UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA INFORMÁTICA EN
TECNOLOGÍAS DE LA INFORMACIÓN

AnalIzA. Aplicación inteligente para procesamiento de datos, creación de modelos IA y generación de datos sintéticos

TRABAJO FIN DE GRADO

Febrero - 2026

AUTOR: Aina Caselles Salvà
DIRECTOR: Jesús Javier Rodríguez Sala

RESUMEN

Este trabajo presenta el desarrollo de una aplicación interactiva orientada a facilitar el análisis, preparación y modelado de datos de manera accesible e interactiva, sin necesidad de conocimientos avanzados en programación. La herramienta está concebida como una solución integral que permite a cualquier usuario cargar un conjunto de datos y recorrer, de forma guiada, las distintas etapas fundamentales de un proyecto de ciencia de datos.

En primer lugar, la aplicación incorpora un módulo de análisis exploratorio de datos (EDA) que permite visualizar estadísticas descriptivas, distribuciones, relaciones entre variables y posibles anomalías. A través de una interfaz intuitiva, el usuario puede explorar las características del conjunto de datos y comprender su estructura sin necesidad de escribir código. Además, el sistema incorpora un mecanismo de asistencia basado en inteligencia artificial que analiza el dataset y sugiere mejoras relacionadas con la limpieza, transformación o preparación de los datos. De este modo, el usuario no solo observa la información, sino que recibe orientación activa para optimizarla antes de su utilización en modelos predictivos.

La herramienta también incluye un módulo de entrenamiento de modelos básicos de aprendizaje automático, orientado principalmente a tareas de clasificación y regresión. Tras cargar el dataset, el usuario puede seleccionar la variable objetivo, y el sistema ofrece recomendaciones sobre el tipo de modelo más adecuado en función de la naturaleza del problema. Aunque el usuario mantiene el control en la elección final, la plataforma proporciona apoyo en la toma de decisiones, reduciendo la complejidad técnica del proceso. Una vez entrenado el modelo, se presentan métricas de evaluación que permiten analizar su rendimiento y se ofrece la posibilidad de realizar predicciones de prueba, así como descargar el modelo generado.

Adicionalmente, la aplicación incorpora una funcionalidad para la generación de datasets sintéticos, permitiendo definir variables y crear conjuntos de datos personalizados para pruebas, experimentación o validación.

En conjunto, el proyecto propone una herramienta que simplifica y democratiza el acceso a procesos fundamentales del análisis y modelado de datos. Su principal aportación reside en integrar, en una única plataforma y bajo un enfoque guiado e interactivo, tareas que tradicionalmente requieren conocimientos técnicos especializados, acercando así el aprendizaje automático y la preparación de datos a un público más amplio.

AGRADECIMIENTOS

Quiero agradecer a todas las personas que han formado parte de este camino, por haber sido piezas esenciales en esta trayectoria.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Capítulo 1
Introducción

1.1.- AUTOMATIZACIÓN DEL ANÁLISIS DE DATOS Y LA GENERACIÓN ACCESIBLE DE MODELOS DE INTELIGENCIA ARTIFICIAL

En los últimos años, la Inteligencia Artificial (IA) y la Ciencia de Datos han experimentado un crecimiento significativo, impulsado principalmente por el aumento exponencial del volumen de información disponible y la necesidad de extraer valor de los datos de forma rápida y eficiente.

En este contexto, la automatización de procesos analíticos, que incluye tareas como el análisis exploratorio de datos, el preprocesamiento, la selección de características y la generación de modelos predictivos, ha adquirido una relevancia creciente, dando lugar a herramientas capaces de reducir la complejidad técnica asociada a estas tareas. Este avance

ha permitido que tanto profesionales especializados como usuarios sin experiencia profunda en programación o modelado puedan acceder a soluciones de IA de manera más sencilla y sistemática.

En este capítulo se describe la evolución de estas tecnologías y su impacto en la democratización del análisis de datos, ofreciendo un marco conceptual que justifica la creación de herramientas automatizadas orientadas a facilitar el procesamiento, el modelado y la simulación de conjuntos de datos.

1.1.1.- Análisis exploratorio de datos y extracción de conocimiento (EDA)

El análisis exploratorio de datos (Exploratory Data Analysis o EDA) constituye una etapa fundamental dentro del proceso de ciencia de datos ya que permite comprender la estructura, calidad y características principales de un conjunto de datos antes de aplicar técnicas de modelado o aprendizaje automático. Su objetivo principal es obtener una visión inicial de los datos, facilitando la identificación de patrones, relaciones entre variables, valores atípicos, datos faltantes y posibles inconsistencias [1][2].

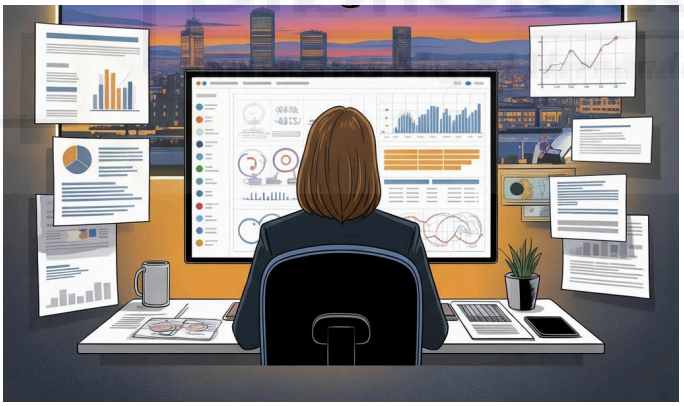


Figura 1.1. Análisis Exploratorio de Datos (generada con Leonardo AI)

El EDA se apoya en un conjunto de técnicas estadísticas descriptivas y métodos de visualización que permiten transformar los datos brutos en información significativa y accionable. Estas técnicas proporcionan al analista una base sólida para formular hipótesis preliminares, evaluar la calidad de los datos y detectar posibles problemas que podrían afectar al rendimiento de modelos posteriores.

Una correcta fase de análisis exploratorio resulta clave para la extracción de conocimiento, ya que influye directamente en decisiones posteriores como la selección de variables relevantes, la aplicación de técnicas de preprocesamiento o la elección del modelo de aprendizaje más adecuado. Sin embargo, este proceso suele requerir conocimientos

técnicos especializados y una intervención manual significativa, lo que puede suponer una barrera para usuarios sin formación en estadística o programación.

En este contexto, la automatización del análisis exploratorio de datos surge como una solución para reducir dicha complejidad, permitiendo que usuarios de distintos perfiles puedan comprender y analizar sus datos de forma guiada, estructurada y accesible, sin necesidad de un conocimiento técnico profundo.

1.1.2.- Generación y entrenamiento de modelos de inteligencia artificial

Una vez comprendidos y analizados los datos mediante el análisis exploratorio, el siguiente paso natural dentro del proceso de ciencia de datos es la construcción de modelos de aprendizaje automático que permitan extraer patrones más complejos y realizar tareas predictivas o descriptivas. Estos modelos constituyen el núcleo de muchas aplicaciones de inteligencia artificial, ya que permiten transformar la información contenida en los datos en predicciones, clasificaciones o agrupaciones que apoyan la toma de decisiones [3].

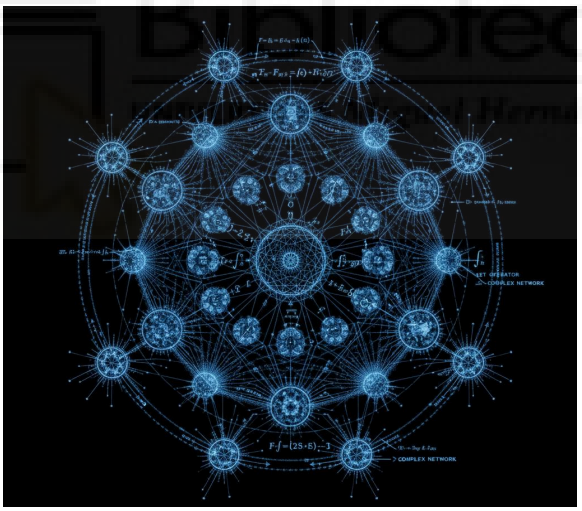


Figura 1.2. Generación de modelos de inteligencia artificial (generada con Leonardo AI)

El entrenamiento de un modelo implica exponerlo a los datos con el objetivo de que aprenda patrones subyacentes que le permitan generalizar a nuevos casos no observados. Para ello, es habitual dividir los datos en distintos subconjuntos, como entrenamiento y validación, que permiten evaluar el rendimiento del modelo y detectar posibles problemas como el sobreajuste o la falta de capacidad de generalización.

En la práctica, la construcción de modelos de inteligencia artificial suele requerir un proceso iterativo en el que se combinan decisiones sobre el tipo de algoritmo, la configuración de sus parámetros y las características de los datos utilizados. Este proceso

se apoya directamente en los resultados obtenidos durante el análisis exploratorio, que condicionan aspectos clave como la selección de variables, el tratamiento de valores atípicos o la necesidad de aplicar técnicas de preprocesamiento adicionales.

1.1.3.- Generación de datos sintéticos

A la hora de entrenar un modelo de IA para dar solución a un determinado problema, es habitual encontrarse con la falta de datos que se ajusten adecuadamente a los requisitos del análisis. Esta limitación puede deberse a la escasez de muestras disponibles, restricciones de acceso, o a la sensibilidad de la información. En este contexto, surge la necesidad de recurrir a la generación de datos sintéticos como una alternativa viable.

Los datos sintéticos son conjuntos de datos generados artificialmente con el objetivo de reproducir determinadas propiedades estadísticas y estructurales de datos reales. Su generación permite crear muestras controladas que preservan distribuciones, relaciones entre variables y patrones relevantes, sin necesidad de utilizar directamente información real, lo que resulta especialmente útil en entornos donde la disponibilidad de datos es limitada [4] [5] [6].

Figura 1.3. Generación de datos sintéticos (generada con Leonardo AI)

Este tipo de datos se emplea habitualmente en el entrenamiento y validación de modelos de inteligencia artificial, así como en la evaluación de algoritmos y la experimentación con distintos escenarios. De este modo, los datos sintéticos facilitan el desarrollo, prueba y comparación de soluciones basadas en datos, permitiendo analizar el comportamiento de los modelos en condiciones controladas.

Existen múltiples enfoques para la generación de datos sintéticos, que abarcan desde métodos estadísticos sencillos, basados en distribuciones teóricas o técnicas de muestreo,

hasta técnicas más avanzadas apoyadas en modelos generativos capaces de capturar dependencias complejas entre variables. La elección del método depende del tipo de datos, del nivel de realismo requerido y de la finalidad del conjunto de datos generado.

La utilización de datos sintéticos se relaciona directamente con las fases previas del proceso de ciencia de datos. El análisis exploratorio permite identificar las características que deben preservarse durante la generación, mientras que los modelos de inteligencia artificial entrenados sobre estos datos ofrecen un marco para evaluar comportamientos y resultados antes de su aplicación sobre datos reales. De este modo, los datos sintéticos se integran como un componente relevante dentro del flujo completo de análisis y modelado.

1.2.- JUSTIFICACIÓN DEL PROYECTO

En los últimos años, con la llegada de las nuevas tecnologías, nuestra capacidad de generar datos ha superado con creces nuestra capacidad para analizarlos. Tanto organizaciones como profesiones individuales de cualquier sector se encuentran a diario con grandes volúmenes de información cuya exploración, limpieza y modelado requieren conocimientos técnicos avanzados y tiempos de procesamiento elevados. Esta situación dificulta la extracción de valores de los datos y limita el uso de la IA en entornos donde podría aportar mejoras significativas.

Ante esta situación, surge la necesidad de desarrollar herramientas que no solo automaticen el análisis, sino que acompañen al usuario explicando cada paso del proceso, ofreciendo resultados comprensibles y permitiendo la interacción de manera intuitiva. Una plataforma capaz de guiar, simplificar y hacer transparente el análisis de datos permite que cualquier persona, independientemente de su formación técnica, pueda centrarse en su trabajo, comprender el significado de sus datos y tomar decisiones sin necesidad de dominar técnicas de programación o modelos de inteligencia artificial. En cada parte, tecnologías como el Big Data, combinadas con herramientas y técnicas de inteligencia artificial (IA), resultan fundamentales para transformar estos datos en conocimiento útil que ayuda a tomar decisiones más acertadas y oportunas.

Este proyecto surge de la motivación personal por desarrollar una herramienta que permita mejorar la eficiencia en cualquier sector que genere datos y busca aprovechar los conocimientos adquiridos tanto en la carrera como en un máster privado que he realizado. En particular, la aplicación ha desarrollar buscará integrar técnicas de IA en el ámbito sanitario para mejorar el análisis de datos clínicos, acelerar el diagnóstico y optimizar la gestión de los recursos médicos, contribuyendo así a una atención más rápida, segura y efectiva.

1.3.- OBJETIVOS

El objetivo principal de este proyecto es desarrollar una aplicación que permita a profesionales dentro de cualquier ámbito explorar, interpretar y aprovechar datos mediante técnicas de inteligencia artificial, sin necesidad de tener conocimientos especializados en IA, ofreciendo una experiencia de uso guiada y accesible. Más concretamente, a continuación se detallan mis objetivos específicos y personales:

- **Objetivos específicos:**

    1. Diseñar una interfaz de usuario intuitiva, centrada en la experiencia del usuario.

    2. Integrar un flujo guiado de uso que acompañe al usuario paso a paso desde la carga y exploración de los datos hasta la obtención e interpretación de los datos.

- **Objetivos personales:**

    1. Adquirir conocimientos y habilidades prácticas en el desarrollo de aplicaciones que integren técnicas de inteligencia artificial.

    2. Aprender a diseñar, consumir e integrar APIs en el desarrollo de software.

    3. Profundizar en el uso de herramientas y lenguajes relevantes para el desarrollo de aplicaciones interactivas basadas en IA, tales como Python, empleados dentro de frameworks de desarrollo web (como FastAPI), y bibliotecas de aprendizaje automático.

    4. Fomentar el pensamiento orientado al usuario, poniendo énfasis en la accesibilidad, simplicidad y utilidad de las soluciones tecnológicas para profesionales no técnicos.

Capítulo 2 Antecedentes y estado de la cuestión

En las últimas décadas, los avances en Inteligencia Artificial y ciencia de datos han transformado de forma significativa la manera en que se analizan, interpretan y explotan grandes volúmenes de información. Este crecimiento se sustenta en tres elementos principales: el aumento exponencial de datos disponibles, el desarrollo de algoritmos y técnicas de aprendizaje automático y el incremento de la capacidad computacional para procesar información compleja en tiempos razonables. Esta convergencia ha impulsado la adopción de enfoques automatizados para la extracción de conocimiento, la generación de modelos predictivos y la creación de conjuntos de datos generados artificialmente, respondiendo a desafíos como la escasez de datos etiquetados y la necesidad de acelerar procesos analíticos complejos [7].

El presente capítulo tiene como objetivo revisar los principales antecedentes y el estado actual de las tecnologías relacionadas con el análisis automatizado de datos, la generación de modelos de inteligencia artificial y la producción de datos sintéticos.

2.1.- SITUACIÓN ACTUAL DE LA INTELIGENCIA ARTIFICIAL

En la actualidad, la inteligencia artificial se articula principalmente en torno al aprendizaje automático (Machine Learning) y, en particular, al aprendizaje profundo (Deep Learning), técnicas que permiten a los sistemas aprender patrones y relaciones directamente a partir de los datos. Estos enfoques han demostrado un alto rendimiento en tareas como clasificación, regresión, detección de anomalías y agrupamiento, consolidándose como herramientas habituales dentro del proceso de análisis de datos y modelado predictivo en múltiples dominios [8][9].

El desarrollo reciente de la IA ha puesto de manifiesto que el rendimiento de los modelos no depende únicamente del algoritmo empleado, sino en gran medida de la calidad y estructura de los datos utilizados durante el entrenamiento. Esta perspectiva ha favorecido un enfoque *data-centric*, en el que la mejora y gestión de los datos se consideran tan relevantes como la optimización de los modelos mismos, impulsando la investigación sobre técnicas de preprocesamiento automatizado, selección de características y evaluación de la calidad de los conjuntos de datos [8][10].

En paralelo, se ha producido una notable evolución hacia la completa automatización de distintas etapas del flujo de trabajo en ciencia de datos. Tareas cruciales como la limpieza y transformación de datos, la selección de modelos apropiados, el ajuste de hiperparámetros o la rigurosa evaluación de resultados han comenzado a integrarse en sistemas capaces de ejecutarlos de forma sistemática, reduciendo la dependencia de intervención manual y favoreciendo la reproducibilidad de los experimentos. Esta importante tendencia ha dado lugar a enfoques conocidos como Automated Machine Learning (AutoML), ampliamente estudiados en literatura reciente como soluciones eficaces para sistematizar y acelerar procesos analíticos complejos [7][11].

Otro elemento clave es el creciente uso de datos sintéticos como complemento o sustituto de datos reales. Estas técnicas permiten generar conjuntos de datos controlados que facilitan la experimentación, el entrenamiento inicial de modelos y la validación de enfoques en escenarios donde los datos reales son insuficientes o difíciles de obtener. Su integración dentro de los flujos de análisis refleja una visión más flexible y modular del proceso de modelado.

En conjunto, la situación actual de la inteligencia artificial se caracteriza por una combinación de modelos cada vez más potentes y una creciente necesidad de herramientas que estructuren, automaticen y faciliten el acceso al análisis y al modelado de datos. Este contexto tecnológico constituye la base sobre la que se desarrollan soluciones orientadas a

simplificar el proceso completo de generación de conocimiento a partir de datos, línea en la que se enmarca el presente trabajo.

2.2.- HERRAMIENTAS DISPONIBLES EN EL MERCADO

El avance de la inteligencia artificial y de la ciencia de datos ha propiciado el desarrollo de múltiples herramientas que facilitan el análisis automatizado de datos, la construcción de modelos de aprendizaje automático y la generación de conjuntos de datos sintéticos. Estas soluciones abarcan desde plataformas de *Automated Machine Learning* (AutoML) hasta bibliotecas de análisis exploratorio y generación de datos sintéticos, permitiendo abordar tareas complejas sin necesidad de intervención manual intensiva.

A continuación, se presentan algunas de las herramientas más representativas actualmente disponibles en el mercado, agrupadas según su función principal: análisis exploratorio de datos, automatización del modelado y generación de datos sintéticos.

2.2.1.- Herramientas de análisis exploratorio de datos automatizado

El análisis exploratorio de datos (Exploratory Data Analysis, EDA) es una fase fundamental dentro de cualquier flujo de trabajo de ciencia de datos, ya que permite comprender la estructura, calidad y patrones presentes en un conjunto de datos antes de aplicar cualquier técnica de modelado. Hoy en día, existen diversas herramientas que automatizan este proceso, facilitando la generación de análisis iniciales de forma rápida, sistemática y reproducible. A continuación, se describen algunas de las soluciones más relevantes:

- **Pandas Profiling / yData Profiling:** Biblioteca de Python ampliamente utilizada para generar reportes automáticos de conjuntos de datos. Proporciona estadísticas descriptivas, análisis de distribuciones, correlaciones, detección de valores atípicos, identificación de datos faltantes y alertas sobre posibles problemas de calidad. Su uso requiere conocimientos básicos de programación y manejo de la librería Pandas [12][13][14].

