

**Proyecto Fin de Máster**

Máster en Big Data, Inteligencia Artificial y Data Science

**HealthBite:**

Aplicación con Visión Artificial y Lenguaje Natural para la nutrición personalizada a partir de ingredientes y estado del usuario

Septiembre 2025

**Autor:** Óscar Xu Zhou

Guión:

* Resumen
* 1) Introducción
  + 1.1 Contexto y justificación del proyecto
  + 1.2 Objetivos del proyecto
* 2) Flujo de la aplicación y algoritmos utilizados
* 3) Análisis exploratorio de datos (EDA)
* 4) Creación del modelo YOLO
  + 5.1) Obtención de datasets
  + 5.2) Entrenamiento del modelo
  + 5.3) Evaluación del modelo y ejemplos de predicción
* 5) Creación de modelo NLP
  + 6.1) Obtención de datasets
  + 6.2) Entrenamiento del modelo
  + 6.3) Evaluación del modelo y ejemplos de predicción
* 6) Algoritmo de recomendación y RAG LLM como juez
  + 7.1) Combinación de los modelos YOLO + NLP
  + 7.2) Traducción del dataset utilizado de recetas
  + 7.3) Algoritmo de recomendación por reglas y puntuación
  + 7.4) RAG LLM como juez final y prompt engineering
* 7) Conclusiones y resultados
  + 8.1) Limitaciones actuales
  + 8.2) Trabajo futuro

# **Introducción**

## Contexto y justificación del proyecto

El desperdicio alimenticio es uno de los grandes problemas que afronta la sociedad actual. Se estima que en la Unión Europea los hogares generan un 54% del desperdicio total, lo que equivale a aproximadamente 72 kilos de desperdicio por persona. (Eurostat, 2024). Este fenómeno no solo implica pérdidas económicas, sino también un fuerte impacto ambiental, dado que se desaprovecha toda la energía invertida en la producción y la descomposición de los alimentos en vertederos genera gases de efecto invernadero.

Diversos estudios señalan que el uso cotidiano de la nevera está relacionado con desperdicio alimenticio. Un estudio sociológico descubrió que, en muchos casos, los productos quedan olvidados o quedan ocultos por la posición, causando que caduquen sin ser consumidos (Heidenstrøm & Hebrok, 2020) Por otra parte, otro estudio reveló que el uso de la nevera es asociado a un aumento del 24.35% de desperdicio alimenticio (Longqiang Zhao, Min, Wang, & Yu, 2024). Estos datos evidencian la necesidad de soluciones que ayuden a la optimización del consumo de los alimentos almacenados en la nevera.

Adicionalmente al contexto de desperdicio, la nutrición personalizada se ha consolidado como un campo emergente alcanzando grandes mejoras sobre todo con los últimos avances en inteligencia artificial. Aunque existen aplicaciones móviles que ofrecen recomendaciones dietéticas, una gran mayoría se han enfocado a proporcionar información tras un escaneo códigos de barras o proponer recetas aleatorias o genéricas, mayoritariamente orientadas a dietas saludables o fitness. Estas soluciones rara vez tienen en cuenta los ingredientes realmente disponibles en el hogar ni cubren las necesidades dietéticas específicas del usuario.

Este proyecto propone el desarrollo de HealthBite, una aplicación inteligente basada en visión artificial, algoritmos de lenguaje natural y modelos de lenguaje de gran escala (LLM). Su objetivo es recomendar recetas que, por un lado, prioricen el uso de los ingredientes disponibles en la nevera para reducir el desperdicio alimenticio, y por otro, contribuyan a mejorar la salud dietética del usuario al cubrir carencias nutricionales potenciales inferidas a partir de descripciones de estado anímico/físico. De este modo, la propuesta combina sostenibilidad, innovación tecnológica y bienestar personal en una única solución escalable.

## Objetivos del proyecto

**Objetivo principal:**

Desarrollar una aplicación inteligente capaz de recomendar recetas considerando tanto los ingredientes disponibles en la nevera del usuario como sus posibles deficiencias nutricionales, inferidas a partir de descripciones de su estado anímico o físico. Adicionalmente, será capaz de explicar las razones de cada recomendación.

