



**Universitat Autònoma de Barcelona**

**Escola de Enginyeria**

**Grado en Ingeniería de Datos**

**Sistema Interactivo para el Seguimiento de Salud y Predicción  
de Hábitos**

**Alumno: Pau Montesinos Cáliz**

**Tutor: Oriol Cortés Comellas**

**Fecha: 10/03/2025**

## 1. Introducción

En la actualidad, la salud se ha vuelto un pilar indiscutible dentro de la sociedad debido a los desafíos relacionados con los malos hábitos, como una mala alimentación o la falta de ejercicio, un aumento considerable de la preocupación por las enfermedades tanto crónicas como agudas, o las tendencias de una población a querer cada vez una calidad de vida superior.

Debido a este hecho, las tecnologías con orientación a aspectos de la salud han sufrido un auge desmedido, mayormente debido a esta constante necesidad de evolución de los estándares de vida.

Dentro de las tecnologías que pueden ayudar en el día a día a abordar este tipo de problemas, nos encontramos con aquellas que, a través de los datos de los pacientes, pueden predecir si tienen alguna enfermedad o si, a lo largo del tiempo con, estas medidas, tienen riesgo de padecerlas.

Este trabajo, tiene este mismo objetivo, el desarrollo de una plataforma de salud que pueda ayudar al que lo necesite.

## 2. Objetivos

El objetivo principal de este proyecto es desarrollar una plataforma la cual pueda recoger los hábitos y datos sobre las personas para, posteriormente, hacer un análisis y revisar su estado de salud, dándole la probabilidad de poder tener cierta enfermedad y como podría cambiar sus hábitos en caso de que lo necesite. Para cumplir este objetivo, se establecerán los siguientes puntos clave.

- Identificar uno o varios datasets útiles.
- Revisar y limpiar las impurezas innecesarias de los datasets elegidos.
- Utilizar los datos para entrenar varios modelos de predicción tanto de Machine Learning como de Deep Learning.
- Evaluar los modelos entrenados para poder escoger el que más se ajuste al propósito final.
- Crear una plataforma para poder utilizar el modelo escogido anteriormente y que, así, el cliente final pueda hacer uso de este modelo de forma fácil e intuitiva.

### 3. Modelos y metodología de estudio

Para realizar las predicciones que se requieren, se probarán distintos modelos, tanto de Machine Learning como de Deep Learning, para buscar entre ellos cual nos da los mejores resultados. Aunque la lista puede hacerse más grande en un futuro, los modelos que inicialmente se probarán son:

- Machine Learning
  - Regresión logística: Aún ser el más básico de los modelos, tarda poco en entrenar y puede llegar a sorprender.
  - Random Forest: Superior a Decision Trees ya que evita el overfitting de estos.
  - Gradient Boosting Decision Trees: Este es un modelo basado en Decision Trees que son potenciados por gradientes. Es un algoritmo veloz, eficaz y escalable, por lo que su uso puede ofrecernos buenos resultados.
- Deep Learning
  - Multilayer Perceptron: Funciona bien con grandes cantidades de datos con patrones complejos no lineales
  - Sparse Attention-Based Neural Networks: Es interesante su uso ya que en cada capa de atención solo escoge las características que le parecen más importantes.

Ejecutados los modelos, es importante tener en cuenta como se van a evaluar. En este caso estamos intentando predecir enfermedades médicas. Podemos ver dos problemas con los modelos, que estos den falsos positivos o que estos den falsos negativos.

Si seguimos por esta rama, un tratamiento aplicado a un falso positivo se podría arreglar, más si no se le aplica este tratamiento a un falso negativo, la persona podría acabar falleciendo. Por esto, lo que intentaremos reducir es este número de falsos negativos. En este contexto, por tanto, escogeremos la métrica de recall (sensibilidad), que nos dirá la relación de falsos negativos en comparación de todos los negativos y, en un ultima instancia, si toda va bien, la accuracy (exactitud), para reducir el número de falsos totales.

### 4. Alcance

El objetivo de este trabajo es ver hasta qué punto se pueden predecir algunas enfermedades revisando los hábitos y algunos parámetros más sobre cierto número

de personas. Hasta este punto, parece clara la cosa, más habría que establecer los límites de este proyecto.

