



**MÁSTER EN DATA SCIENCE AND BUSINESS ANALYTICS-  
FULL TIME | PRESENCIAL 3ra ed.**

# **Alimentación inteligente, sistema recomendador de cesta de compra.**

**TFM elaborado por:**

**Daniel Monsalve Buendía**

**Mariana Teresa Olavarri Niño**

**Aranza Nayeli Pizano Ocampo**

**Fernando Villalba Muñoz**

**Tutor/a de TFM:**

**Abel González Durán**

**Madrid, España, 8 de Septiembre del 2022**

# Resumen

En este trabajo de fin de máster se realizó un estudio sobre los productos alimenticios en los supermercados más conocidos de España, con el fin de crear un sistema recomendador sobre una de las principales necesidades de las personas, su alimentación. Mediante el análisis exploratorio de datos se seleccionó un conjunto de características relevantes, como la composición nutricional y el precio de cada uno de los productos. Seguidamente se procesaron los datos con el fin de obtener sugerencias de productos para los distintos usuarios y sus objetivos nutricionales y económicos, finalmente el consumidor toma decisiones basadas en datos y se demuestra de manera gráfica en un tablero interactivo.

Palabras clave: *API, CSV, ETL, Fuzzy Match, Herramientas de visualización, Nutri-score, Sistema recomendador, Web scraping.*

## Abstract

In this master's thesis, a study was carried out on food products in the best-known supermarkets in Spain, in order to create a recommender system on one of the main needs of people, their food. Through the exploratory data analysis, a set of relevant characteristics were selected, such as the nutritional composition and the price of each of the products. The data was then processed to obtain suggested products for the different users and their nutritional and financial goals, finally the consumer makes decisions based on data and it is shown graphically in an interactive dashboard.

Key Words: *API, CSV, ETL, Fuzzy Match, Nutri-score, Recommender system, Visualization tools, Web scraping.*

# Índice

Capítulo 1. Marco de Referencia	9
1.1 Introducción	9
1.2 Estado del arte	10
1.3 Planteamiento del problema	11
1.4 Justificación	12
1.5 Alcances y limitaciones	13
1.6 Objetivos	13
1.6.1 Objetivo general	13
1.6.2 Objetivos específicos	13
Capítulo 2. Modelo de negocio	14
2.1 B2B	14
2.2 B2C	15
2.3 Aspecto económico	15
2.3.1 Costes	15
2.3.2 Ingresos	15
2.3.3 Beneficios	16
Capítulo 3. Marco Teórico	16
3.1 Alimentación y nutrición	16
3.2 Alimentación en España	18
3.3 Bases de datos de alimentos	22
3.3.1 Open Food Facts	22

3.3.2 BEDCA	24
3.4 Aplicaciones de nutrición	25
3.4.1 Yuka	26
3.4.2 Fitia	27
3.5 Obtención de precios	28
3.5.1 Soysuper	29
3.5.1 Scraping	30
3.5.2 API WhiteBox	30
3.6 Sistemas recomendadores	31
Capítulo 4. Desarrollo del proyecto	33
4.1 Stack tecnológico, arquitectura y justificación	33
4.2 Información nutricional - Open Food Facts	35
4.3 Precios y supermercados	36
4.3.1 Web Scraping	37
4.3.2 API White Box	38
4.4 Carga a la base de datos a través de Pentaho	41
4.4.1 ETL, archivos por lotes.	41
4.4.2 ETL con Pentaho	43
4.5 Base de datos	47
4.6 Sistema recomendador	50
Capítulo 5. Resultados	57
5.1 Resultados esperados	57
5.2 Resultados	57

Capítulo 6. Conclusiones	66
6.1 Conclusiones del proyecto	66
6.2 Competencias desarrolladas o aplicadas	69
Capítulo 7. Siguiendo pasos	71
Referencias	73
Anexos	75
Anexo 1: Descarga&limpieza_openfood.ipynb	75
Anexo 2: downloader.py	76
Anexo 3: writer.py	92
Anexo 4: process.py	107
Anexo 5: Script creación base de datos SQL Text File	108
Anexo 6: Perfiles alimentación.ipynb	111
Anexo 7: Reglas_Perfiles.txt	115
Anexo 8: Sistema recomendador red neuronal.ipynb	117
Anexo 9: Get_data_usuarios	123
Anexo 10: Get_data	125
Anexo 11: Tablero de visualización general	127
Anexo 11.1: Cantidad de productos por supermercado	127
Anexo 11.2: Precio promedio por categoría	127
Anexo 11.3: Porcentaje de productos por categoría Nutri-Score	127
Anexo 11.4: Precio promedio por supermercado	128
Anexo 12: PyDiaScraper	128

# Índice de ilustraciones

Ilustración 1: Distribución del gasto en alimentación 2020 .....	19
Ilustración 2: Distribución del gasto en alimentación 2020 .....	20
Ilustración 3: Ejemplo de producto en OPF .....	23
Ilustración 4: Ejemplo de tabla nutricional en OPF .....	23
Ilustración 5: Base de Datos Española de Composición de Alimentos .....	24
Ilustración 6: Información nutricional de las aceitunas en BEDCA .....	25
Ilustración 7: interfaz de Yuka .....	26
Ilustración 8: Configuración inicial de Fitia .....	27
Ilustración 9: Interfaz de Fitia .....	28
Ilustración 10: Interfaz de Soysuper .....	29
Ilustración 11: Filtros colaborativos y basados en contenido .....	51

# Índice de figuras

Figuras 1 Grafico de estimaciones de beneficios .....	16
Figuras 2: Diagrama de flujo del proceso ETL .....	33
Figuras 3: Tratamiento de datos de open food facts .....	36
Figuras 4: Flujo para realizar web scraplicacióning .....	38
Figuras 5: Plataforma para descarga de whitebox .....	39
Figuras 6: Dataset supermercado ordenado por precio.....	40
Figuras 7: Tratamiento de datos dataset supermercados.....	41
Figuras 8: Proceso ETL .....	42
Figuras 9: archivos por lotes .....	42
Figuras 10: Transformación Fuzzy_Match .....	45
Figuras 11: Tabla Fuzy_match.....	45
Figuras 12: Transformación merge .....	46
Figuras 13: Error en la unión por clave de pentaho .....	46
Figuras 14: Job ETL.....	47
Figuras 15: Output de la red neuronal.....	58
Figuras 16: productos recomendados bajo en calorías .....	60
Figuras 17: productos recomendador ahorro .....	61
Figuras 18: Filtros para dashboard.....	63
Figuras 19: visualización de perfil: alto rendimiento .....	64
Figuras 20 visualización de perfil balanceado .....	65
Figuras 21: Información general en el dashboard .....	67

## Índice de tablas

Tabla 1: Estimados de locación supermercados .....	40
Tabla 2: tabla alimentos .....	48
Tabla 3: Tabla Merge .....	49
Tabla 4: Open Food .....	50
Tabla 5: supermercado .....	50
Tabla 6: Rating y significado .....	52
Tabla 7: Perfil centrado en la pérdida de peso .....	53
Tabla 8: Perfil centrado en la creación de masa muscular .....	54
Tabla 9: Perfil centrado en el alto rendimiento deportivo .....	54
Tabla 10: Perfil centrado en una dieta balanceada .....	55
Tabla 11: Perfil centrado en el ahorro económico .....	55
Tabla 12: Estadísticos descriptivos .....	56
Tabla 13: Usuarios y sus productos calificados positivamente para matriz basada en contenido .....	62
Tabla 14: Usuario, perfil, recomendaciones entregadas por el sistema.....	63
Tabla 15: precio promedio por categoría.....	68
Tabla 16: Nutri Score, Supermercado y precio promedio .....	69



# Capítulo 1. Marco de Referencia

## 1.1 Introducción

El objetivo principal de este proyecto es generar un producto que le permita a los usuarios tener recomendaciones de productos basados en sus objetivos nutricionales y económicos. En la actualidad podemos encontrar una gran diferencia en la calidad y los precios de los alimentos según su proveedor, por lo que nuestro objetivo será brindar información al consumidor para que conozca las ventajas y desventajas de los productos a consumir y en que supermercado se encuentra a un mejor precio. A través de un sistema recomendador se brinda una herramienta importante a los usuarios para conocer opciones o elementos de su interés.

La malnutrición es el problema que se desea atacar, por lo tanto, nuestra solución estaría orientada a mejorar la salud de la población española en este sentido. Según datos del INE (Instituto Nacional de Estadística) realizados en el año 2020, el sobrepeso y la obesidad afectan a un 16.5% de hombres y un 15.5% de mujeres con más de 18 años. Además, el grupo de edad comprendido entre 35 y 74 años es el más afectado, con un 44.9% de hombres y un 30.6% de mujeres que padecen de sobrepeso.

Para analizar esta problemática es necesario mencionar sus causas, y una de ellas es el desconocimiento sobre el contenido del producto, por ello, en noviembre del año 2018 se inició la implantación de Nutri-Score, un etiquetado nutricional basado en un código de colores, que categoriza el valor nutricional de los alimentos y bebidas, buscando que el usuario pueda hacer una toma de decisiones mejor informada.

Este proyecto se realizó por el interés de implementar un consumo más consciente e informado en la vida de los españoles, brindando una herramienta que permitiera introducir una serie de perfiles, que, basados en los objetivos de cada uno de los usuarios, permitiera obtener una selección de productos según sus necesidades.

La estructura técnica del proyecto inició con la búsqueda de bases de datos, de las cuales se eligieron Open Food Facts y la API WhiteBox. La primera se utiliza para la categorización de los alimentos y sus propiedades nutricionales, mientras que la API es utilizada para la obtención de precios en los distintos supermercados. A continuación, pasamos a la limpieza y depuración de ambas fuentes de datos, utilizando Pentaho para actualizar y aplicar las transformaciones necesarias que resultaron en la unión de ambas tablas.

Por último, planteamos diferentes perfiles de usuario con los que deseamos aplicar el sistema recomendador. La característica principal es que analizan gustos y preferencias de un usuario en específico, lo cual, genera mayor satisfacción, ya que se personifica el consumo. Estos sistemas actúan como asistentes y animan a seguir descubriendo productos, haciendo que la experiencia sea una actividad más agradable.

## **1.2 Estado del arte**

Existen diferentes programas que buscan la mejoría de la dieta utilizando la información nutricional genérica de los productos, algunos ejemplos son: EICoCo, Yuka, MyRealFood. Las tres poseen un objetivo en común: ofrecer al consumidor información para mejorar la escogencia de productos.

EICoCo es una aplicación gratuita que tiene como objetivo crear una alimentación más consciente. Esta evalúa un producto específico partiendo del Sistema Nutri-score y la Clasificación NOVA, que ordena los alimentos en cuatro grupos en función de su “grado de procesamiento”. El usuario al utilizar CoCo crea un histórico con los productos anteriormente escaneados.

Yuka sigue tres criterios para la clasificación de productos: su calidad nutricional Nutri-Score (60%), la presencia de aditivos (30%) y si el producto es ecológico (10%). Ya obteniendo esto, el producto escaneado establece una clasificación de malo, mediocre, bueno y excelente. De igual forma, en el caso de ser una mala puntuación la aplicación recomienda otro más saludable.

Finalmente, MyRealFood basa su análisis del producto escaneado en la clasificación NOVA y también toma en cuenta las opciones de la EFSA (Autoridad Europea de Seguridad Alimentaria). Esta aplicación lanza una advertencia cuando un producto tiene un alto contenido de grasas saturadas, azúcares, sal o energía, y al igual que Yuka se brinda una opción con mejor contenido nutricional.

Por otro lado, las comparaciones con otro trabajo de final de máster se presentaron en La Laguna al (2018) por el estudiante Wehbe Nuez Eduardo. El proyecto mencionado busca recopilar y centralizar alimentos con su respectiva información nutricional y su precio e integrar un sistema E-Commerce, donde los pequeños productores puedan vender su producto de manera gratuita.

Las aplicaciones y estudios realizados anteriormente desean resolver un problema en común, este es brindar una solución que genere una mejora en los hábitos alimenticios. No obstante, las aplicaciones anteriores muestran el contenido nutricional pero no el precio de mercado y algunas de ellas no generan recomendaciones de productos al menos de que la evaluación de alguno se encuentre en una mala categoría.

Lo diferencial de este TFM respecto a las soluciones anteriormente presentadas es que nuestro proyecto se trata de un producto “all-in-one”, se presenta una solución que integra comparativa de precios y valores nutritivos de más de 7500 productos de los principales supermercados españoles como el resto de apps, pero que además cuenta con un sistema recomendador personalizado para el usuario que hace las veces de asistente de la cesta de la compra, siguiendo criterios como el tipo de nutrición, dieta o presupuesto para recomendar siempre la mejor opción.

### **1.3 Planteamiento del problema**

Con la intención de aportar una solución al sector de la salud, nos centramos en la problemática que enfrenta la población española referente al consumo desmedido de productos con baja calidad nutricional.

En este sentido, algunos de problemas actuales relativos a la alimentación según Burgos (2007) son:

- Incorporación de hábitos y alimentos extraños a nuestro medio y costumbres.
- Aumento desmedido del consumo de proteínas derivadas de la carne.
- Exceso o escaso uso del pescado en la alimentación cotidiana.
- Exceso de azúcares refinados: postres, comida chatarra.
- Alto consumo de productos industriales y precocidos.
- Incorporación de bebidas gaseosas en sustitución de agua.

Todos estos hábitos de alimentación inadecuados generan efectos adversos sobre la salud de la población. Los datos revelan el alto índice de enfermedades relacionadas con el sobrepeso y la obesidad, tales como la diabetes, hipertensión y enfermedades cardíacas. La alimentación es uno de los pilares fundamentales para mantener una buena salud, no solo es necesario mantener una dieta balanceada y saludable, sino que, además, es necesario tener una correcta orientación de lo que se debe consumir y las consecuencias que esto puede traer a nuestra salud.

Para ayudar a resolver este problema se ha pensado en ofrecer un recomendador de productos que estuviera basado en el objetivo nutricional de cada uno de los usuarios. De esta forma, las personas podrán llevar una alimentación más consciente, estarán al corriente de los productos que se encuentran en el mercado, y conocerán los precios que estos puedan tener.

### **1.3 Premisa**

La aplicación de un sistema recomendador basado en los gustos del usuario, puede incitar a la comunidad española a un consumo más consciente en alimentos con mejor calidad y un precio ajustable a sus necesidades. El uso de nuestro producto proporcionará las herramientas necesarias para que los usuarios tomen decisiones más informadas y que se ajusten a sus objetivos alimentarios y económicos.

### **1.4 Justificación**

La elección de productos es cada vez más complicada ya que los proveedores ponen a disposición más opciones de donde escoger, sin embargo, es importante conocer el contenido nutricional de los alimentos que se consumen a diario, ya que estos impactan de manera directa en la salud. Para facilitar la elección de mejores alimentos se implementaron estrategias para reducir la obesidad a nivel nacional como la herramienta de Nutri Score, que permite visualizar un indicador que agrupa distintos y más complejos valores nutricionales para que todo tipo de usuario pueda entenderlo.

Nutri Score es un proyecto de la Oficina Europea de Seguridad Alimentaria y de la ONG de salud y alimentación "Open Food France", que ha sido puesto en marcha por el Ministerio de Sanidad. Este es un sistema de puntuación utilizado en varios países de Europa para evaluar la calidad nutricional de los alimentos, de forma sencilla y visible para el consumidor. El objetivo del Nutri-Score es simplificar la evaluación de los alimentos y hacerla más fácil para el consumidor. El objetivo es que el consumidor pueda tener una guía sencilla para comprar mejor y para tomar decisiones más saludables.

El desarrollo de este proyecto se centra en ofrecer una herramienta a la población española que funcione como recomendador de productos basados en su aporte nutricional y el precio. Como se menciona, se utiliza el Nutri-Score como principal fuente para dictaminar la calidad de un producto y los distintos supermercados más conocidos a nivel nacional para la comparativa de costos.

## 1.5 Alcances y limitaciones

### Alcances

- Contar con bases de datos para la categorización de alimentos y precios de venta de los productos.
- Brindar una herramienta que permita introducir perfiles determinados.
- Crear un sistema de recomendación que les permita obtener productos de acuerdo con sus objetivos nutricionales.
- Mostrar al usuario una lista de productos recomendados y su precio en diferentes supermercados.

### Limitaciones

- Las bases de datos para la categorización de alimentos y precios de venta de los productos pueden estar incompletas y no contener todos los productos en el mercado.
- Se cuenta con una memoria y espacio limitado en los ordenadores locales.
- La capacidad de recomendación está limitada a la cantidad de datos disponibles.

## 1.6 Objetivos

### 1.6.1 Objetivo general

Desarrollar una herramienta que permita a los usuarios obtener sugerencias personalizadas de productos alimenticios basados en sus objetivos nutricionales y económicos, además brindar información sobre en qué supermercado encontrarán dichos artículos y su precio, para que el usuario pueda mejorar sus hábitos alimenticios, lograr un ahorro si lo desea, así como reducir sus tiempos de compra.

### 1.6.2 Objetivos específicos

1. Identificar las bases de datos de alimentos y precios de venta de los productos disponibles.
2. Realizar la limpieza de ambas bases de datos.
3. Determinar la calidad de los productos utilizando el sistema de puntuación Nutri Score.
4. Comparar los precios de los productos en diferentes supermercados.
5. Recomendar al usuario los productos más saludables y económicos de acuerdo a sus objetivos nutricionales.

