por ciento [imagen 5.10]. Además, al finalizar la época se muestra un panel de métricas con los valores obtenidos.

Finalizando la ejecución del entrenamiento se indica al usuario que inserte mediante teclado un nombre para el fichero que contendrá el modelo entrenado. Dicho fichero será almacenado en la ruta de modelos de la aplicación `ruta_aplicacion/mentalapp/models/` con extensión `.pt`² [imagen 5.11].

En el caso de selección de evaluación de modelo se pide la selección de un modelo almacenado en fichero con extensión `.pt` y su tecnología asociada. Posteriormente se evalúa el modelo con los datos cargados con tal propósito y se

²Extensión estándar para ficheros cuyo contenido es la estructura de una red neuronal con los pesos y sesgos.



Figura 5.12: AIMentalHealth evaluación de red neuronal

muestra el panel de métricas resultantes, además del porcentaje de datos analizado en tiempo real [imagen 5.12].

5.4.5. Historia de usuario 007

«Como usuario especializado, quiero configurar los parámetros del modelo para personalizar los entrenamientos y evaluaciones.»

Historia de usuario relacionada con los requisitos: [RF-012](#), [RF-015](#), [RNF-006](#)

Añadiendo un acceso al menú principal se ha implementado un menú secuencial de configuración de los diversos parámetros [opción 4 de la imagen ??]. Al iniciar la ejecución de la aplicación la carga de los parámetros es transparente al usuario para poder utilizar/entrenar/evaluar modelos directamente.

Ejecutando la opción de configuración de parámetros, por cada parámetro configurable se le pedirá al usuario que ingrese un valor, mostrando entre paréntesis el valor actual (correspondiente al predeterminado si aún no se ha modificado). Si el usuario inserta un valor no válido se muestra el mensaje de error y se vuelve a pedir un valor para el parámetro. En el caso de que el usuario inserte un valor válido o vacío, permaneciendo el valor existente en esta última casuística, se

```
Opción: 4
Por favor indique el número máximo de palabras que admita el modelo (200): entrada no valida
Please enter a valid integer number
Por favor indique el número máximo de palabras que admita el modelo (200):
Por favor indique el tamaño de lote (16): 4
Por favor indique el número de épocas (iteraciones) (3): 1
Por favor indique el número de procesos paralelos (1): 3
Por favor indique el tanto por uno de drop out para BERT (0.3):
```

Figura 5.13: AIMentalHealth configuración de parámetros

```
¿Estás seguro de los cambios realizados? [y/n]: entrada no válida
Please enter Y or N
¿Estás seguro de los cambios realizados? [y/n]:
Please enter Y or N
¿Estás seguro de los cambios realizados? [y/n]: N
Por favor indique el número máximo de palabras que admita el modelo (200):
Por favor indique el tamaño de lote (4):
Por favor indique el número de épocas (iteraciones) (1):
Por favor indique el número de procesos paralelos (3):
Por favor indique el tanto por uno de drop out para BERT (0.3):
¿Estás seguro de los cambios realizados? [y/n]: y
```

Figura 5.14: AIMentalHealth confirmación de los cambios en la configuración

repite el proceso con el siguiente parámetro [imagen 5.13].

Al definir el valor de todos los parámetros se muestra un mensaje indicando si se quiere perpetuar los cambios realizados en la configuración. Si la respuesta es afirmativa se regresa al menú principal y en caso contrario, se reinicia la configuración de parámetros [imagen 5.14].

5.4.6. Historia de usuario 008

«Como usuario especializado, quiero saber los resultados de las métricas estándar en AI para estudiar un modelo.»

Historia de usuario relacionada con los requisitos: [RF-007](#), [RNF-005](#)

Al efectuar un entrenamiento o evaluación de un modelo de red neuronal, por cada etapa (una en el caso de evaluación) se muestra por pantalla un panel con las métricas estándar y los valores medidos durante la ejecución del entrenamiento.