En primer lugar, este será un proyecto meramente educativo, no será oficial ni podrá ser considerado definitivo para diagnosticar ciertas enfermedades.

En segundo lugar, uno de los datasets proviene de datos extraídos de Estados Unidos, por lo que extrapolar esta información a otros países con un nivel cultural y socioeconómico diferentes podría ser un verdadero problema.

Por último, la limitación de recursos, ya sea tiempo o capacidad de entrenamiento, podría tener influencia en el modelo a escoger finalmente y, por tanto, en los resultados finales.

## **5. Herramientas**

Para desarrollar el proyecto se hará uso de diferentes herramientas para poder cumplir todos los objetivos. Estas herramientas serán en su mayoría en Python y ayudarán en todo el proceso.

- Pandas: se encargará de la lectura y manipulación de datasets.
- Scikit-learn: Proporciona las herramientas adecuadas para implementar los modelos de Machine Learning.
- HTML y CSS: Para crear la aplicación web, se usarán HTML y CSS como base.
- Flask: A través de ella podremos renderizar los archivos HTML y CSS para poder crear una aplicación web sencilla.
- Sqlite3: Con ella se creará la base de datos usada en el proyecto.
- Plotly: Con esta biblioteca de Python visualizaremos los datos que necesitemos.
- Pytorch/Keras: Se usarán en la creación de modelos de Deep Learning en el caso que proceda.