- **Sweetviz:** Herramienta enfocada a la creación de informes visuales intuitivos y comparativos. Permite contrastar diferentes datasets, como conjuntos de entrenamiento y validación, facilitando la identificación de sesgos, desviaciones en las distribuciones y problemas de representatividad. Destaca por su presentación

gráfica clara y su enfoque visual, aunque su personalización requiere ciertos conocimientos técnicos [15][16].

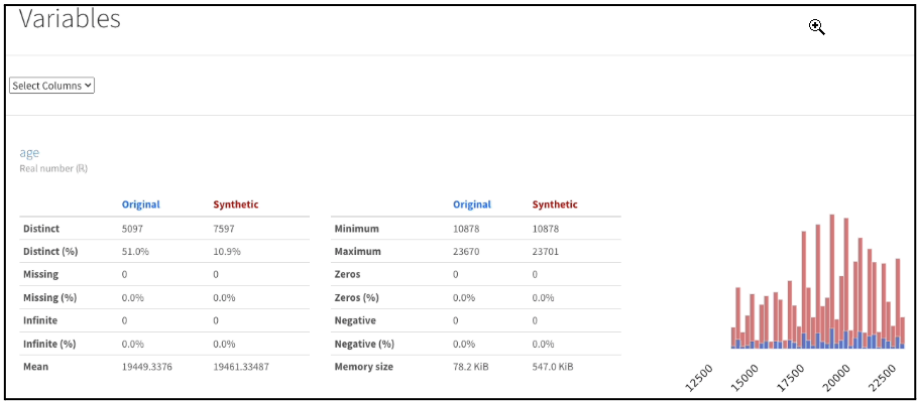


Figura 2.1. Ejemplo de uso de YDataProfiling 4.18 (Fuente: YDataProfiling)

Figura 2.2. Ejemplo de uso de Sweetviz (Fuente: geeksforgeeks)

- **AutoViz:** Solución de código abierto que automatiza la generación de visualizaciones relevantes en función del tipo de variables presentes en el dataset. Su objetivo es acelerar la comprensión global de los datos sin necesidad de configuraciones complejas, generando gráficos y resúmenes estadísticos de manera automática [17].

- **Dtale**: Herramienta interactiva que combina análisis exploratorio automatizado con exploración visual dinámica a través de una interfaz web. Permite profundizar en los datos una vez detectados patrones o anomalías, manteniendo un equilibrio entre automatización y control del usuario, ideal para exploraciones más detalladas [18][19].

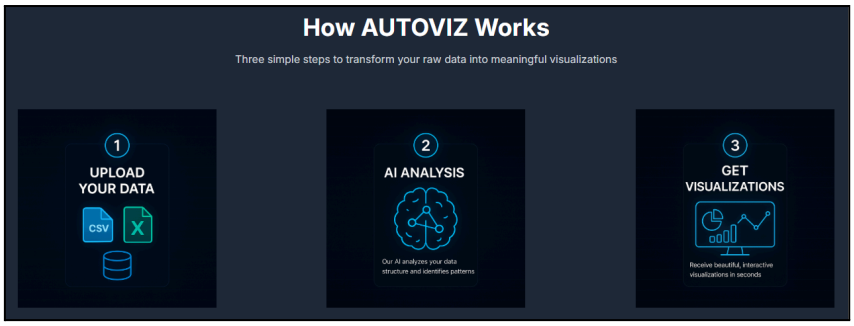


Figura 2.3. Flujo de uso de AutoViz (Fuente: AutoViz)

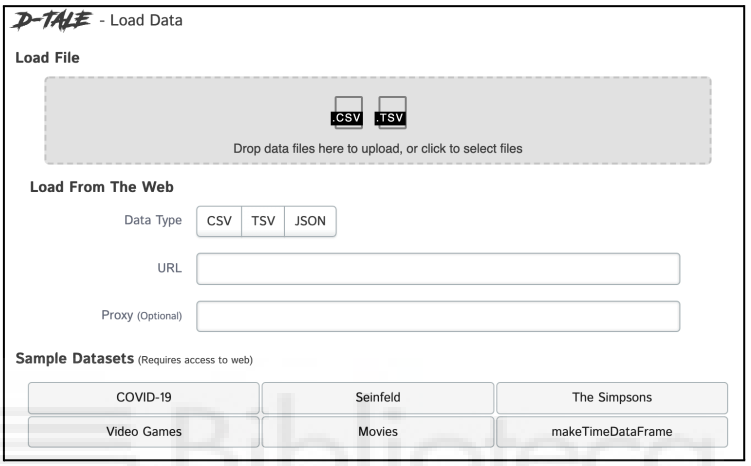


Figura 2.4. Carga de datos D-TALE (Fuente: [19])

- **DataRobot**: Plataforma empresarial orientada a la preparación y comprensión de datos que integra EDA automatizado con procesos guiados de limpieza y transformación. Su enfoque se centra en garantizar la calidad y consistencia de los datos antes de su uso en procesos de modelado, siendo especialmente útil en entornos corporativos donde se requiere reproducibilidad y escalabilidad [20].

Figura 2.5. Ejemplo EDA DataRobot (Fuente: DataRobot)

2.2.2.- Plataformas de AutoML

Las plataformas de Automated Machine Learning (AutoML) permiten automatizar etapas avanzadas del flujo de trabajo analítico, incluyendo la selección de algoritmos, la ingeniería de características y la optimización de hiperparámetros, acelerando la construcción de modelos predictivos de manera sistemática y reproducible. Entre las plataformas más destacadas se encuentran:

- **Google Cloud AutoML**: Proporciona herramientas especializadas para clasificación de imágenes, texto y datos tabulares. Destaca por sus interfaces gráficas accesibles y por su capacidad para procesar grandes volúmenes de datos, lo que la hace adecuada tanto para usuarios con conocimientos limitados en programación como para entornos empresariales de gran escala [21].

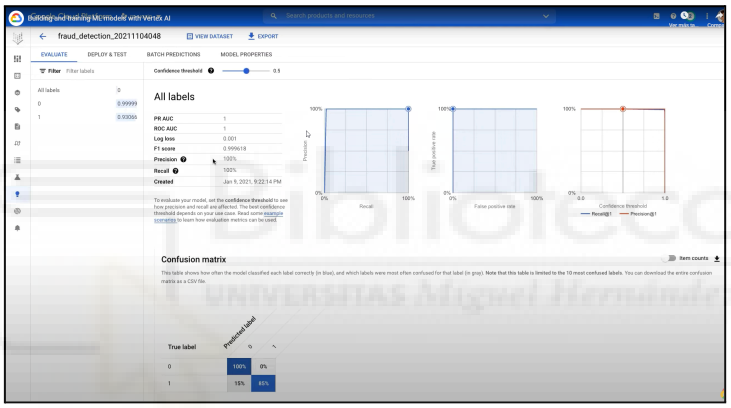


Figura 2.6. Creación y entrenamiento de modelos Vertex AI (Fuente: Google Cloud)

- **H2O.ai Driverless AI**: Plataforma que automatiza el preprocesamiento, la selección de variables y características, y proporciona explicaciones interpretables de los modelos generados. Es especialmente útil en aplicaciones de negocio y salud, ofreciendo una combinación de eficiencia, rendimiento y soporte para técnicas avanzadas como interpretabilidad y scoring de modelos [22].

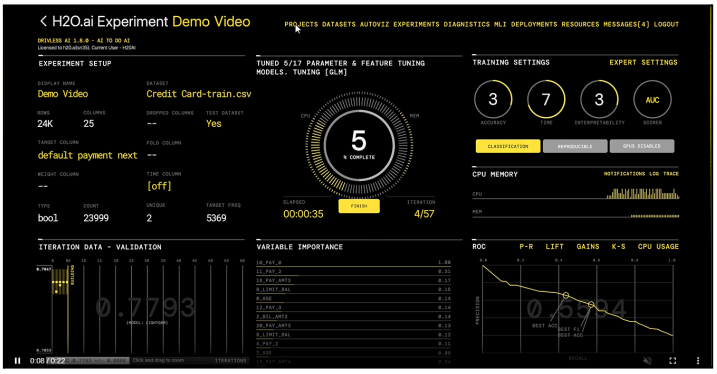


Figura 2.7. Simulación funcionamiento H2O.ai (Fuente: H2O.ai)

- **Microsoft Azure Machine Learning**: Integra pipelines de AutoML con notebooks colaborativos y capacidades de despliegue en entornos corporativos. Su enfoque facilita la integración de modelos en sistemas existentes y permite a equipos multidisciplinarios colaborar en la construcción, validación y despliegue de modelos predictivos [23].

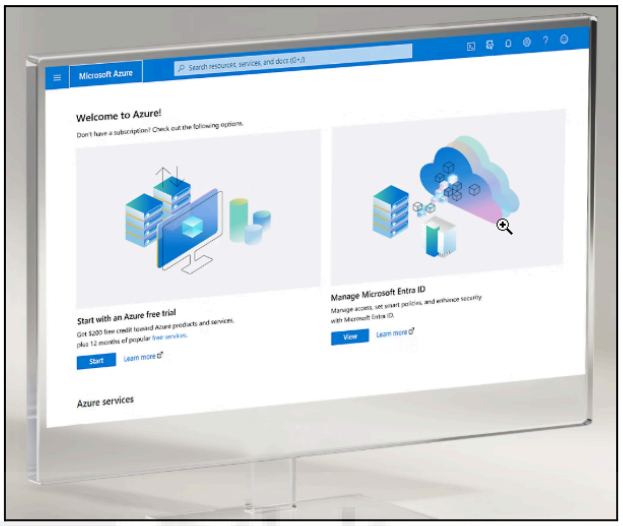


Figura 2.8. Microsoft Azure. (Fuente: Microsoft)

2.2.3.- Herramientas de generación de datos sintéticos

La generación de datos sintéticos se ha consolidado como una estrategia fundamental para entrenar y validar modelos de inteligencia artificial cuando los datos reales son escasos, incompletos o presentan restricciones de privacidad. Estas herramientas permiten crear conjuntos de datos artificiales que preservan relaciones estadísticas, distribuciones y correlaciones relevantes del dataset original, facilitando la experimentación y el desarrollo de modelos de forma segura y reproducible. Entre las soluciones más destacadas se encuentran:

- **Synthpop:** Paquete desarrollado en R para generar datos tabulares sintéticos preservando relaciones estadísticas entre variables. Es especialmente útil en contextos académicos y de investigación, permitiendo simular datasets con estructuras realistas, aunque su uso requiere conocimientos en programación en R y estadística [24].

- **Gretel.ai:** Plataforma SaaS de NVIDIA que permite generar datos sintéticos tabulares y de series temporales, asegurando la preservación de correlaciones y protegiendo la privacidad de los datos originales. Destaca por su enfoque en la

facilidad de uso y la disponibilidad de interfaces gráficas, aunque algunas funciones avanzadas requieren suscripción empresarial [25].

- **CTGAN/ SDV (Synthetic Data Vault)**: Son Bibliotecas de Python basadas en modelos generativos (GANs y VAE) que permiten crear datasets sintéticos a partir de datos tabulares complejos, considerando incluso posibles dependencias no lineales y correlaciones entre los atributos del conjunto de ejemplo. Estas herramientas ofrecen gran flexibilidad y capacidad de modelado de estructuras complejas, pero requieren conocimientos técnicos en Python y aprendizaje automático para su configuración y optimización [26].

2.2.4.- Resumen

El análisis de las herramientas actualmente disponibles nos muestra cómo el ecosistema de la ciencia de datos y la inteligencia artificial ha evolucionado hacia soluciones cada vez más especializadas y potentes, pero también fragmentadas. Existen herramientas consolidadas orientadas al análisis exploratorio de datos, capaces de generar estadísticas descriptivas y visualizaciones de forma automática, así como plataformas de aprendizaje automático que permiten entrenar modelos mediante enfoques AutoML.

Sin embargo, la mayoría de estas soluciones requieren conocimientos técnicos en estadística, programación o aprendizaje automático, y suelen enfocarse en etapas aisladas del flujo de trabajo. Además, priorizan el rendimiento técnico sobre la interpretabilidad, centrándose en la configuración de parámetros o la interpretación de métricas sin ofrecer explicaciones que permitan comprender plenamente el proceso. Esta situación dificulta que usuarios no especializados puedan abordar de forma integrada el análisis, la modelización y la simulación de datos, y resalta la necesidad de herramientas que combinen estas funcionalidades bajo un enfoque accesible y guiado (ver resumen de las características de las herramientas analizadas en la tabla 2.1).

2.3.- VALORACIÓN

Tras el análisis de las herramientas revisadas, se observa que muchas ofrecen soluciones especializadas y de alto rendimiento para tareas concretas del flujo de trabajo en ciencia de datos, como el análisis exploratorio, el entrenamiento de modelos o la generación de datos sintéticos. Algunas incorporan interfaces gráficas e interacción con los datos, facilitando la exploración, pero en general requieren conocimientos técnicos en estadística, programación o aprendizaje automático, y no integran de forma completa todas las fases del proceso.

Ninguna de las herramientas estudiadas proporciona, de manera sencilla y accesible, una guía completa para que usuarios sin conocimientos técnicos puedan realizar un análisis exploratorio de datos (EDA), entrenar modelos y generar datos sintéticos desde una única interfaz integrada. Esta carencia evidencia una oportunidad clara para el desarrollo de una solución que unifique estas funciones y acompañe al usuario paso a paso en el análisis y la interpretación de sus datos.

Tabla 2.1: Tabla comparativa de herramientas

Herramienta	EDA Automatizado	Generación de modelos IA	Datos sintéticos	Accesibilidad para usuarios no técnicos	Principales ventajas	Principales limitaciones
Pandas Profiling (yData)	✔	✖	✖	Baja	Informes EDA completos y automáticos	Requiere programación y conocimientos estadísticos
Sweetviz	✔	✖	✖	Baja	Visualizaciones claras y comparativas	Uso exclusivo mediante código
AutoViz	✔	✖	✖	Baja	Generación rápida de gráficos	Escasa interpretabilidad y control limitado
Dtale	✔	✖	✖	Media	Exploración interactiva de datos	Necesita entorno técnico previo
DataRobot	✔	✔	✖	Media	Plataforma AutoML madura y potente	Coste elevado y enfoque empresarial
Google Cloud AutoML	✖	✔	✖	Media	Escalabilidad y automatización en la nube	Dependencia del ecosistema cloud
H20.ai Driverless AI	✔	✔	✖	Media	Alta automatización del modelado	Plataforma propietaria, curva de aprendizaje
Azure Machine Learning AutoML	✖	✔	✖	Media	Integración con servicios Azure	Requiere conocimientos ML y cloud
Synthpop	✖	✖	✔	Baja	Generación estadística de datos sintéticos	Uso técnico y limitada flexibilidad
Gretel.ai	✖	✖	✔	Media	Plataforma SaaS orientada a datos sintéticos	Menor control interno del proceso
CTGAN/SDV	✖	✖	✔	Baja	Modelos generativos avanzados	Alta complejidad técnica y computacional

Para ilustrar esta brecha, se plantea una valoración práctica mediante dos enfoques complementarios. En primer lugar, se examinará cómo algunas herramientas existentes (por ejemplo, Pandas Profiling, Sweetviz o AutoViz) permiten generar análisis exploratorios automatizados y visualizaciones, señalando sus ventajas y limitaciones en términos de accesibilidad, explicabilidad y cobertura funcional. En segundo lugar, se realizará un ejemplo utilizando modelos de inteligencia artificial generativa, como ChatGPT, para producir un análisis exploratorio guiado a partir de un dataset de ejemplo. Esta prueba permite observar cómo una IA puede generar un flujo de análisis explicativo e interactivo, acercándose al tipo de experiencia que se pretende ofrecer con la herramienta propuesta, y sirve para identificar fortalezas y limitaciones que justifican la necesidad de un sistema propio que integre todas las funcionalidades de manera accesible e intuitiva.

2.3.1.- Análisis exploratorio automatizado con Pandas Profiling

Pandas Profiling es una librería ampliamente utilizada en el ecosistema Python que permite generar informes automáticos de análisis exploratorio de datos a partir de un conjunto de datos estructurado. Su objetivo es ofrecer una visión rápida de la calidad, distribución y relaciones entre variables, mediante estadísticas descriptivas y visualizaciones generadas de forma automática.

Para evaluar sus capacidades, se utilizó un dataset de ejemplo y se generó un informe automático de EDA empleando la configuración por defecto de la herramienta. El informe resultante incluye información sobre tipos de variables, valores faltantes, distribuciones estadísticas, correlaciones y posibles anomalías en los datos.

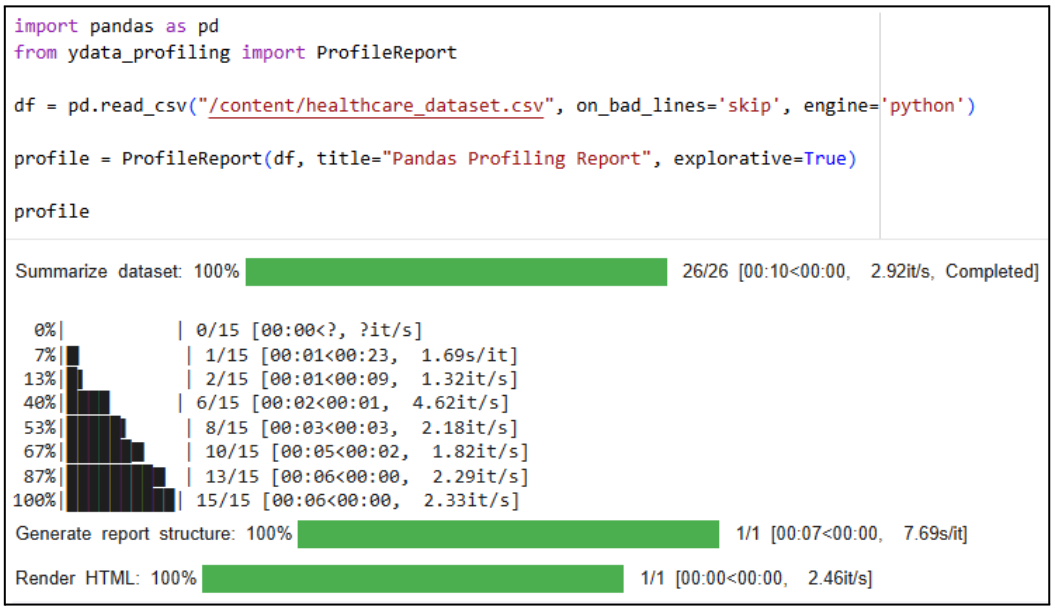


Figura 2.9. Ejecución de ydata_profiling

Los resultados muestran que la herramienta es capaz de identificar de forma rápida patrones relevantes, problemas de calidad de los datos y relaciones entre variables, proporcionando un análisis exhaustivo con un esfuerzo mínimo por parte del usuario una vez configurado el entorno de ejecución.