## **Capítulo 2. Modelo de negocio**

La misión de nuestro negocio es ofrecer a nuestros usuarios alternativas y recomendaciones de alimentos en base a distintos objetivos, para asistirle en el proceso de hacer la lista de la cesta de la compra o comprando en supermercados online.

Nuestra visión es impulsar una alimentación más saludable y económica accesible para todo el mundo.

Nuestros valores tienen en cuenta la sostenibilidad, accesibilidad y privacidad de nuestros usuarios y medio ambiente.

Tras el desarrollo del proyecto, consideramos que el servicio se puede ofrecer en forma de producto a distintos tipos de clientes que detallamos a continuación.

Nuestro producto fácilmente se puede enfocar a clientes B2B (business to business), grandes superficies y retailers online y B2C (business to client) en forma de aplicación a usuarios particulares.

### **2.1 B2B**

Casi todas las grandes superficies tienen presencia online y servicio de compra por internet para sus productos, algunos ejemplos son Carrefour, Día, que incluso se ha integrado en Amazon, Hipercor o Mercadona que tienen un servicio a domicilio. Es normal en las tiendas físicas encontrar distintos estímulos, como puede ser la forma de colocar ciertos productos en el lineal, usar colores, música u otros elementos para guiar al consumidor en el proceso de compra, por lo general son técnicas efectivas, pero difícilmente personalizables por su naturaleza física.

Realizando el estudio de los distintos supermercados, nos dimos cuenta de que la experiencia online es muy estándar para diferentes tipos de clientes y no se incentiva la compra de productos distintos a los que el consumidor tiene pensado comprar. Esto genera una oportunidad de negocio para nuestro servicio, ya que implementar un sistema de recomendación de productos personalizable para el consumidor puede incrementar las ventas y traducirse en mayores ingresos para el supermercado.

## **2.2 B2C**

Nuestro servicio dirigido a usuarios particulares se ofrecería mediante aplicación y como extensión de navegador para aportar un mayor valor añadido a servicios de terceros de listas o listas de la compra. La aplicación ofrecería valor añadido al usuario con sistema de alertas de precios, recomendaciones diarias en base a sus gustos y patrones de consumo, y con recomendación activa a la hora de crear listas de compra basado en los distintos objetivos que tenga el consumidor, como hacer dieta o ahorrar.

## **2.3 Aspecto económico**

### **2.3.1 Costes**

Tenemos que diferenciar por un lado los costes operativos o variables, que serían todos aquellos derivados de la propia actividad y uso, y los costes fijos.

En cuanto a los costes variables, hemos incurrido en costes de 150€ mensuales en el hosting de la base de datos MySQL en Google cloud con una máquina adaptada al entorno de desarrollo, por lo que tener dos entornos, uno de producción y otro de desarrollo, escalado a las necesidades de la plataforma, podemos presupuestar 300€ mensuales en entorno DES y 500€ en el entorno PRO-OPERATIVO.

En cuanto a los costes fijos, es necesario el desarrollo de un sitio web y una aplicación móvil Android y iOS más su mantenimiento, de acuerdo con los precios de mercado, por un tiempo de desarrollo de 2 a 3 meses los costes ascienden a 15000€. Necesitamos un presupuesto en marketing de 100€ mensuales para anuncios en medios especializados de alimentación saludable y para el desarrollo del producto e investigación de mercado es necesario un “Data Scientist” y un “Data Engineer” cuyos salarios son de alrededor a 60.000-80.000€ anuales para cada uno.

### **2.3.2 Ingresos**

Deberíamos diferenciar por una parte los clientes B2B de los B2C. Para los B2B los ingresos tendrían lugar a partir de contratos bilaterales por duración de un año tras ser negociados, estimamos que el servicio totalmente mantenido por nosotros se puede ofertar entre 300.000 y 500.000€.

Para los usuarios particulares, la app se ofrece en modelo de suscripción por 10€ mensuales, que es lo común para apps similares como las vistas en el capítulo 2.4.

### 2.3.3 Beneficios

El gasto total del primer año asciende a 147.000€, suponiendo que no conseguimos ningún contrato B2B, sería necesario tener una base de usuarios que paguen la suscripción mensual durante un año de 1225 clientes para conseguir el break even.

Para el segundo año y consecutivos los gastos son de 132.000€, estimando que, si se consigue un contrato B2B, ya sería suficiente para cubrir el “break even” e incluso generar beneficios positivos. *En la figura 1* se presenta un escenario a tres años de la evolución de los beneficios del negocio en base a estimaciones.

#### Escenario estimaciones de beneficios a 3 años

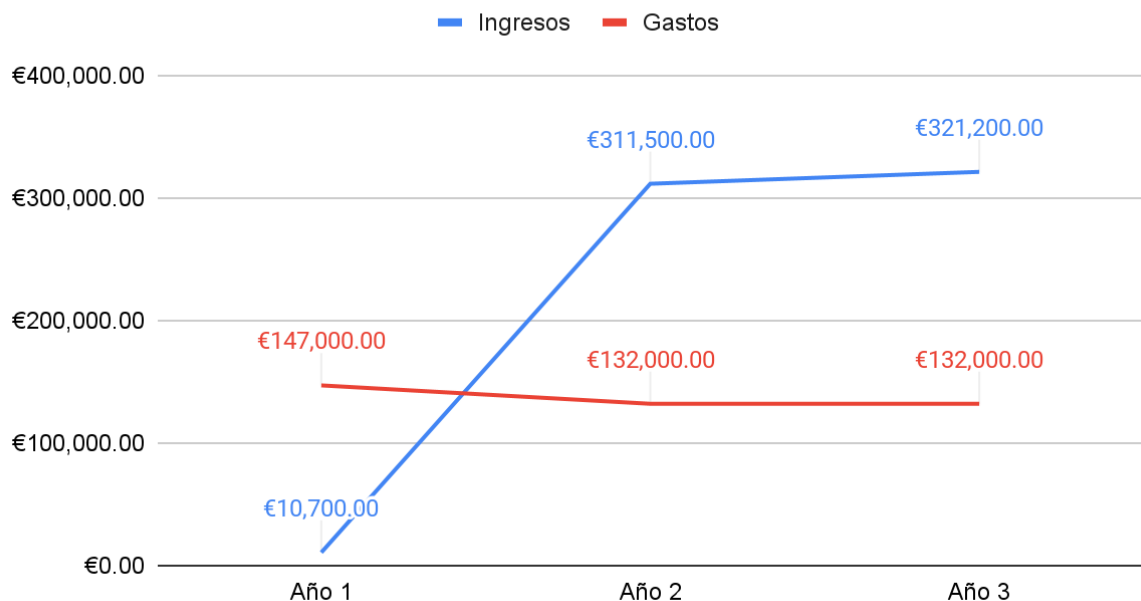


Figura 1 Gráfico de estimaciones de beneficios

## Capítulo 3. Marco Teórico

### 3.1 Alimentación y nutrición

La alimentación es un fenómeno cultural muy relevante tanto desde el punto de vista individual como colectivo. No existe una alimentación equilibrada ideal que se pueda



extrapolar a toda la población, sino que varía en cada individuo según sus condiciones socioeconómicas, la disponibilidad de alimentos en el área donde se reside, los gustos personales, los hábitos de vida y alimentación, las habilidades culinarias, las creencias religiosas, la existencia de alguna enfermedad, las situaciones fisiológicas especiales y, por supuesto, la edad, el sexo y la antropometría del sujeto (Martínez et al., 2005).

La alimentación de un individuo es equilibrada cuando, teniendo en cuenta los factores anteriores, contiene todos los alimentos necesarios para conseguir un estado nutricional óptimo. Este estado es aquel en el que la alimentación cubre los siguientes objetivos:

- Aporte de calorías suficientes para llevar a cabo los procesos metabólicos y el trabajo físico necesarios.
- Suministro de suficientes nutrientes con funciones plásticas y reguladoras.
- Mantenimiento o consecución del peso ideal.
- Equilibrio entre las cantidades de cada uno de los nutrientes entre sí.

Además de una alimentación equilibrada se requiere que sea nutritiva. Los nutrientes son sustancias contenidas en los alimentos, que son necesarias para vivir. Se dividen en dos grupos: macronutrientes y micronutrientes.

Los macronutrientes son aquellas sustancias que proporcionan energía al organismo para un buen funcionamiento, y otros elementos necesarios para reparar y construir estructuras orgánicas, para promover el crecimiento y para regular procesos metabólicos. Este grupo está constituido por:

- Proteínas: Son los componentes de las estructuras de las células.
- Grasas: Son el nutriente energético por excelencia.
- Hidratos de Carbono: Son una fuente importante de energía y proceden fundamentalmente de los vegetales.

Los micronutrientes son sustancias que no aportan energía, pero son esenciales para el buen funcionamiento de nuestro organismo. En este grupo encontramos:

- Vitaminas:
  - Hidrosolubles: son ocho vitaminas del grupo B y la vitamina C.
  - Liposolubles: vitaminas A, D, K o E.

- Minerales y oligoelementos: en este grupo se encuentran el calcio, fósforo, magnesio, sodio, potasio, cloro, azufre, hierro, yodo, cinc, cobre, cromo, selenio y flúor.

Una dieta equilibrada aporta a nuestro organismo las vitaminas y minerales necesarios para su buen funcionamiento (Álvarez, 2020).

## 3.2 Alimentación en España

Para conocer el contexto de la alimentación en España tenemos a Mercasa, que realiza desde 1998 una publicación anual llamada “Alimentación en España. Producción, Industria, Distribución y Consumo”. Para la elaboración de estos informes, Mercasa cuenta con la colaboración del Ministerio de Agricultura, Pesca y Alimentación

Comenzamos por establecer un gasto alimentario en España, tanto sector hogares como hostelería, de acuerdo con la información aportada por el Ministerio de Agricultura, Pesca y Alimentación, se determina que el gasto total en alimentación y bebidas ascendió a 102.082,8 millones de euros en 2020.

La participación de los hogares en este gasto se cifra en 79.348,3 millones de euros un 77,7% mientras que los establecimientos de hostelería y restauración alcanzaron un gasto de 22.734,5 millones de euros un 22,3%, como podemos observar en la *ilustración 1*. El volumen de gasto y el reparto de este marcan notables diferencias con respecto a los ejercicios anteriores motivadas por los efectos de la COVID-19.

Durante el año 2020 se produjo un descenso del gasto alimentario con respecto al año anterior (-3,2%), motivado por la notable minoración del consumo extra doméstico (-36,8%) puesto que en la alimentación del hogar existe un efecto positivo y compensador (14,2%).

<b>GASTO TOTAL ALIMENTACIÓN</b>  102.082,8 millones de euros Δ 2020-19: -3,2%	<b>GASTO ALIMENTACIÓN EN EL HOGAR</b>  79.348,3 millones de euros Δ 2020-19: 14,2%	<b>COMERCIO ESPECIALIZADO</b> 13.849,0 millones de euros (17,4%) Δ 2020-19: 17,7%
		<b>SUPERMERCADOS</b> 47.292,1 millones de euros (59,6%) Δ 2020-19: 12,8%
		<b>HIPERMERCADOS</b> 10.424,6 millones de euros (13,1%) Δ 2020-19: 10,4%
		<b>ECONOMATOS Y COOPERATIVAS</b> 233,6 millones de euros (0,3%) Δ 2020-19: 23,9%
		<b>MERCADILLOS</b> 540,4 millones de euros (0,7%) Δ 2020-19: -12,3%
		<b>VENTA A DOMICILIO</b> 569,1 millones de euros (0,7%) Δ 2020-19: 40,9%
		<b>AUTOCONSUMO</b> 1.419,7 millones de euros (1,8%) Δ 2020-19: 1,7%
		<b>VENTA INTERNET</b> 1.742,1 millones de euros (2,2%) Δ 2020-19: 71,6%
		<b>OTROS CANALES DE VENTA</b> 3.277,7 millones de euros (4,2%) Δ 2020-19: 19,4%
		<b>GASTO ALIMENTACIÓN EXTRADOMÉSTICO</b>  22.734,5 millones de euros Δ 2020-19: -36,8%
<b>RESTAURANTES</b> 5.933,7 millones de euros (26,1%)		
<b>BARES Y CAFETERÍAS</b> 9.025,6 millones de euros (39,7%)		
<b>PANADERÍAS Y PASTELERÍAS</b> 431,9 millones de euros (1,9 %)		
<b>TIENDAS CONVENIENCIA Y ESTACIONES SERVICIO</b> 932,1 millones de euros (4,1%)		
<b>HOTELES</b> 113,7 millones de euros (0,5%)		
<b>MÁQUINAS DISPENSADORAS</b> 500,2 millones de euros (2,2%)		
<b>SERVICIOS EN LA EMPRESA</b> 386,5 millones de euros (1,7%)		
<b>OTROS CANALES DE VENTA</b> 2.273,4 millones de euros (10,0%)		

Ilustración 1: Distribución del gasto en alimentación 2020

En cuanto a la demanda de alimentación de los hogares españoles siguen primando los productos frescos; la carne supone un 20,4% sobre el gasto total, las patatas, frutas y hortalizas frescas un 19,0%, los pescados un 12,9%, seguido por el gasto alimentario la leche y derivados lácteos con un 11,0% sobre el gasto total, en menor escala les siguen el gasto en pan con el 4,6%, los productos de bollería y pastelería 3,8%, el aceite de oliva 1,5% o los platos preparados 4,3%. A continuación, se presentan estos datos en la ilustración 2.

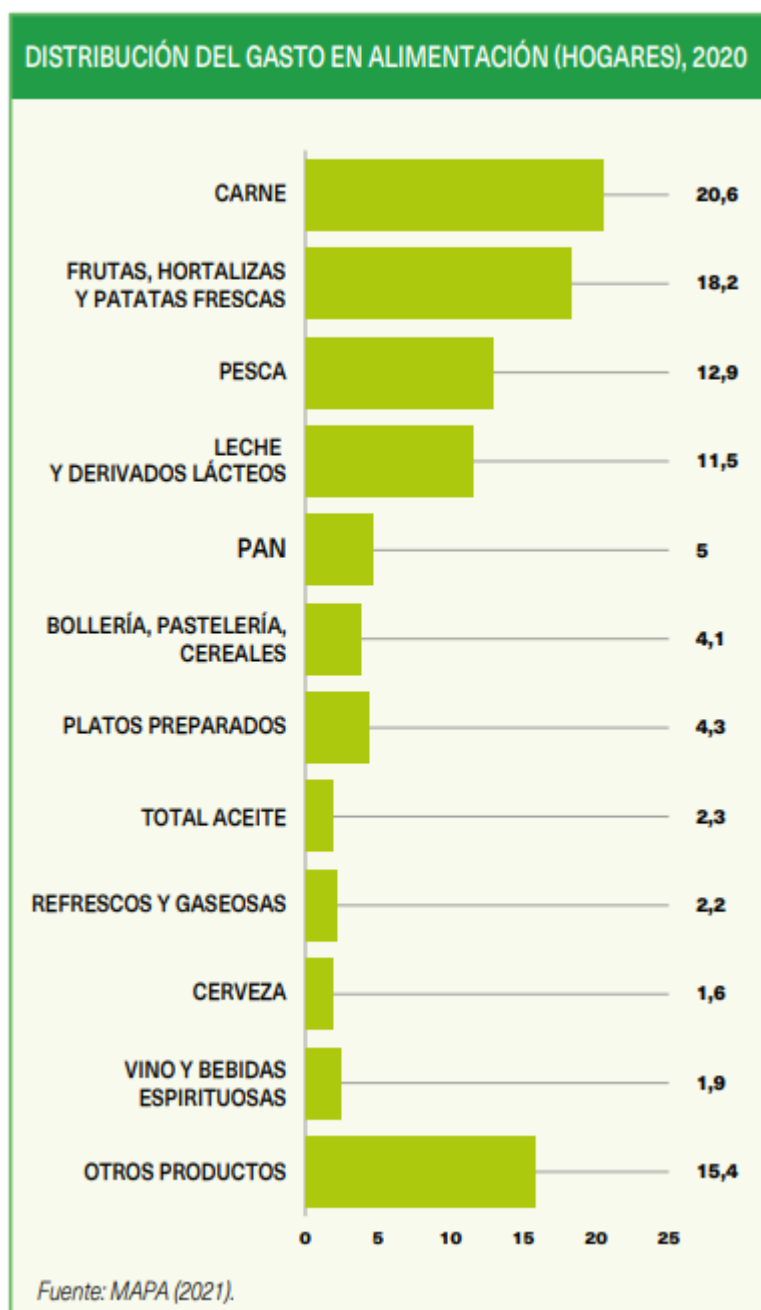


Ilustración 2: Distribución del gasto en alimentación 2020

Durante el año 2020, el gasto por persona en alimentos y bebidas para consumo en el hogar se cifra en más de 1.716 euros (1.716,3 euros, concretamente). En consecuencia, la media de gasto mensual en productos alimentarios se situó en 143 euros, aunque los efectos de la COVID-19 elevaron notablemente el gasto en alimentación en determinados meses del año (por ejemplo, marzo).

La carne es el producto más demandado y cada español gastó el año 2020 la suma de 349,5 euros en los 49,9 kilos per cápita consumidos. El gasto por individuo en productos

del mar asciende a 221,5 euros y supone un consumo de 24,8 kilos por persona. Las frutas y hortalizas, tanto frescas como transformadas, tienen un protagonismo notable en la demanda del consumidor español.