Siendo un proceso transparente para el usuario hasta la finalización de la etapa, la medición se realiza con la cuenta de los distintos casos de la matriz de confusión. Estos casos son:

- **True Negative (TN):** Casos negativos clasificados como negativos.
- **False Positive (FP):** Casos negativos clasificados como positivos.
- **False Negative (FN):** Casos positivos clasificados como negativos.

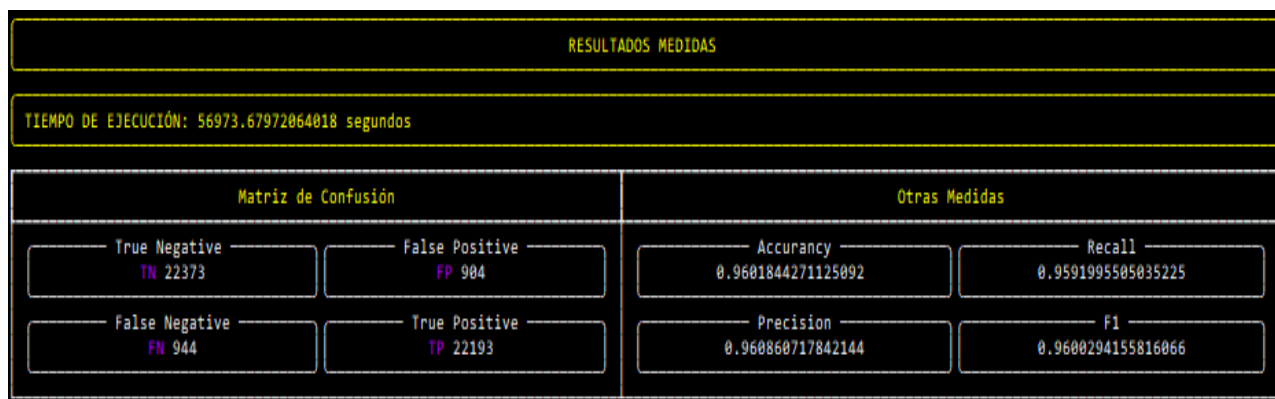


Figura 5.15: AIMentalHealth panel de métricas

- **True Positive (TP)**: Casos positivos clasificados como positivos.

Finalizando el entrenamiento/evaluación, con los resultados de las métricas anteriores se calculan:

- **Accuracy**³: $(TP + TN) / (TP + FP + FN + VN)$
- **Recall**⁴: $TP / (TP + FN)$
- **Precision**⁵: $TP / (TP + FP)$
- **F1**: $(2 * Precision * Recall) / (Precision + Recall)$

Por último se imprime por consola el panel de métricas estructurado cuya disposición consiste en una cabecera con el tiempo de ejecución y una tabla dividida en dos partes: lado izquierdo dedicado a la matriz de confusión y lado derecho dedicado a las métricas restantes. Cada métrica posee un cuadro gráfico con título correspondiente al nombre de la medida y su valor en el interior. La estructura en su conjunto es apreciable en la imagen 5.15.

5.4.7. Historia de usuario 009

«Como usuario especializado, quiero que se cargue un dataset predeterminado para poder entrenar un modelo.»

Historia de usuario relacionada con los requisitos: [RF-008](#), [RF-010](#), [RNF-004](#)

³Exactitud

⁴Sensibilidad

⁵Precisión

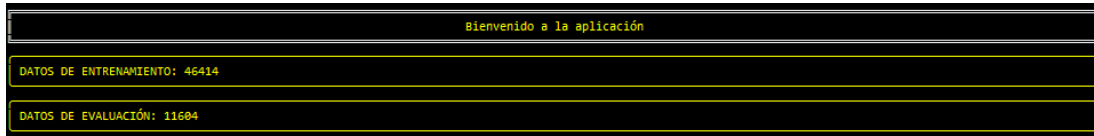


Figura 5.16: AIMentalHealth carga inicial del dataset predeterminado

Al inicio de la aplicación se carga el dataset predeterminado de la aplicación disponible en la siguiente ruta: `ruta_aplicacion/mentalapp/mod_dataset/files/`. La carga de datos consiste en la lectura de un fichero .csv y por cada línea, aplicando paradigma funcional, se crea un objeto mensaje el cuál consiste en un texto en inglés para clasificar y una clasificación (non-suicide o suicide). Una estructura de datos cuyo contenido son todos los objetos creados constituye el dataset.