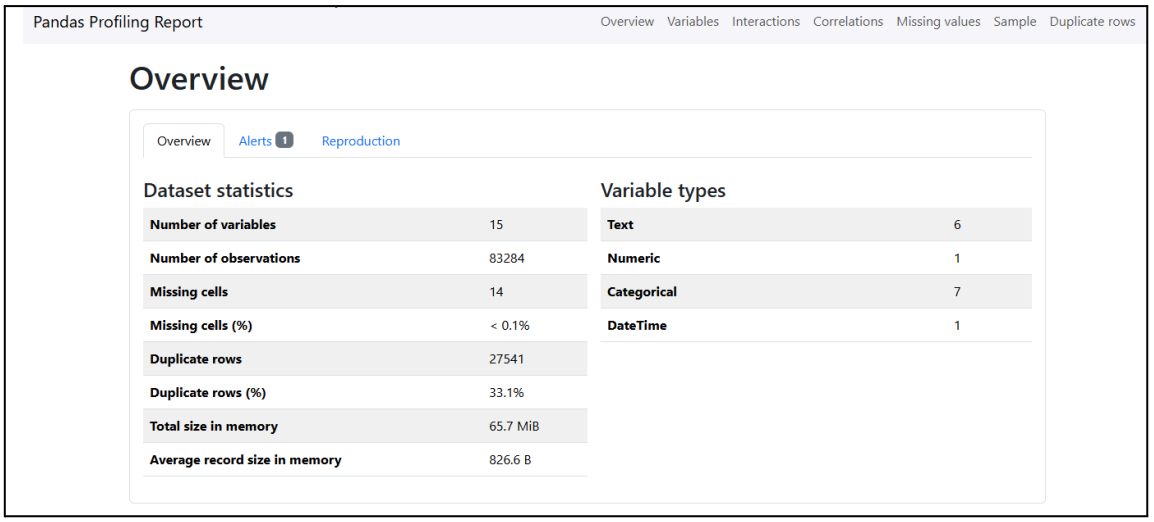


Figura 2.10. Vista previa de los resultados

No obstante, el uso de Pandas Profiling requiere conocimientos previos en programación y en el manejo de entornos de análisis de datos, ya que la generación del informe depende de la carga del dataset mediante código y de la interpretación de métricas estadísticas. Por este motivo, aunque la herramienta resulta muy útil para analistas y científicos de datos, su accesibilidad para usuarios sin formación técnica es limitada.

2.3.2.- Análisis exploratorio guiado mediante IA generativa

Con el objetivo de explorar enfoques alternativos al uso de herramientas tradicionales de análisis exploratorio, se realizó una prueba utilizando un modelo de inteligencia artificial generativa, concretamente ChatGPT Go, para generar un análisis exploratorio de datos a partir del mismo conjunto de datos del apartado anterior. Este enfoque permite evaluar hasta qué punto una IA conversacional puede guiar al usuario en la comprensión de los datos de forma interactiva y explicativa, sin requerir conocimientos técnicos previos ni el uso de herramientas de programación.

En la interacción realizada, se solicitó de forma genérica la realización de un análisis exploratorio de datos, sin especificar de manera explícita la necesidad de generar visualizaciones. No obstante, el modelo inició su respuesta presentando directamente gráficos de distribución y una descripción visual de las variables, lo que pone de manifiesto una tendencia a priorizar elementos gráficos como parte del análisis inicial.

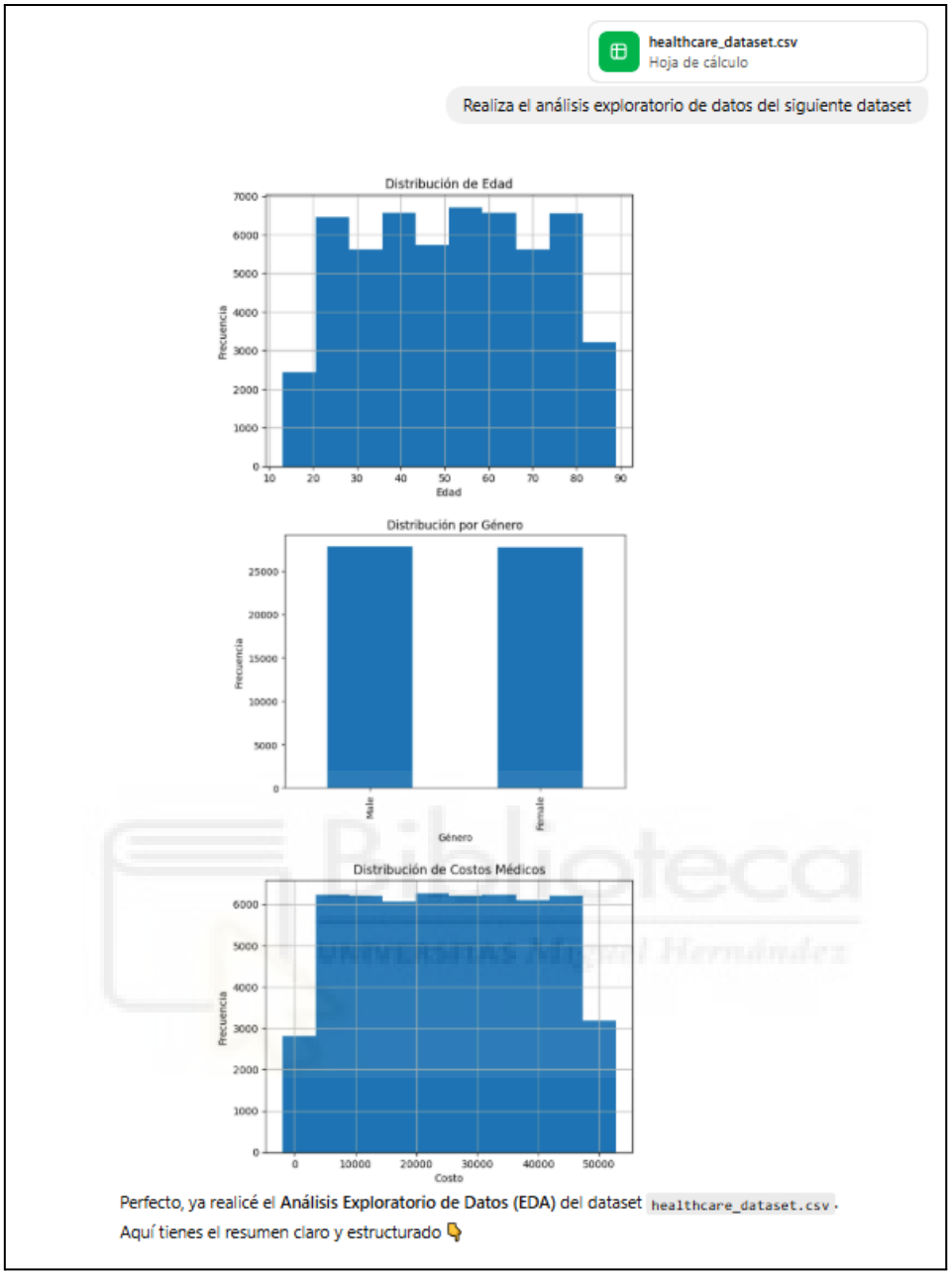


Figura 2.11. Solicitud de EDA a ChatGPT Go y el inicio de su respuesta

A continuación, ofreció una descripción general del conjunto de datos, identificando el número de registros, las variables disponibles, sus tipos (numéricas, categóricas y temporales) y la ausencia de valores nulos.

El análisis incluyó una exploración univariada de variables relevantes como la edad, el género y los costes médicos, destacando aspectos como la forma de las distribuciones, la presencia de asimetrías y la posible existencia de valores atípicos. Asimismo, se abordaron variables categóricas clave, como las condiciones médicas o los resultados de pruebas clínicas, acompañadas de interpretaciones en lenguaje natural que contextualizan su posible uso en tareas de clasificación, segmentación o predicción.

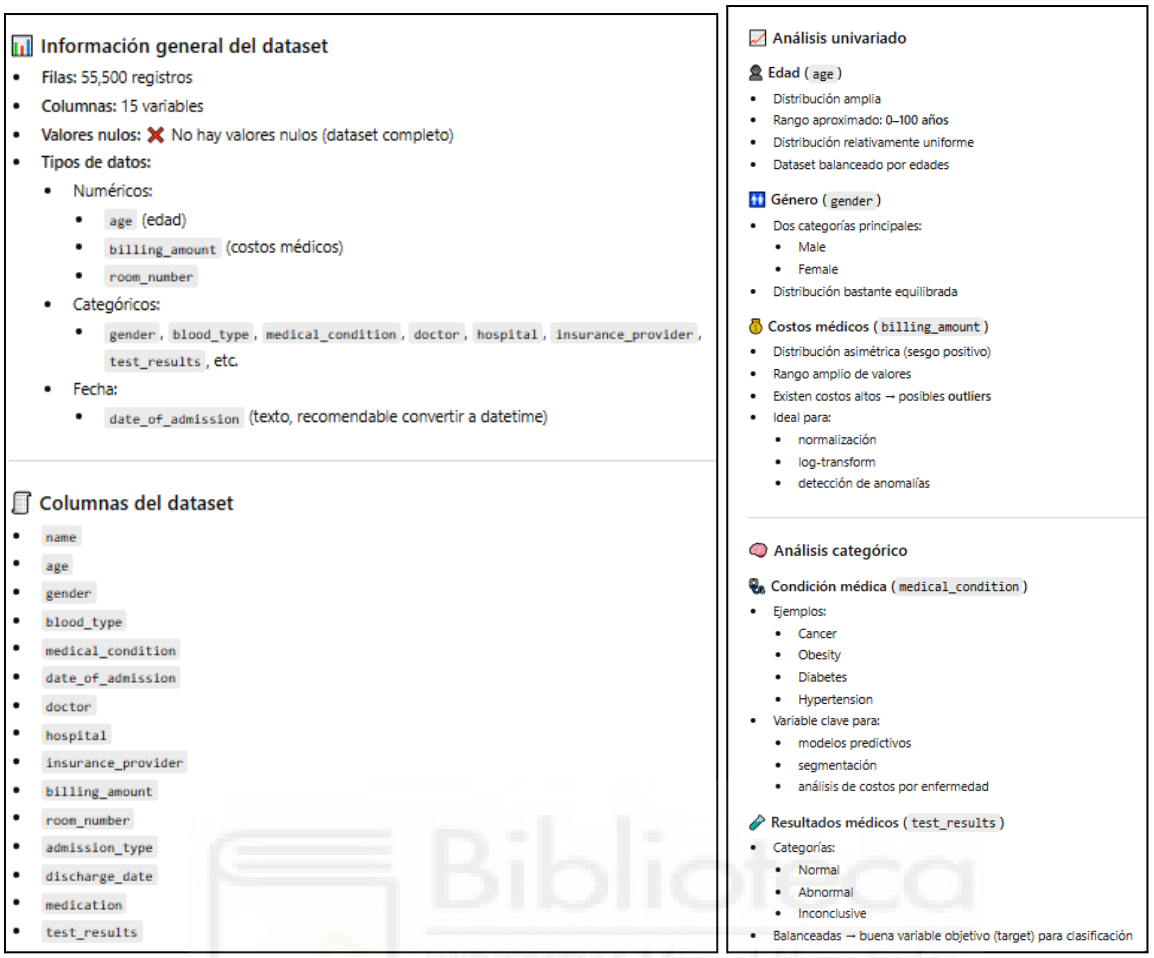


Figura 2.12. Presentación de los resultados EDA ofrecidos por ChatGPT Go

Uno de los principales puntos fuertes observados en esta prueba es la capacidad del modelo generativo para estructurar automáticamente el análisis y presentar los resultados de forma clara y accesible, sin requerir configuración previa ni intervención técnica por parte del usuario. La interacción conversacional permite además profundizar en el análisis mediante preguntas sucesivas, lo que refuerza su utilidad como herramienta de apoyo y orientación.

Sin embargo, esta aproximación presenta limitaciones relevantes. El análisis generado no se ejecuta directamente sobre los datos reales en un entorno controlado por el usuario, lo que impide garantizar la reproducibilidad y exactitud de los resultados. Además, no existe una integración directa con etapas posteriores del flujo de trabajo, como el entrenamiento de modelos predictivos o la generación de datos sintéticos, quedando el análisis limitado a una función descriptiva y orientativa.

En conjunto, esta experiencia pone de manifiesto el potencial de la inteligencia artificial generativa como interfaz accesible para la exploración inicial de datos y la interpretación de resultados, especialmente para usuarios no técnicos. Al mismo tiempo, evidencia la necesidad de una solución integrada que combine la ejecución real del análisis exploratorio con una presentación visual guiada y comprensible, y que permita enlazar de forma natural

con fases posteriores del proceso de modelado y simulación de datos, objetivo central del sistema propuesto en este trabajo.

2.3.3.- Comparativa y conclusiones de la valoración

La evaluación práctica de las herramientas disponibles muestra un patrón claro: las soluciones tradicionales de análisis exploratorio de datos, como Pandas Profiling, ofrecen un alto grado de automatización y profundidad técnica, generando estadísticas y visualizaciones completas de manera eficiente. Sin embargo, su uso requiere conocimientos en programación y manejo de entornos de datos, lo que limita su accesibilidad para usuarios no especializados.

Por otro lado, los modelos de inteligencia artificial generativa, representados en esta prueba por Chat GPT, facilitan un análisis descriptivo guiado y comprensible, con explicaciones en lenguaje natural y capacidad de interacción. Esta aproximación reduce la barrera técnica y permite a los usuarios centrarse en la interpretación de los resultados. No obstante, carece de ejecución directa sobre los datos reales, reproducibilidad y vinculación con etapas posteriores del flujo de trabajo, como el entrenamiento de modelos o la generación de datos sintéticos.

En conjunto, la comparación evidencia que ningún enfoque aislado cubre de manera integral todas las necesidades de un proceso completo de análisis de datos: desde el EDA automatizado hasta la integración de modelos de inteligencia artificial y la generación de datos sintéticos en un entorno accesible y guiado. Este análisis refuerza la motivación del presente trabajo: desarrollar una aplicación que combine estas capacidades, ofreciendo una solución unificada, interactiva y comprensible, orientada tanto a usuarios técnicos como no técnicos, y que permita avanzar de manera segura y estructurada desde la exploración de los datos hasta la obtención de conocimiento accionable, que permita tomar decisiones.

Capítulo 3 Hipótesis de trabajo

A lo largo de este capítulo se describen las hipótesis de trabajo sobre las que se fundamenta el desarrollo del presente proyecto, entendiendo por hipótesis de trabajo el conjunto de tecnologías, herramientas, lenguajes de programación y librerías, arquitecturas de software, metodologías y estándares que condicionan y sustentan la implementación de la aplicación final desarrollada.

El proyecto consiste en el desarrollo de una aplicación web interactiva basada en una arquitectura cliente-servidor, en la que se separan claramente las responsabilidades del frontend y del backend. Para ello, se ha optado por el uso de tecnologías ampliamente extendidas en el desarrollo web moderno, combinando un backend desarrollado en Python mediante el framework FastAPI y una vista implementada con la librería React como tecnología de frontend fundamental.

Asimismo, en este capítulo se detalla el grado de relevancia de cada uno de los elementos tecnológicos utilizados, dedicando una descripción más concisa a aquellas tecnologías de

uso más común, y profundizando en mayor medida en aquellas que resultan clave para el desarrollo del proyecto o presentan un menor grado de familiaridad.

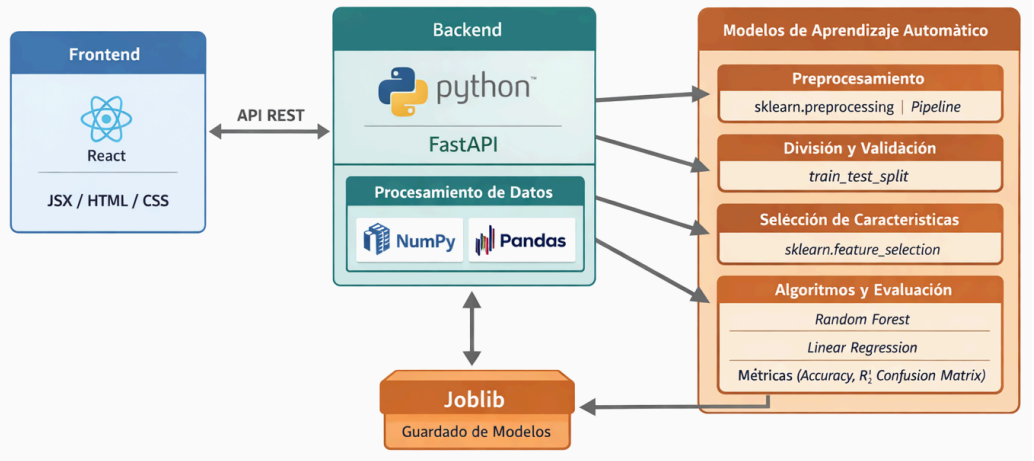


Figura 3.1. Resumen del sistema y de las tecnologías utilizadas

3.1.- TECNOLOGÍAS Y LENGUAJES DE DESARROLLO

En este apartado se describen los lenguajes de programación y las tecnologías principales utilizadas en el desarrollo de la aplicación web, diferenciando entre la parte correspondiente al servidor y la correspondiente al cliente.

3.1.1.- Backend: Python y FastAPI

El desarrollo del backend de la aplicación se ha realizado utilizando el lenguaje de programación **Python**, debido a su simplicidad, legibilidad y amplio uso en el ámbito del desarrollo de servicios web y aplicaciones backend [27].

Para la implementación de la lógica del servidor se ha empleado el framework **FastAPI**, orientado al desarrollo de APIs REST de forma eficiente y estructurada. FastAPI se basa en el estándar ASGI, lo que permite el uso de programación asíncrona y contribuye a mejorar el rendimiento del sistema, especialmente en aplicaciones que gestionan múltiples peticiones concurrentes [28].

Una de las principales características de FastAPI es el uso de tipado estático mediante anotaciones de tipo en Python, lo que permite la validación automática de los datos de entrada y salida de los distintos endpoints. Este enfoque reduce la probabilidad de errores, mejora la robustez del sistema y facilita el mantenimiento del código. Además, FastAPI

genera de forma automática documentación interactiva de la API, lo que resulta de gran utilidad tanto durante el desarrollo como en futuras fases de ampliación y mejora del proyecto.

La comunicación entre el frontend y el backend se realiza mediante el protocolo HTTP, siguiendo el paradigma REST, y utilizando el formato JSON para el intercambio de información entre ambas capas del sistema [29][30]. Este modelo de comunicación favorece el desacoplamiento entre cliente y servidor y permite la integración de distintos clientes sin necesidad de modificar la lógica del backend.

3.1.2.- Frontend: React, JSX, HTML y CSS

La interfaz de usuario de la aplicación se ha desarrollado utilizando **React**, una librería de JavaScript orientada a la creación de interfaces de usuario dinámicas basadas en componentes reutilizables. Este enfoque permite dividir la aplicación en unidades independientes, favoreciendo una estructura modular del código y facilitando tanto el mantenimiento como la evolución futura de la aplicación [31].

React emplea un modelo de renderizado declarativo que permite actualizar de forma eficiente la interfaz de usuario en función de los cambios en el estado de la aplicación. Esta característica contribuye a mejorar el rendimiento y la experiencia de usuario, especialmente en aplicaciones web interactivas.

Para la definición de los componentes se utiliza **JSX**, una extensión de sintaxis que combina JavaScript con una estructura similar a HTML, permitiendo describir de manera clara y declarativa la estructura de la interfaz de usuario. El uso de JSX facilita la integración entre la lógica de la aplicación y su representación visual, mejorando la legibilidad del código [32].

Asimismo, se emplea **HTML** para la estructuración del contenido y **CSS** para la definición del estilo y la presentación visual de la aplicación. Estas tecnologías permiten diseñar una interfaz coherente y adaptable a distintos dispositivos y tamaños de pantalla, contribuyendo a una experiencia de usuario consistente [33][34].

3.1.3.- Librerías de análisis de datos y aprendizaje automático

El sistema desarrollado incorpora un conjunto de librerías especializadas en procesamiento de datos y aprendizaje automático con el objetivo de automatizar el análisis exploratorio de conjuntos de datos y permitir el entrenamiento de modelos predictivos de forma guiada.