Durante 2020, en términos medios, cada individuo consumió 99,7 kilos de frutas frescas, 96,0 kilos de hortalizas y patatas frescas y 14,5 kilos de frutas y hortalizas transformadas; en cifras de gasto per cápita, el consumo referido supuso 170,5 euros, 155,4 euros y 32,0 euros, respectivamente. El gasto y el consumo en leche líquida y derivados lácteos también resulta notable en los hogares españoles. Por persona, se cuantifica un consumo de 74,0 litros de leche y un gasto de 51,1 euros mientras que los productos lácteos alcanzan, también en cifras per cápita, un gasto de 138,2 euros y un consumo de 37,4 kilos.

En términos medios, durante 2020 cada español consumió 32,8 kilos de pan, 14,2 kilos de bollería y pastelería y 16,8 kilos de platos preparados que, en términos de gasto per cápita, supusieron 78,2 euros, 65,6 euros y 73,5 euros, respectivamente.

El aceite de oliva, el vino y los huevos son alimentos arraigados en la cultura gastronómica española y, por tanto, con una presencia generalizada en la demanda de los hogares. En 2020 cada español consumió 8,9 litros de aceite de oliva y gastó 26,3 euros en este producto; el consumo per cápita de vino y derivados ascendió a 10,1 litros y supuso un gasto de 30,6 euros por persona; finalmente, el gasto en huevos llegó a 22,9 euros y se corresponde con los 155 huevos consumidos de media por persona.

El consumo alimentario aparece condicionado por las diferentes características que tienen los individuos que realizan su demanda. Esto es, el tamaño de la población de residencia, el número de personas que componen el hogar, el nivel socioeconómico, la presencia o no de niños en la familia, la situación en el mercado laboral del encargado de realizar las compras o la edad de este son variables que intervienen significativamente en la decisión de compra de alimentos y bebidas (Sanz De La Torre et al., 2021).

En la actualidad gran medida de la población se está desarrollando de una serie de malos hábitos alimenticios: como el aumento en el consumo de comida chatarra y procesada, mientras que la comida casera y los alimentos de origen natural se reducen. Parte de la población no está al tanto de estas opciones alimenticias, que podrían ayudarles a lograr un mejor desempeño, tanto en la escuela como en su vida diaria (Burgos, 2007).

El aislamiento social o las cuarentenas declaradas por los gobiernos para evitar la propagación del COVID-19, aunado a las dificultades económicas asociadas con esto por la pérdida de empleos, entre otras situaciones, ha llevado a las familias a realizar un confinamiento en muchos casos sin poder cubrir sus requerimientos calóricos y nutricionales mínimos, poniendo en riesgo su seguridad alimentaria y su estado nutricional, que con el paso de los días tiende a empeorar. Esto se agrava cuando las personas por diferentes motivos no tienen claridad en qué tipo de alimentos comprar para mantener una dieta saludable, cómo prepararlos adecuadamente, o simplemente no cuentan con recursos para adquirirlos, por lo que posiblemente privilegian la compra de alimentos altos en carbohidratos y grasas, pues generan saciedad, son económicos y rendidores, pero también se sabe que aportan muchas calorías y son pobres en micronutrientes (Deossa Restrepo et al., 2020).

### **3.3 Bases de datos de alimentos**

#### **3.3.1 Open Food Facts**

Open Food Facts (OPF) es una base de datos de productos alimenticios con ingredientes, alérgenos, propiedades nutricionales y toda la información que podemos encontrar en las etiquetas de los productos. Aparte de la colaboración voluntaria de los usuarios, OPF utiliza una serie de aplicaciones que aportan datos adicionales a la base de datos entre la que se destaca Yuka como la más importante; Yuka es una aplicación móvil que escanea los productos alimentarios y cosméticos para descifrar sus etiquetas.

OPF tiene un registro de 278.996 productos a la venta en España, indicando la marca, los datos nutricionales más importantes y algunas características diferenciadoras como el país de venta, envases empleados, etiquetas o certificaciones, ingredientes y una larga lista de datos adicionales excluyendo el precio.

La base de datos de Open Food Facts está publicada como datos abiertos (Open Data) bajo la licencia Open Database Licence. Cualquier persona puede usarla para cualquier propósito (Open Food Facts, 2022).

**Nutri-Score A**  
Calidad nutricional muy buena

NOVA no calculado  
Nivel de procesamiento de alimentos desconocido

**Eco-Score B**  
Bajo impacto ambiental

Elige la información que prefieres ver primero. [Editar tus preferencias de alimentos](#)

### Características del producto

**Cantidad:** 550 g

**Envase:** Caja, en:Cardboard, Paquete, en:Green dot, en:Triman, fr:Boîte de carton, fr:Etui en carton, fr:Pensez au tri!

**Marcas:** QUAKER, Oats

**Categorías:** Alimentos y bebidas de origen vegetal, Alimentos de origen vegetal, Cereales y patatas, Desayunos, Cereales y derivados, Cereales para el desayuno, Copos, Copos de cereales, en:Rolled flakes, Copos de avena, fr:Alimentos de origen vegetal, fr:Alimentos y bebidas de origen vegetal, fr:Cereales para el desayuno, fr:Cereales y derivados, fr:Cereales y patatas, fr:Copos, fr:Copos de avena, fr:Copos de cereales, fr:Desayunos

**Etiquetas, certificaciones, premios:** 100% natural, Punto Verde, Sin azúcares añadidos, Sin conservantes, Nutriscore, Nutriscore A, fr:Contribue-a-reguler-le-cholesterol

Ilustración 3: Ejemplo de producto en OPF

☒ Diferencia en % ☐ valor para 100 g/ 100 ml

→ Nota: para cada nutriente, el promedio no es el de todos los productos de la categoría, sino el de todos los productos para los cuales se especifica la cantidad de nutrientes.

Información nutricional	Como se vende por 100 g / 100 ml	Como se vende por porción (40g)	Comparado con: fr:cereales-y-derivados
Energía	1570 kJ (375 kcal)	628 kJ (150 kcal)	+64 %
Grasas	8 g	3,2 g	+269 %
Grasas saturadas	1,5 g	0,6 g	+226 %
Hidratos de carbono	60 g	24 g	+43 %
Azúcares	1,1 g	0,44 g	-62 %
Fibra alimentaria	9 g	3,6 g	+154 %
Proteínas	11 g	4,4 g	+53 %
Sal	0,01 g	0,004 g	-82 %
Vitamina B1 (Tiamina)	0,9 mg	0,36 mg	
Fósforo	380 mg	152 mg	
Hierro	3,8 mg	1,52 mg	
Magnesio	110 mg	44 mg	
Fruits, vegetables, nuts and rapeseed, walnut and olive oils (estimate from ingredients list analysis)	0 %	0 %	
Beta-glucano	3,6 g	1,44 g	

Ilustración 4: Ejemplo de tabla nutricional en OPF

### 3.3.2 BEDCA

Es una red de Centros de investigación públicos, Administración e Instituciones privadas cuyo objetivo es el desarrollo y mantenimiento de la Base de Datos Española de Composición de Alimentos.

Esta base de datos está construida con los estándares europeos desarrollados por la Red de Excelencia Europea EuroFIR y se incorporará a otras Bases de Datos Europeas dentro la Asociación EuroFIR AISBL encargadas de elaborar una plataforma unificada y con estándares de calidad de las Bases de Datos de Composición de Alimentos Europeas y su interconexión a través de servicios WEB.

Esta base de datos cuenta con 2713 alimentos sin especificar marca alguna, tienen una amplia descripción de la información nutricional incluyendo los macronutrientes y vitaminas de los productos como se muestra en la *ilustración 5*.



The screenshot shows the BEDCA website interface. At the top, there is a header with the BEDCA logo, a decorative image of a sunflower, and the text 'Base de Datos Española de Composición de Alimentos' and 'Spanish Food Composition Database'. Below the header is a navigation bar with buttons for 'Inicio', 'Fuentes', and 'Consulta'. The main content area is titled 'LISTADO DE ALIMENTOS DE LA CONSULTA' and displays a table of food items. The table has three columns: 'Id', 'Nombre', and 'Name'. The items listed are various types of oils, including cotton oil, peanut oil, coconut oil, rape oil, wheat germ oil, sunflower oil, grape seed oil, cod liver oil, flaxseed oil, walnut oil, olive oil, extra virgin olive oil, extra virgin olive oil (organic), palm oil, sesame oil, and soybean oil.

Id	Nombre	Name
746	Aceite de algodón	Cotton oil
747	Aceite de cacahuete	Peanut oil
748	Aceite de coco	Coconut oil
753	Aceite de colza	Rape oil
749	Aceite de germen de trigo	Wheat germ oil
2541	Aceite de girasol	Sunflower oil
754	Aceite de grano de uva	Grape seed oil
755	Aceite de hígado de bacalao	Cod liver oil
2542	Aceite de lino	Flaxseed oil
750	Aceite de nuez	Walnut oil
2543	Aceite de oliva	Olive oil
2544	Aceite de oliva virgen extra	Extra virgin olive oil
2545	Aceite de oliva virgen extra, producción ecológica	Extra virgin olive oil, organic
751	Aceite de palma	Palm oil
752	Aceite de sésamo	Sesame oil
2546	Aceite de soja	Soybean oil

Ilustración 5: Base de Datos Española de Composición de Alimentos



**Información de composición (por 100 g de porción comestible)**

Componente	Valor	Unidad	Fuente
<b>Proximales</b>			
alcohol (etanol)	0	g	38
energía, total	502 (120)	kJ (kcal)	236
grasa, total (lípidos totales)	12.5	g	7
proteína, total	1.3	g	7
agua (humedad)	78	g	7
<b>Hidratos de Carbono</b>			
fibra, dietética total	4.8	g	38
carbohidratos	1	g	38
<b>Grasas</b>			
ácido graso 22:6 n-3 (ácido docosahexaenóico)	-	-	-
ácidos grasos, monoinsaturados totales	8.7	g	7
ácidos grasos, poliinsaturados totales	0.6	g	38
ácidos grasos saturados totales	2.6	g	38
ácido graso 12:0 (láurico)	-	-	-
ácido graso 14:0 (ácido mirístico)	-	-	-
ácido graso 16:0 (ácido palmítico)	-	-	-
ácido graso 18:0 (ácido esteárico)	-	-	-
ácido graso 18:1 n-9 cis (ácido oléico)	-	-	-
colesterol	0	mg	38
ácido graso 18:2	-	-	-
ácido graso 18:3	-	-	-
ácido graso 20:4 n-6 (ácido araquidónico)	-	-	-
ácido graso 20:5 (ácido eicosapentaenóico)	-	-	-
<b>Vitaminas</b>			
Vitamina A equivalentes de retinol de actividades de retinos y carotenoides	48	ug	38
Vitamina D	0	ug	38
Vitamina E equivalentes de alfa tocoferol de actividades de vitámeros E	1.48	mg	38
folato, total	10.4	ug	38
equivalentes de niacina, totales	0.77	mg	38
riboflavina	0.05	mg	38
tiamina	0.03	mg	38

*Ilustración 6: Información nutricional de las aceitunas en BEDCA*

## 3.4 Aplicaciones de nutrición

La tecnología puede convertirse en una gran aliada para aquellos que busquen mejorar su salud a través de una correcta nutrición, abriendo camino a un acceso de información nutricional de calidad, fiable y que pueda apoyar en la toma de decisiones más informadas, a continuación, se presentan aplicaciones similares al producto desarrollado en este trabajo para comparar lo que ya existe, sus fortalezas y lo que aún no se ha desarrollado.

### 3.4.1 Yuka

De acuerdo a la información proporcionada en su página web yuka.io, Yuka es una aplicación que analiza los productos alimentarios y te explica la evaluación de cada producto en una ficha de producto detallada, en caso de escanear productos mediocres o malos, también recomienda con total independencia productos similares y mejores para la salud, además muestra un historial de los productos escaneados para poder identificar el impacto de cada producto sobre la salud mediante un código sencillo de colores. Esta página como ya se mencionó en el capítulo 2.3.1 es una de las principales fuentes de alimentación de la base de datos de OPF.

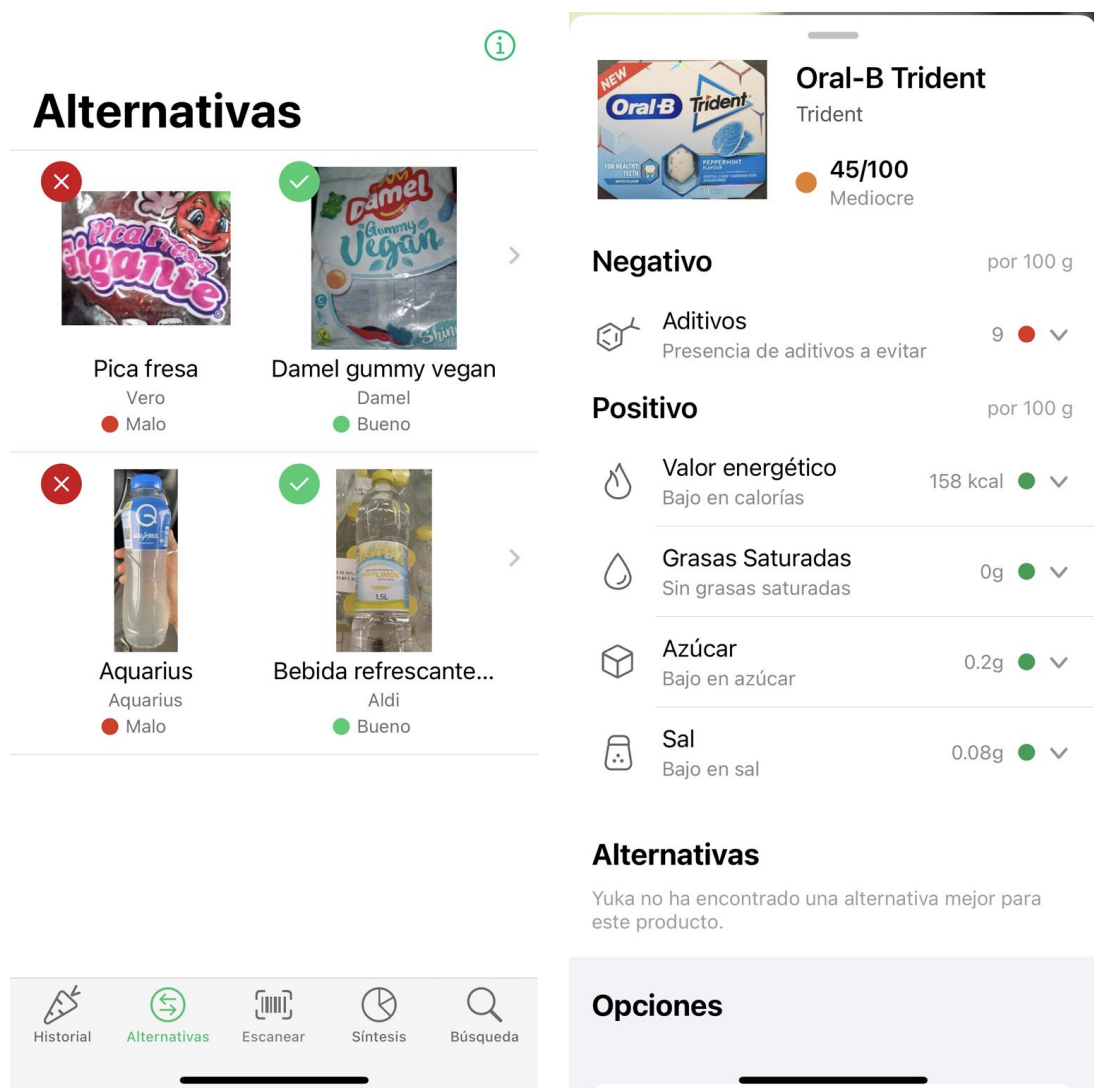


Ilustración 7: interfaz de Yuka

En 2019 contaban con una base de datos de más de 600.000 alimentos los cuales sólo puedes visualizar si escaneas el código de barras de los productos en cuestión o eres

usuario premium para poder tener acceso a la lista de productos. Esta aplicación tiene poca información nutricional de los productos comparada con otras fuentes y no incluye el precio de los productos.

### 3.4.2 Fitia

Fitia es una aplicación de planeación y seguimiento de alimentación diaria que planea nuestro menú diario basándose en alimentos que nos gustan y un objetivo seleccionado al iniciar la aplicación como podemos ver en la *ilustración 8*. Incluye recetas sencillas, con cantidades fáciles de medir y muy completas a nivel de macronutrientes en la versión de pago. Permite llevar un seguimiento de peso y adaptar los planes a las exigencias del usuario.



Ilustración 8: Configuración inicial de Fitia

La aplicación ofrece en su versión gratuita un menú diario que consiste de los alimentos y las porciones que debes incluir en cada comida, así como la posibilidad de añadir manualmente los alimentos y porciones consumidos para poder darle un seguimiento a los macronutrientes y calorías consumidas en el día, el resto de datos nutricionales son exclusivos de la versión de pago, cuenta con la opción de escanear productos así como la posibilidad de ver la marca cuando se genera una búsqueda específica. Con la versión gratuita no se pueden ver menús de otros días, no tiene la opción de hacer lista de compra y no incluye el precio de los productos, en la *ilustración 9* podemos ver cómo es la interfaz diaria de la aplicación.