Posteriormente el dataset se divide entre datos de entrenamiento y evaluación siguiendo la proporción 80 % y 20 % respectivamente.

Todo el proceso definido es transparente para el usuario y solamente es mostrado por pantalla el número de datos de entrenamiento y evaluación una vez cargados, y en dos cuadros de texto independientes y secuenciales [imagen5.16].

5.4.8. Historia de usuario 011

«Como usuario especializado, quiero que los datos se dividan en entrenamiento y evaluación para poder estudiar un modelo.»

Historia de usuario relacionada con los requisitos: **RF-010**

Al cargar un dataset para un evento relacionado con la inteligencia artificial y concretamente con el área de las redes neuronales es imperativo la división de los datos para el correcto funcionamiento de la red resultante. Esto se debe a que se necesita un grupo de datos independientes de control para evaluar un modelo una vez haya sido entrenado con el resto de datos disponibles, además de que la cantidad de datos de entrenamiento sea superior a los datos de evaluación y ambos grupos estén equilibrados, es decir, que posean datos para todos los casos posibles. En este caso en específico consiste en que ambos grupos posean casos de Non-Suicide y Suicide.

En la división del dataset se destina un 80 % de los datos a entrenamiento y un 20 % a evaluación independientemente del número de casos disponibles. Dicha división se realiza de forma aleatoria y equilibrada gracias a la librería Sklearn y concretamente se utiliza la función `train_test_split`.

5.5. Pruebas configuraciones de red neuronal

Las pruebas realizadas en la aplicación han consistido en experimentar con las posibilidades que ofrece respecto al entrenamiento de modelos.

Utilizando la aplicación simulando ser un usuario experimentado en la informática y en el desarrollo de redes neuronales se han entrenado múltiples modelos diferenciando la tecnología y algún parámetro de la configuración. De esta forma se han conseguido múltiples modelos disponibles para su uso en la aplicación y se han evaluado para comprobar cuál es el que ofrece el mejor ratio entre tiempo de entrenamiento y resultado de métricas en la evaluación.

Los modelos almacenados siguen la siguiente lógica de nomenclatura:

AAAA_BBBB_CCCC_DDDD_EEEE_FFFF_GGGG_HHHH_IIII

Correspondiendo cada sección:

- **AAAA** → Código numérico del modelo.
- **BBBB** → Tecnología utilizada.
- **CCCC** → Longitud en palabras del input.
- **DDDD** → Tamaño de lote.
- **EEEE** → Número de épocas.
- **FFFF** → Número de procesos paralelos.
- **GGGG** → Drop out.
- **HHHH** → Porcentaje de datos sobre el total del dataset.
- **IIII** → Porcentaje de datos destinados a entrenamiento.

Los resultados de las pruebas se pueden observar en el [apéndice B](#) disponible en este documento. Estos resultados reflejan la aplicación de paradigma concurrente en el entrenamiento de redes neuronales y cómo afectan al tiempo de entrenamiento además de mostrar qué tecnología es mejor para la problemática a la que se destina la aplicación.

Resumiendo el estudio disponible en el apéndice, la elección de la tecnología afecta considerablemente al tiempo de entrenamiento siendo ROBERTA la más costosa en esta medida. Por otra parte, el número de hilos no demuestra una diferencia considerable en el tiempo de entrenamiento ni efecto alguno en el resto de métricas.

6

Conclusiones y trabajos futuros

El objetivo de esta sección es reflejar las conclusiones extraídas del trabajo realizado y los posibles caminos que se pueden seguir para su desarrollo en siguientes etapas.