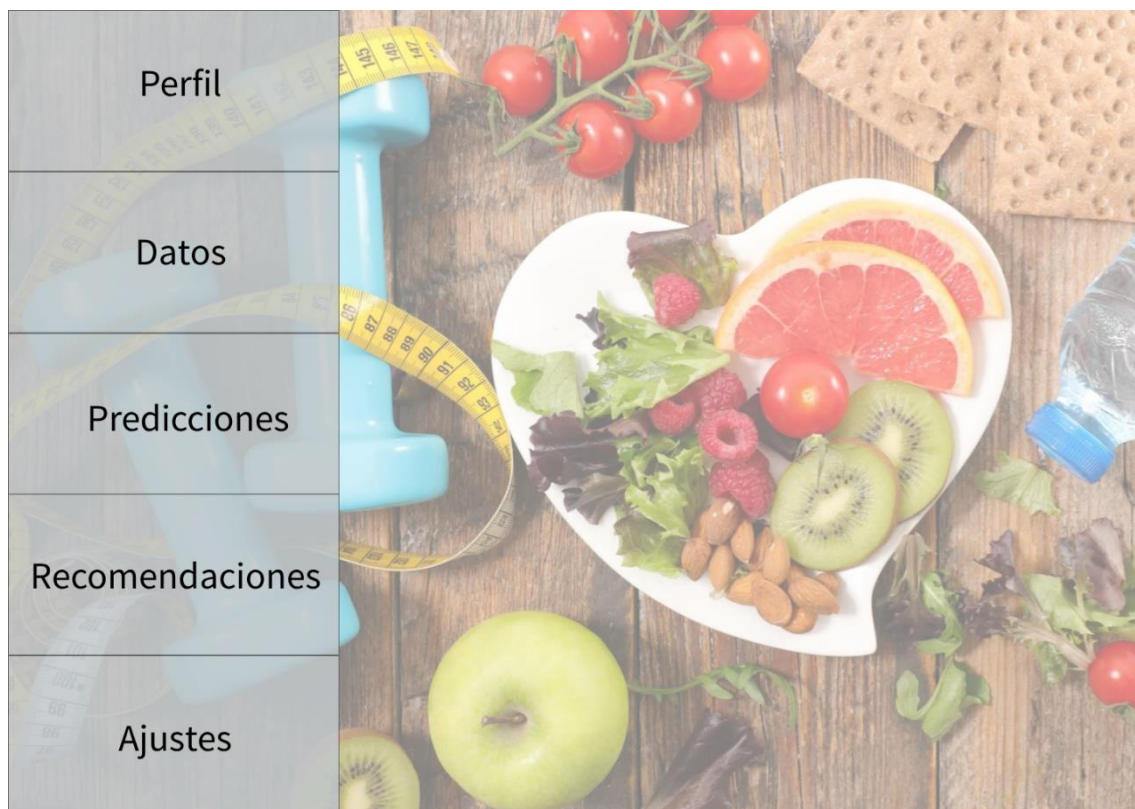
## **6. Mini Diseño de producto final**

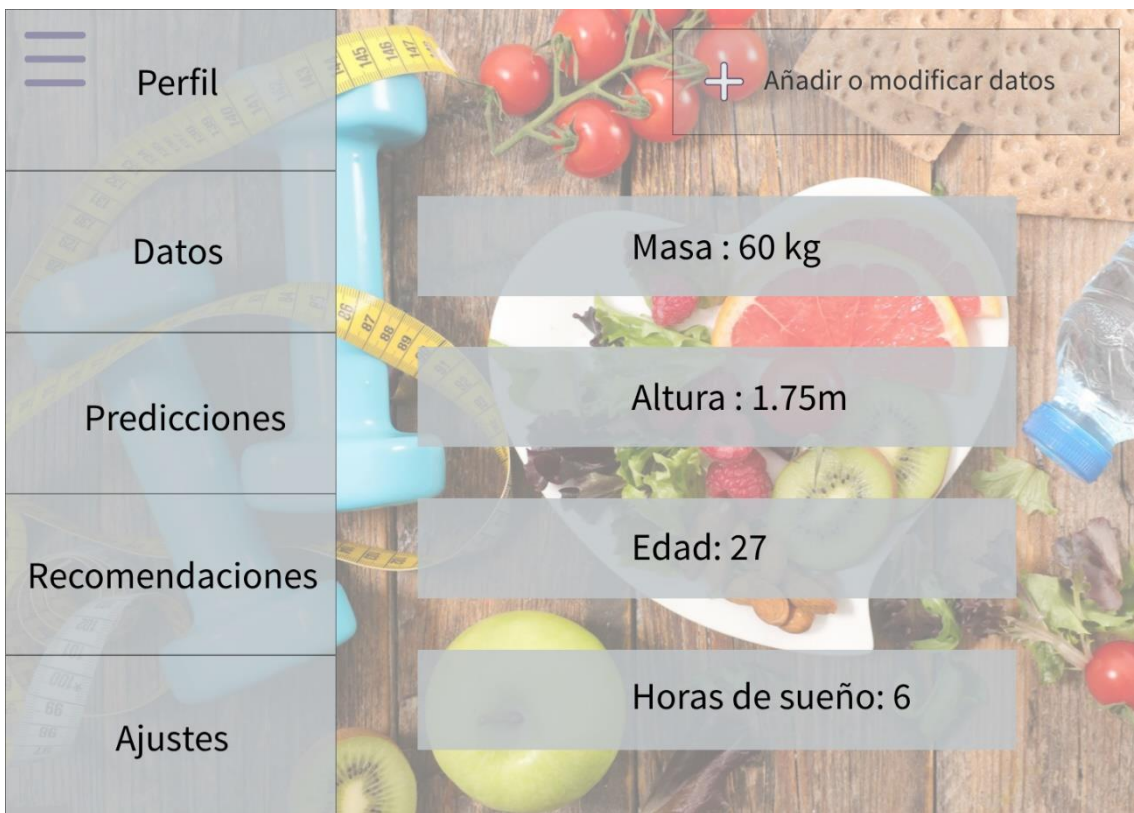
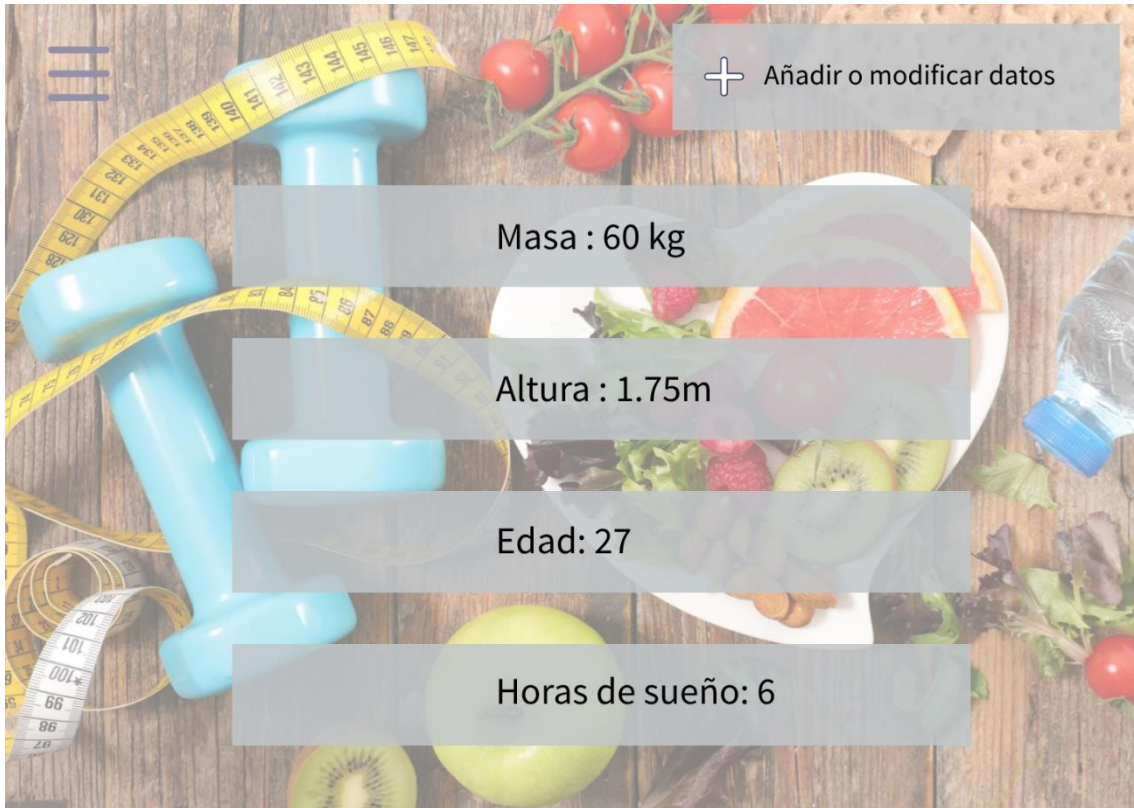
El diseño inicial, el cual puede sufrir cambios sustanciales a lo largo del tiempo, se encuentra dentro del siguiente proyecto desarrollado con la plataforma PenPot.