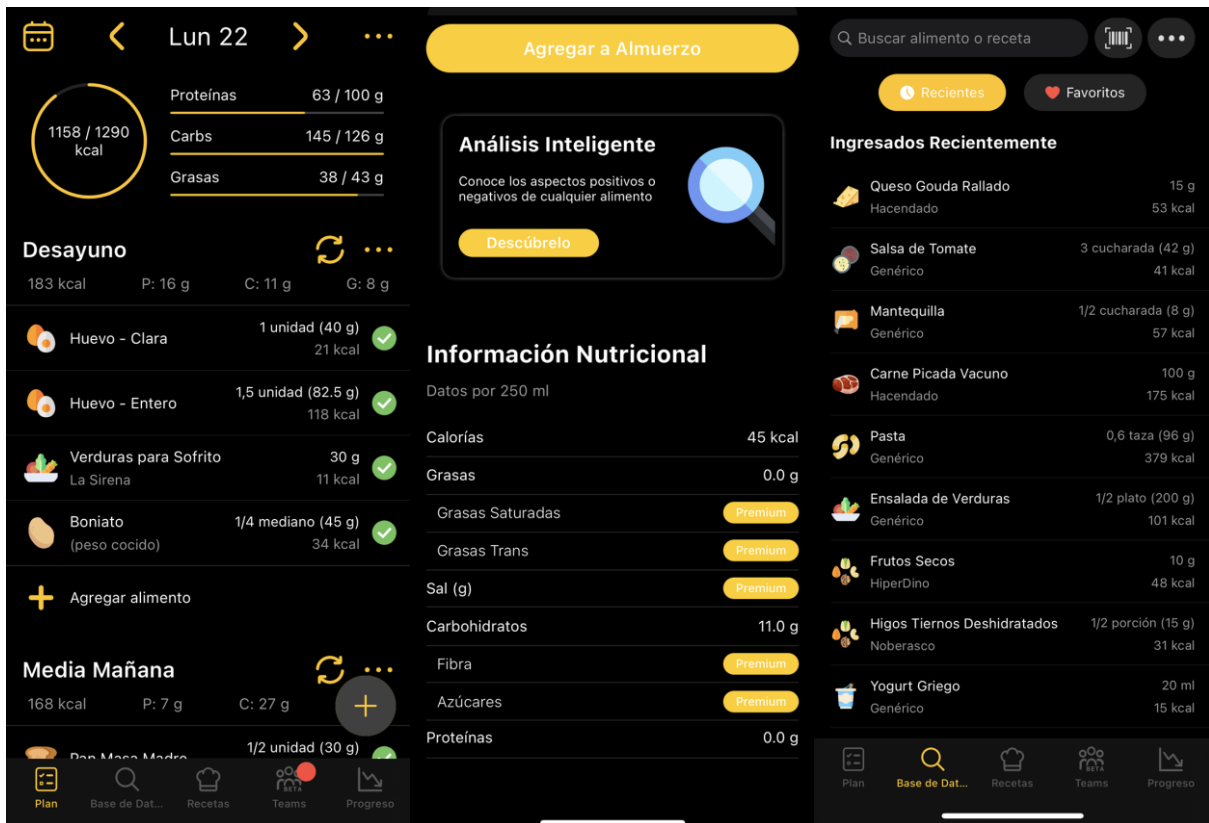


Ilustración 9: Interfaz de Fitia

### 3.5 Obtención de precios

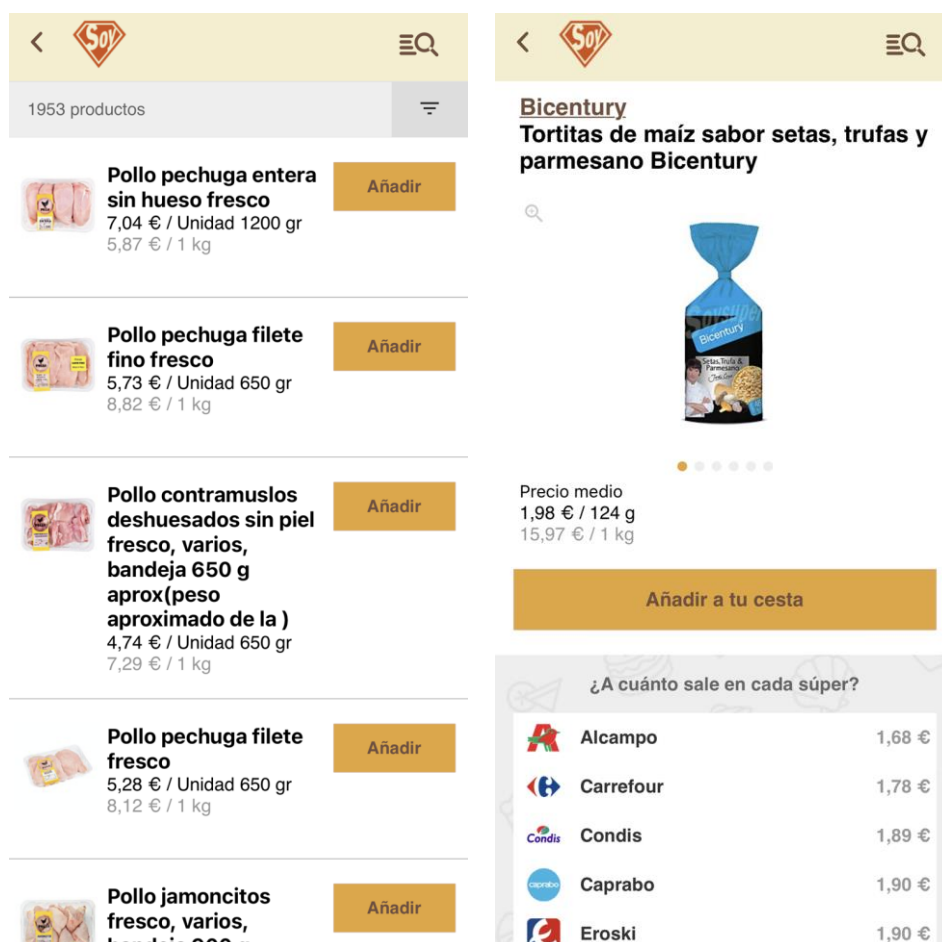
Para obtener los precios de los productos que encontramos en las bases de datos de alimentos se investigaron varios métodos de adquisición. La mayoría de las compañías no hace públicos estos datos, así que se buscaron diversas alternativas para obtener esta información

Buscando desarrollar un producto que aporte valor, no solo sería imprescindible incorporar la dimensión de valores nutricionales y productos que se recomienden proactivamente en función de la dieta que quiera llevar a cabo el usuario, si no también aportar valor directamente proporcionando productos reales que se pudieran comprar en los tres principales supermercados de España (Mercadona, Dia y Carrefour) con los precios actualizados. Los precios incluyen la dimensión económica que nos parecía igual de importante para que el usuario pueda tomar las mejores decisiones desde el punto de vista económico, nutrimental y de conveniencia.

### 3.5.1 Soysuper

Es una plataforma online y en aplicación que permite comparar los precios entre los principales supermercados de España, utilizando la ubicación actual en la que se encuentra el usuario para localizar los supermercados próximos, de modo que permite visualizar cualquier producto y realizar la comparativa del mismo entre los supermercados análogos.

Tiene una gran variedad de supermercados para poder comparar los diferentes precios de los productos, cuenta con un total de 199.341 productos alimenticios y de otros rubros, clasificados por categorías, en cada producto podemos ver su precio, precio de referencia y comparativa si se encuentra en más de un supermercado, pero no cuenta con información nutricional de los artículos.



The screenshot displays the Soysuper app interface. The left panel shows a list of chicken products with their prices and 'Añadir' (Add) buttons. The right panel shows a detailed view of a product, 'Tortitas de maíz sabor setas, trufas y parmesano Bicentury', with its price and a comparison table for other supermarkets.

**Left Panel Product Listings:**

- Pollo pechuga entera sin hueso fresco**: 7,04 € / Unidad 1200 gr, 5,87 € / 1 kg. **Añadir**
- Pollo pechuga filete fino fresco**: 5,73 € / Unidad 650 gr, 8,82 € / 1 kg. **Añadir**
- Pollo contramuslos deshuesados sin piel fresco, varios, bandeja 650 g aprox(peso aproximado de la )**: 4,74 € / Unidad 650 gr, 7,29 € / 1 kg. **Añadir**
- Pollo pechuga filete fresco**: 5,28 € / Unidad 650 gr, 8,12 € / 1 kg. **Añadir**
- Pollo jamoncitos fresco, varios, bandeja 900 g**: **Añadir**

**Right Panel Product Details:**

**Bicentury Tortitas de maíz sabor setas, trufas y parmesano Bicentury**

Precio medio: 1,98 € / 124 g, 15,97 € / 1 kg

**Añadir a tu cesta**

**¿A cuánto sale en cada súper?**

Supermercado	Precio
Alcampo	1,68 €
Carrefour	1,78 €
Condis	1,89 €
Caprabo	1,90 €
Eroski	1,90 €

Ilustración 10: Interfaz de Soysuper

### 3.5.1 Scraping

El web scraping, también conocido como extracción o recolección web, es una técnica para extraer datos de la World Wide Web (WWW) y guardarlos en un sistema de archivos o base de datos para su posterior recuperación o análisis. Por lo general, los datos web se “scrapean” utilizando el Protocolo de transferencia de hipertexto (HTTP) o a través de un navegador web. Esto se puede hacer manualmente por un usuario o automáticamente usando un bot o un web crawler.

Debido al hecho de que una enorme cantidad de datos se genera constantemente en la WWW, el web scraping es ampliamente reconocido como una técnica eficiente y poderosa para recopilar grandes datos (Mooney et al. 2015; Bar-Ilan 2001).

El proceso de “scrapeo” de datos de Internet se puede dividir en dos pasos secuenciales; adquirir recursos web y luego extraer la información deseada de los datos adquiridos. Específicamente, el web scraping comienza con una solicitud HTTP para adquirir recursos de un sitio web en específico.

Esta solicitud se puede formatear en una URL que contenga una consulta GET o en un fragmento de mensaje HTTP que contenga una query POST. Una vez que el sitio web de destino reciba y procese correctamente la solicitud, el recurso solicitado se recuperará del sitio web y luego se enviará de vuelta al programa de web scraping.

El recurso puede estar en varios formatos, como páginas web creadas a partir de HTML, fuentes de datos en formato XML o JSON, o datos multimedia como imágenes, archivos de audio o video (Zhao, 2017).

### 3.5.2 API WhiteBox

La empresa Whitebox en marzo del 2022 hizo público a través de su LinkedIn que, dada la actual situación de inflación y desabastecimiento, muchos los contactaron en relación al data set de productos de supermercados que hay en DataMarket que contiene:

- Los 3 principales supermercados (80% del mercado español).
- 46.000 productos diferentes.
- Evolución del precio con un año de histórico y resolución diaria.
- Metadatos como el tipo de packaging y el precio por unidad.
- Archivo CSV de ~1.5GB.

- Volumen estimado: 50000 registros cada 24 h
- Histórico: disponible desde 2021-03

Dicha información fue distribuida de diferentes maneras, inicialmente se montó una API en Tinybird con el objetivo de construir aplicaciones sobre ellos, o conectar herramientas de visualización pero la empresa al final optó por distribuir el CSV a través de Google drive con actualizaciones diarias y al que se puede acceder desde su página web: <https://datamarket.es/#productos-de-supermercados-dataset> en este punto de acceso podemos descargar el data set completo diariamente con el histórico desde marzo del 2021 con los productos de Mercadona, Día y Carrefour, también se puede descargar la información seccionada mensualmente y por supermercado.

### 3.6 Sistemas recomendadores

Un sistema de recomendación es un sistema inteligente que proporciona a los usuarios una serie de sugerencias personalizadas sobre un determinado tipo de producto. Se trata de uno de los algoritmos de “machine learning” más útiles para las empresas online, mejorando la experiencia de los usuarios y la toma de decisiones de las empresas.

Aunque se ha dicho que una de las características de los sistemas recomendadores es que muestran sugerencias personalizadas para cada usuario, es importante señalar la existencia de los recomendadores “Hand-picked” o recomendadores basados en popularidad. Herramientas muy simples que basa su recomendación sobre una lista cerrada de “favoritos”. Por ejemplo, un sistema que recomienda los artículos, videos o películas más populares. El problema de estos sistemas es que no tiene en cuenta el papel de los usuarios, no siendo un servicio personalizado, aunque sí se reconocen comúnmente como sistemas recomendadores.

Desde un punto de vista técnico, los sistemas recomendadores se pueden entender como un filtro que deja pasar aquella información que le va a resultar de interés al usuario y va a desechar aquella que le pueda resultar indiferente. De esta manera, siguiendo esta definición de filtrado, los recomendadores se pueden clasificar según los siguientes filtros:

- Filtrado demográfico: proporciona recomendaciones atendiendo a características demográficas (edad, género, etc.).
- Filtrado basado en contenido: proporciona recomendaciones en base a aquellos ítems que se consideran similares a los que el usuario mostró interés en algún momento. Para hacer esto, los sistemas de filtrado basados en contenido calculan la similitud entre diferentes ítems.
- Filtrado colaborativo: Estos sistemas se basan en la idea de que a usuarios similares les gustarán productos similares. Este filtro si tiene en cuenta el comportamiento habitual de las personas, dando valor a las similitudes entre usuarios y productos simultáneamente. Este tipo de filtrado se puede categorizar a su vez en sistemas colaborativos basados en memoria y basados en modelos.
  - Métodos basados en memoria: también se pueden subclasificar en “user-base” e “item-based”. Los user-based identifican los productos (ítems) en base a la puntuación dada por otros usuarios, mientras que los item-based identifican usuarios similares en base a la similitud de los productos. Su principal inconveniente es que necesitan un número mínimo de usuarios para realizar la recomendación.
  - Métodos basados en modelos: utiliza algoritmos de aprendizaje automático para encontrar patrones. Mejora el rendimiento en cuanto a la predicción porque da un fundamento más intuitivo. Funcionan usando las evaluaciones de los usuarios afines para calcular la elección del usuario activo
- Métodos de filtrado híbridos: estos métodos hacen una mezcla de los anteriores. Permiten hacer recomendaciones en base a un conjunto más grande de información, siendo habitual utilizar para situaciones de “cold-start”, en las que es difícil realizar recomendaciones con el filtro colaborativo cuando los registros de votación son pocos, por lo tanto, las ventajas de los sistemas híbridos son que a menudo superan la precisión de los sistemas basados en el contenido y de filtrado colaborativo, no sufren el problema del arranque en frío, no tienen un sesgo de popularidad, pueden recomendar elementos con características poco frecuentes y utilizan la retroalimentación implícita para reducir el problema de la escasez. Sin embargo, como contra, es difícil de implementar. (Breve Ramírez, 2021)

En este contexto, podemos definir un filtro como el algoritmo matemático que “decide” cuál es la recomendación óptima basada en los datos que le entregamos. (González, 2022)



## Capítulo 4. Desarrollo del proyecto

### 4.1 Stack tecnológico, arquitectura y justificación

Para la implementación del proyecto contábamos con información nutricional de diversas fuentes, mencionadas en el capítulo anterior, pero se decidió trabajar con Open Food Facts ya que incluye una diversa fuente de alimentos, la mayor parte de sus productos presentan información nutricional muy completa con respecto a los macronutrientes y micronutrientes, y a su vez, presenta gran variedad de productos que se pueden encontrar en los mercados españoles, cuyos datos provienen de la colaboración de los usuarios y de aplicaciones como Yuka.

La información nutricional será complementada con la base de datos de Whitebox para añadir a la información de los productos la marca, si es que aplica, url de donde comprar, el supermercado, el precio, el precio de referencia y fecha que permite hacer un histórico de los precios.

Con estas dos bases de datos unidas utilizando una medida de similitud, se realizará la carga a una base de datos hosteada en Google cloud, haciendo la unión y la carga a la base de datos a través de Pentaho Data Integration, tal y como se muestra en la siguiente Figura, siendo descrito a detalle en los siguientes capítulos.

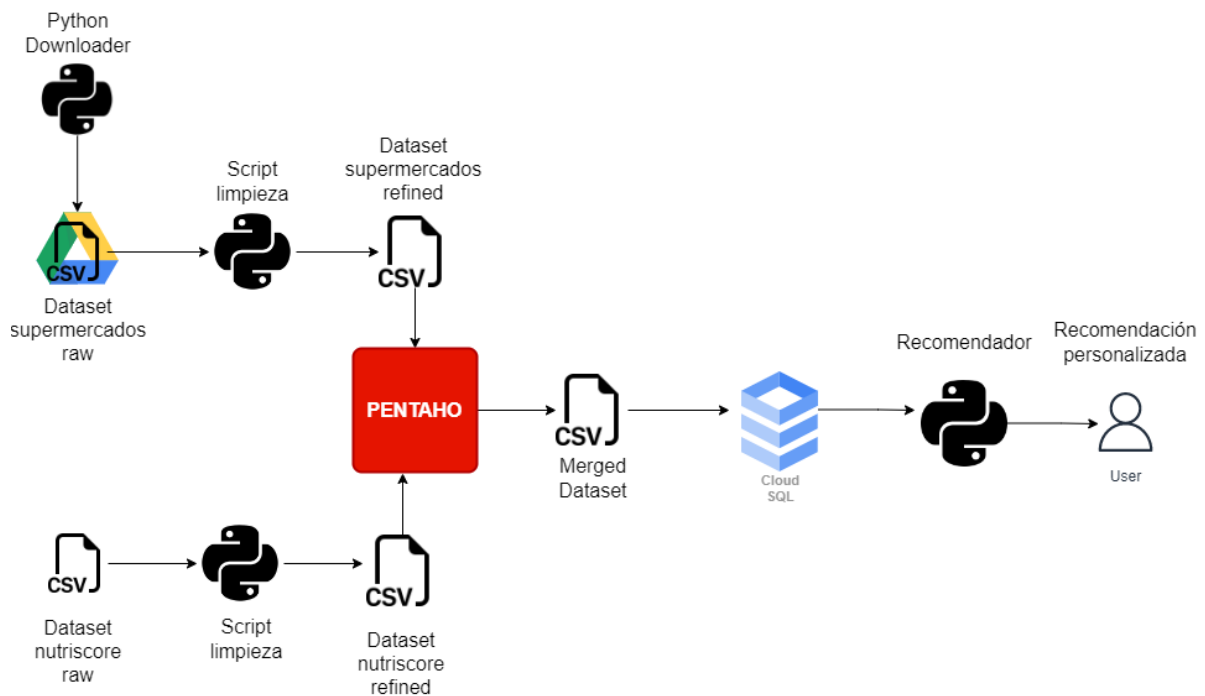


Figura 2: Diagrama de flujo del proceso ETL

Como se ha explicado brevemente, el stack tecnológico usado en este proyecto se compone de scripts de Python, la herramienta de ETL Pentaho, una base de datos SQL hospedado en cloud y el uso de keras como capa de alto nivel para el sistema recomendador.