6.1. Conclusiones

AIMentalHealth cumple la función de herramienta de apoyo en investigación de IA y en el análisis psicológico de un paciente. Dependiendo del objetivo perseguido el usuario utilizará ciertas funciones disponibles. Poniendo un ejemplo, un estudiante de IA utilizará en gran medida la opción de configuración de parámetros para personalizar sus entrenamientos.

Sin embargo, esta aplicación no oculta funcionalidades según el tipo de usuario que la utilice por lo que beneficiará en gran medida a personas que estén interesadas en ambos objetivos.

El estudio y desarrollo de inteligencias artificiales está en auge en los últimos años destacando el pico de interés provocado por la empresa OpenAI y la publicación de su herramienta chatGPT al público general, acercando así a nuevos estudiantes (especializados o no) al área del machine learning. Al poder utilizar esta herramienta de forma gratuita la empresa ha provocado que un área de estudio muy concreta haya conseguido llegar al público general independientemente de la profesión o cargo.

AIMentalHealth facilitará el estudio a nuevos estudiantes de inteligencia artificial debido a su fácil uso y a la baja necesidad de recursos. Debido a estas facilidades cabe la posibilidad de que más personas se interesen por este estudio además de permitir a los ya interesados avanzar más rápido en sus estudios. De esta manera se incrementará el número de personas formadas pudiendo abastecer la demanda de especialistas de AI que existe en el mercado actual.

Por otra parte, destacando su importancia desde la reciente pandemia del COVID-19, la inteligencia emocional está presente en la sociedad actual. Problemas que hace unos años se clasificaban como nervios o estrés se diagnostican hoy en día con nombres propios y se personaliza el tratamiento a cada paciente. Desgraciadamente, la saturación del sistema sanitario relacionado con el tratamiento psicológico está muy saturado y muchos de los pacientes no pueden permitirse sesiones privadas aún cuando las necesitan urgentemente. El conjunto de estos factores desemboca en citas para diagnóstico muy tardías y en el deterioro del estado de los pacientes sin tratamiento. Cada año más de estos casos terminan en suicidio.

La herramienta desarrollada en este trabajo permitirá a los especialistas recibir un informe con el análisis de los textos escritos por un paciente ya hecho. Este proceso agiliza el análisis secuencial que tiene que realizar un profesional de la salud por cada paciente pudiendo delegar esta tarea a un personal menos especializado que cree el informe. Se logra que una parte del diagnóstico del paciente se realice de manera más rápida permitiendo descolapsar el sistema sanitario a medio y largo plazo, pudiendo abarcar a más personas en un futuro.

6.2. Trabajos futuros

La aplicación desarrollada en este trabajo es una sencilla interfaz de texto desde el punto de vista del usuario. El primer paso del desarrollo ha sido centrado en diseñar una aplicación útil para el ser humano implementado su lógica principal y funcionalidades auxiliares.

Los próximos pasos en el desarrollo de la aplicación deben de estar enfocados en la accesibilidad del producto. Mediante diseñadores y especialistas en áreas como la interacción persona-ordenador se puede diseñar una aplicación de escritorio con las mismas funcionalidades actuales o incluso una app para dispositivos móviles. Además, publicando el producto como página web siguiendo la vía de chatGPT se puede abrir el uso de la aplicación al gran público debido a la ausencia de complicaciones producidas por una instalación.

Por otro lado, si el objetivo futuro seguido es apoyar preferentemente a la comunidad de desarrolladores AI se puede desarrollar widgets con usos específicos y publicarlos en las principales plataformas de distribución de conocimientos

referentes a la inteligencia artificial. Poniendo como ejemplo la plataforma que ha ofrecido los recursos necesarios para el desarrollo de este proyecto, Hugging Face permite la publicación de este tipo de producto software bajo el nombre de *Spaces*.