Estas herramientas constituyen el núcleo lógico del backend, ya que permiten transformar datos sin procesar en información estructurada y generar modelos capaces de realizar predicciones sobre nuevos datos.

3.1.3.1. Procesamiento y manipulación de datos

Para el tratamiento inicial de los datos se emplean las librerías **NumPy** y **Pandas**, ampliamente utilizadas en el ámbito del análisis de datos en Python.

NumPy proporciona estructuras de datos optimizadas para el cálculo numérico, especialmente arrays multidimensionales, así como operaciones matemáticas vectorizadas de alto rendimiento. Esto permite realizar transformaciones estadísticas y operaciones sobre grandes volúmenes de datos de forma eficiente [35].

Pandas, por su parte, permite trabajar con estructuras tabulares denominadas *DataFrame*, que facilitan la manipulación estructurada de la información. Mediante esta librería se implementan funcionalidades como la identificación de tipos de variables, detección y gestión de valores nulos, cálculo de estadísticas descriptivas y transformación de datos [36].

Ambas librerías son fundamentales para preparar los datos antes de su utilización en algoritmos de aprendizaje automático.

3.1.3.2. Modelado y algoritmos de aprendizaje automático

Para el desarrollo de modelos predictivos se emplea la librería **Scikit-learn**, que proporciona implementaciones eficientes de algoritmos supervisados, herramientas de preprocesamiento y métricas de evaluación [37].

Dentro de esta librería se utilizan los siguientes módulos:

a) Preprocesamiento:

En este módulo encontramos las herramientas necesarias para realizar las transformaciones que nos garanticen que los datos estén en el formato adecuado antes del entrenamiento.

- **Variables categóricas**: por lo general, los algoritmos de IA requieren datos en formato numérico, por lo que es muy común este tipo de transformación. Para

ello, empleamos **OrdinalEncoder**, cuando se trata de categorías que siguen un orden lógico y **OneHotEncoder**, cuando no [38] [39].

- **Variables numéricas:** los datos numéricos también necesitan ser transformados para que los modelos puedan aprender mejor. Por tanto, aplicamos técnicas como **StandardScaler**, que normaliza los datos restando la media y dividiendo entre la desviación estándar, generando valores con media 0 y desviación 1. Y por otro lado, tenemos **MinMaxScaler**, que escala los datos a un rango específico, generalmente [0,1], preservando las relaciones entre valores [40] [41].

b) División y validación:

Para evaluar la capacidad predictiva de los modelos se divide el conjunto de datos en entrenamiento y validación:

- **train_test_split**: separa los datos en dos subconjuntos, típicamente 70-80% entrenamiento y 20-30% validación [42].

- **cross_val_score**: realiza validación cruzada, entrenando el modelo múltiples veces con distintas particiones del conjunto de datos para obtener una medida robusta de su desempeño [43].

c) Selección de características:

Este proceso se emplea para reducir la dimensionalidad del conjunto de datos, eliminar variables poco relevantes y mejorar tanto la eficiencia computacional como la capacidad de generalización de los modelos.

En este módulo, empleamos principalmente **SelectKBest** [44] junto con medidas basadas en información mutua. La información mutua cuantifica el grado de dependencia entre cada variable independiente y la variable objetivo, permitiendo identificar aquellas características con mator capacidad predictiva. Las técnicas utilizadas son **mutual_info_classif** (clasificación) y **mutual_info_regression** (regresión) [45] [46].

d) Modelos supervisados:

El sistema emplea modelos de aprendizaje supervisado seleccionados en función del tipo de variable objetivo.

Cuando la variable objetivo es categórica, el problema se aborda como una tarea de clasificación, utilizando principalmente **RandomForestClassifier**, modelo basado en

ensemble que combina múltiples árboles de decisión para mejorar la precisión y reducir el sobreajuste [47].

Cuando la variable objetivo es numérica, el problema se trata como una tarea de regresión, empleando **LinearRegression**, modelo lineal interpretable adecuado para relaciones aproximadamente lineales, y **RandomForestRegressor**, método que promedia múltiples árboles de decisión para capturar relaciones no lineales [48] [49].

Adicionalmente, se incorporan técnicas de optimización de hiperparámetros mediante **GridSearchCV** y **RandomizedSearchCV**, que permiten evaluar distintas configuraciones del modelo utilizando validación cruzada, seleccionando aquella que maximiza métricas como la exactitud o el coeficiente de determinación (R^2) [50] [51].

e) Métricas de evaluación:

Para medir el rendimiento de los modelos se utilizan las siguientes métricas:

Tabla 3.1. Métricas de evaluación de los modelos de aprendizaje automático

Tipo de problema	Métricas	Descripción
Clasificación	accuracy_score	Proporción de predicciones correctas sobre el total [52]
	confusion_matrix	Matriz que muestra predicciones correctas vs incorrectas por categoría [53]
Regresión	r2_score	Determina qué proporción de la variabilidad de la variable objetivo es explicada por el modelo [54]
	mean_squared_error	Error cuadrático medio de las predicciones [55]
	mean_absolute_error	Promedio de errores absolutos [56]

3.1.3.3. Visualización de gráficos

Para la generación de representaciones visuales se han empleado las librerías **Matplotlib** y **Seaborn** [57] [58].

Matplotlib constituye la base para la creación de gráficos en Python, permitiendo un control detallado sobre la configuración de figuras, ejes y estilos. Por su parte, Seaborn se apoya en Matplotlib y proporciona una interfaz de alto nivel que simplifica la creación de gráficos estadísticos más elaborados, como mapas de calor, distribuciones o gráficos de correlación, mejorando además la estética y claridad visual de las representaciones.

3.1.3.4. Persistencia de modelos

Para la serialización y almacenamiento de modelos entrenados se utiliza la librería Joblib, que permite guardar y cargar modelos de manera eficiente [59]. De esta manera, podemos permitir al usuario descargar los modelos entrenados y reutilizarlos dentro del flujo.

3.2.- ARQUITECTURA Y HERRAMIENTAS DE DESARROLLO

En este apartado se describe la arquitectura general del sistema y las herramientas utilizadas durante el proceso de desarrollo.

3.2.1.- Arquitectura cliente-servidor

La aplicación sigue una arquitectura **cliente-servidor,** en la que se establece una separación clara entre la capa de presentación y la capa de lógica de negocio. El frontend actúa como cliente, siendo el responsable de la interacción con el usuario y de la representación de la información, mientras que el backend funciona como servidor, encargándose del procesamiento de las peticiones, la ejecución de la lógica de negocio y la gestión de los datos [60].

La comunicación entre ambas capas se realiza mediante el protocolo HTTP, a través de una API REST implementada en el backend, utilizando formatos estándar para el intercambio de información. Este enfoque permite desacoplar el cliente del servidor, facilitando el mantenimiento del sistema y posibilitando su escalabilidad y evolución futura.

Figura 3.2. Representación de la arquitectura cliente-servidor

3.2.2.- Herramientas y entorno de desarrollo

Para el desarrollo del proyecto se han utilizado diversas herramientas software que han facilitado la implementación, ejecución y depuración de la aplicación web.

Como entorno de desarrollo integrado se ha utilizado **Visual Studio Code**, un editor de código ampliamente extendido que proporciona soporte para múltiples lenguajes de programación, extensiones específicas para Python y JavaScript, y herramientas de depuración que facilitan el desarrollo tanto del backend como del frontend [61].

En la parte correspondiente al backend, desarrollada con Python y FastAPI, se ha empleado **pip** como gestor de dependencias, permitiendo la instalación y gestión de las librerías necesarias para el correcto funcionamiento de la aplicación [62]. La ejecución del servidor se ha realizado mediante **Uvicorn**, un servidor ASGI ligero y de alto rendimiento, recomendado para aplicaciones desarrolladas con FastAPI [63].

Para el desarrollo del frontend se ha utilizado **npm** como gestor de paquetes, encargado de la instalación y gestión de las dependencias del proyecto React [64]. La creación y configuración inicial del proyecto se ha llevado a cabo mediante **Vite**, una herramienta moderna de construcción y desarrollo que permite una puesta en marcha rápida de aplicaciones web y ofrece un entorno de desarrollo eficiente con recarga automática de los cambios realizados [65].

El desarrollo de la aplicación se ha realizado en un entorno basado en el sistema operativo **Windows**, que ha proporcionado el soporte necesario para la ejecución de todas las herramientas y tecnologías empleadas.

Capítulo 4 Metodología y resultados

El desarrollo del presente proyecto se ha estructurado siguiendo un enfoque iterativo–incremental, que permite la incorporación progresiva de mejoras y ajustes a medida que se avanza en la construcción del sistema. Este enfoque resulta especialmente adecuado para proyectos de carácter práctico y experimental, como el desarrollo de un asistente para análisis exploratorio de datos, generación de modelos de inteligencia artificial y simulación de datos, donde los módulos requieren refinamiento continuo según los resultados obtenidos.

El proyecto se ha organizado en distintas fases: planificación, captura de requisitos, diseño del sistema, implementación, pruebas y validación de resultados. Para ofrecer una visión clara de la distribución temporal de estas etapas, se ha elaborado un diagrama de Gantt,

que refleja la duración estimada de cada fase y la superposición de tareas, permitiendo visualizar el carácter iterativo del desarrollo.

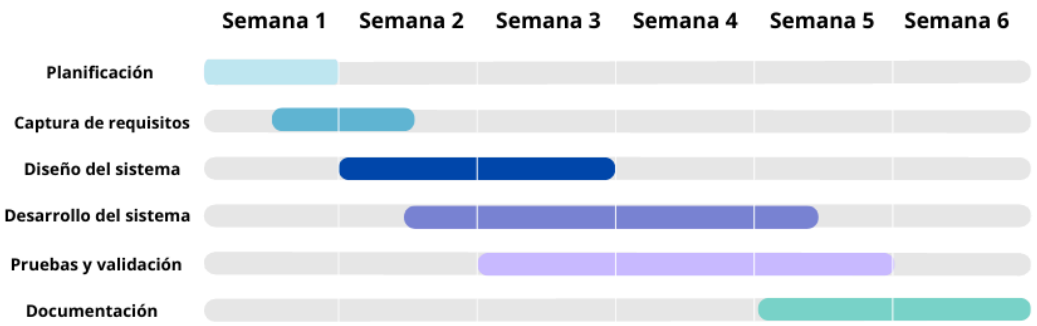


Figura 4.1. Diagrama de Gantt

A lo largo de este capítulo se presenta una visión del proceso de desarrollo, integrando de manera ordenada todas las fases del proyecto.

4.1.- PLANIFICACIÓN DEL PROYECTO

La planificación del proyecto se ha llevado a cabo siguiendo un modelo de ciclo de vida **iterativo–incremental**, coherente con la naturaleza modular del sistema desarrollado y con la necesidad de integrar progresivamente los distintos componentes funcionales [66].

Este modelo organiza el desarrollo en etapas claramente definidas: planificación, análisis de requisitos, diseño, desarrollo, pruebas y documentación; permitiendo, al mismo tiempo, la revisión y mejora continua de los módulos implementados. A diferencia de los modelos estrictamente secuenciales, el enfoque iterativo facilita la adaptación del sistema a medida que se validan sus funcionalidades. La Figura 4.2 muestra de forma esquemática el ciclo de vida adoptado.

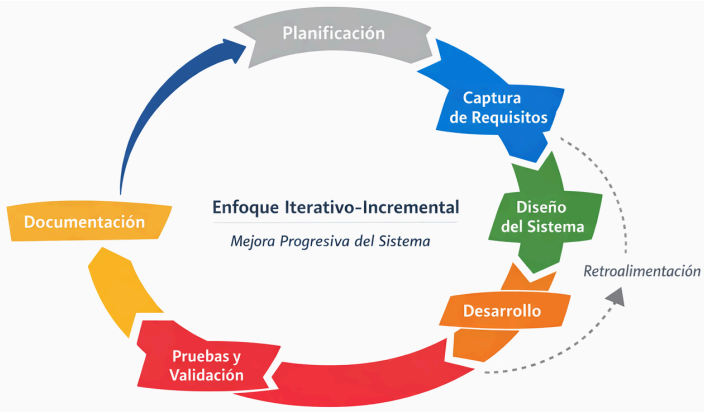


Figura 4.2. Modelo de ciclo de vida iterativo-incremental

4.2.- CAPTURA DE REQUISITOS

En este apartado se identifican y describen los usuarios del sistema, así como las acciones que pueden realizar dentro de la aplicación. La información se organiza mediante plantillas de descripción de roles y casos de uso que se ofrecen en la asignatura Ingeniería del Software [67].

4.2.1.- Actor

El sistema está diseñado actualmente para un único tipo de usuario: **Usuario estándar**, sin necesidad de autenticación ni registro previo. Este usuario puede acceder directamente a la aplicación web y utilizar todas las funcionalidades disponibles.

Tabla 4.1. Descripción del actor usuario estándar

Actor	Usuario estándar
Descripción	Usuario que accede a la aplicación web sin necesidad de registro. Puede subir datasets, preprocesar datos, entrenar modelos, visualizar gráficos, simular datos y descargar resultados. Diseñado para usuarios sin conocimientos técnicos avanzados.
Casos de uso relacionados	C.U.1, C.U.2, C.U.3, C.U.4, C.U.5, C.U.6, C.U.7

4.2.2.- Casos de uso

Se presentan a continuación los siete casos de uso del sistema, mostrando cómo el usuario estándar interactúa con las funcionalidades principales de la aplicación. Cada caso de uso describe de forma estructurada los pasos necesarios para completar una acción específica, sus condiciones de ejecución y los resultados esperados, ofreciendo una visión del comportamiento del sistema:

Tabla 4.2. C.U.1 - Cargar dataset (EDA)

C.U.1	Cargar dataset (Módulo EDA)
Actores	Usuario estándar
Descripción	El usuario carga un archivo en formato CSV para su utilización en el módulo análisis exploratorio de datos (EDA). El sistema valida el formato del archivo sea CSV y procesa la información básica, mostrando un resumen inicial del dataset.
Dependencias	Ninguna. Este caso de uso independiente
Precondición	- El usuario se encuentra en la página Análisis EDA. - El módulo de EDA está activo y listo para recibir un archivo CSV.
Secuencia normal	1. El usuario pulsa el botón “Seleccionar archivo” o arrastra un archivo hacia la sección correspondiente. 2. El sistema verifica que el archivo tenga formato CSV. 3. El sistema muestra el nombre del archivo y desplaza la vista a la sección inferior. 4. El sistema presenta información básica del dataset: número de filas, columnas, tamaño y tipos de datos. 5. El dataset queda disponible para que el usuario realice análisis exploratorio, limpieza y visualización.
Poscondición	Dataset cargado y disponible para análisis exploratorio.
Excepciones	- Si el archivo no es CSV → el sistema muestra un mensaje de error y no permite continuar - Si el archivo está vacío → el sistema muestra un aviso y no permite continuar.
Rendimiento	La carga y validación debe completarse en tiempo razonable para datasets de tamaño moderado. Pueden existir limitaciones implícitas debido a la memoria disponible del sistema.
Frecuencia	Alta. Se espera que el usuario cargue un dataset cada vez que inicia un análisis en EDA.
Importancia	Alta. Constituye el punto de entrada funcional del módulo de análisis exploratorio.
Urgencia	Alta. Es indispensable para el correcto funcionamiento del módulo EDA.
Estado	Implementado
Estabilidad	Media. Depende del tamaño y la calidad del dataset; para datasets muy grandes o con inconsistencias, el tiempo de procesamiento puede verse afectado.
Comentarios	Este caso de uso establece el flujo inicial del usuario en el módulo EDA y condiciona todas las operaciones posteriores de limpieza, exploración y visualización de los datos. Es crítico para garantizar que el análisis pueda iniciarse correctamente y de manera flexible.

Tabla 4.3. C.U.2 - Análisis Exploratorio de Datos (EDA)

C.U.2	Análisis Exploratorio de Datos (EDA)
Actores	Usuario estándar
Descripción	El usuario realiza un análisis exploratorio del dataset cargado. El sistema muestra automáticamente un resumen inicial del dataset y permite navegar sobre el resto de información. El usuario puede explorar, limpiar y visualizar los datos mediante gráficos y estadísticas básicas. También puede decidir eliminar columnas irrelevantes, preparar el dataset y descargarlo para uso posterior.
Dependencias	C.U.1 – Dataset cargado y disponible en el módulo EDA.
Precondición	- El usuario se encuentra en el módulo EDA - Dataset cargado en formato CSV y disponible en el módulo EDA.
Secuencia normal	1. El sistema muestra la información relevante para el análisis exploratorio de los datos y ofrece sugerencias de mejora. 2. El usuario explora la información disponible mediante botones. 3. El usuario decide aplicar cambios. 4. El sistema actualiza la información y visualizaciones de manera interactiva tras cada acción. 5. El usuario puede repetir el paso 3 para refinar el dataset. 6. El usuario puede descargar el dataset procesado. 7. El sistema ofrece la opción de ir a “entrenamiento de datos”.
Poscondición	- Dataset modificado según las acciones del usuario y disponible para descarga y uso en el módulo de entrenamiento de modelos. - Cambios aplicados se reflejan en el dataset descargado.
Excepciones	Columnas con valores no procesables → el sistema recomienda eliminarlas.
Rendimiento	Las operaciones de análisis y visualización deben completarse en tiempo razonable según el tamaño del dataset y la memoria disponible.
Frecuencia	Alta. Cada vez que el usuario realiza un análisis de un dataset.
Importancia	Alta. Constituye la funcionalidad principal del módulo de EDA.
Urgencia	Alta. Es necesario para que el usuario comprenda y prepare los datos antes de entrenamiento o simulación.
Estado	Implementado
Estabilidad	Media. Depende de la calidad y tamaño del dataset; datasets muy grandes o con datos inconsistentes pueden afectar tiempos de visualización y análisis.
Comentarios	Este caso de uso refleja un flujo interactivo e iterativo, donde el usuario puede explorar, limpiar y preparar datos con la asistencia de recomendaciones del sistema, garantizando que el dataset final sea comprensible y utilizable en etapas posteriores, como entrenamiento de modelos de IA.

Tabla 4.4. C.U.3 - Cargar y evaluar dataset (Entrenamiento)

C.U.3	Cargar y evaluar dataset (Módulo Entrenamiento)
Actores	Usuario estándar
Descripción	El usuario carga un archivo en formato CSV para entrenar un modelo de IA. El sistema muestra un resumen del dataset y realiza una evaluación de calidad. En función de dicha evaluación, el sistema puede habilitar la continuación del proceso, recomendar la limpieza con EDA o redirigir al módulo de simulación de datos si no es apto para entrenamiento.
Dependencias	Ninguna.
Precondición	- El usuario se encuentra en el módulo de Entrenamiento de modelos. - El sistema está listo para recibir un archivo CSV.
Secuencia normal	1. El usuario selecciona o arrastra un archivo CSV. 2. El sistema verifica el formato y procesa el archivo. 3. El sistema muestra automáticamente información básica del archivo y los resultados de la evaluación de calidad. 4. En función de los resultados obtenidos en la evaluación, el sistema habilita las opciones correspondientes: continuar, redirigir a EDA o a simulación de datos.
Poscondición	- Dataset evaluado - Botón “Continuar” habilitado si el dataset es apto para entrenamiento. - En caso contrario, opción de redirección habilitada según el resultado de la evaluación.
Excepciones	- Si el archivo no es CSV → el sistema muestra un mensaje de error - Si el archivo está vacío → el sistema muestra un aviso
Rendimiento	- La carga y validación debe completarse en tiempo razonable para datasets de tamaño moderado. - Puede existir limitación implícita por la memoria del sistema.
Frecuencia	La carga y validación deben completarse en tiempo razonable para datasets de tamaño moderado. Puede existir limitación implícita por la memoria disponible del sistema.
Importancia	Media-alta. Se ejecuta cada vez que el usuario desea entrenar un modelo.
Urgencia	Alta. Sin una evaluación adecuada del dataset no puede garantizarse un entrenamiento correcto.
Estado	Implementado
Estabilidad	Media. Puede verse afectada por el tamaño del dataset y por la complejidad de las validaciones realizadas.
Comentarios	Este caso de uso introduce una validación más estricta que la realizada en el módulo EDA, ya que el entrenamiento requiere una estructura de datos coherente y adecuada. La evaluación automática permite asistir al usuario en la toma de decisiones antes de iniciar el proceso de modelado.