1. Python, el uso de Python se encuentra presente en casi cualquier paso del TFM, hemos decidido usar este lenguaje y no otro por la versatilidad que ofrece, ya que con él hemos desarrollado conectores a Google drive desde donde descargamos los orígenes de datos, hemos realizado análisis, manipulación y limpieza de datos además de servir como framework para keras y el recomendador. En el caso de los scripts de descarga, limpieza, etc. Se ejecutan en procesos bash que contienen los scripts necesarios de Python, se ha comprobado que es la forma más eficiente sin interrumpir la ejecución.
2. Pentaho es una herramienta de ETL con una interfaz gráfica muy intuitiva que acepta múltiples orígenes de datos y cuenta con todas las funciones requeridas para nuestro proyecto, decidimos usar esta y no otras alternativas como airflow por su simplicidad y porque puede ejecutar sobre una arquitectura local. Pentaho funciona como orquestador de los procesos Python y también realiza las cargas a nuestra BASE DE DATOS SQL que es eje central del proyecto la cual alimenta al sistema recomendador. Pentaho también realiza el matcheo de todos los orígenes de datos a través de algoritmos de similitud, ya que no se disponen de ID únicos en ambos orígenes para llevar a cabo la relación.
3. Base de datos SQL, nuestras necesidades son de una base de datos relacional ya que todos los registros tienen el mismo formato y los mismos campos. Conociendo las necesidades, podíamos optar por MySQL o PostgreSQL, pero al conocer la sintaxis de MySQL y cumplir todas nuestras necesidades optamos por MySQL. Tras probar varias alternativas, optamos por una arquitectura serverless en cloud, Google cloud SQL, la justificación es: escalable, no necesitamos aprovisionar máquinas ni realizar ningún tipo de mantenimiento a la base de datos, se crean copias de seguridad automáticas, no es necesario tener una máquina local exclusivamente para la base de datos y a través de una ip pública limitando las conexiones, nos podemos conectar fácilmente a MySQL desde cualquier origen (Pentaho, IDEs de MySQL)
4. Keras, es una capa de alto nivel de tensorflow que permite definir redes neuronales sin entrar a detalles que se pueden automatizar o predefinir gracias al uso de keras.

El sistema recomendador nos provee de la inteligencia necesaria para preseleccionar alimentos en bruto que después con refinados por reglas de negocio definidas, aun no siendo una pieza central de nuestro proyecto, las recomendaciones por redes neuronales proveen la individualidad necesaria a los usuarios para ajustar la experiencia única a cada comprador en base a sus gustos.

Todos los archivos que contienen estos desarrollos se pueden encontrar en el siguiente repositorio de GitHub: [tfm\\_supermercados \(github.com\)](https://github.com/tfm-supermercados) .

## **4.2 Información nutricional - Open Food Facts**

Open Food Facts nos permite una variedad de opciones para consultar o descargar su base de datos, todas ellas explicadas en su página web. Ofrecen la posibilidad de utilizar MongoDBdump, una API, exportación a CSV y JSON o descargas en formato RDF (Open Food Facts, 2022).

En un primer momento tratamos de descargar nuestra muestra haciendo uso de la API. Sin embargo, nos dimos cuenta de que la API estaba diseñada para realizar consultas sobre productos en específico, por lo que no era una opción que pudiéramos utilizar. Finalmente nos decantamos por descargar su base de datos en formato CSV y más adelante “limpiar” y extraer nuestra muestra a través de un notebook de Python.

Con el CSV de Open Food Facts disponible, en el análisis exploratorio de datos lo primero que se realiza es una selección de las columnas que nos aportan valor, como el nombre, la marca, la categoría del artículo, los países donde se vende el artículo y su información nutricional más relevante.

En dicha selección de columnas hacemos un filtrado de aquellas que no son nulos, en categorías como Nutri-Score, nombre y marca, ya que sin esta información dicho registro no nos aporta el valor que se requerimos. Por último, aplicamos un filtro más sobre la categoría de países para quedarnos con los registros de productos que tengan venta en España.

Con el objetivo de eliminar los productos duplicados se hace un cambio sobre la capitalización del nombre de producto y su categoría a minúsculas, posteriormente se

cambian los caracteres especiales con excepción de la ñ y las letras acentuadas por su letra base, ya que encontramos casos donde un registro de Jamón y jamón tenían la misma marca e información nutricional, pero al eliminar los duplicados no se eliminaban ya que *drop\_duplicates* no es case sensitive.

Lo descrito previamente se realizó de la siguiente manera:

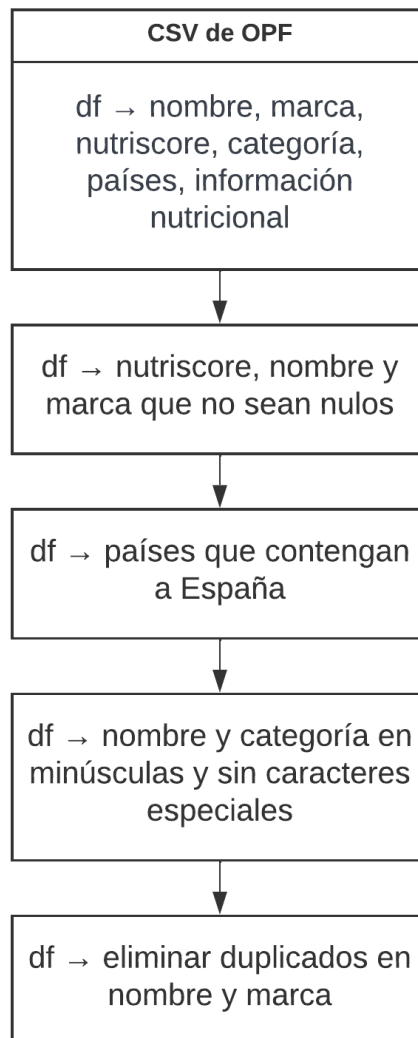


Figura 3: Tratamiento de datos de Open Food Facts

## 4.3 Precios y supermercados

De los métodos revisados en el capítulo 2.5 para obtener precios se decidió desarrollar el web scraping y el uso del data set que Whitebox ofrece a dominio público, la adquisición de datos y su tratamiento se describe a continuación.

### 4.3.1 Web Scraping

La primera opción considerada y que se desarrolló para el supermercado Dia fue el método de web scraping.

Este es un método manual que debe ser lanzado todos los días o según la periodicidad deseada para obtener los datos actualizados. A partir del sitemap del supermercado DIA: <https://www.dia.es/compra-online/sitemap.xml> se listan todos los productos con su link. Un sitemap es un archivo en formato XML que contiene todas las url de una web, generalmente se usan para optimizar el rendimiento de indexadores y buscadores de contenido, esto nos evita tener que paginas las webs para capturar todas las url de producto, haciendo el proceso mucho más eficiente.

Usando regex se extraen la categoría a la que pertenece el producto y su url del XML, una url de producto por ejemplo sería del siguiente formato: <https://www.dia.es/compra-online/despensa/lacteos-y-huevos/yogures/p/113374>, `regex` = "`<loc>(https:\\V.*?\\Vcf)</Vloc>`". La siguiente función permite "scrapear" los productos de dos maneras, a través de un archivo en formato JSON con los productos deseados, o aquellas categorías que elijamos con todos los productos que pertenezcan a ellas.

El resultado del proceso es un CSV que contiene el nombre del producto, su categoría, precio, url y marca temporal de cuando fue "scrapeado". Estos datos junto a la BASE DE DATOS de Open Food son la base que alimentará al sistema recomendador de productos alimenticios. Para realizar esto se sigue el siguiente diagrama y código del Anexo 12: *PyDiaScraper*

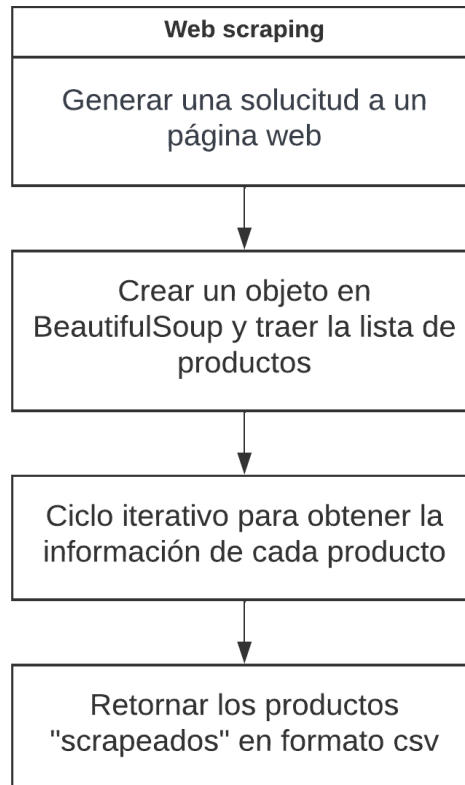


Figura 4: Flujo para realizar web scraping

#### 4.3.2 API White Box

La adquisición del data set está disponible para su descarga en el sitio web de Whitebox <https://datamarket.es/#productos-de-supermercados-dataset> tal como se muestra en la

**Figura 5** el CSV está conformado por las siguientes columnas:

- category: Categoría del producto.
- text\_fieldsdescription: Información adicional del producto (formato de empaquetado, etc.).
- date\_rangeinsert\_date: Fecha de extracción de la información.
- text\_fieldsname: Nombre del producto.
- calculateprice: Precio absoluto del producto en €. En caso de existir algún tipo de descuento aparecerá el menor precio disponible.
- calculatereference\_price: Precio unitario (por unidad de medida del producto, €/Kg, €/L, etc.).
- text\_fieldsreference\_unit: Unidad de referencia del producto (Kg, L, etc.).
- text\_fieldssupermarket: Supermercado al que pertenece el producto.



Figura 5: Plataforma para descarga de White Box

Una vez descargado el data set se procede a realizar el análisis exploratorio de los datos con Python, donde el primer inconveniente que encontramos es que tenemos artículos de todo tipo, no solo alimentos, por lo que la primera transformación se centró en reducir las 1501 categorías que no fueran alimentos utilizando una lista negra de categorías no alimenticias, como jardinería, farmacia, limpieza, etc., después de ejecutar esta limpieza nos quedamos con 953 categorías

A continuación se revisan las variables categóricas donde al igual que con el data set de OPF se pasan a minúsculas las columnas *category* y *name* con el propósito de que al hacer el match con OPF se reduzcan los duplicados y repeticiones debido a que algún producto se encuentre capitalizado y el mismo se pueda encontrar solo con minúsculas, siguiendo esta lógica se han reemplazado todos los caracteres especiales con excepción de la ñ y las letras acentuadas para reducir al máximo los productos que pudieran repetirse por detalles de capitalización o causar conflicto de codificación.

La limpieza de duplicados se debe hacer a conciencia ya que se tienen productos duplicados porque el mismo producto se vende en más de un supermercado o porque se repite en la consulta del día siguiente, así que se toma en consideración el nombre, el

supermercado y la fecha del registro para eliminar los registros duplicados, reduciendo el 5% del data set.

Con los registros sin duplicados procedemos a estudiar los estimados de locación para ver la distribución de los datos con el precio y precio de referencia obteniendo los resultados de la *Tabla 1*.

	Promedio	Mediana	25%	50%	75%	Max
Price	4.91	1.89	1.19	1.89	3	7915.05
Reference price	11.30	5.56	2.54	5.56	10.3	16633.3

Tabla 1: Estimados de locación supermercados

Claramente se nota que hay una diferencia grande entre el cuartil 3 y 4 tanto en el precio como en el precio de referencia, por lo que se procede a ordenar el data set por precio y observar los patrones, con esto pudimos ver que los productos cuya descripción era granel tenían el precio de 99 kilogramos de producto, así que con un *loc* se reemplazó el precio del artículo por el de referencia para que todo el producto a granel fuera el costo de 1 kilogramo de este.

Al revisar nuevamente los datos ordenados por precio vemos que los precios altos son congruentes con el tipo de artículo como vemos en la siguiente *figura*:

```
food.sort_values(by=['price'], ascending=False)
```

	url	supermarket	category	name	description	price	reference_price	reference_unit	date
6724813	https://tienda.mercadona.es/product/58323/jamo...	mercadona-es	charcuteria_y quesos_jamon_serrano	Jamón de bellota ibérico 100% Campo Extremadura	Pieza	479.75	50.50	kg	2022-05-31
7240468	https://tienda.mercadona.es/product/58323/jamo...	mercadona-es	charcuteria_y quesos_jamon_serrano	Jamón de bellota ibérico 100% Campo Extremadura	Pieza	479.75	50.50	kg	2022-07-07
6860971	https://tienda.mercadona.es/product/58323/jamo...	mercadona-es	charcuteria_y quesos_jamon_serrano	Jamón de bellota ibérico 100% Campo Extremadura	Pieza	479.75	50.50	kg	2022-06-09
6744839	https://tienda.mercadona.es/product/58323/jamo...	mercadona-es	charcuteria_y quesos_jamon_serrano	Jamón de bellota ibérico 100% Campo Extremadura	Pieza	479.75	50.50	kg	2022-06-01
6804224	https://tienda.mercadona.es/product/58323/jamo...	mercadona-es	charcuteria_y quesos_jamon_serrano	Jamón de bellota ibérico 100% Campo Extremadura	Pieza	479.75	50.50	kg	2022-06-06

Figura 6: Dataste supermercado ordenado por precio

Posteriormente se revisan los precios de referencia donde de igual manera encontramos artículos del tipo gourmet como el azafrán cuyo precio de referencia el día 26 de agosto del 2022 es de 6.375,00 €/Kg así que a esta categoría no se le aplica ninguna



transformación. Por último, se descarga un CSV con los datos tratados para su carga a la base de datos.

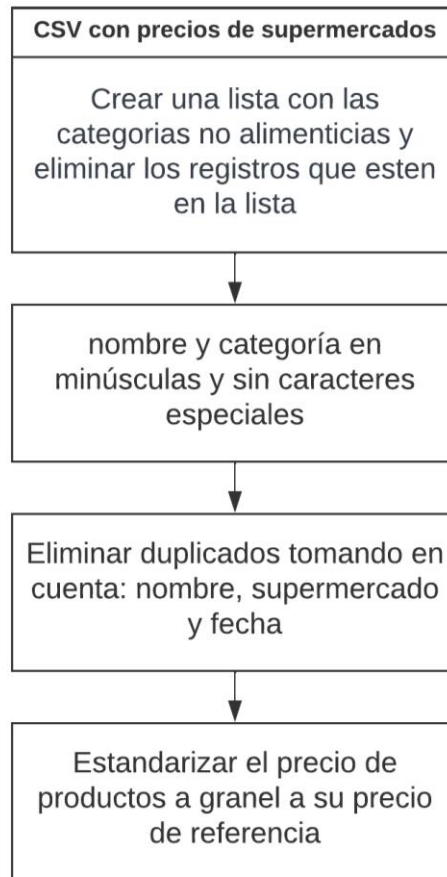


Figura 7: Tratamiento de datos data set supermercados

## 4.4 Carga a la base de datos a través de Pentaho

### 4.4.1 ETL, archivos por lotes.

Dentro de nuestras dos fuentes de datos, consideramos que nuestro data set de supermercados, a diferencia de los datos de Open Food Facts, si necesita ser actualizado más a menudo. Esto se debe a que los precios ofrecidos por cada uno de los supermercados cambian con frecuencia, mientras que los valores nutricionales raramente lo hacen.

En este sentido, hemos optado por automatizar la actualización de datos de supermercado, manteniendo nuestro CSV con los datos nutricionales originales.

Tanto para la descarga y la limpieza del data set de supermercados hemos optado por utilizar diferentes scripts de Python, que se ejecutan a través de dos archivos por lotes (archivos

.bat). En un primer momento nos planteamos ejecutar los scripts de Python mediante un “Shell” en el Job de Pentaho, sin embargo, nos encontramos con que debíamos actualizar el enlace de descarga (ofrecido por la empresa WhiteBox) directamente sobre nuestro código, por lo que no era completamente automático.

Finalmente decidimos dividir nuestro proceso ETL en dos pasos: Por un lado, la descarga y limpieza de datos de supermercado mediante archivos por lotes y, por otro lado, la carga de ese mismo resultado en nuestra base de datos y posteriores transformaciones.