**Objetivos específicos:**

* Realizar un análisis exploratorio de datos (EDA) sobre resultados de encuestas de consumición alimentario en hogares de España, con el fin de obtener mejor entendimiento sobre patrones y productos más consumidos
* Creación de un modelo de visión artificial (YOLO) para identificar automáticamente los ingredientes disponibles en la nevera a partir de imágenes
* Entrenar un modelo de procesamiento del lenguaje natural (NLP) capaz de clasificar descripciones del estado anímico/físico del usuario en síntomas y asociarlas con posibles deficiencias nutricionales
* Implementar un sistema de traducción automática a gran escala mediante modelos NLP para adaptar el dataset de recetas del inglés al español
* Construir un sistema de puntuación que permita ordenar las recetas en función de su afinidad al contexto específico del usuario (ingredientes disponibles y necesidades nutricionales).
* Desarrollar e integrar un LLM con RAG, ajustado mediante técnicas de prompt engineering, que actúe como juez final para validar que las condiciones de recomendación se cumplen y genere una explicación clara para el usuario.
* Integrar todos los modelos en una aplicación interactiva con Streamlit, lista para ser utilizada en un entorno real por usuarios finales.

# A diagram of a diagram AI-generated content may be incorrect.**Flujo de la aplicación y algoritmos utilizados**

*Imagen 1: Diagrama de flujo de la estructura y sistema de la aplicación*

Tal y como se puede observar en el diagrama de flujo, el proceso hasta llegar a la recomendación personalizada final de recetas pasa por varias fases:

**Fase de inputs:**

En esta fase se obtienen los inputs necesarios para los modelos a utilizar posteriormente:

* Se recogen dos inputs del usuario: una foto de su nevera y una descripción breve de su estado físico/anímico.
* Se carga internamente un dataset de recetas (2+ millones de registros) que servirá como base para las recomendaciones. **Fase de entrenamiento y aplicación de los modelos:**

**Fase de entrenamiento y aplicación de modelos**

En este paso se entrenarán y se aplicarán tres modelos principales:

* **Detección de objetos *(YOLO, You Only Look Once):*** es un modelo de detección de objetos basado en redes convolucionales que predice clases de objetos a partir de cajas delimitadas. Se ha fine-tuneado con fotos curadas de ingredientes en la nevera combinando datos públicos con fotos sintéticas generadas por scripts de Python y data augmentation.
* **NLP para síntomas (base BERT):** se utiliza un modelo *Transformer* capaz de entender descripciones en español y clasificar a partir de del texto de estado síntomas y posibles deficiencias nutricionales. Se ha fine-tuneado con descripciones anímico-físcas con un dataset sintético compilado a partir de fuentes médicas y clínicas
* Por último, también se utiliza el **modelo Marian MT**, modelo encoder-decoder para la traducción masiva del dataset de recetas de más de 2 millones de filas del inglés al español.

**Outputs de los modelos:**

En esta fase, se recogen las salidas de los diferentes modelos:

* Ingredientes disponibles (a partir de la foto de nevera usando YOLO)
* Síntomas y posibles deficiencias en nutrientes (a partir de la descripción)
* Recetas en español (a partir de la traducción masiva)

Estos resultados alimentarán el sistema de puntuación

**Sistema de puntuación:**

Se ha desarrollado un algoritmo que puntúa y hace un ranking de cada receta en base a una serie de métricas. Se evalúan aspectos como: si la receta requiere ingredientes que ya dispone el usuario en la nevera, si la receta contiene ingredientes que ayudarían con la deficiencia de nutrientes del usuario, etc.

**Recomendación final:**

En esta última fase, se toman las 10 recetas mejor puntuadas y un LLM (Large Language Model), con RAG (Retrieval-Augmented Generation), actúa como juez para seleccionar las tres mejores finales. El modelo recupera evidencia relevante (ingredientes detectados, síntomas/deficiencias y fichas de las recetas) y se guía por instrucciones (prompting) con criterios explícitos. Con ese contexto, reevalúa las candidatas y descarta las que incumplen restricciones o requieren ingredientes críticos ausentes. Devuelve las tres recetas elegidas junto con una explicación breve del razonamiento.

# **Análisis exploratorio de datos (EDA)**

Con el objetivo final de construir una aplicación inteligente capaz de recomendar recetas personalizadas, el primer paso es realizar un análisis exploratorio de datos para entender qué categorías de alimentos tiene mayor consumo en la sociedad española. Como para la aplicación se construirá un modelo YOLO capaz de identificar ingredientes en fotografías de neveras, necesitamos priorizar una cantidad manejable de clases. Dado que las combinaciones de ingredientes pueden ser infinitas, esto puede suponer costes computacionales muy elevados a la hora del entrenamiento del modelo. Por ello, se he decidido fijar un tope de 30 clases a entrenar y predecir para equilibrar cobertura y coste computacional. Para no sacrificar calidad el EDA se apoya en resultados de encuestas oficiales para seleccionar los alimentos más consumidos y con mayor crecimiento, de modo que la aplicación aporte valor pese a la escala reducida.