## Diseño Inicial

En las siguientes imágenes se puede ver el producto igual, pero, si se quiere interactuar con él, será necesario entrar al link establecido anteriormente.



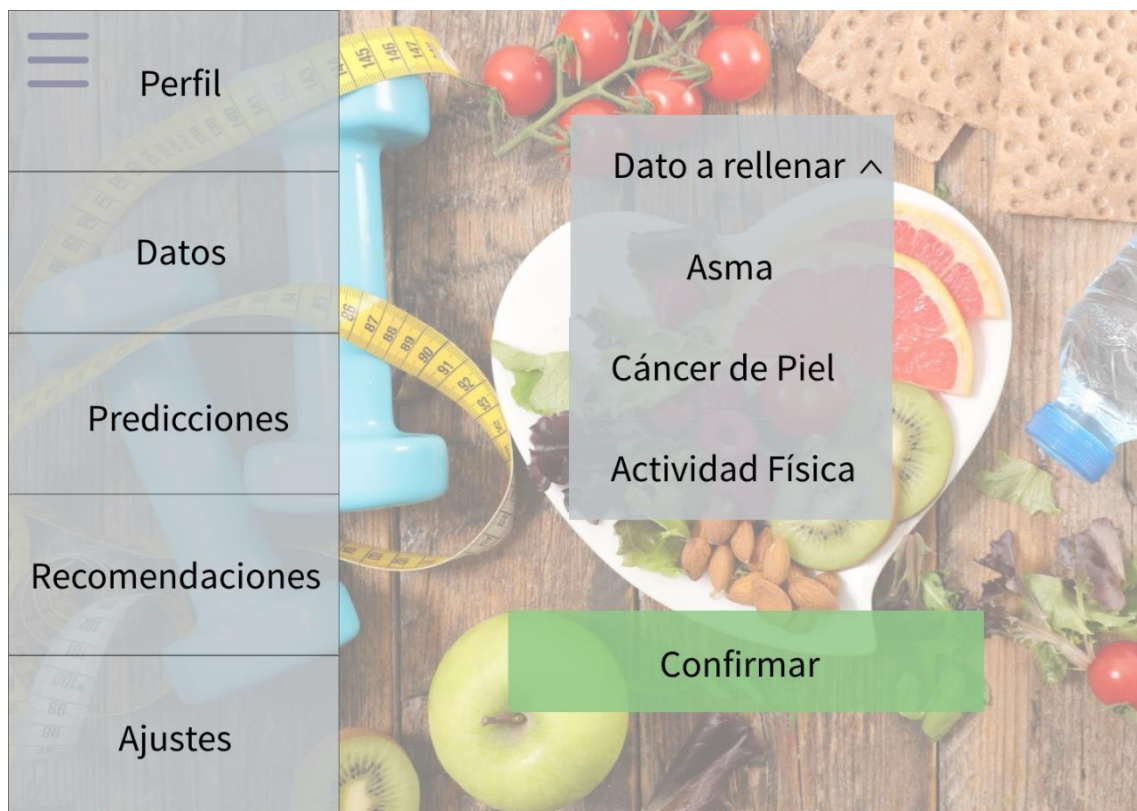
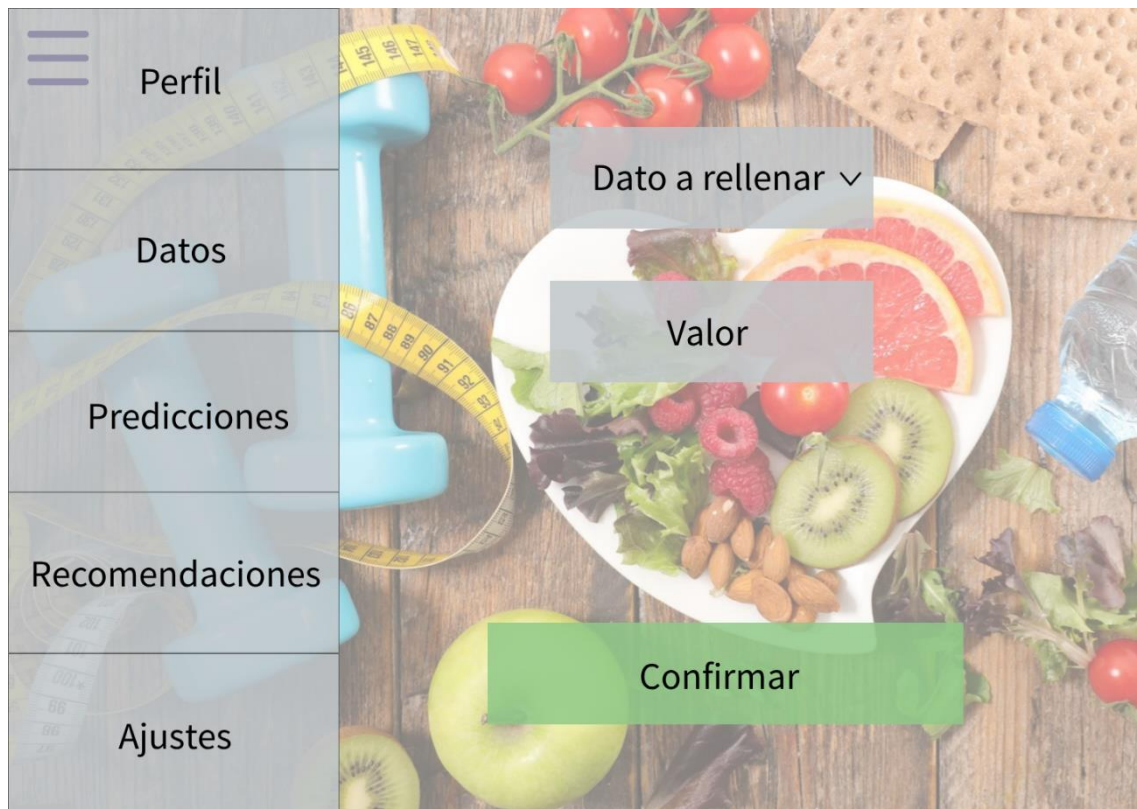