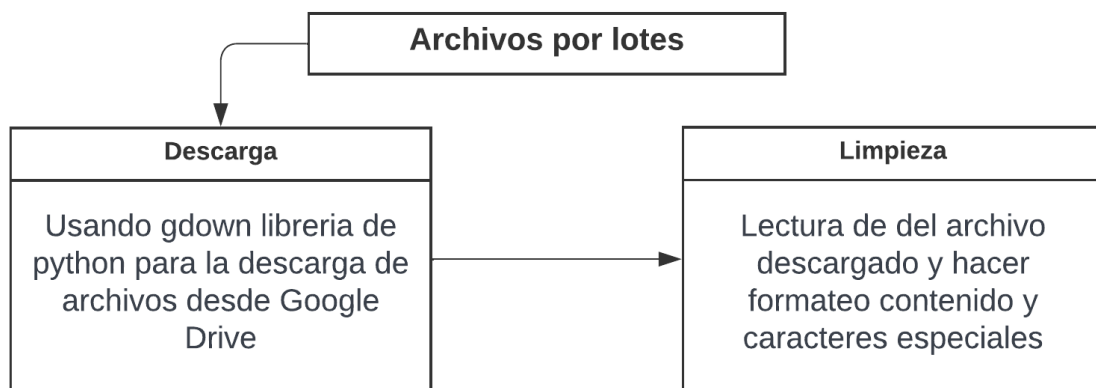


Figura 8: Proceso ETL

Parte fundamental es la limpieza del data set, en nuestro archivo transform, que nos permite descartar los productos ofrecidos por los supermercados que no cumplen la condición de alimentos, además de sustituir caracteres especiales.

En *main* llama al *script* de transformación y de limpieza para tener finalmente un data set con marca temporal de los datos actualizados.

El resultado de este primer paso se guarda en una carpeta en local (output) que se ve de la siguiente manera:

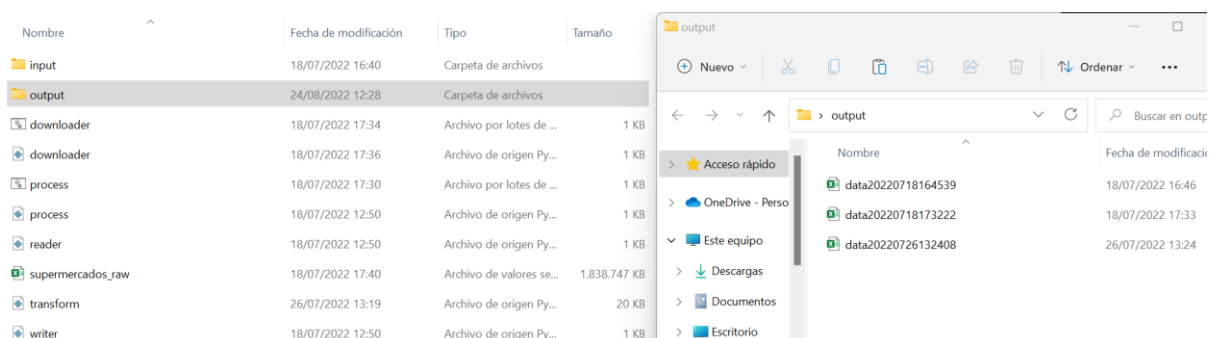


Figura 9: archivos por lotes

## 4.4.2 ETL con Pentaho

Para automatizar el proceso ETL hemos utilizado Pentaho Data Integration en su versión open source. Esta herramienta dispone de aplicaciones potentes para la extracción, transformación y carga de datos en diversas plataformas de bases de datos. Basado en Java, permite interactuar a través de una interfaz gráfica bastante intuitiva (Spoon), permitiendo generar diferentes transformaciones (colección de pasos en un proceso ETL) y trabajos (colección de transformaciones) sin escribir apenas código.

Nuestro objetivo al utilizar Pentaho ha sido generar un trabajo que ejecute todas las transformaciones necesarias para actualizar nuestra base de datos. En este sentido, nuestro archivo KJB (Job) ejecuta las siguientes 3 transformaciones (archivos KTR) de forma consecutiva:

- Actualiza la tabla de productos con la información que ofrece cada supermercado.
- Aplicamos el algoritmo Jaro Winkler, necesario para relacionar nuestras dos fuentes de datos, generando una tabla que llamamos fuzzy\_match, y que utilizaremos en el proceso de unión.
- Por último, ejecuta una sentencia de SQL para insertarla en nuestra tabla merge la unión de ambas tablas.

Es importante recordar que, mientras el proceso de extracción y limpieza de los datos de supermercado se encuentra en un archivo .bat, utilizando scripts de Python. Más adelante, tanto la carga como las transformaciones para realizar el merge se realizan a través del Job de Pentaho.

¿Por qué utilizamos el algoritmo Jaro Winkler?

Una vez cargadas nuestras dos fuentes de datos en nuestro servidor MySQL, tenemos que buscar la manera de unir dos tablas sin columnas en común. Ambas tablas contienen tanto el nombre del producto como la categoría y aunque no coinciden exactamente se aproximan bastante.

Debido a la cantidad de valores faltantes en la columna de categoría nos centraremos únicamente en las columnas de nombre del producto de ambas tablas (name en supermercado y product\_name en openfood).

Pentaho nos ofrece la posibilidad de calcular la similitud entre 2 entidades a través de un fuzzy\_match, una técnica que permite calcular el nivel de paridad entre textos. Para esta

operación Pentaho ofrece varios algoritmos para calcular esta similitud de varias maneras, entre los que destacan los siguientes:

- Levenshtein y Damerau-Levenshtein: calculan la distancia entre dos cadenas observando cuántos pasos de edición se necesitan para llegar de una cadena a otra. El primero sólo tiene en cuenta las inserciones, eliminaciones y sustituciones. El segundo añade la transposición. La puntuación indica el número mínimo de cambios necesarios.
- Jaro y Jaro Winkler: calculan un índice de similitud entre dos cadenas. El resultado es una fracción entre cero, que indica que no hay similitud, y uno, que indica una coincidencia idéntica.
- Needleman Wunsch: calcula la similitud de dos secuencias y se utiliza principalmente en bioinformática. El algoritmo calcula una penalización por lagunas.
- Metaphone, Double Metaphone, SoundEx y Refined SoundEx: son algoritmos fonéticos que intentan emparejar las cadenas en función de su sonido. Cada uno de ellos se basa en el idioma inglés y no sería útil para comparar otros idiomas.

Según los expertos, el algoritmo de Levenshtein funciona mejor que Jaro-Winkler para comparar textos de más de una palabra, sin embargo, Jaro-Winkler es más rápido (Srinivas kulkarni, 2021). Debido al tamaño de nuestra muestra y a la limitación de nuestro equipo preferimos decantarnos por el algoritmo de Jaro-Winkler.

Al aplicar este algoritmo podemos generar una tabla en la que, en este caso, la columna `producto_name` esté presente tanto en la tabla original de supermercados como en la nueva tabla de openfood (ahora llamada `fuzzy_match`) lo que nos permitirá realizar un `inner join` más adelante.

Este paso se encuentra en la transformación llamada también `fuzzy_match`, que a su vez se encuentra integrada en el Job.

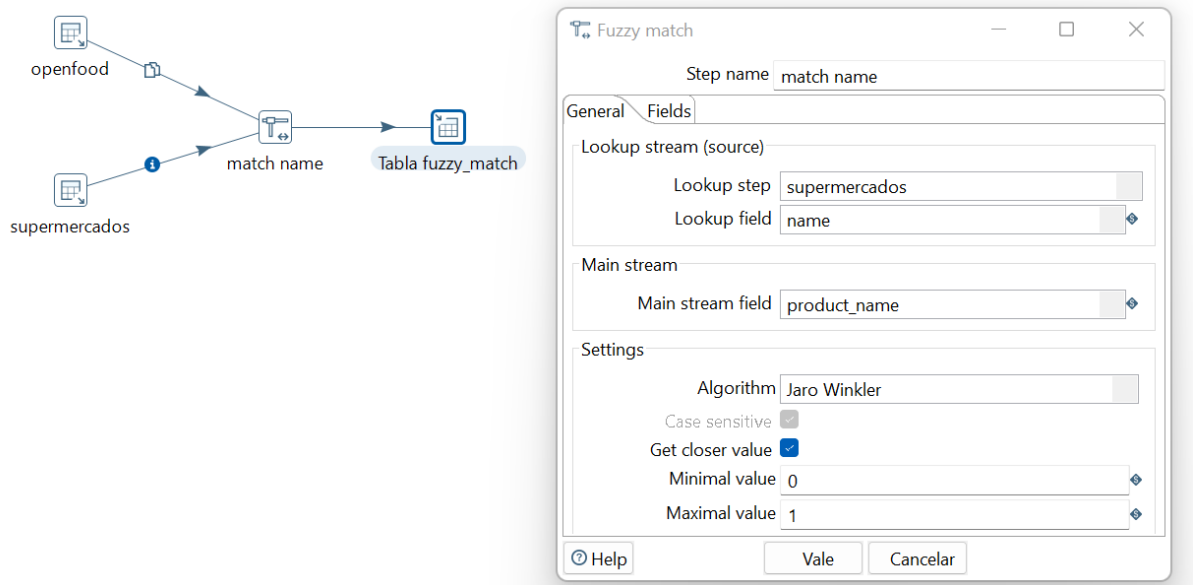


Figura 10: Transformación Fuzzy\_Match

La tabla resultante es la siguiente:

The screenshot shows the 'alimentos' database interface. The 'fuzzy\_match' table is selected, and the following SQL query is executed:

```
1 • SELECT name,product_name,medida_similitud FROM alimentos.fuzzy_match;
```

The result grid displays the following data:

name	product_name	medida_similitud
NESTLE Caja roja bombones bolsa 100 gr	nestle caja roja bolsa bombones 100 gr	0.9736842105263158
Bizcocho de azucar 470 gr	bizcocho de azucar	0.944
Galletas de mantequilla lata 500 gr	galletas de mantequilla	0.9314285714285714
Barra de pan 250 gr	barra de pan	0.9263157894736842
Morcilla de burgos pieza 320 gr	morcilla de burgos de arroz	0.9242831541218638
GREFUSA pipas tijuana bolsa 165 gr	grefusa pipas tijuana	0.9235294117647059
CHEETOS gustosines bolsa 96 gr	cheetos gustosines	0.92
SCHWEPES tonica lata 33 cl	schweppes tonica	0.9185185185185185
SCHWEPES citrus lata 33 cl	schweppes citrus	0.9185185185185185
Bizcocho de limon 350 gr	bizcocho limon	0.9166666666666667
KAIKU Caffé latte light vaso 230 ml	kaiku caffè latte light big	0.9149206349206348
CHEETOS pandilla bolsa 20 gr	cheetos pandilla	0.9142857142857143
Champiñon laminado bandeja 250 gr	champiñon laminado la vereda	0.9140865800865801
SIMON LIFE naranja botella 1.5 l	simon life naranja	0.9125
MIOS chips de maiz y cebolla bolsa 150 gr	mios (chips de maiz & cebolla)	0.9117073170731708

Figura 11: Tabla Fuzy\_match

La última transformación que realizará el Job será la de unir esta tabla con la de supermercados, dando lugar a la tabla que finalmente utilizemos para nuestro sistema recomendador.

La transformación “merge” ejecuta el script de SQL necesario para unir ambas tablas.

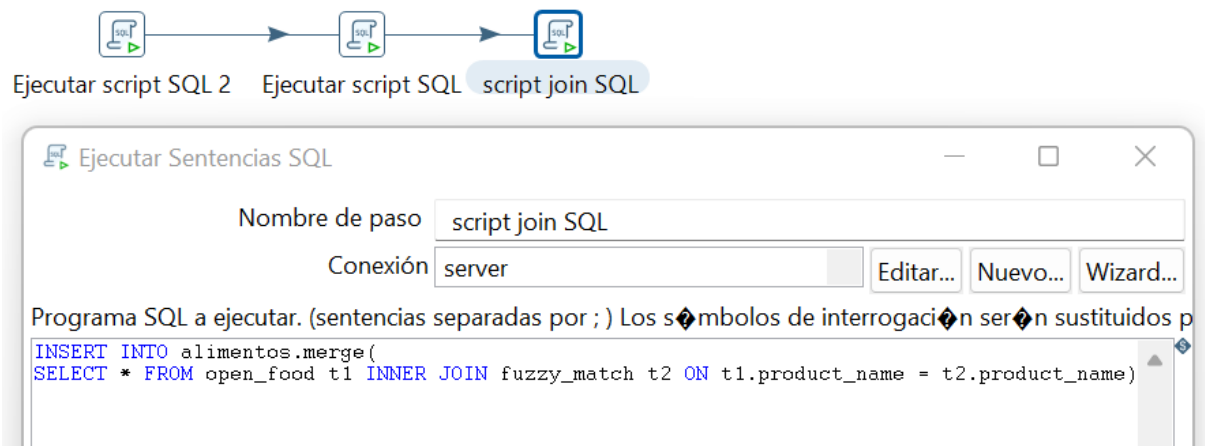


Figura 12: Transformación merge

Lo ideal sería realizar la unión de ambas tablas utilizando una de las herramientas de unión que proporciona Pentaho. Sin embargo, este método no funcionó en nuestro caso y optamos por utilizar scripts de SQL.

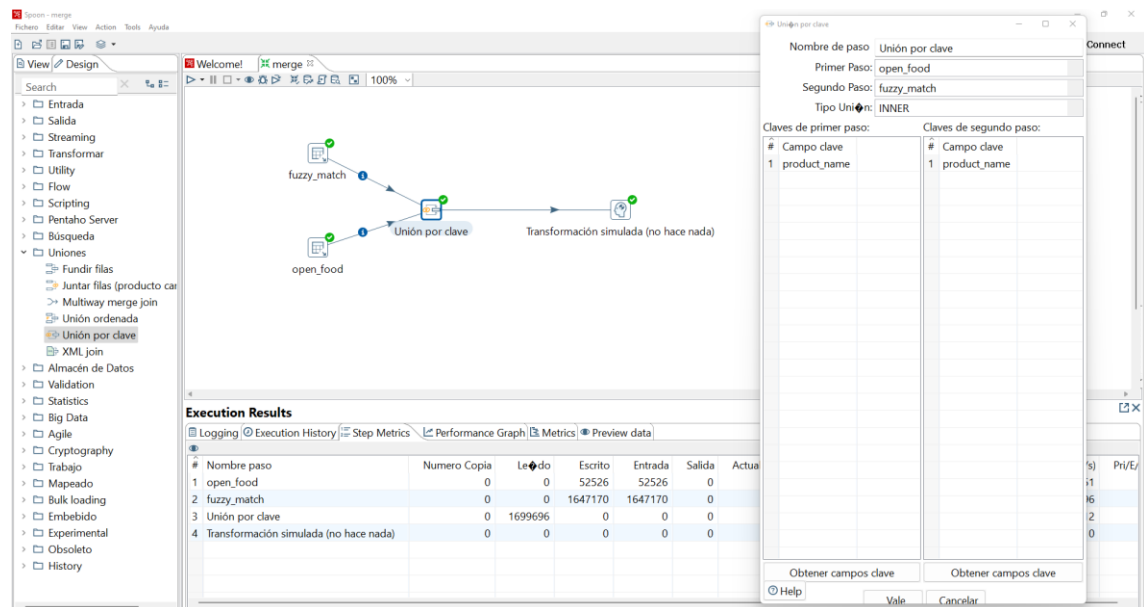


Figura 13: Error en la unión por clave de Pentaho

Y como ya hemos indicado, todo este proceso se encuentra integrado en el siguiente Job:

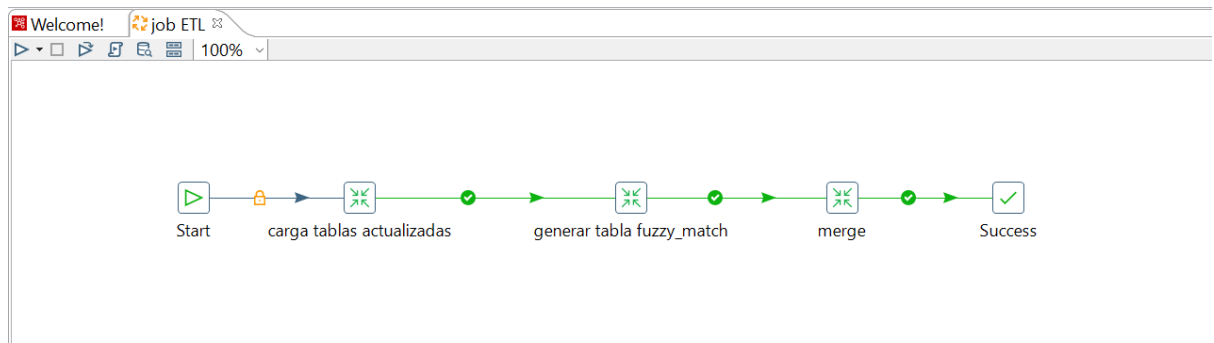


Figura 14: Job ETL

## 4.5 Base de datos

El proyecto basado en datos gira alrededor de una base de datos MySQL creada en Google cloud. Consideramos varias bases de datos con free hosting como PlanetScale, el único inconveniente es que no permitía conexiones a Pentaho o MySQL workbench y acabamos optando por una base de datos MySQL en Google cloud.

Aprovisionamos una máquina con 35gb, 3.75gb de memoria y 2 CPU's que nos aseguran un rendimiento estable y escalado a nuestras necesidades. Las conexiones se hacen a través de IP's autorizadas para asegurar cierto nivel de seguridad y controlar las conexiones.