Tabla 4.5. C.U.4 - Entrenar modelo de IA

C.U.4	Entrenar modelo de IA
Actores	Usuario estándar
Descripción	El usuario entrena un modelo de IA a partir de un dataset ya validado. El sistema permite seleccionar la variable objetivo (target), identifica automáticamente el tipo de problema (clasificación o regresión) y muestra modelos compatibles. Se selecciona y ejecuta el modelo, y se presentan métricas de rendimiento para evaluar la calidad del modelo.
Dependencias	C.U.3 – Dataset validado y botón "Continuar" habilitado.
Precondición	- El dataset ha sido evaluado como apto para entrenamiento. - El usuario ha pulsado "Continuar".
Secuencia normal	1. El sistema muestra la sección de selección de variable objetivo (target) y sugiere una. 2. El usuario selecciona la variable target. 3. El sistema determina automáticamente el tipo de problema (clasificación o regresión) en función del tipo de la variable objetivo. 4. El sistema muestra modelos compatibles y sugiere. 5. El usuario selecciona el modelo. 6. El sistema inicia el entrenamiento. 7. Se muestran métricas de rendimiento (según el tipo de problema). 8. El sistema habilita las opciones posteriores: realizar predicciones o descargar resultados (definidas en casos de uso independientes).
Poscondición	- Modelo entrenado. - Métricas de rendimiento disponibles para su consulta. - Funcionalidades de predicción y descarga habilitadas.
Excepciones	- Target no seleccionado → no se permite continuar. - Error durante entrenamiento → mensaje de error y reintentar.
Rendimiento	El tiempo de ejecución dependerá del tamaño del dataset y del modelo seleccionado (tiempo razonable para datasets de tamaño moderado).
Frecuencia	Media. Se ejecuta cuando el usuario dispone de un dataset validado y desea generar un modelo predictivo.
Importancia	Alta. Constituye el núcleo funcional del módulo de entrenamiento.
Urgencia	Alta. Es imprescindible para generar modelos predictivos.
Estado	Implementado
Estabilidad	Media. Puede verse afectada por el tamaño del dataset, la complejidad del modelo y la disponibilidad de recursos del sistema.
Comentarios	Este caso de uso se centra exclusivamente en el proceso de entrenamiento del modelo. Las acciones posteriores, como la realización de predicciones o la descarga del modelo, se definen en casos de uso independientes para mantener la modularidad y claridad del sistema.

Tabla 4.6. C.U.5 - Realizar predicciones

C.U.5	Realizar predicciones
Actores	Usuario estándar
Descripción	El usuario utiliza un modelo de inteligencia artificial previamente entrenado para generar predicciones a partir de nuevos datos. El sistema solicita los valores de las variables de entrada correspondientes, procesa la información a través del modelo seleccionado y muestra la predicción obtenida. Esta funcionalidad permite validar de manera inmediata la utilidad del modelo generado y aplicar sus resultados.
Dependencias	C.U.4 – Modelo entrenado disponible para su uso.
Precondición	- El usuario ha entrenado un modelo de IA (C.U.4) y este está disponible para realizar predicciones. - Ya están cargadas las variables de entrada que requiere el modelo.
Secuencia normal	1. El usuario accede a “predicción” en el módulo de entrenamiento. 2. El sistema muestra los campos correspondientes a las variables de entrada utilizadas durante el entrenamiento del modelo. 3. El usuario introduce los valores deseados en cada campo. 4. El usuario pulsa el botón “Predecir”. 5. El sistema procesa los datos a través del modelo entrenado. 6. El sistema muestra la predicción obtenida, junto con información relevante sobre la confianza o métricas asociadas, si procede.
Poscondición	- Resultado de la predicción disponible y mostrado al usuario.
Excepciones	
Rendimiento	La predicción debe completarse en tiempo inmediato o muy corto, garantizando la interactividad de la herramienta, incluso para datasets de tamaño moderado.
Frecuencia	Media/Alta. Cada vez que el usuario desee probar un modelo entrenado con nuevos datos.
Importancia	Alta. Permite al usuario validar el modelo y generar resultados prácticos a partir del sistema.
Urgencia	Alta. Clave para el flujo completo de análisis y entrenamiento.
Estado	Implementado.
Estabilidad	Alta. Depende de que los datos de entrada sean consistentes y el modelo esté correctamente entrenado.
Comentarios	Este caso de uso se centra únicamente en generar predicciones con modelos entrenados. No incluye la descarga de modelos ni la modificación de los datos de entrenamiento, que se abordan en casos de uso independientes. La interfaz debe ser sencilla, clara y accesible para usuarios sin conocimientos técnicos avanzados, mostrando resultados y advertencias de manera comprensible.

Tabla 4.7. C.U.6 - Generar/Simular dataset

C.U.6	Generar/Simular dataset
Actores	Usuario estándar
Descripción	El usuario crea un dataset simulado personalizado para ser utilizado en análisis exploratorio o entrenamiento de modelos de IA. El sistema permite definir las variables, tipos de datos, rangos de valores y categorías, generando un dataset coherente que puede ser descargado o usado directamente en el sistema.
Dependencias	Ninguna. Este caso de uso es independiente.
Precondición	El usuario se encuentra en el módulo de simulación de datos.
Secuencia normal	1. El sistema muestra la interfaz de simulación con opciones para definir variables y sus características. 2. El usuario define las variables y sus propiedades. 3. El sistema valida la coherencia de las definiciones (sin duplicados de nombres, rangos correctos, categorías consistentes). 4. El usuario confirma la generación del dataset. 5. El sistema genera el dataset simulado según los parámetros definidos.
Poscondición	Dataset simulado generado.
Excepciones	Conflictos en nombres o tipos → el sistema solicita ajustes antes de generar el dataset.
Rendimiento	La generación del dataset debe completarse en tiempo razonable, incluso para datasets de tamaño moderado.
Frecuencia	Media. Dependerá de la necesidad de generar datasets de prueba o entrenamiento.
Importancia	Alta. Constituye el núcleo funcional del módulo de entrenamiento.
Urgencia	Alta. Permite al usuario crear datasets personalizados.
Estado	Implementado.
Estabilidad	Alta. Depende de la coherencia de las definiciones del usuario..
Comentarios	Este caso de uso complementa la preparación de datos, permitiendo al usuario generar datasets consistentes y adaptados a sus necesidades sin depender de fuentes externas.

Tabla 4.8. C.U.7 - Descargar resultados

C.U.7	Descargar resultados
Actores	Usuario estándar
Descripción	El usuario descarga los resultados generados dentro del módulo activo del sistema: - En análisis EDA: dataset procesado (CSV). - En entrenamiento: modelo entrenado (archivo PKL). - En simulación datos: dataset generado (CSV). El sistema prepara el archivo en el formato correspondiente y permite su descarga local.
Dependencias	Resultados disponibles en el módulo activo: EDA, Entrenamiento de modelos o Simulación de datos.
Precondición	El usuario ha completado las acciones necesarias dentro del módulo activo y existen resultados listos para descargar.
Secuencia normal	1. El usuario activa la opción “Descargar archivo” dentro del módulo. 2. El sistema inicia la descarga y notifica al usuario al completarse.
Poscondición	Archivo descargado correctamente y disponible para uso externo.
Excepciones	- Error en la descarga → el sistema notifica al usuario y permite reintentar.
Rendimiento	La descarga debe completarse en tiempo razonable según el tamaño del archivo y los recursos disponibles.
Frecuencia	Media-Alta. Se espera que el usuario descargue resultados tras finalizar el flujo de cada módulo.
Importancia	Alta. Permite al usuario conservar los resultados generados en el módulo.
Urgencia	Media-Alta. Es una acción complementaria que cierra el flujo operativo dentro de cada módulo.
Estado	Implementado.
Estabilidad	Alta. Depende del tamaño del archivo y de la conexión del usuario.
Comentarios	Este caso de uso representa la funcionalidad de descarga contextual, disponible en cada módulo una vez que los resultados estén listos.

4.2.3.- Requisitos funcionales

A continuación, se detallan los requisitos funcionales del sistema, organizados por módulo. Estos describen las funcionalidades que el usuario estándar puede realizar, así como las condiciones y restricciones asociadas a cada operación.

4.2.3.1. Módulo Análisis exploratorio de datos (EDA)

- Carga de dataset: El sistema deberá permitir la carga de archivos en formato CSV para su análisis.
- Visualización de datos: El sistema mostrará estadísticas descriptivas básicas (número de filas y columnas, tipos de datos, valores nulos y categorías). Asimismo, se generarán visualizaciones gráficas como histogramas, heatmaps, boxplots, scatterplots y matrices de correlación. El usuario podrá seleccionar subconjuntos de filas para su visualización.
- Limpieza y transformación: El sistema permitirá la eliminación de columnas, la imputación básica de valores nulos (media, mediana o moda) y la conversión de tipos de datos cuando sea necesario para facilitar análisis posteriores.
- Descarga del dataset procesado: El dataset resultante podrá descargarse en formato CSV.

4.2.3.2. Módulo Entrenamiento de modelos de IA

- Carga y evaluación del dataset: El sistema permitirá la carga de archivos CSV y mostrará un resumen del dataset. Se realizará una evaluación básica para determinar su idoneidad para el entrenamiento. En función del resultado, se permitirá continuar con el proceso o se sugerirá el uso de otros módulos.
- Selección del target: El sistema sugerirá una variable objetivo, aunque el usuario podrá seleccionar libremente cualquier variable disponible.
- Selección y entrenamiento de modelos: El sistema determinará automáticamente si el problema corresponde a clasificación o regresión según la variable objetivo seleccionada. En función de ello, se mostrarán los modelos disponibles (Random Forest Classifier, Random Forest Regressor, XGBoost y Linear Regression). Se sugerirá un modelo por defecto, pero la decisión final corresponderá al usuario.
  Tras el entrenamiento, se presentarán las métricas de rendimiento correspondientes.
- Predicciones manuales: El sistema permitirá introducir valores manuales para las variables predictoras con el fin de generar predicciones utilizando el modelo entrenado en la sesión actual. Esta funcionalidad estará disponible hasta que se entrene un nuevo modelo o se reinicie la sesión.

- Descarga del modelo entrenado: El modelo generado podrá descargarse para su uso externo.

4.2.3.3. Módulo Simulación de datos

- Creación del dataset: El sistema permitirá definir variables numéricas o categóricas, incluyendo su nombre, rango de valores, probabilidades y parámetros de dispersión. Se podrán establecer configuraciones básicas para la generación de datos coherentes.
- Descarga del dataset generado: El dataset simulado podrá descargarse en formato CSV.

4.2.3.4. Requisitos generales

- Limitación de tamaño: Para garantizar un rendimiento adecuado en entornos con recursos limitados, el sistema estará orientado al procesamiento de datasets de tamaño moderado (hasta aproximadamente 20.000 filas).
- Interfaz y usabilidad: El sistema proporcionará retroalimentación visual tras cada operación, facilitando la comprensión del proceso por parte del usuario.

4.3.- DISEÑO

En este apartado se describe la arquitectura y estructura interna de la aplicación, así como el diseño de sus componentes y la interacción entre ellos. Se incluyen los diagramas UML más relevantes, como diagramas de clases, secuencia, estados y actividad, que permiten visualizar la organización del sistema y el flujo de datos y operaciones. Además, se presenta la relación entre interfaces de usuario, servicios y módulos, mostrando cómo se gestionan las funcionalidades principales de manera modular, consistente y mantenible. Los diagramas sirven para explicar tanto la lógica de negocio como la interacción del usuario con la aplicación, facilitando la comprensión técnica y la planificación de la implementación.

4.3.1.- Diagrama de clases

El diagrama de clases refleja la estructura estática de la aplicación, mostrando las clases principales, sus responsabilidades y relaciones. Permite visualizar cómo se organiza el sistema, qué entidades participan en cada módulo y cómo interactúan entre sí, ofreciendo

una perspectiva clara de la arquitectura orientada a objetos y de la separación de responsabilidades entre interfaz y servicios.

El sistema se estructura en torno a un punto de entrada central, "**MainBackend**", que gestiona los endpoints y enruta las solicitudes a los servicios correspondientes de cada módulo: "**ServicioEDA**", "**ServicioEntrenamiento**" y "**ServicioSimulación**". Cada módulo cuenta con su interfaz independiente, encargada de mostrar información, permitir la interacción del usuario y enviar peticiones al backend. La "**InterfazPrincipal**" actúa como coordinadora, proporcionando navegación unificada y acceso a todos los módulos desde un único punto de entrada.

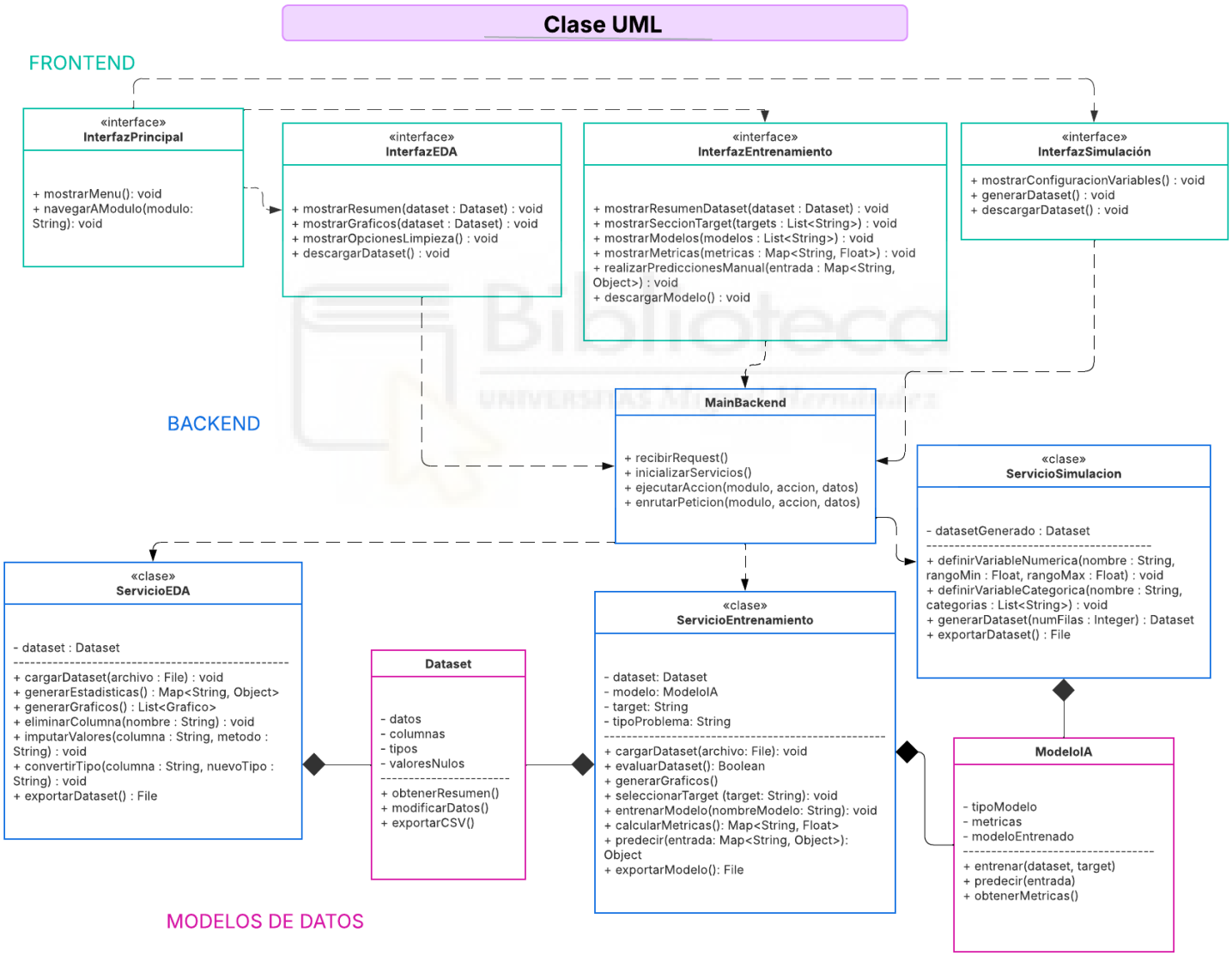


Figura 4.3. Diagrama de clases

El módulo de EDA permite explorar, limpiar y visualizar datasets mediante estadísticas y gráficos interactivos, así como aplicar transformaciones básicas y descargar el dataset

procesado. El módulo de Entrenamiento facilita la selección de la variable objetivo, sugiere modelos según el tipo de problema (clasificación o regresión), entrena los modelos disponibles y permite realizar predicciones manuales con los modelos generados. El módulo de Simulación ofrece herramientas para definir variables numéricas o categóricas, establecer rangos, probabilidades y correlaciones, generar datasets personalizados y descargarlos en formato CSV.

Los servicios procesan los datos y modelos en memoria durante la sesión, manteniendo la independencia entre módulos y evitando el intercambio automático de información, lo que garantiza consistencia y evita conflictos. Esta arquitectura centralizada simplifica la gestión del flujo de datos y la lógica de negocio, manteniendo un sistema modular, escalable y fácil de mantener, donde cada componente cumple funciones claramente definidas dentro del flujo operativo del usuario, y la "**InterfazPrincipal**" asegura la coordinación y coherencia en la experiencia de uso.

4.3.2.- Diagrama de secuencia

En este apartado se presentan los diagramas de secuencia que ilustran el flujo de interacción entre el usuario, las interfaces y los servicios del sistema durante las operaciones más relevantes. Los diagramas muestran paso a paso cómo se manejan la carga de archivos CSV, el análisis exploratorio de datos, el entrenamiento de modelos de IA y la generación de datasets simulados. Cada diagrama refleja la comunicación entre los actores y los componentes internos, permitiendo visualizar cómo se procesan las solicitudes, se realizan validaciones y se actualiza la información, garantizando la consistencia y el correcto funcionamiento del sistema.

4.3.2.1. Carga y validación de archivo CSV

El usuario inicia su interacción desde la Interfaz Principal (landing page), que funciona como punto de entrada a los distintos módulos: Análisis Exploratorio de Datos (EDA), entrenamiento de modelos IA y simulación de datasets. Desde esta interfaz, el usuario selecciona el módulo que desea utilizar.

Al acceder, por ejemplo, al módulo EDA, el usuario selecciona o arrastra un archivo CSV, el sistema valida su formato, y si es correcto, presenta el nombre del archivo cargado como respuesta y habilita la sección que muestra información del dataset. En caso opuesto, aparece un mensaje de error y no se muestra más información. En la figura 4.4. se puede observar el diagrama de secuencia que representa esta interacción.