Un camino paralelo es destinar los futuros esfuerzos de los desarrolladores en hacer pruebas reales con los modelos entrenados junto con especialistas interesados en el proyecto. En el estado actual de la aplicación se puede ejecutar este trabajo sin complicaciones excepto las posibles producidas por la instalación en un inicio. El apoyo de los especialistas en sanidad y el estudio del desempeño del aplicativo en la actividad diaria permitirá refinar diversos aspectos en la estructura o en el flujo de uso.

Por último, los ingenieros del software con interés en AIMentalHealth pueden realizar sprints más pequeños que implementando una interfaz gráfica para la aplicación. El desarrollo se puede enfocar en la ampliación de tecnologías de modelos para ofrecer su uso a los usuarios. Como añadido se pueden realizar nuevas pruebas variando los parámetros de configuración y los modelos para observar qué combinación se adapta mejor a la problemática tratada.

Bibliografía

- [1] Observatorio del Suicidio en España. (2020). Prevención del suicidio. Recuperado de <https://www.fsme.es/observatorio-del-suicidio-2020/>
- [2] Observatorio del Suicidio en España 2022 (datos definitivos diciembre 2023). (s. f.). Prevención del suicidio. <https://www.fsme.es/observatorio-del-suicidio-2022-definitivo/>
- [3] Madrid Salud. (2022, 8 noviembre). Datos de Madrid - Prevención del suicidio. Prevención del suicidio. <https://prevenciondelsuicidio.es/datos-de-madrid/>
- [4] Junta de Gobierno de la Ciudad de Madrid. (2022). Plan de actuación de prevención del suicidio del ayuntamiento de Madrid 2023-2024.
- [5] IEEE. (1998). IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998). Institute of Electrical and Electronics Engineers.
- [6] GitHub: Let's build from here. (s. f.). GitHub. <https://github.com/>
- [7] Manage your team's projects from anywhere — Trello. (s. f.). <https://trello.com/>
- [8] Manifiesto Ágil. (2001). Manifiesto para el Desarrollo Ágil de Software. Recuperado de <http://agilemanifesto.org/>
- [9] Home. (s. f.). Scrum.org. <https://www.scrum.org/>
- [10] Visual Studio Code - Code editing. Redefined. (2021, 3 noviembre). <https://code.visualstudio.com/>
- [11] Inteligencia Artificial - Un Enfoque Moderno (2^a ed.). Russell, S.; Norvig, P. (2003): Prentice Hall Hispanoamericana
- [12] Inteligencia Artificial - Una Nueva Síntesis. Nilsson, N. (2001): McGraw-Hil
- [13] Shneiderman, B., & Plaisant, C. (s. f.). Diseño de interfaces de usuario: estrategias para una interacción persona-computadora efectiva.

-
- [14] TIOBE Index - TIOBE. (2022, 3 junio). TIOBE. <https://www.tiobe.com/tiobe-index/>
- [15] Lutz, M. (2018). Learning Python. O'reilly.
- [16] Hugging Face – The AI community building the future. (n.d.). Huggingface.co. <https://huggingface.co>
- [17] API reference — pandas 1.1.4 documentation. (n.d.). Pandas.pydata.org. <https://pandas.pydata.org/docs/reference/index.html>
- [18] NumPy Reference — NumPy v1.23 Manual. (n.d.). Numpy.org. <https://numpy.org/doc/stable/reference/index.html#reference>
- [19] scikit-learn. (2019). scikit-learn: machine learning in Python. Scikit-Learn.org. <https://scikit-learn.org/stable/>
- [20] Introduction — Rich 12.6.0 documentation. (n.d.). Rich.readthedocs.io. <https://rich.readthedocs.io/en/stable/introduction.html>
- [21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. arXiv (Cornell University), 30, 5998-6008. <https://arxiv.org/pdf/1706.03762v5>
- [22] Git. (s. f.). <https://git-scm.com/>
- [23] Atlassian. (s. f.). Flujo de trabajo de GitFlow — Atlassian Git Tutorial. <https://www.atlassian.com/es/git/tutorials/comparing-workflows/gitflow-workflow>
- [24] Principios del manifiesto ágil. (s. f.). <https://agilemanifesto.org/iso/es/principles.html>

Apéndices



Estudio modelos neuronales

Documento en el que se reflejan las métricas obtenidas en las pruebas realizadas en la aplicación AIMentalHealth.