La base de datos alimentos contiene 4 tablas:

1. fuzzy\_match que funciona como una copia de seguridad de las tablas después del merge.

Nombre de campo	Tipo de dato	Descripción
url	varchar	Enlace al producto
supermercado	varchar	Supermercado al que pertenece el producto
category	varchar	Categoría del producto
name	varchar	Nombre del producto de la tabla supermercado
description	varchar	Descripción del producto
price	float	Precio del producto
reference price	float	Precio por unidad, por

		ejemplo, euro por kilo o por litro
reference_unit	varchar	Unidad del producto
product_name	varchar	nombre del producto a matchear de open food
medida de similitud	double	Índice de 0 a 1 que muestra cómo de similares son los productos matcheados

Tabla 2: tabla alimentos

2. Merge que es la última versión con los datos actualizados de supermercados y alimentos cruzados.

Nombre de campo	Tipo de dato	Descripción
id_food	int	id de producto
product_name	varchar	nombre de producto
brands	varchar	marca
categories_en	varchar	categoría del alimento
countries_en	varchar	país
nutriscore_grade	varchar	Índice Nutri-Score
food_groups_en	varchar	categoría alimenticia primer nivel
main_category_en	varchar	categoría alimenticia segundo nivel
energy-kcal_100	int	calorías por 100 gramos
fat_100g	int	grasas por 100 gramos
saturated-fat_100g	int	grasas saturadas por 100 gramos
carbohydrates_100g	int	carbohidratos por 100 gramos
sugars_100g	int	azúcares por 100 gramos
proteins_100g	int	proteínas por 100 gramos
salt_100g	int	sal por 100 gramos
url	varchar	Link al producto



supermercado	varchar	Supermercado al que pertenece el producto
category	varchar	Categoría del producto
name	varchar	Nombre del producto de la tabla supermercado
description	varchar	Descripción del producto
price	float	Precio del producto
reference price	float	Precio por unidad, por ejemplo euro por kilo o por litro
reference_unit	varchar	Unidad del producto
product_name	varchar	nombre del producto a matchear de open food
medida de similitud	double	Índice de 0 a 1 que muestra cómo de similares son los productos matcheados

Tabla 3: Tabla Merge

3. open\_food, esta tabla contiene la carga de los datos con valores nutricionales de todos los alimentos a los que previamente se hizo una limpieza de datos para solo seleccionar los alimentos que se venden en España.

Nombre de campo	Tipo de dato	Descripción
id_food	int	id de producto
product_name	varchar	nombre de producto
brands	varchar	marca
categories_en	varchar	categoría del alimento
countries_en	varchar	país
nutriscore_grade	varchar	Índice Nutri-Score
food_groups_en	varchar	categoría alimenticia primer nivel
main_category_en	varchar	categoría alimenticia segundo nivel
energy-kcal_100	int	calorías por 100 gramos

fat_100g	int	grasas por 100 gramos
saturated-fat_100g	int	grasas saturadas por 100 gramos
carbohydrates_100g	int	carbohidratos por 100 gramos
sugars_100g	int	azúcares por 100 gramos
proteins_100g	int	proteínas por 100 gramos
salt_100g	int	sal por 100 gramos

Tabla 4: Open Food

- supermercado, la tabla contiene el histórico de precios y alimentos de los 3 supermercados mencionados, la ingesta de esta tabla es diaria.

Nombre de campo	Tipo de dato	Descripción
url	varchar	Link al producto
supermercado	varchar	Supermercado al que pertenece el producto
category	varchar	Categoría del producto
name	varchar	Nombre del producto
description	varchar	Descripción del producto
price	float	Precio del producto
reference price	float	Precio por unidad, por ejemplo euro por kilo o por litro
reference_unit	varchar	Unidad del producto
date	date	Fecha en formato YYYY-MM-DD

Tabla 5: supermercado

## 4.6 Sistema recomendador

Hemos decidido seleccionar los registros que cruzaban con una medida de similitud superior a 0,88 tras un exhaustivo estudio de los registros que hacían el merge correctamente. Tras esta selección tenemos 7.574 productos y nos sentimos cómodos con la calidad de los datos, lo cual es importante para conseguir un buen modelo o sistema de recomendación.

Una vez se obtienen los datos limpios y unificados para el marco de datos de alimentos donde existen las siguientes columnas: 'id\_food', 'product\_name', 'brands', 'nutriscore\_grade', 'main\_category\_en', 'energy\_kcal\_100g', 'fat\_100g', 'saturated\_fat\_100g', 'carbohydrates\_100g', 'sugars\_100g', 'proteins\_100g', 'salt\_100g', 'url', 'supermarket', 'price', 'reference\_price', 'reference\_unit', 'medida\_similitud'. Es necesario el planteamiento de cuál es el sistema recomendador más beneficioso según nuestras necesidades.

Inicialmente recurrimos a un recomendador basado en contenido, ya que no contábamos con una base de usuarios para realizar un filtrado colaborativo. Esencialmente, desarrollamos un recomendador basado en redes neuronales con 2 inputs de entrada, la categoría del alimento y el índice Nutri-Score. Los datos de entrenamiento eran todos los valores nutricionales. El objetivo era que a partir de estos dos inputs nos diera una serie de componentes, que tras aplicar unas reglas de negocio y un perfilado sobre los alimentos que cumplieran con los requisitos del output se ofreciera al usuario opciones que cumplieran sus preferencias. Los datos estaban estandarizados y vectorizados, ya que una red neuronal únicamente acepta datos numéricos. Tras testear el sistema, nos dimos cuenta que el proceso no aportaba la inteligencia suficiente ni el valor añadido que buscábamos integrando un método de algoritmia avanzada, por lo que pivotamos y probamos un recomendador basado en filtro colaborativo.

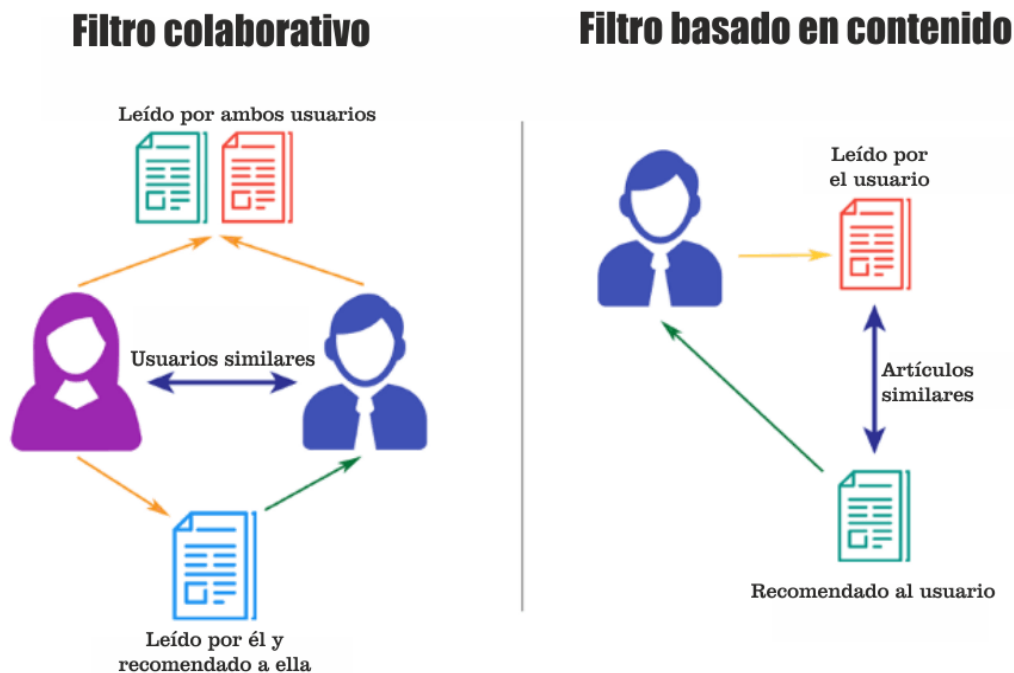


Ilustración 11: Filtros colaborativos y basados en contenido

Como se expone anteriormente el recomendador basado en un filtrado colaborativo está más alineado con la solución que queremos proponer, ya que posee de la inteligencia colectiva de la base de usuarios que contará el sistema cuando salga a producción. Los usuarios votan según su agrado del producto consumido, esto de la siguiente manera:

Rating	Significado
1	Malo
2	Regular
3	Bueno
4	Excelente

Tabla 6: Rating y significado

También es importante resaltar que para la realización de la base de datos de usuarios se utilizó aleatoriedad al momento de asignar la calificación, ya que no queremos condicionar el sistema basándonos en los perfiles especificados, este demo se realizó para un total de 100 usuarios para consecuentemente poder contar con una base de datos con la cual podemos entrenar el sistema. De igual forma, cuando la aplicación cuente con una base de datos reales se debe entrenar de nuevo el modelo para que la calidad de las recomendaciones sea superior y aporten valor.

En base a estas valoraciones se hace la comparativa del historial de un usuario respecto a “n” usuarios identificando patrones de preferencias. Por lo que, es posible predecir cuál será el rating de un producto/s no votado/s en base a su historial y como de parecido es con otros usuarios que han votado ese producto y tienen preferencias similares.

Con este sistema conseguimos obtener una serie de productos que son probables de gustar al usuario, a los que más tarde depuramos con las reglas de perfilado los cuales son:

1. Baja en calorías
2. Volumen
3. Alto rendimiento
4. Balanceado
5. Ahorro.

Con este sistema de recomendación híbrido basado en filtrado colaborativo conseguimos lo mejor de las dos partes, ya que la lista de alimentos propuestos por el recomendador es probable de gustar al usuario y después se aplica la capa de inteligencia extra para proponer la mejor opción dependiendo del objetivo particular que tenga un cliente en un momento concreto.

Como hemos visto el sistema recomendador devuelve como outputs valores nutricionales de productos en función del índice nutricional (Nutri-Score) y la categoría de alimento que deseemos. Estos valores se corresponden a los campos de la tabla merge: sugars\_100g, carbohydrates\_100g, price, etc. Con este output, es posible filtrar qué alimentos son candidatos dentro de la categoría alimenticia seleccionada, después para refinar el resultado se han definido unos criterios a nivel de negocio que llamamos perfiles alimenticios.

El usuario deberá seleccionar previo a recibir la recomendación cuál es su objetivo entre todos los perfiles disponibles, esto se explicará a continuación:

1. Perfil centrado en la pérdida de peso: como el propio nombre indica, los tipos de alimentos recomendados son bajos en calorías, nutritivos. Los criterios definidos son los siguientes.

Concepto	Valor
Nutriscore	A, B
Kcal	<130
Fat	<1
Saturated Fat	0
Carbohydrates	<6
Sugars	<5
Proteins	NA
Salts	<1
Price	NA

Tabla 7: Perfil centrado en la pérdida de peso

2. Perfil centrado en la creación de masa muscular: alimentos altos en proteínas e hidratos.

Concepto	Valor
Nutri-Score	A, B, C, D
Kcal	NA
Fat	<3
Saturated Fat	<1
Carbohydrates	NA
Sugars	<10
Proteins	Ordenar en función de proteínas DESC
Salts	<5
Price	NA

Tabla 8: Perfil centrado en la creación de masa muscular

3. Perfil centrado en el alto rendimiento deportivo: alimentos con alta carga de hidratos y azúcar.

Concepto	Valor
Nutri-Score	A, B, C
Kcal	Ordenar en función de kcal DESC
Fat	<3
Saturated Fat	<1
Carbohydrates	Ordenar en función de carbohydrates DESC
Sugars	Ordenar en función de sugars DESC
Proteins	NA
Salts	NA
Price	NA

Tabla 9: Perfil centrado en el alto rendimiento deportivo

4. Perfil centrado en una dieta balanceada: Deben ser productos económicos a la vez de sanos y nutritivos.

Concepto	Valor
Nutri-Score	A, B, C, D
Kcal	<86
Fat	<1
Saturated Fat	0
Carbohydrates	<6
Sugars	<1
Proteins	<4
Salts	<0.5
Price	=<2

Tabla 10: Perfil centrado en una dieta balanceada

5. Perfil centrado en el ahorro económico.

Concepto	Valor
Nutri-Score	NA
Kcal	<245
Fat	<3
Saturated Fat	<1
Carbohydrates	<28
Sugars	<4
Proteins	<12
Salts	<1
Price	Ordenar en función de price ASC

Tabla 11: Perfil centrado en el ahorro económico

Para definir estos perfiles nos hemos basado principalmente en los estadísticos descriptivos de nuestro data set, estudiando la distribución de los valores nutricionales y precio, junto a sus rangos intercuartílicos. Esto sumado a la sensibilidad y conocimiento del tema que estamos tratando, los perfiles desarrollados fueron 5 que recogen las principales preocupaciones alimenticia de la mayoría de los usuarios que usarían el recomendador de alimentos.

Los descriptivos están comprendidos entre 0 y 100 a excepción de kcal y price, se observa que grasas saturadas y sal tienen una concentración baja en la mayoría de los alimentos, por el contrario, los carbohidratos es el componente más abundante en la mayoría de los alimentos. De media en los supermercados estudiados el precio es de 2,3 euros.

Descriptivo	kcal	fat	saturated fat	carbohydrates	sugars	proteins	salt	price
media	262.9	12.79	4.66	23.11	9.53	9.12	1.2	2.3
min	0	0	0	0	0	0	0	0.13
25%	86	1	0	2	1	2	0	1
50%	222	6	1	9	2	7	1	1.55
75%	374	19	5	46	9	14	1	2.39
max	61304	100	95	100	100	100	100	351

Tabla 12: Estadísticos descriptivos



## Capítulo 5. Resultados

### 5.1 Resultados esperados

Se espera crear un sistema de recomendación alimenticio que entregue diversas alternativas de productos al usuario basado en diversos objetivos nutricionales y económicos que éste pueda tener, este sistema contará con la información nutricional completa del producto, así como dónde adquirirlo y a qué precio se espera que lo encuentre.

Este sistema será creado con una red neuronal basada en el usuario, entrenada y alimentada de la unión de un data set que provee la información nutricional del producto y otro que brinda la información económica actualizada diaria o semanalmente. La información será gestionada en una base de datos en la nube

### 5.2 Resultados

Se consiguió una base de datos que se puede actualizar diariamente para entregarle al usuario el valor económico de su cesta de compra personalizada a sus necesidades y metas alimenticias. En esta base de datos se tienen un total de 7.574 productos distintos con una medida de similitud igual o mayor al 88% entre las dos tablas que conforman dicha base de datos, la cantidad de productos pueden ir variando, dependiendo del día de la consulta ya que los supermercados pueden añadir o descontinuar algún producto.

Se creó una red neuronal con Keras de filtro colaborativo usuario a usuario, que se puede revisar en el *Anexo 8*, esta red recibe como entrada el usuario y la puntuación que le ha dado a diversos alimentos y entrega productos que le puedan gustar al usuario basándose en las puntuaciones que otros usuarios con gustos similares le hayan dado a otros productos. El filtro colaborativo (usuario a usuario) basado en memoria trata de imitar el comportamiento de los seres humanos cuando buscan una recomendación, buscar personas con gustos similares y seguir su recomendación.

Esta red calcula la similitud entre los 100 usuarios sintéticos, elige a los usuarios más próximos, estima el valor de los artículos valorados por los usuarios más cercanos y que el usuario no haya votado, para entregar los productos mejor estimados, como lo vemos en la Figura 15 tenemos los alimentos mejor calificados por el usuario 72, estos artículos no necesariamente tienen que ser saludables o de buen puntaje en el Nutri-Score, y de acuerdo a sus preferencias el sistema genera un dataframe de N recomendaciones de productos que le pueden gustar al usuario.

Recomendaciones para el usuario: 72

=====

Alimentos que el usuario ha calificado positivamente

-----

anacardo crudo : b

judia verde plana : a

aceitunas manzanilla gazpachas con hueso : d

atun en aceite de girasol : c

ciruelas desecadas sin hueso : a

-----

Recomendaciones para el usuario

-----

	id_food	alimentos	nutriscore
0	3874	chuleta de lomo de cerdo	b
1	47147	panecillo	b
2	5711	mollete de antequera	c
3	3739	alas partidas de pollo	c
4	25911	agua mineral natural	a
5	15518	harina de espelta integral	a
6	25293	azucar blanco	d
7	4658	lechuga iceberg	a
8	24901	cerveza sin alcohol	c
9	15383	alubias con chorizo	c
10	33798	tinto de verano limon sin alcohol y sin azucar	b

Figura 15: Output de la red neuronal

Una vez el sistema recomendador entrega las sugerencias generales de productos, al usuario se le pregunta cuál es su objetivo nutricional entre:

1. Baja en calorías
2. Volumen
3. Alto rendimiento
4. Balanceado
5. Ahorro

Siguiendo las reglas previamente explicadas en el *capítulo 3.6* el recomendador le entregará al usuario una sugerencia personalizada de artículos para su cesta de compra, obteniendo la información del artículo, sus valores nutricionales más importantes, un link de compra, el supermercado donde encontrará sus artículos y un precio actualizado para poder tomar decisiones informadas para sus hábitos de consumo alimenticio.