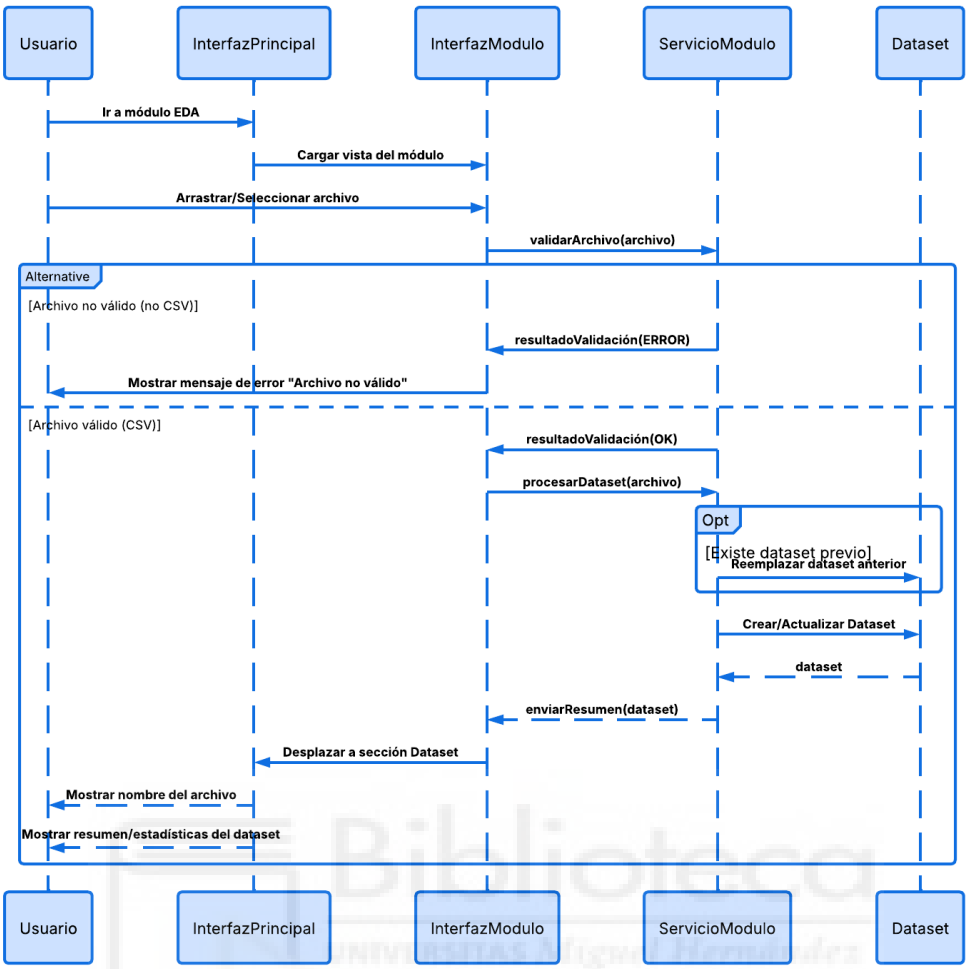


Figura 4.4. Diagrama de secuencia - Carga y validación de archivo CSV

4.3.2.2. Módulo EDA

En la Figura 4.5 se representa el flujo de interacción en el módulo Análisis exploratorio EDA. El usuario puede navegar a través de los distintos botones que le permiten acceder a distinta información relevante del dataset.

Al mismo tiempo, puede realizar acciones sobre los datos, como eliminar columnas, imputar valores o aplicar transformaciones básicas. Este ciclo de exploración y modificación se repite de forma iterativa hasta que el usuario considera que el dataset está preparado. Finalmente, el sistema permite descargar el dataset procesado en formato CSV, conservando los cambios aplicados.

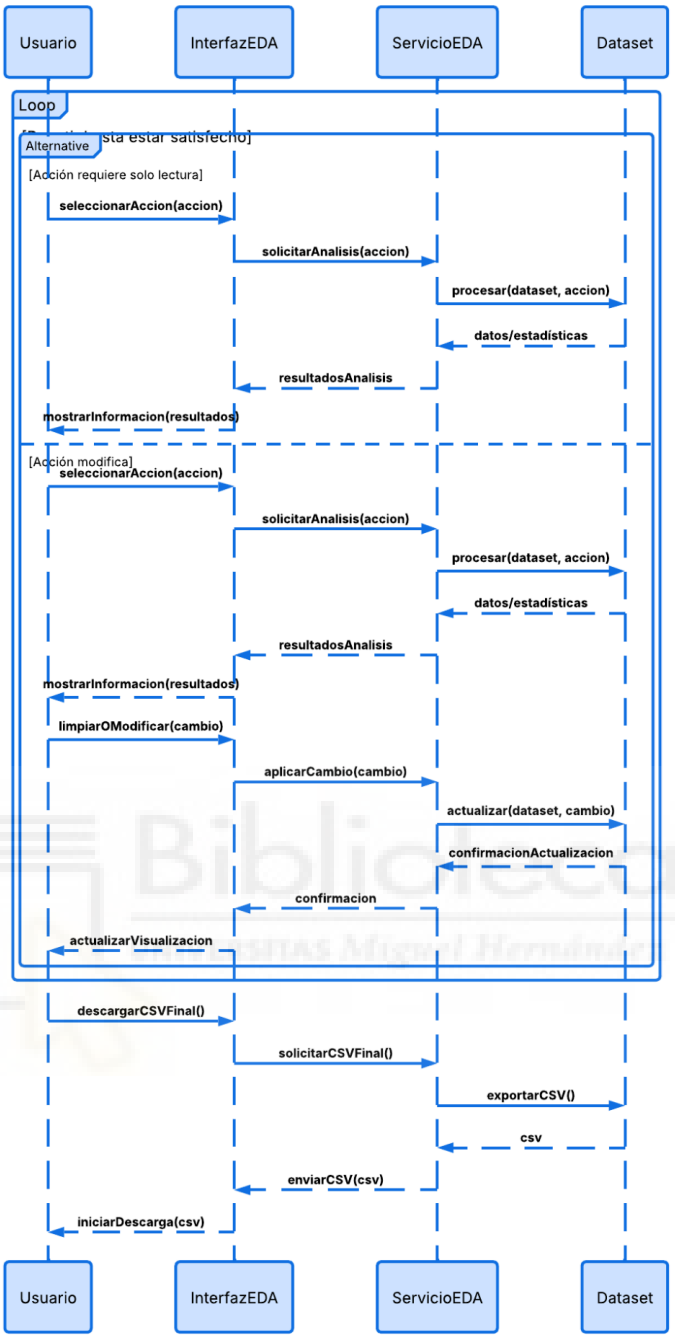


Figura 4.5. Diagrama de secuencia - Módulo EDA

4.3.2.3. Entrenamiento de modelo de IA

El siguiente diagrama (Figura 4.6) muestra la interacción que se produce en el módulo de Entrenamiento de Modelos de IA entre el actor principal (usuario) y los componentes del sistema implicados.

El flujo comienza cuando el usuario carga un archivo en el sistema. Este es evaluado para determinar si cumple los requisitos necesarios para el entrenamiento (estructura válida,

presencia de variables adecuadas, formato correcto, etc.). Si el dataset no es apto, el sistema informa al usuario y no permite continuar con el proceso, pudiendo redirigirlo a otros módulos para su revisión o preparación.

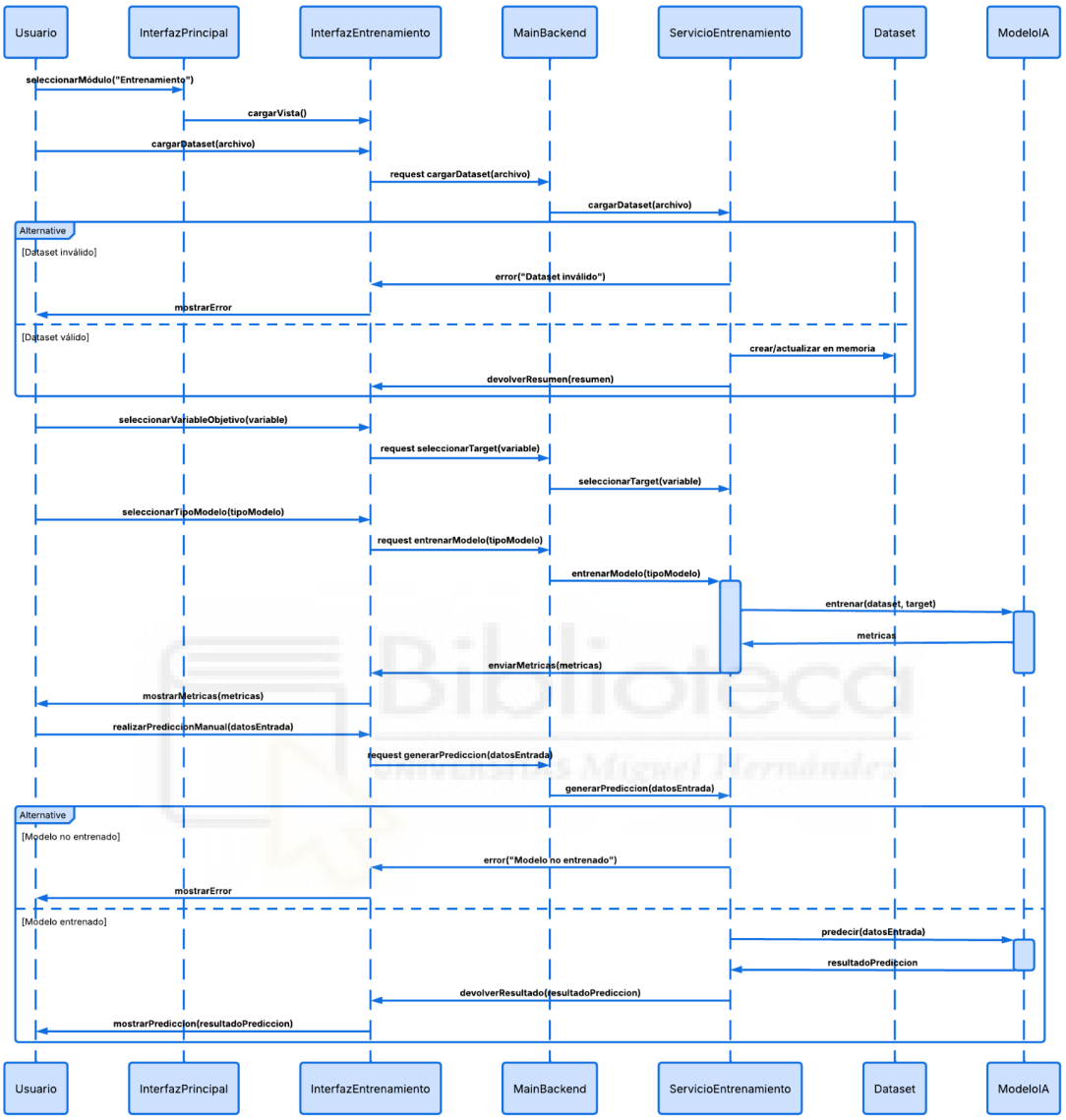


Figura 4.6. Diagrama de secuencia del entrenamiento del modelo de IA

En caso de validación satisfactoria, el usuario accede a la fase de selección de la variable objetivo (target). Una vez tomada esta decisión, el sistema habilita la selección del modelo de aprendizaje automático. Tras elegir el modelo, el usuario inicia el proceso de entrenamiento.

Si durante el entrenamiento se produce algún error, el sistema devuelve un mensaje informativo indicando la causa. Si el proceso finaliza correctamente, se muestran las métricas de rendimiento del modelo y se habilitan las opciones para descargarlo o realizar predicciones manuales directamente desde la interfaz.

4.3.2.4. Simulación de dataset

El proceso comienza cuando el usuario accede al módulo de simulación y el sistema carga la vista correspondiente. A partir de ese momento, se inicia un flujo iterativo en el que el usuario puede definir distintas variables, indicando su nombre, tipo (numérica o categórica) y los parámetros necesarios para su generación.

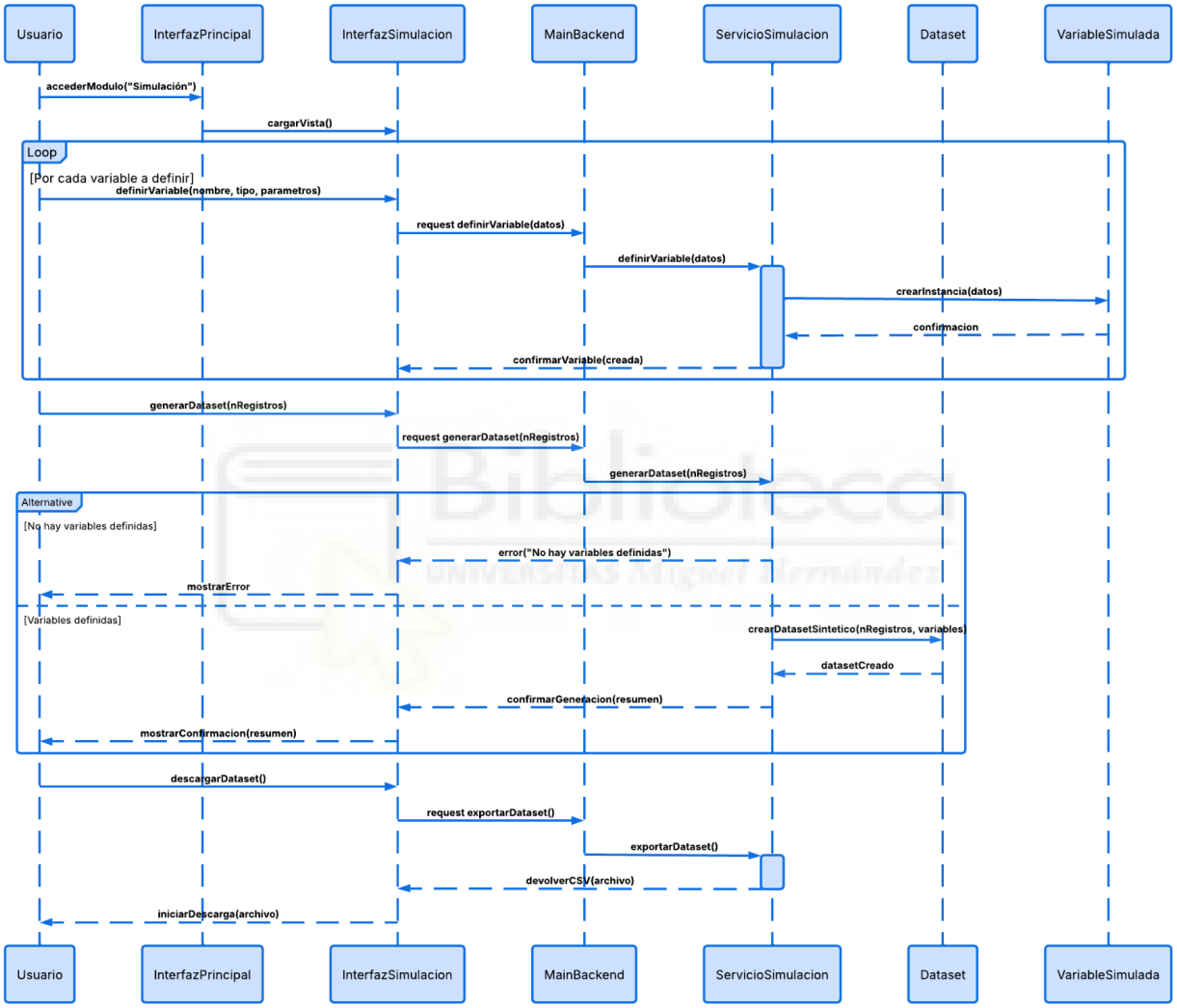


Figura 4.7. Diagrama de secuencia del módulo simulación de dataset

Cada vez que se crea una variable, el sistema valida los datos introducidos y la añade a la configuración del dataset. Este proceso se repite hasta que el usuario decide finalizar la definición de variables.

Cuando el usuario solicita la generación del dataset, la vista envía la petición al controlador principal, que delega la operación en el servicio de simulación. Si no existen variables definidas, el sistema bloquea la operación y muestra un mensaje de advertencia. En caso

contrario, el servicio genera el dataset conforme a los parámetros establecidos. Finalmente, el dataset generado se devuelve a la vista y el usuario puede descargarlo en formato CSV.

4.3.3.- Diagrama de estados

Los diagramas de estados permiten representar el comportamiento dinámico del sistema mediante la definición de los distintos estados en los que puede encontrarse una interfaz o módulo, así como las transiciones que se producen entre ellos en función de las acciones del usuario o eventos internos. A diferencia de los diagramas de secuencia o de actividad, este tipo de diagrama se centra en la evolución del estado interno del sistema, mostrando cómo responde ante diferentes situaciones.

4.3.3.1. Diagrama de estados para interfaz EDA

El módulo EDA comienza en un estado inicial en el que no existe ningún dataset cargado. Cuando el usuario sube un archivo válido, el sistema transita al estado "**DatasetCargado**". Desde este estado, el usuario puede realizar visualizaciones o aplicar operaciones de limpieza y transformación, lo que provoca transiciones a estados intermedios donde el dataset es modificado. Una vez procesado, el sistema permite la descarga del dataset, manteniéndose en un estado operativo hasta que se cargue un nuevo archivo o se reinicie la sesión.

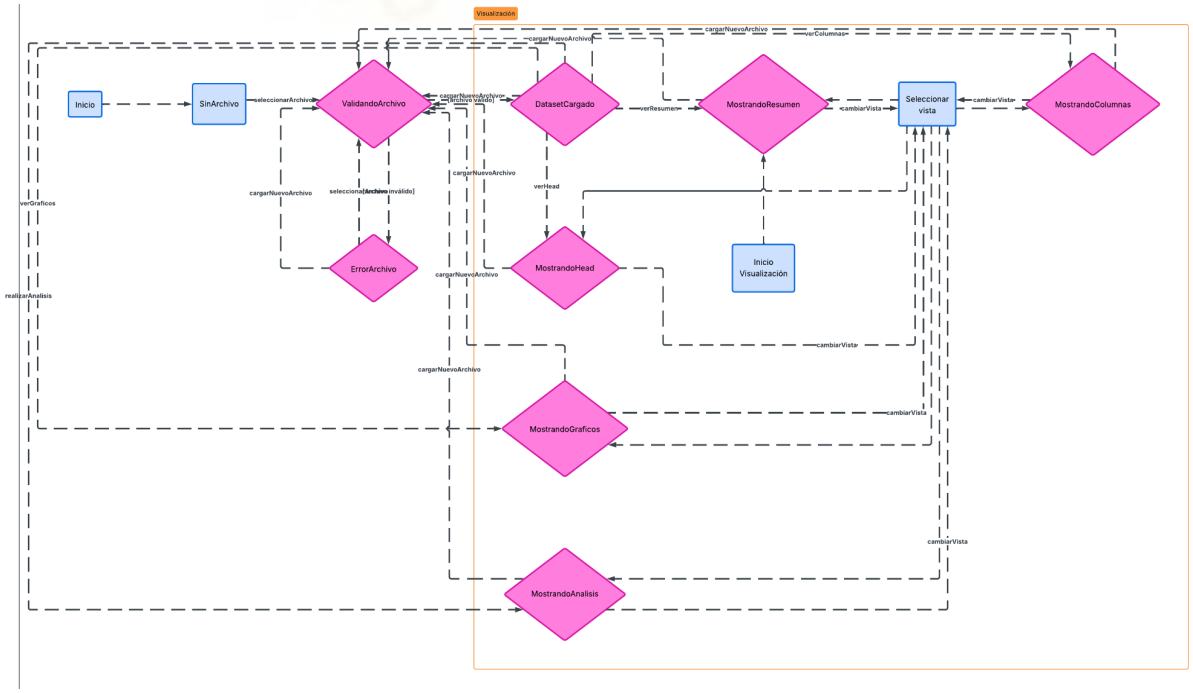


Figura 4.8. Diagrama de estados interfaz EDA

4.3.3.2. Diagrama de estados para la interfaz entrenamiento de modelos

El módulo de entrenamiento de modelos comienza en el estado "**SinDataset**", en el que el sistema aún no cuenta con ningún dataset cargado. Desde este estado, el usuario carga un archivo, lo que provoca la transición al estado "**ValidandoDataset**", donde el sistema comprueba que los datos son aptos para el entrenamiento. Validado el dataset, se pasa al estado "**DatasetValido**", donde el usuario puede seleccionar la variable objetivo, pasando el sistema al estado "**SeleccionandoTarget**". A continuación, se elige un modelo de IA, generando la transición al estado "**SeleccionandoModelo**".