**Obtención de dataset:**

Los datos han sido obtenidos combinando diferentes resultados de encuestas a hogares sobre consumo de alimentos publicados periódicamente (en archivo Excel, anual o semestralmente) por el Ministerio de Agricultura, Pesca y Alimentación (MAPA). Se ha construido un script de Python que recorre cada archivo preprocesando y estandarizando formatos y esquemas y combina todos los archivos en un único DataFrame (para más información, véase el anexo). El dataset incluye métricas como consumo per cápita, gasto per cápita y penetración de categoría (porcentaje de hogares consumidores).

**Depuración y procesamiento realizado:**

Además de estandarizar, se ha realizado una depuración y filtro del set. Los datos publicados contienen muchas filas que no son representativas o de utilidad. Se ha decidido codificar cada fila del dataset para poder filtrar después y obtener solamente filas que son de interés.

En primer lugar, se codifica cada fila basándonos en el nombre original en categorías diferentes del 0 al 3. Solamente nos quedaremos con la categoría 1. A continuación se muestra la lógica de codificación:

* Categoría 0: el nombre es demasiado genérico (ej.: “total alimentación”, “total carnes”).
* Categoría 1: el nombre es representativo (ej.: “Huevos”, “Miel”).
* Categoría 2: el nombre es demasiado específico y puede estar englobado en otra fila (ej.: “Huevos a granel”, “Huevos envasados”).
* Categoría 3: nombres ambiguos que no aportan valor (ej.: “Otras carnes”, “Otras aves”).

En segundo lugar, se refinan algunos nombres para mejor legibilidad (ej: T.Huevos UNDS -> “Huevos”). Por último, dado que hay demasiados nombres únicos, realizar un análisis exploratorio sería difícil. Se ha decidido agrupar diferentes alimentos bajo categorías de la pirámide alimenticia.

Se muestra dos ejemplos de cómo quedarían las filas del dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre original | Categoría codificada | Nombre refinado | Categoría de pirámide alimenticia |
| T.Huevos Unds | 1 | Huevos | Lácteos, Derivados y Huevos |
| Cons.pescado atún | 1 | Conserva atún | Conservas y precocinados |

**Análisis de datos:**

El análisis exploratorio se centra en identificar productos con mayor penetración en hogares, mayor consumo per cápita y mayor crecimiento porcentual (tanto en penetración como en consumo). Con ello cubrimos los alimentos históricamente populares y, a la vez, los que muestran adopción acelerada. A modo ilustrativo se presentan algunas gráficas (para el conjunto completo, véase el anexo, sección XXX).

A graph of different colored bars

AI-generated content may be incorrect.

A graph showing a line

AI-generated content may be incorrect.

*\*El axis y representa el crecimiento porcentual, que para algunos productos es considerablemente alto*

Tras analizar las gráficas y los números, se han utilizado los siguientes criterios para decidir qué clases escoger e integrar dentro del proyecto para este primer prototipo:

* **Criterio de mayor métrica:** Priorizar productos con valores altos en las métricas observadas (penetración, consumo per cápita, crecimiento).
* **Criterio contextual:** Se debe tener en cuenta que las clases serán las que podrá predecir nuestro futuro modelo de detección de objetos en la nevera. Hay ciertos alimentos que, a pesar de tener buenas métricas, probablemente no se almacenarían en la nevera (e.j: conservas no abiertas de lata, pasta, etc.). Estos alimentos se descartan.
* **Criterio transfer-learning:** Favorecer clases con presencia en datasets públicos para aprovechar mejor el aprendizaje por transferencia.

A continuación, se ofrece un resumen de los alimentos seleccionados, en total 30 clases:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Verduras y hortalizas** | **Pescados y mariscos** | **Lácteos, derivados y huevos** | **Carnes** | **Frutas** |
| Tomate  Cebolla  Patata  Lechuga/Endivia  Zanahorias  Calabacines  Pepino  Champiñones  Brocoli  Coliflor | Merluza  Gambas/Langostinos  Mix marisco/molusco  Lubina  Salmón | Leche  Huevos  Yogur  Queso  Mantequilla | Carne pollo  Carne cerdo  Carne vacuno  Salchichas  Carne pavo | Plátano  Aguacate  Sandía  Limón  Manzanas  Aguacate |