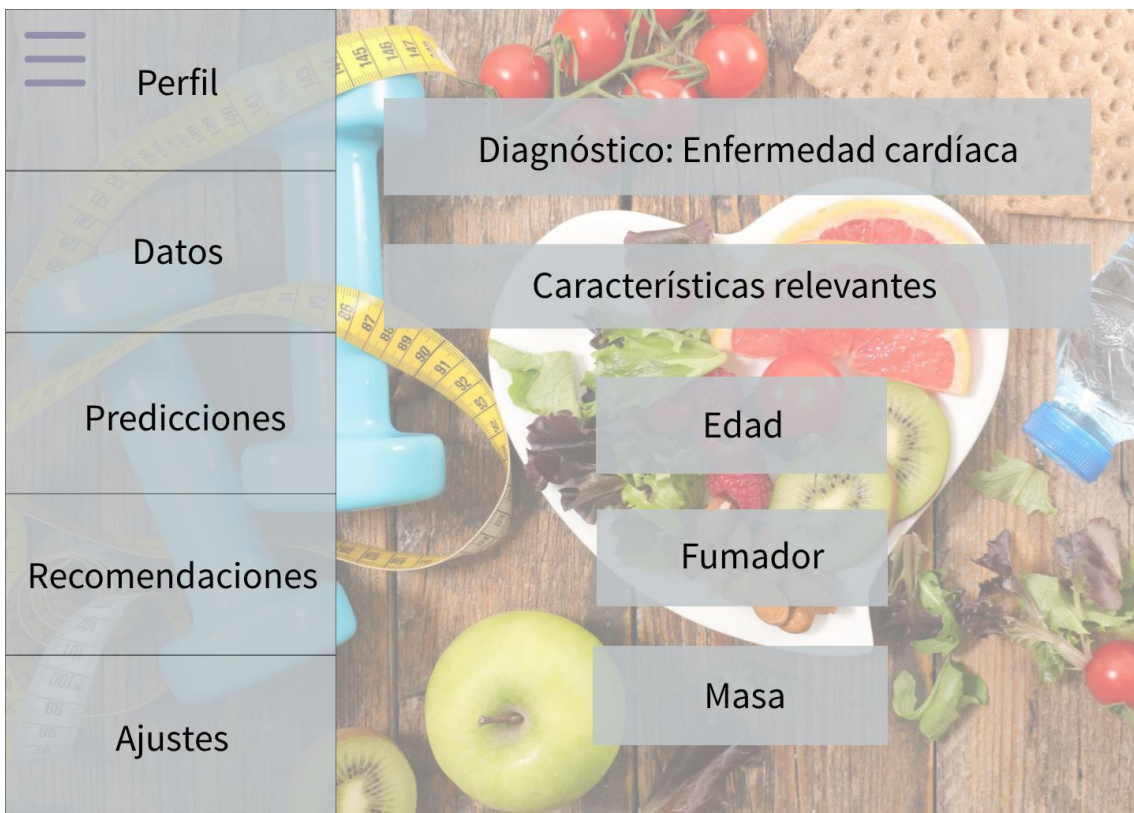














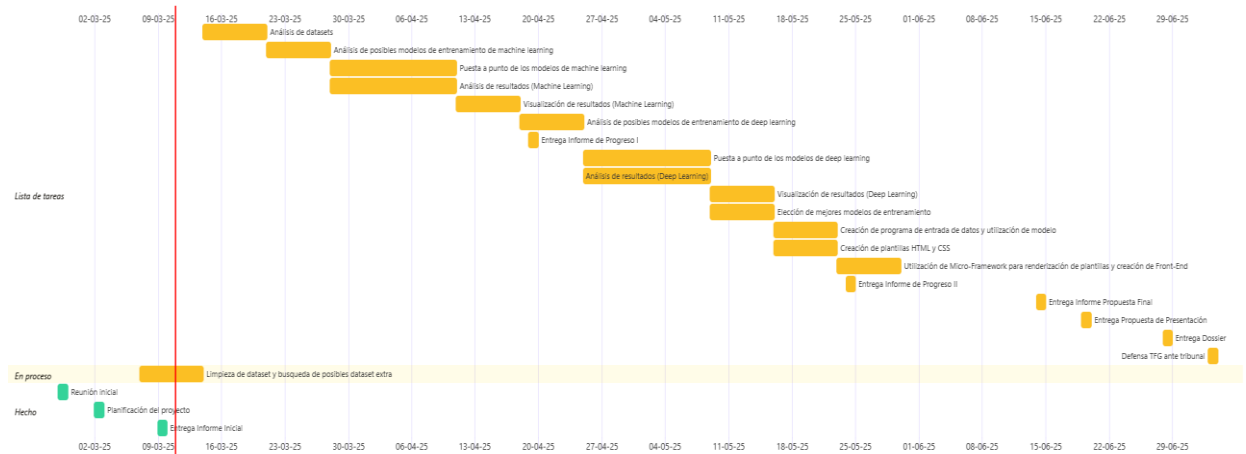


## 7. RoadMap

Para realizar el proyecto, se seguirán las tareas ya propuestas dentro del siguiente tablero de la aplicación Trello, siguiendo la metodología Kanban.

[Tablero Trello](#)

En este diagrama de Gantt también se define el trayecto que se seguirá para realizar el proyecto



## 8. Bibliografía

[1] Pandas, “pandas - Python Data Analysis Library,” *Pydata.org*, 2024.

<https://pandas.pydata.org/> (accessed Feb. 28, 2025).

[2] S. Developers, “scikit-learn: machine learning in Python — scikit-learn 1.6.1 documentation,” *Scikit-learn.org*, 2025. <https://scikit-learn.org/stable/> (accessed Feb. 28, 2025).

[3] WHATWG, “HTML Standard,” *Whatwg.org*, 2025. <https://html.spec.whatwg.org/> (accessed Mar. 02, 2025).

[4] W3C, “Cascading Style Sheets,” *W3.org*, 2024. <https://www.w3.org/Style/CSS/> (accessed Mar. 02, 2025).

[5] P. Projects, “Welcome to Flask — Flask Documentation (3.1.x),” *Palletsprojects.com*, 2025. <https://flask.palletsprojects.com/en/stable/> (accessed Mar. 03, 2025).