A.1. Introducción

El propósito del presente documento es persistir los resultados obtenidos en las investigaciones oficiales realizadas mediante la aplicación AIMentalHealth.

Teniendo como objetivo conseguir la mejor configuración de modelo para el problema de clasificación presentado por la aplicación se han realizado diversos estudios creando modelos con variaciones en sus parámetros de creación, entrenamiento y evaluación.

Las pruebas se han realizado en una máquina Linux con las siguientes características:

- **GPU:** Información extraída con el mandato `lspci | grep -i vga`. Ver Imagen [A.1](#)
- **RAM:** Información extraída con el mandato `free -h`. Ver Imagen [A.2](#)

```
00:02.0 VGA compatible controller: Intel Corporation HD Graphics 6000 (rev 09)
```

Figura A.1: Información de GPU

	total	used	free	shared	buff/cache	available
Mem:	7,6Gi	218Mi	7,0Gi	1,0Mi	408Mi	7,2Gi
Swap:	4,0Gi	0B	4,0Gi			

Figura A.2: Información de memoria RAM

```
Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Address sizes: 30 bits physical, 48 bits virtual
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Vendor ID: GenuineIntel
Model name: Intel(R) Core(TM) i5-5250U CPU @ 1.60GHz
CPU family: 6
Model: 61
Thread(s) per core: 2
Core(s) per socket: 2
Socket(s): 1
Stepping: 4
CPU(s) scaling MHz: 43%
CPU max MHz: 2700,0000
CPU min MHz: 500,0000
BogoMIPS: 3192.45
Flags: fpu vme de pse tsc mtr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe syscall nx pdpe1gb rdtscp lm constant_tsc arch_
performance pbs bits rep_good nopl stopleg nonstop_tsc cpuid aperfmperf pni pclmulqdq dtes64 monitor ds_cpl vmx est ts2 sse3 sdbg fma cx16 xtpr pdcm pcid sse4_1 sse4_2
xapic movbe popcnt tsc_deadline_timer aes xsave avx f16c rdrand lahf_lm abm 3dnowprefetch cpuid_fault epb invpcid_single pti ssbd ibrs stibp tpr_shadow vmmi flex
priority ept vpid ept_ad fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid rdseed adx smap intel_pt xsaveopt dtherm ida arat pln pts md_clear flush_lid
Virtualization features:
Virtualization: VT-x
Caches (sum of all):
L1d: 64 KiB (2 instances)
L1i: 64 KiB (2 instances)
L2: 512 KiB (2 instances)
L3: 3 MiB (1 instance)
NUMA:
NUMA node(s): 1
NUMA node0 CPU(s): 0-3
Vulnerabilities:
Itlb multihit: KVM: Mitigation: VMX disabled
L1tf: Mitigation: PTE Inversion; VMX conditional cache flushes, SMT vulnerable
Mds: Mitigation: Clear CPU buffers; SMT vulnerable
Meltdown: Mitigation: PTI
Mmio stale data: Unknown: No mitigations
Retbleed: Not affected
Spec store bypass: Mitigation: Speculative Store Bypass disabled via prctl
Spectre v1: Mitigation: usercopy/swapgs barriers and __user pointer sanitization
Spectre v2: Mitigation: Retpolines, IBPB conditional, IBRS_FW, STIBP conditional, RSB filling, PBR5B-eIBRS Not affected
Srbds: Mitigation: Microcode
Tsx async abort: Not affected
```

Figura A.3: Información de memoria CPU

- CPU: Información extraída con el mandato `lscpu`. Ver Imagen [A.3](#)