Una vez seleccionadas las clases, se han analizado los resultados por grupos socioeconómicos para ver si hay demasiado desbalance en métricas en las clases de productos escogidas a predecir posteriormente en el modelo YOLO. Se han comparado específicamente los porcentajes de penetración en el hogar por poder económico y por la edad de los participantes de la encuesta. Podemos observar en las siguientes gráficas las siguientes conclusiones:

* Todos los productos seleccionados tienen mayor % de penetración en hogares con nivel alto y menor porcentaje en hogares de nivel bajo. Sin embargo, en general no hay demasiada disparidad
* En cuanto a edad, observamos que el porcentaje de penetración también está bastante distribuido por todos los grupos de edad, a excepción de los productos de pescado, que alcanzan solamente un 30% de particulares de menos de 30 años.

En general, las clases escogidas de las diferentes categorías tienen alto porcentaje de consumo y no hay demasiado desbalance entre grupos sociodemográficos.

A graph showing different colored bars

AI-generated content may be incorrect.

A graph showing different colored bars

AI-generated content may be incorrect.

# **Creación del modelo YOLO**

En esta sección se describe la construcción y ajuste de un detector de objetos YOLO para identificar ingredientes en fotografías de neveras. YOLO es un modelo de visión por computador basado en redes neuronales convolucionales que, en una única pasada por la imagen, predice simultáneamente cajas delimitadoras, confianza y clase de cada objeto, permitiendo detección en tiempo real. (Redmon, 2015)

la construcción del set de datos utilizado posteriormente para entrenar el modelo de detección de objetos YOLO (You Only Look Once), un algoritmo de visión computerizada que permite localizar e identificar objetos mediante cajas delimitadas (bounding boxes en inglés).

5.2) Entrenamiento del modelo

* + 5.3) Evaluación del modelo y ejemplos de predicción

## Obtención de datasets

El primer paso para construir el modelo YOLO definitivo ha sido obtener el set de datos. El set de datos definitivo se construirá a partir de una combinación de:

* **Datasets públicos:** Se han conseguido varios datasets con fotografías de ingredientes en la nevera, principalmente descargados de la plataforma *Roboflow (plataforma simlar a Kaggle)*. Los diferentes datasets públicos han sido combinados en uno definitivo, no obstante, previamente se ha estandariza el documento data.yaml donde cada clase se mapea a su id correspondiente para la consistencia.

A pesar de haber datasets públicos, no se disponen de fotografías de todas las 30 clases de alimentos seleccionadas previamente en el análisis exploratorio. Adicionalmente, algunas clases, a pesar de aparecer en algunas fotografías, están poco representadas.

* **Datasets sintéticos:** dado que no existen fotografías de todas las clases necesarias, se ha decidido generar a partir de un script de Python fotos sintéticas de ingredientes en las neveras. El script toma imágenes individuales de ingredientes, les elimina el fondo y las coloca aleatoriamente sobre distintos fondos de neveras vacías para simular fotos realistas que podría haber tomado un usuario. Esta técnica nos permite suplementar los datasets públicos cuyas fotos no incluyen algunas de las clases que necesitamos identificar.

A diagram of a refrigerator

AI-generated content may be incorrect.Se muestra a continuación una representación gráfica del proceso de generación de datos sintéticos.

* En primer lugar, se descargan fotografías de ingredientes que no se han podido encontrar en datasets públicos o que tienen poca representación. Estas fotos son procesadas utilizando la librería *rembg* eliminando los fondos que contengan.
* En segundo lugar, se descargan fotos de neveras vacías
* Una vez listas las fotos de nevera, se delimitan cajas que sirven como coordenadas de áreas donde se le permite al script de Python posicionar ingredientes aleatoriamente respetando sin sobrepasar el grado de solapamiento definido y sin posicionar demasiados ingredientes en la misma área.
* Con todo listo, un script de Python (véase el script en el anexo XXXX) toma diferentes ingredientes sin fondo y los posiciona aleatoriamente en las áreas delimitadas de diferentes fotos de nevera para simular fotos realistas.
* A refrigerator with a few objects falling

  AI-generated content may be incorrect.Por último, para aumentar la diversidad del dataset sintético, se usa la función *augment\_synthetic\_dataset (véase anexo XXX)* que aplica técnicas de aumentación (rotaciones, inversiones, cambios de tono de color, ruido gaussiano, etc.) ya que las imágenes reales tomadas por los usuarios pueden tener ángulos, iluminación, orientaciones diferentes, etc. La técnica de *data augmentation* ayuda a simular situaciones más realistas y también evita que el modelo simplemente memorice patrones.