Se inicia el entrenamiento, en el estado "**EntrenandoModelo**", si finaliza correctamente, se muestran las métricas de rendimiento en el estado "**MostrandoMétricas**". Desde este punto, el usuario puede realizar predicciones manuales y descargar el modelo entrenado, alcanzando el estado "**PrediccionManual**". Finalmente, se habilita el estado "**ModeloDescargable**" para que el usuario pueda realizar la descarga del modelo.

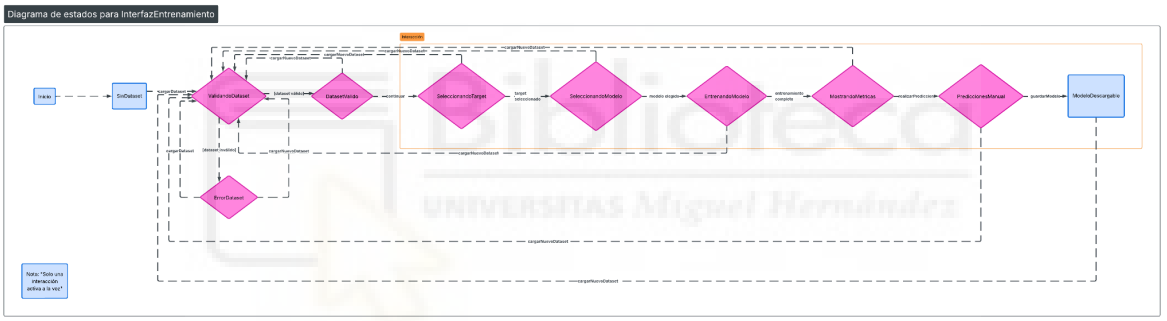


Figura 4.9. Diagrama de estado entrenamiento de modelos

4.3.4.- Diagrama de actividad

Los diagramas de actividad representan el flujo de acciones y decisiones dentro de un sistema o módulo, mostrando cómo se desarrolla un proceso paso a paso. Permiten visualizar las acciones del usuario y del sistema, los puntos de decisión, las bifurcaciones y los posibles caminos alternativos, incluyendo errores o rutas de corrección. A diferencia de los diagramas de estados, que muestran los estados internos del sistema, los diagramas de actividad reflejan la secuencia de operaciones y la lógica del flujo de trabajo.

4.3.4.1. Diagrama de actividad para el entrenamiento del modelo

El flujo comienza cuando el usuario carga un dataset en el sistema. El archivo se valida para comprobar que sea un CSV y que la calidad de los datos sea suficiente. Si el dataset

no cumple los requisitos, se muestra un mensaje de error y se ofrece la posibilidad de corregir los datos en el módulo EDA o, si no es posible, redirigir al módulo de simulación y finalizar el proceso. Si la validación es exitosa, se habilita el botón para continuar.

A continuación, el sistema sugiere un target por defecto, aunque el usuario puede seleccionar cualquier variable como objetivo. Luego se muestran los modelos disponibles según el tipo de problema, recomendando uno por defecto, pero la elección final queda en manos del usuario. Al pulsar "entrenar", el sistema inicia el entrenamiento del modelo. Si se produce algún error durante esta fase, se muestra un mensaje de error y el flujo termina; si el entrenamiento es exitoso, se presentan las métricas de rendimiento.

Finalmente, el usuario puede realizar predicciones manuales y/o descargar el modelo entrenado, pudiendo realizar ambas acciones en el orden que desee antes de finalizar. Este flujo asegura un entrenamiento guiado, seguro y controlado, integrando validaciones, recomendaciones y rutas alternativas para errores o datos insuficientes.

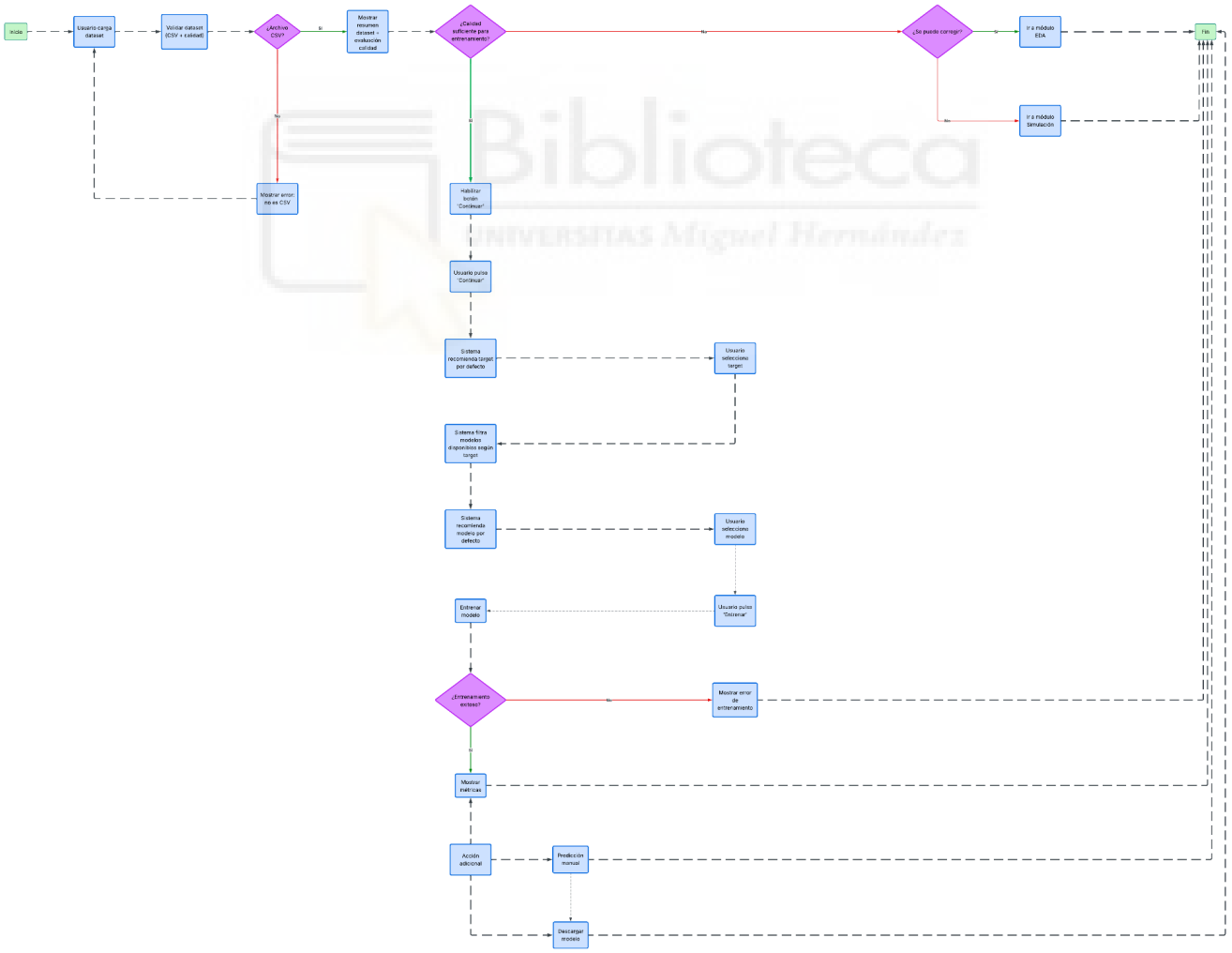


Figura 4.10. Diagrama de actividad entrenamiento de modelos

4.3.5.- Diseño de la interfaz gráfica

Antes de la implementación, se realizaron esbozos y croquis de las pantallas de la aplicación con el objetivo de planificar la disposición de los elementos, la organización de la información y el flujo de navegación. Estos diseños preliminares permitieron definir cómo se presentarían los distintos módulos y funcionalidades, asegurando que la experiencia de usuario fuera intuitiva y coherente desde el inicio.

A continuación, se presentan los principales elementos de diseño y los flujos visuales de las interfaces de los módulos de Análisis Exploratorio de Datos (EDA) y de Entrenamiento de Modelos de IA que se desarrollaron al inicio previo al desarrollo.

4.3.5.1 Esbozos de la interfaz gráfica

El diseño inicial del módulo de Análisis Exploratorio de Datos se planteó con una estructura organizada en secciones navegables. La idea principal era que, una vez cargado el dataset, el usuario fuera redirigido a una interfaz estructurada como un menú con diferentes apartados, accesibles mediante botones o pestañas. Este enfoque permitía dividir las funcionalidades en bloques claros y evitar la sobrecarga de información en una única pantalla.

La primera sección visible tras la carga del dataset sería la sección de información general, donde se mostrarían los datos básicos del archivo, como el número de filas y columnas, el tamaño del archivo y un resumen de los tipos de variables. Esta vista inicial tenía como objetivo ofrecer al usuario una comprensión rápida del contenido y la estructura del dataset antes de realizar cualquier análisis más profundo.

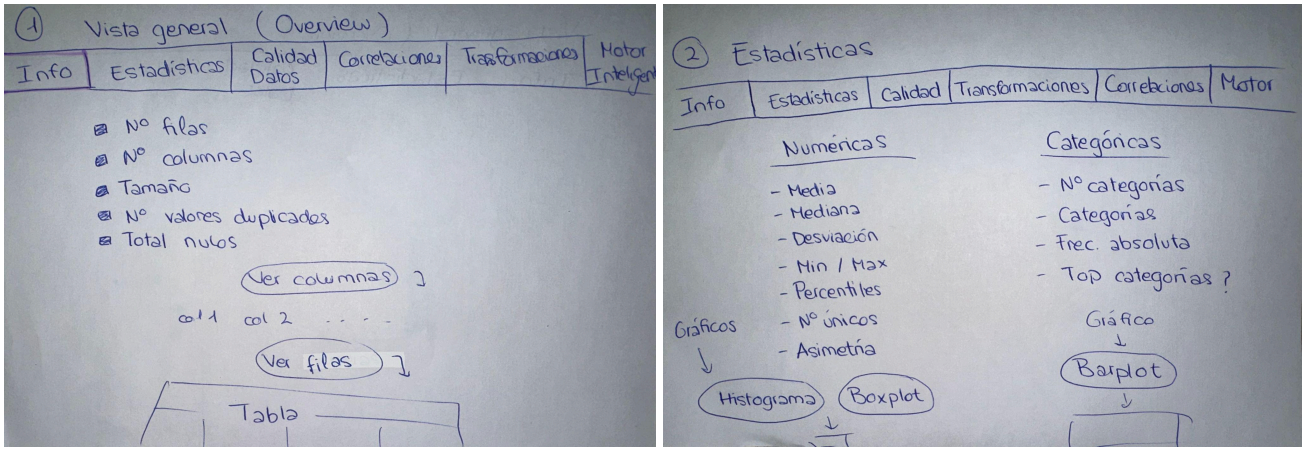


Figura 4.11. Esbozo interfaz secciones Vista general y Estadísticas (Módulo EDA)

4.3.5.2 Diseño del flujo

Además de la realización de esbozos de la interfaz, también se elaboraron esbozos para plantear el flujo que debía seguir el proceso en cada módulo.Este esquema inicial permitió definir de forma conceptual las distintas etapas del entrenamiento antes de su implementación, organizando la secuencia de acciones y decisiones que estructurarían el módulo.

El diagrama presentado refleja esta primera visión del flujo de entrenamiento, que posteriormente fue refinada durante el desarrollo, pero que sirvió como base para estructurar la lógica interna y la experiencia de usuario del sistema.

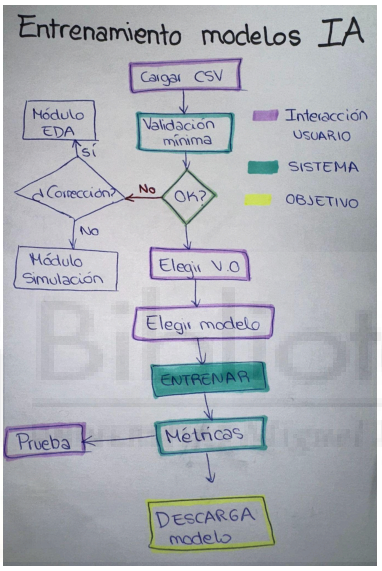


Figura 4.12. Esbozo del flujo del módulo Entrenamiento

4.4.- IMPLEMENTACIÓN

En este apartado se presenta la implementación final de la aplicación, mostrando las principales pantallas desarrolladas y explicando brevemente las funcionalidades más relevantes de cada módulo. A través de capturas de pantalla comentadas se pretende ilustrar cómo se materializaron los diseños y flujos previamente definidos, así como la estructura final de la interfaz.

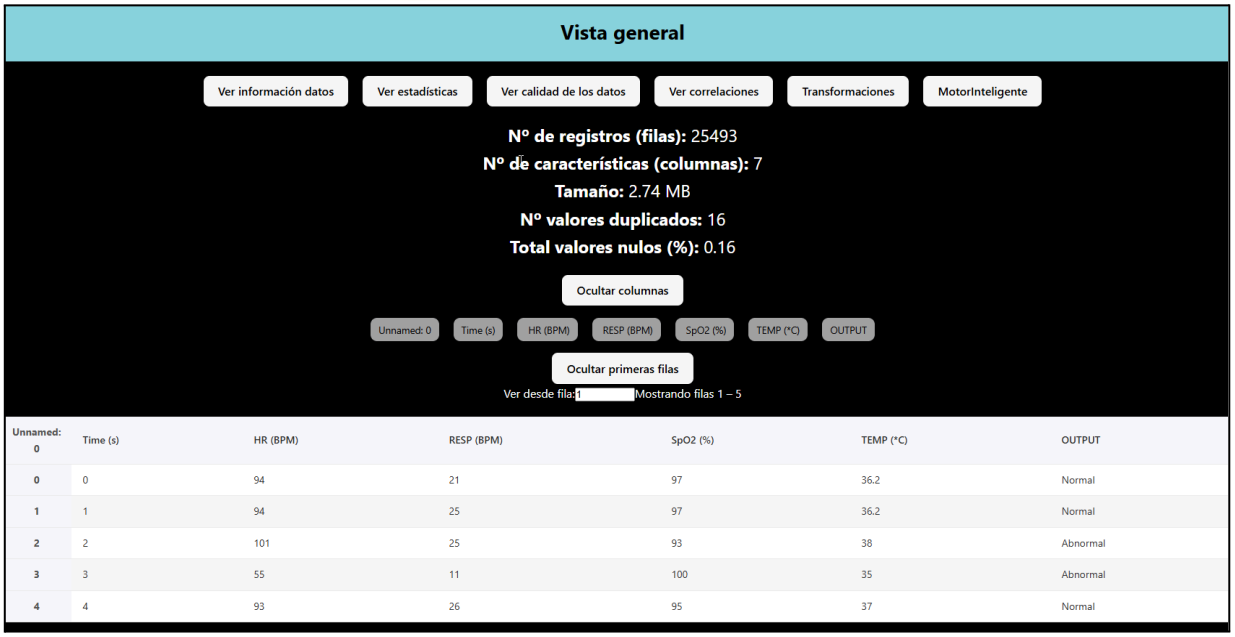
Asimismo, se destacan aquellos elementos que resultan especialmente relevantes desde el punto de vista funcional, centrándose en la experiencia de usuario y en la integración de los distintos módulos del sistema.

4.4.1. Módulo de Análisis Exploratorio de Datos (EDA)

La primera funcionalidad que encuentra el usuario es la pantalla de carga del dataset. En esta vista se permite subir archivos en formato CSV, iniciando así el proceso de análisis. La interfaz ha sido diseñada de forma sencilla y clara, facilitando la comprensión del paso inicial del flujo.

Figura 4.13. Pantalla principal Análisis Exploratorio de Datos

Una vez cargado el dataset, el usuario accede al menú principal del módulo EDA, donde se muestran distintas secciones navegables. Estas permiten consultar información general del dataset, estadísticas descriptivas y diferentes visualizaciones gráficas.



Unnamed: 0	Time (s)	HR (BPM)	RESP (BPM)	SpO2 (%)	TEMP (°C)	OUTPUT
0	0	94	21	97	36.2	Normal
1	1	94	25	97	36.2	Normal
2	2	101	25	93	38	Abnormal
3	3	55	11	100	35	Abnormal
4	4	93	26	95	37	Normal

Figura 4.14. Vista general Análisis exploratorio de datos EDA

La primera sección que se muestra es la "**Vista general**", que tiene como objetivo situar al usuario en contexto con el dataset cargado. En esta pantalla inicial se proporciona un resumen rápido de la información básica: número de filas y columnas, tipos de datos y tamaño del archivo. Además, se incluyen botones que permiten visualizar los nombres de las columnas o explorar las filas de la base de datos, ofreciendo un acceso rápido a los elementos esenciales del dataset.

A continuación, el usuario puede navegar hacia otras secciones del módulo, como la de "**Estadísticas descriptivas**" o la de "**Visualizaciones**". Por ejemplo, en la sección de "**Correlaciones**" se presenta una matriz que permite identificar relaciones entre variables numéricas, ofreciendo una visión clara de la estructura interna de los datos.