A.2. Modelos

Modelos existentes con su configuración básica identificados con un código único. Ver Tabla [A.1](#)

A.3. Entrenamiento

Entrenamiento de modelo identificado por el código representativo de este. Todos los entrenamientos han sido realizados con los mismos datos y se muestran los valores de las métricas ofrecidos por la aplicación. Ver Tabla [A.2](#)

A.4. Evaluación

Evaluación de modelo identificado por el código representativo de este. Todos las evaluaciones han sido realizadas con los mismos datos y se muestran los valores de las métricas ofrecidos por la aplicación. Ver Tabla [A.3](#)

Código	Modelo	Porcentaje datos	Tokenizador	Longitud input (Nº palabras)	Tamaño de lote	Épocas	Workers	Drop out
0001	BERT	25	BERT	200	16	1	5	0.3
0002	ALBERT	25	ALBERT	200	16	1	5	0.3
0003	ROBERTA	25	ROBERTA	200	16	1	5	0.3
0004	BERT	25	BERT	200	16	1	1	0.3
0005	ALBERT	25	ALBERT	200	16	1	1	0.3
0006	ROBERTA	25	ROBERTA	200	16	1	1	0.3
0007	BERT	25	BERT	200	16	1	3	0.3
0008	ALBERT	25	ALBERT	200	16	1	3	0.3
0009	ROBERTA	25	ROBERTA	200	16	1	3	0.3

Tabla A.1: Configuraciones modelos de inteligencia artificial

Código	Porcentaje datos	Número datos	Tiempo (segundos)	Matriz de confusión							
				True Negative	False Positive	False Negative	True Positive	Exactitud	Sensibilidad	Precisión	F1
0001	80	46414	55288,00633	22373	904	944	22193	0,96018	0,95920	0,96086	0,96003
0002	80	46414	46716,91757	22761	516	635	22502	0,97520	0,97255	0,97758	0,97506
0003	80	46414	81841,53777	22796	481	575	22562	0,97725	0,97515	0,97913	0,97713
0004	80	46414	51279,79382	22373	904	944	22193	0,96018	0,95920	0,96086	0,96003
0005	80	46414	46136,06444	22761	516	635	22502	0,97520	0,97255	0,97738	0,97506
0006	80	46414	85017,14165	22796	481	575	22562	0,97725	0,97515	0,97913	0,97713
0007	80	46414	55154,40477	22373	904	944	22193	0,96018	0,95920	0,96086	0,96003
0008	80	46414	46183,44849	22761	516	635	22502	0,97520	0,97255	0,97758	0,97506
0009	80	46414	95501,43697	22796	481	575	22562	0,97725	0,97515	0,97913	0,97713

Tabla A.2: Entrenamiento de modelos

Código	Porcentaje datos	Número datos	Tiempo (segundos)	Matriz de confusión								F1
				True Negative	False Positive	False Negative	True Positive	Exactitud	Sensibilidad	Precisión		
0001	20	11604	3550,59255	5653	149	160	5642	0,97337	0,97242	0,97427	0,97335	
0002	20	11604	4066,12965	5735	67	118	5684	0,98406	0,97966	0,98835	0,98399	
0003	20	11604	3538,71513	5764	38	75	5726	0,99026	0,98707	0,99341	0,99023	
0004	20	11604	3504,78262	5653	149	160	5642	0,97337	0,97242	0,97427	0,97335	
0005	20	11604	4033,52740	5735	67	118	5684	0,98406	0,97966	0,98835	0,98399	
0006	20	11604	3527,45223	5764	38	75	5726	0,99026	0,98707	0,99341	0,99023	
0007	20	11604	3536,91044	5653	149	160	5642	0,97337	0,97242	0,97427	0,97335	
0008	20	11604	4022,00663	5735	67	118	5684	0,98406	0,97966	0,98835	0,98399	
0009	20	11604	3539,71039	5764	38	75	5726	0,99026	0,98707	0,99341	0,99023	

Tabla A.3: Evaluación de modelos