**A graph of blue bars with black text

AI-generated content may be incorrect.A graph with blue bars

AI-generated content may be incorrect.**Procedemos a comparar gráficamente cómo ha evolucionado el conteo de imágenes en las que aparece cada ingrediente. Podemos observar que antes de la creación del dataset sintético (gráfico izquierdo) había clases con muy poca o nula representación como pepino, limón, sandía, salmón, etc. Tras incorporar el dataset sintético a los datasets públicos, observamos que ahora todas las clases tienen cierta representación.

## Entrenamiento del modelo

Se han entrenado tres modelos diferentes en total. Cabe destacar que todos los modelos entrenados **parten de los pesos prefijos de la base YOLO V11** (small? Nano?, mencionar que implica esto, de cuantos gigabytes estamos hablando, poner descripción de modelo) que han sido preentrenados con el dataset COCO (Common Objects in Context) que contiene más de 330 mil imágenes y alrededor de 1,5 millones de instancias de objetos de utilidad general.

En nuestro caso, se reentrena el modelo para poder adaptarlo a nuestro problema y contexto específicos, la detección de diferentes clases de ingredientes en fotos de nevera. Los tres modelos entrenados han sido los siguientes:

* ***Modelo 1 - 5 epochs sin datos sintéticos*:** Se ha entrenado un modelo base sencillo de tan solo 5 epochs solamente con los datasets públicos (sin el sintético generado) para comprobar cuál sería el rendimiento base.
* ***Modelo 2 - 10 epochs con datos sintéticos*:** Se ha entrenado un modelo de 10 epochs donde se ha incluido un dataset generado de manera sintética para cubrir ingredientes con baja o nula representación.
* ***Modelo 3 - 100 epochs con datos sintéticos*:** Se ha entrenado un modelo de 100 epochs incluyendo el dataset sintético y con tamaño de imagen 768.

Los detalles específicos de entrenamiento como los parámetros se pueden observar en el anexo XXX.

Añadir info sobre:  **Especificación del modelo**: versión exacta (p. ej., YOLOv11 **n/s/m/l/x**), *commit*/release y archivo de **config** usado.

 **Splits y datos**: nº de imágenes por split (**train/val/test**), **30 clases**, criterio de partición (estratificado por clase/condición), y % de **sintético vs. público** en cada experimento.

## Evaluación del modelo y ejemplos de predicción

Estos son los resultados obtenidos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Métrica |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* Explicar qué significa cada métrica
* Poner la comparación gráfica, las curvas
* Las matrices de confusiones?
* Ejemplo de una nevera predicha

**\*\*Ejemplo ilustrativo:\*\***

1) Nuestro objetivo es identificar ["pepino", "tomate", "huevos"] en fotos de nevera

2) De datasets públicos, hemos encontrado fotos de nevera disponibles con ["pepino"]

3) Con un script de python, construimos datasets sintéticos donde aleatoriamente (para simular una situación realista) se colocan fotos de huevos y tomate en diferentes fondos de nevera.

4) Al combinar 2 y 3 obtenemos un dataset final con todas las clases de comida que intentamos identificar con el modelo YOLO

Poner antes y después (las gráficas de conteo)

Cabe destacar que YOLO suele utilizar pesos iniciales preentrenados en el dataset COCO (Common Objects in Context), que contiene más de 330.000 imágenes y alrededor de 1,5 millones de instancias de objetos de utilidad general.

En este trabajo, se emplearán dichos pesos como punto de partida mediante transfer learning, realizando posteriormente un fine-tuning del modelo para adaptarlo específicamente a la detección de ingredientes en la nevera.

En este notebook se han hecho las siguientes iteraciones/entrenamientos a la hora de entrenar el modelo YOLO con fotos específicas de ingredientes en la nevera:

- *\*`Modelo 1 - 5 epochs sin datos sintéticos`\**: Se ha entrenado un modelo base sencillo de tan solo 5 epochs solamente con los datasets públicos (sin el sintético generado) para ver cuál sería el rendimiento base.

- *\*`Modelo 2 - 10 epochs con datos sintéticos`\**: Se ha entrenado un modelo de 10 epochs donde se ha incluido un dataset generado de manera sintética para cubrir

- *\*`Modelo 3 - 100 epochs con datos sintéticos`\**: Se ha entrenado un modelo de 100 epochs incluyendo el dataset sintético y con tamaño de imagen 768.

ANEXO

A group of graphs on a white background

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.A group of colorful bars

AI-generated content may be incorrect.