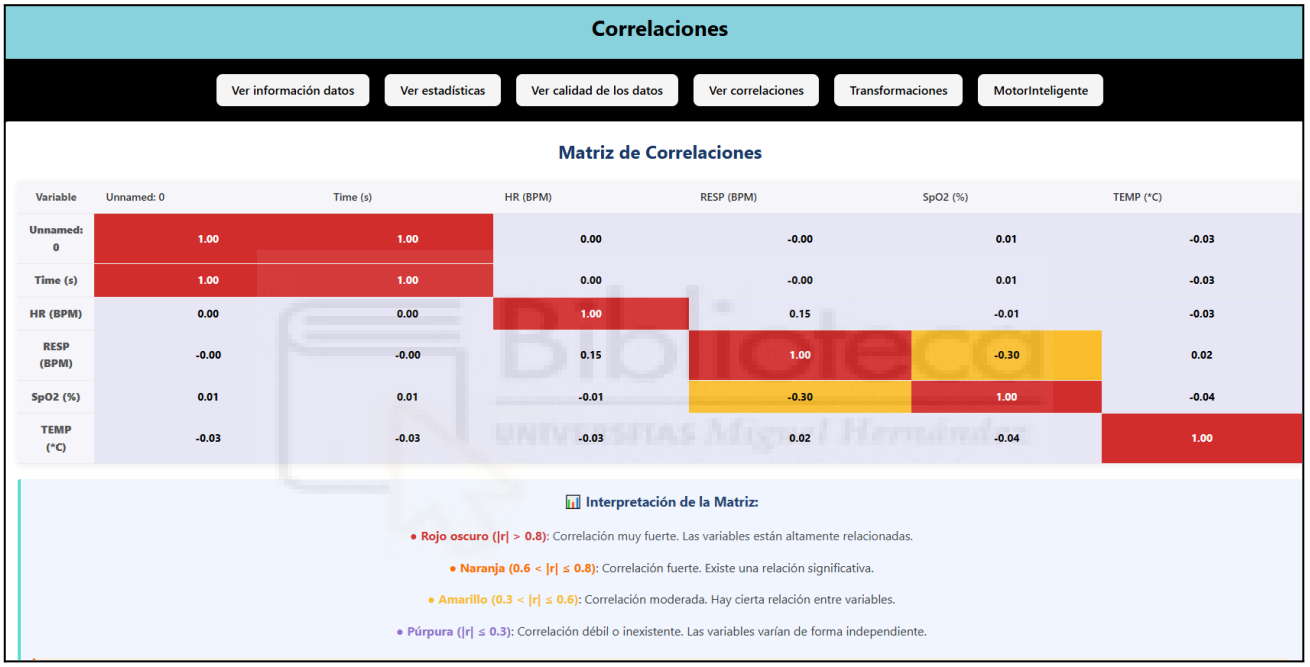


Figura 4.15. Sección correlaciones Análisis Exploratorio EDA

Este módulo también incluye herramientas básicas de limpieza y transformación de datos, así como la posibilidad de descargar el dataset procesado, completando así el ciclo de análisis exploratorio.

4.4.2. Módulo de entrenamiento de modelos de IA

El flujo inicial de este módulo también comienza con la carga del dataset, siguiendo un procedimiento similar al módulo EDA. Una vez cargado, el sistema valida que el archivo sea CSV y evalúa la calidad de los datos. En función de esta evaluación, se presenta al usuario un resumen breve de la información del dataset y de los resultados de la evaluación de calidad.

A continuación, el flujo continúa con la selección de la variable objetivo (target). El sistema sugiere automáticamente un target, pero la decisión final recae siempre en el usuario. Seguidamente, se muestran los modelos disponibles, recomendando uno por defecto en función del tipo de problema detectado (clasificación o regresión). Una vez seleccionado el modelo, el usuario inicia el entrenamiento.

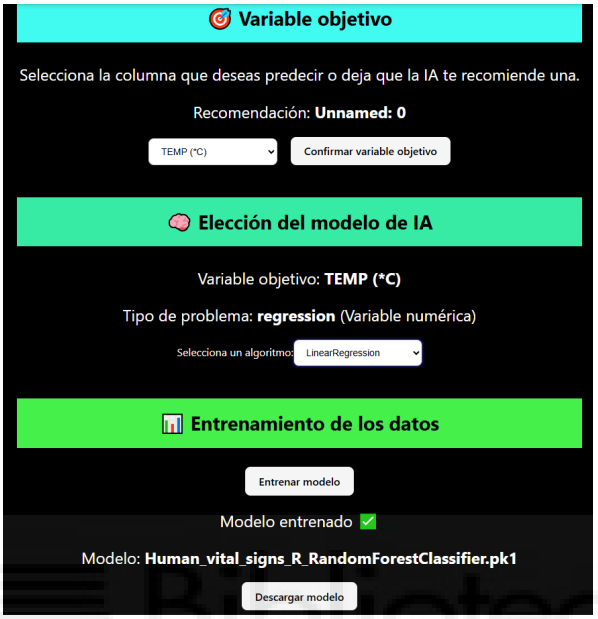


Figura 4.16. Selección de variable objetivo, modelo de IA y entrenamiento de los datos

Si durante el entrenamiento se produce algún error, el sistema muestra un mensaje de alerta y el flujo termina. En caso contrario, al finalizar, se presentan las métricas de rendimiento del modelo y se habilitan las acciones finales: realizar predicciones manuales y descargar el modelo entrenado, que constituyen los resultados más relevantes para el usuario.

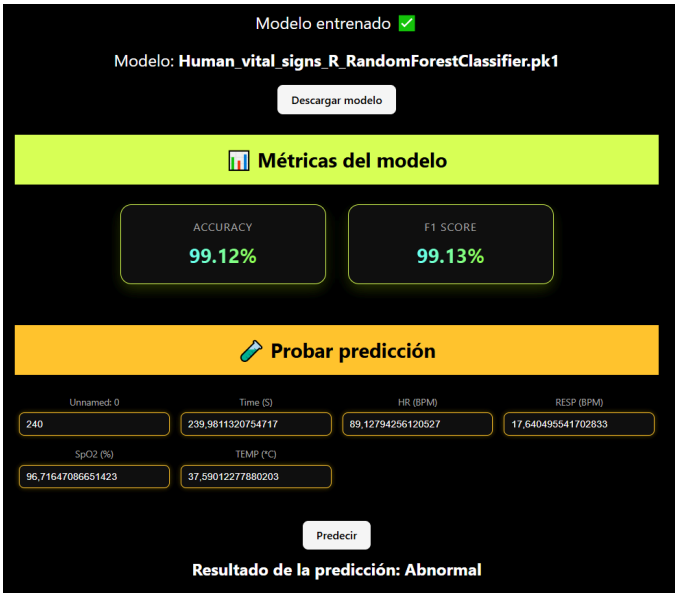


Figura 4.17. Rendimiento y predicción del modelo generado

4.4.3. Módulo de simulación de datos

El módulo de simulación permite generar datasets desde cero mediante la definición de variables personalizadas. Inicialmente, se presenta una pantalla donde el usuario puede comenzar a configurar las variables que compondrán el dataset.

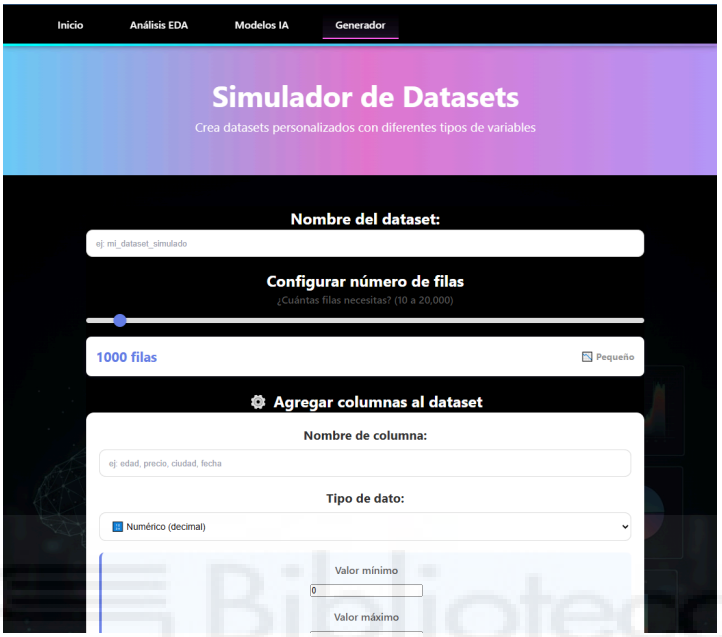


Figura 4.18. Interfaz simulador de datasets

A medida que se añaden variables, el sistema muestra las variables que se han añadido dando la posibilidad de eliminarlas.

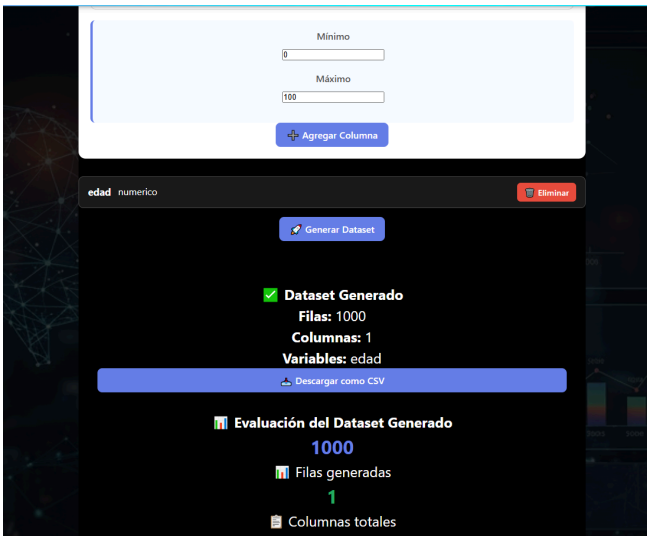


Figura 4.19. Adición de variables y generación del dataset

Cuando el usuario esté satisfecho con las variables, el sistema genera el dataset simulado y permite su descarga en formato CSV, cerrando así el flujo de creación de datos.

Capítulo 5
Conclusiones y trabajo futuro

5.1.- CONCLUSIONES

El proyecto desarrollado ha logrado cumplir con su objetivo principal de proporcionar una aplicación que permite a profesionales de distintos ámbitos explorar, interpretar y aprovechar datos mediante técnicas de inteligencia artificial, sin requerir conocimientos especializados en la materia. La aplicación ofrece un flujo guiado que acompaña al usuario desde la carga y análisis de los datos hasta la obtención de resultados y predicciones, asegurando accesibilidad y facilidad de uso.

En cuanto a los objetivos específicos planteados:

- La interfaz de usuario se ha diseñado de manera intuitiva, centrada en la experiencia del usuario, y facilita la navegación por los distintos módulos.

- El flujo guiado permite realizar análisis exploratorios de datos (EDA), entrenar modelos de IA y generar datasets simulados, cumpliendo con las funcionalidades básicas necesarias para un análisis inicial y la obtención de resultados interpretables.

A nivel personal, el desarrollo de este proyecto ha permitido adquirir competencias prácticas en:

- Desarrollo de aplicaciones que integran técnicas de inteligencia artificial, incluyendo la carga, limpieza y visualización de datos.

- Diseño e integración de APIs y uso de frameworks de desarrollo web como FastAPI, así como de bibliotecas de aprendizaje automático en Python.

- Pensamiento orientado al usuario, enfocándose en accesibilidad, simplicidad y utilidad de la herramienta para profesionales no técnicos.

Si bien el sistema cumple los objetivos esenciales, se identifican áreas de mejora:

- La capacidad de análisis puede ampliarse para permitir estudios más detallados y avanzados de los datasets.

- Sería recomendable realizar pruebas de usabilidad con usuarios reales para evaluar la experiencia de uso y la comprensión del flujo guiado.

- La limitación de memoria al entrenar modelos en entornos con recursos limitados representa un desafío, especialmente al trabajar con datasets grandes o técnicas más complejas.

En general, el proyecto ha demostrado ser funcional y útil como herramienta básica de análisis y simulación de datos mediante IA. Además, se han implementado mecanismos que previenen errores durante el entrenamiento, asegurando que el sistema no ofrezca modelos inadecuados. Esto constituye un avance significativo en la combinación de análisis de datos, simulación y predicción accesible para usuarios no expertos.

5.2.- POSIBLES DESARROLLOS FUTUROS

El sistema desarrollado ofrece una base funcional sólida, pero existen múltiples posibilidades de ampliación que podrían dotar a la aplicación de nuevas funcionalidades y mejorar la experiencia del usuario:

1. **Análisis de datos más especializados:** Se podría incorporar análisis avanzados, como clustering, reducción de dimensionalidad, análisis de series temporales o técnicas estadísticas más complejas, para permitir estudios más detallados y profundos de los datasets.

2. **Ampliación de modelos de IA:** Sería posible incluir nuevos modelos de aprendizaje automático y técnicas de deep learning, lo que ampliaría la capacidad predictiva y permitiría abordar problemas más variados o complejos.

3. **Escalabilidad y almacenamiento:** Integrar servicios en la nube permitiría manejar datasets más grandes y persistir información de manera segura. Además, en un futuro se podrían implementar **sesiones de usuario**, de manera que los modelos entrenados y los datasets procesados se puedan guardar y recuperar directamente dentro del sistema.

4. **Asistencia guiada y herramientas de soporte:** Se podría desarrollar un flujo de ayuda más completo, con sugerencias automáticas, alertas y explicaciones de cada paso, para acompañar al usuario durante todo el proceso de análisis y entrenamiento de modelos.

5. **Mejoras en la interfaz y visualización:** Añadir paneles interactivos, reportes automáticos y visualizaciones más avanzadas ayudaría a que el usuario comprenda mejor los resultados y tome decisiones más informadas. Además, se podría incorporar la generación de informes listos para descargar.

Estas posibles mejoras representan un camino natural de evolución del proyecto, potenciando tanto la funcionalidad como la accesibilidad de la aplicación y su utilidad para profesionales sin experiencia en inteligencia artificial.

Bibliografía

[1] Guía práctica de introducción al análisis exploratorio de datos en Python
https://datos.gob.es/es/conocimiento/guia-practica-de-introduccion-al-analisis-exploratorio-de-datos-en-python
Gobierno de España (Noviembre 2024)

[2] ¿Qué es el análisis exploratorio de datos (EDA)?
https://www.ibm.com/es-es/think/topics/exploratory-data-analysis
IBM

[3] ¿Qué es el entrenamiento de modelos?
https://www.ibm.com/es-es/think/topics/model-training
IBM

[4] Datos sintéticos: ¿Qué son y para qué se usan?
https://datos.gob.es/es/conocimiento/datos-sinteticos-que-son-y-para-que-se-usan
Gobierno de España (Octubre 2023)

[5] Datos sintéticos y protección de datos
https://www.aepd.es/prensa-y-comunicacion/blog/datos-sinteticos-y-proteccion-de-datos
Agencia Española de Protección de Datos (Noviembre 2023)

[6] ¿Qué son los datos sintéticos?
https://www.ibm.com/es-es/think/topics/synthetic-data
IBM

[7] A literature review on automated machine learning
https://link.springer.com/article/10.1007/s10462-025-11397-2
Springer Nature (Noviembre 2025)

[8] Data-centric Artificial Intelligence: A Survey
Zha et al.
https://arxiv.org/abs/2303.10158
arXiv (Marzo 2023)

[9] Synthetic data generation by diffusion models
Jun Zhu
https://academic.oup.com/nsr/article/11/8/nwae276/7740777
National Science Review (Agosto 2024)

[10] Reimagining Synthetic Tabular Data Generation through Data-Centric AI: A Comprehensive Benchmark
Hansen et al.
https://arxiv.org/abs/2310.16981
arXiv (Octubre 2023)

[11] Automated machine learning: past, present and future
https://link.springer.com/article/10.1007/s10462-024-10726-1
Springer Nature (Abril 2024)

[12] pandas-profiling
https://pypi.org/project/pandas-profiling/
PyPI, Python Package Index (Enero 2023)

[13] YDataProfiling
https://docs.profiling.ydata.ai/

[14] Usando Pandas Profiling Para Acelerar Nuestra Exploración de Datos
https://www.datasource.ai/uploads/d8bd6d716a55e75759045076654f51b3.html

[15] sweetviz 2.3.1
https://pypi.org/project/sweetviz/
PyPI (Noviembre 2023)

[16] SweetViz | Automated Exploratory Data Analysis (EDA)
https://www.geeksforgeeks.org/data-analysis/sweetviz-automated-exploratory-data-analysis-eda/

[17] AUTOVIZ
https://www.autoviz.ai/

[18] Dtale
Andrew Schonfeld
https://github.com/man-group/dtale

[19] Dtale
https://pypi.org/project/dtale/

[20] Exploratory Data Analysis (EDA)
https://docs.datarobot.com/en/docs/reference/data-ref/eda-explained.html
DataRobot

[21] AutoML
https://cloud.google.com/automl
GoogleCloud

[22] H2O Driverless AI
https://h2o.ai/platform/ai-cloud/make/h2o-driverless-ai/
H20.ai (2026)

[23] Azure
https://azure.microsoft.com/
Microsoft (2026)

[24] Synthpop
https://www.synthpop.org.uk/

[25] Synthetic Data Generation for Agentic AI
https://www.nvidia.com/en-us/use-cases/synthetic-data-generation-for-agentic-ai/
nvidia

[26] Welcome to the SDV!
https://docs.sdv.dev/sdv
Synthetic Data Vault (Diciembre 2025)

[27] python
https://www.python.org/

[28] FastAPI
https://fastapi.tiangolo.com/

[29] Generalidades del protocolo HTTP
https://developer.mozilla.org/es/docs/Web/HTTP/Guides/Overview
mdn (Julio 2025)

[30] ¿Qué es JSON?
https://www.oracle.com/es/database/what-is-json/
Oracle (Abril 2024)

[31] React
https://es.react.dev/

[32] Escribir marcado con JSX
https://es.react.dev/learn/writing-markup-with-jsx

[33] HTML: Lenguaje de etiquetas de hipertexto
https://developer.mozilla.org/es/docs/Web/HTML
MDN (Junio 2025)

[34] CSS Introduction
https://www.w3schools.com/css/css_intro.asp
W3Schools

[35] NumPy
https://numpy.org/
numpy

[36] Pandas
https://pandas.pydata.org/
pandas

[37] scikit-learn
https://scikit-learn.org/

[38] OrdinalEncoder
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OrdinalEncoder.html
scikit-learn

[39] OneHotEncoder
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
scikit-learn

[40] StandardScaler
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
scikit-learn

[41] MinMaxScaler
https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
scikit-learn

[42] train_test_split
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
scikit-learn

[43] cross_val_score
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html
scikit-learn

[44] SelectKBest
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
scikit-learn

[45] mutual_info_classif
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_classif.html
scikit-learn

[46] mutual_info_regression
https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.mutual_info_regression.html
scikit-learn

[47] RandomForestClassifier
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
scikit-learn

[48] LinearRegression
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
scikit-learn

[49] RandomForestRegressor
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

[50] GridSearchCV
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html
scikit-learn

[51] RandomizedSearchCV
https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
scikit-learn

[52] accuracy_score
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html
scikit-learn

[53] confusion_matrix
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html

scikit-learn

[54] r2_score
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html
scikit-learn

[55] mean_squared_error
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html
scikit-learn

[56] mean_absolute_error
https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html
scikit-learn

[57] Matplotlib
https://matplotlib.org/

[58] Seaborn
https://seaborn.pydata.org/

[59] Joblib
https://joblib.readthedocs.io/en/latest/

[60] Qué es la arquitectura cliente servidor y cómo funciona
https://www.daemon4.com/empresa/noticias/arquitectura-cliente-servidor/
Daemon4 (Julio 2024)

[61] Visual Studio Code documentation
https://code.visualstudio.com/docs
Visual Studio Code

[62] pip
https://pypi.org/project/pip/
PyPI (Enero 2026)

[63] uvicorn
https://uvicorn.dev/
Uvicorn

[64] About npm
https://docs.npmjs.com/about-npm
npm Docs (Octubre 2023)

[65] VITE
https://vite.dev/guide/

[66] Iterative Incremental Model in Designing System
https://www.geeksforgeeks.org/system-design/iterative-incremental-model-in-designing-system/
geeksforgeeks (Marzo 2024)

[67] Ingeniería del Software (2792)
http://umh2792.edu.umh.es/material/teoria/
UMH (Octubre 2014)