[6] Python Software Foundation, “sqlite3 — DB-API 2.0 interface for SQLite databases,” *Python documentation*, 2025. <https://docs.python.org/3/library/sqlite3.html>

(accessed Mar. 03, 2025).

[7] Plotly, “Plotly,” *Plotly.com*, 2025. <https://plotly.com/python/> (accessed Mar. 04, 2025).

[8] K. Team, “Keras documentation: Keras 3 API documentation,” *Keras.io*, 2025. <https://keras.io/api/> (accessed Mar. 04, 2025).

[9] P. Team, “PyTorch documentation — PyTorch 2.6 documentation,” *Pytorch.org*, 2023. <https://pytorch.org/docs/stable/index.html> (accessed Mar. 05, 2025).

[10] IBM and E. K. Erika Russi, “XGBoost,” *Ibm.com*, May 09, 2024. <https://www.ibm.com/es-es/think/topics/xgboost> (accessed Mar. 06, 2025).

[11] Gaurav, “An Introduction to Gradient Boosting Decision Trees - Machine Learning Plus,” *Machine Learning Plus*, Jun. 12, 2021. <https://www.machinelearningplus.com/machine-learning/an-introduction-to-gradient-boosting-decision-trees/> (accessed Mar. 06, 2025).

[12] Z. Xiang, Z. Zhang, and Q. Liu, “Sparse Attention-Based Neural Networks for Code Classification,” *arXiv.org*, 2023. <https://arxiv.org/abs/2311.06575> (accessed Mar. 08, 2025).