Siguiendo con las recomendaciones del usuario 72, si dicho usuario seleccionará un perfil bajo en calorías el sistema le entregaría las siguientes sugerencias

id_food	alimentos	nutriscore	product_name	brands	nutriscore_grade	main_category_en
33798	tinto de verano limon sin alcohol y sin azucar	b	tinto de verano limon sin alcohol y sin azucar	sandevide	b	Non-Alcoholic beverages
50587	tomates	a	tomates	dia	a	Tomatoes
30865	champiñon laminado	a	champiñon laminado	unide	a	Champignon mushrooms

energy_kcal_100g	fat_100g	saturated_fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g
1.0	0.0	0.0	0.0	0.0	0.0	0.0
18.0	0.0	0.0	4.0	3.0	1.0	0.0
18.0	0.0	0.0	1.0	0.0	2.0	1.0

url	supermarket	price	reference_price	reference_unit
https://tienda.mercadona.es/product/66734/tint...	mercadona-es	1.18	0.59	l
https://tienda.mercadona.es/product/69976/toma...	mercadona-es	1.99	1.99	kg
https://tienda.mercadona.es/product/69519/cham...	mercadona-es	1.60	5.00	kg

Figuras 16: productos recomendados bajo en calorías

Si el mismo usuario cambiará sus preferencias a un perfil de ahorro el sistema le arrojará las siguientes recomendaciones:

id_food	alimentos	nutriscore	product_name	brands	nutriscore_grade	main_category_en
24901	cerveza sin alcohol	c	cerveza sin alcohol	estrella de levante	c	Non-alcoholic beers
37458	maiz dulce en grano	a	maiz dulce en grano	el corte ingles	a	Canned sweet corn
4658	lechuga iceberg	a	lechuga iceberg	el mercado de aldi	a	Fresh vegetables
48170	menestra de verduras	a	menestra de verduras	eroski	a	Frozen-vegetable-mixes
33798	tinto de verano limon sin alcohol y sin azucar	b	tinto de verano limon sin alcohol y sin azucar	sandevide	b	Non-Alcoholic beverages
30865	champiñon laminado	a	champiñon laminado	unide	a	Champignon mushrooms
50587	tomates	a	tomates	dia	a	Tomatoes

energy_kcal_100g	fat_100g	saturated_fat_100g	carbohydrates_100g	sugars_100g	proteins_100g	salt_100g
20.0	0.0	0.0	5.0	2.0	0.0	0.0
73.0	1.0	0.0	11.0	5.0	3.0	0.0
14.0	1.0	0.0	1.0	1.0	1.0	0.0
55.0	0.0	0.0	9.0	3.0	3.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0
18.0	0.0	0.0	1.0	0.0	2.0	1.0
18.0	0.0	0.0	4.0	3.0	1.0	0.0
42.0	0.0	0.0	9.0	9.0	1.0	0.0
url	supermarket	price	reference_price	reference_unit		
https://tienda.mercadona.es/product/27136/cerv...	mercadona-es	0.68	2.06	l		
https://www.carrefour.es/supermercado/maiz-dul...	carrefour-es	0.77	1.92	kg		
https://tienda.mercadona.es/product/69670/lech...	mercadona-es	0.79	3.16	kg		
https://www.carrefour.es/supermercado/menestra...	carrefour-es	1.05	1.05	kg		
https://tienda.mercadona.es/product/66734/tint...	mercadona-es	1.18	0.59	l		
https://tienda.mercadona.es/product/69519/cham...	mercadona-es	1.60	5.00	kg		
https://tienda.mercadona.es/product/69976/toma...	mercadona-es	1.99	1.99	kg		
https://www.carrefour.es/supermercado/naranja-...	carrefour-es	3.16	0.79	kg		

Figuras 17: productos recomendador ahorro

Vemos entonces que para el mismo usuario cambiando entre el perfil bajo en calorías que prioriza un enfoque nutricional antes que el económico y el perfil de ahorro que prioriza el costo del producto antes que el nutricional, encontramos una coincidencia de 3 artículos ya que las propuestas entregadas vendrán de un mismo conjunto de recomendaciones y serán filtradas y priorizadas por el conjunto de reglas de cada perfil.

A continuación, se muestra una matriz de resultados con usuarios ficticios a modo de presentación de resultados, se simulan posibles alimentos recomendados a partir del input de productos que han valorado los propios usuarios, el sistema recomendador arroja una nueva data frame que será filtrado y priorizado según el perfil que el usuario elija y le brindará recomendaciones tales como las mostradas en la segunda tabla.

Usuario	Producto 1	Producto 2	Producto 3	Producto 4	Producto 5
23	queso mezcla tierno	cebolla troceada congelada	boquerones en vinagre	jamoncitos de pollo congelados	yogur natural de cabra
62	cacahuete frito con miel	coliflor ultra congelada	mantequilla ecológica pura de irlanda	tomate triturado	champiñones laminados
84	maltín polar	néctar naranja	mayonesa ligera	salmón ahumado	pan tostado con ajo y perejil
93	patatas fritas churrería	bizcocho de cacao	granizado de limón	batido sabor vainilla	filetes de caballa del sur en aceite de oliva
65	agua mineral	salchichón extra loncheado	mortadela con aceitunas	manzana	pan de molde sin corteza

Tabla 13: Usuarios y sus productos calificados positivamente para matriz basada en contenido

Usuario	Perfil	Recomendación 1	Recomendación 2	Recomendación 3	Recomendación 4
23	Volumen	arroz redondo	yogur natural edulcorado 0%	champiñón laminado	coliflor
62	Balanceado	judía verde plana	sopa juliana	tinto de verano limón sin alcohol y sin azúcar	agua mineral natural manantial de solares
84	Bajo en calorías	coliflor	requesón	ensalada variada	champiñones laminados
93	Alto rendimiento	tortitas de maíz	fideos con quinoa sin gluten	harina de maíz	mango
65	Económico	higo pico	tomate rallado	obleas para helado	yogur sabor fresa

Tabla 14: Usuario, perfil, recomendaciones entregadas por el sistema

Adicionalmente se crearon dashboards donde el usuario puede ver todo el listado de productos que cumplen con el perfil que desea, para poder complementar su cesta de compra de una manera informada ya que en estas visualizaciones no solo tiene la información global de los productos, sino que además puede ver:

- El costo promedio por supermercado para los artículos en el perfil de elección.
- Los promedios nutricionales del grupo para comparar sus artículos.
- La distribución de productos según su Nutri-score y su grupo alimenticio.

Estos dashboards podrán ser filtrados como se muestra en la figura 18:

## Filtros

**Grupos Alimentos** ▼

All
▼

**Precio** ▼

0.19

23.67

**Calificación Nutri-Sc...** ▼

All
▼

**Supermercado** ▼

All
▼

Figura 18: Filtros para dashboard

A continuación, podemos ver dos ejemplos de las visualizaciones para los perfiles nutricionales de alto rendimiento y balanceado, logrando que nuestro proyecto sea una herramienta amigable y visual para el usuario.

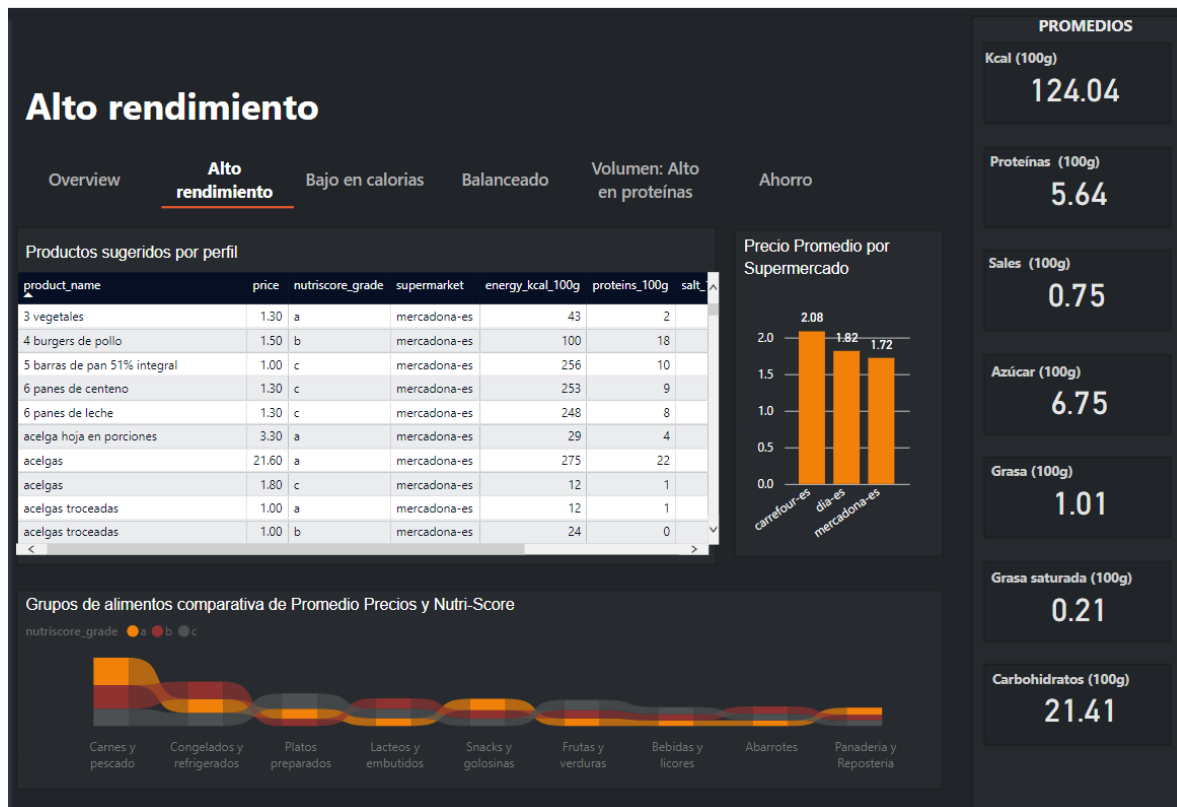


Figura 19: visualización de perfil: alto rendimiento



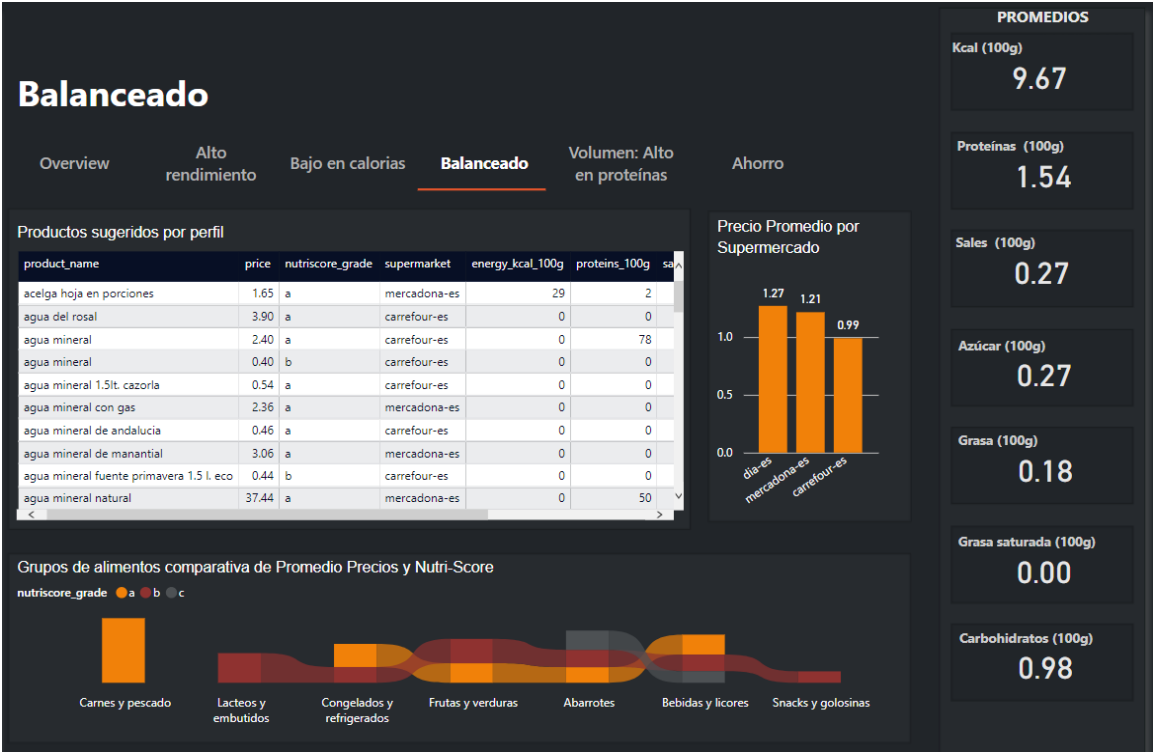


Figura 20 visualización de perfil balanceado

## **Capítulo 6. Conclusiones**

### **6.1 Conclusiones del proyecto**

El estudio de distintas formas de realización de los sistemas recomendadores fue clave para poder acertar el proceso de elección. Después de distintas pruebas comprobamos que el aplicado inicialmente un filtro colaborativo nos beneficia porque el algoritmo se basa en el comportamiento de las personas, y consecuentemente agregarle una capa de inteligencia que se acopla a los objetivos del usuario.

Con este sistema de recomendación híbrido basado en filtrado colaborativo conseguimos lo mejor de las dos partes, ya que la lista de alimentos propuestos por el recomendador es probable de gustar al usuario y después se aplica la capa de inteligencia de negocio para proponer opciones dependiendo del objetivo particular que tenga el cliente en un momento concreto.

Se concluye que el sistema recomendador es funcional ya que devuelve el resultado esperado. Este sistema permite al usuario basarse en su preferencia y en sus objetivos nutricionales para generar una lista de productos que se adapte a sus necesidades, tanto económicas como nutricionales, esta lista puede ser actualizada cada día o a la semana dependiendo de las preferencias del usuario.

Para la realización de este realizamos un reporte donde muestra el antes y después de las recomendaciones basadas en un perfil específico, esto va a permitir al usuario conocer cuál es el proveedor de productos de mejor calidad y a un precio más económico. De igual forma concluimos que cumplimos con nuestro objetivo ya que estamos optimizando el proceso de compra y de elección de supermercado a unos minutos que se realice la ejecución del programa y se impriman los alimentos.

La creación de un reporte de visualización de resultados es de suma importancia ya que se muestra información general sobre algunos datos concluyentes del trabajo, en este proyecto se implementó a través de Power BI y cuenta de 6 pantallas, la primera nos despliega la información general de los productos, como se ve en la figura 21.



Figura 21: Información general en el dashboard

Se conoce que la totalidad de los registros después de su trabajo de limpieza y estandarización es de más de 6.500 valores únicos y la distribución de ellos en cuestión de supermercado el que contiene mayor cantidad de productos es Mercadona (65.55%), seguido por Carrefour (31.91%) y por último El Día (2.54%).

Se realizó una categorización para poder generalizar los datos las cuales son: Abarrotes, bebidas y licores, carnes y pescado, congelados y refrigerados, frutas y verduras, lácteos y embutidos, panadería y repostería, platos preparados y por último Snacks y golosinas. Ver *anexo 11.2*. De esta forma, podemos obtener el precio promedio para cada categoría mostrado en la siguiente tabla.

<b>Categoría</b>	<b>Precio promedio</b>
Carnes y pescado	3.87 €
Lácteos y embutidos	3.82 €
Congelados y refrigerados	2.99 €
Platos preparados	2.29 €
Abarrotes	1.88 €
Frutas y verduras	1.80 €
Snacks y golosinas	1.75 €
Bebidas y licores	1.51 €
Panadería y repostería	1.49 €

*Tabla 15: precio promedio por categoría*

Por otra parte, se concluye que los productos con calificación Nutri-Score están repartidos de la siguiente forma: D(26.46%), C(24.94%), A(19.36%), B(16.85%) y E(12.39%), mostrados de manera ascendente a descendente. Con lo que podemos concluir que las opciones mostradas en los supermercados generalmente son productos que no son beneficiosos para nuestra salud, ya que si realizamos una agrupación de “A y B” sale un sumatorio total de 36.21% y el 63.79% restante son productos que consideramos regulares a malos. *Ver anexo 11.3*

Asimismo, calculamos el precio promedio de la totalidad de productos por supermercado. En la primera posición tenemos a Carrefour siendo el precio más elevado con 2.60€, seguido por Mercadona con 2.24€ y por último al día con 2.02€. De igual forma se debe tomar en cuenta que la cantidad de productos evaluados en los supermercados es distinta entre sí.

Por último, tenemos la clasificación Nutri-Score por el promedio de precio de cada uno de los supermercados, mostradas en la tabla a continuación. *Ver anexo 11.4*

Supermercado	a	b	c	d	e
Carrefour	1.94	2.31	2.65	2.77	3.22
Mercadona	1.69	2.11	1.91	2.30	4.14
Dia	1.92	1.76	2.27	2.16	1.99

Tabla 16: Nutri Score, Supermercado y precio promedio

En la anterior tabla podemos concluir que la agrupación “A y B” que calificamos como más deseable a la hora de tomar en cuenta para una dieta saludable se encuentra en mejor precio en Mercadona. Por lo cual, Mercadona es más barato para adquirir lo necesario para una dieta saludable.

Podemos extraer las siguientes conclusiones:

- Los productos con mejor calificación en cuanto a su aporte nutricional son los que se encuentran en el supermercado Mercadona, seguido de Dia y finalmente Carrefour.
- Los productos con peor calificación en cuanto a su aporte nutricional son los que se encuentran en el supermercado Carrefour, seguido de Mercadona y finalmente Dia.
- Mercadona es el supermercado más económico en cuanto a productos saludables.
- Carrefour es el supermercado más económico en cuanto a productos no saludables (E).
- El supermercado Dia ofrece productos a un precio medio.

## 6.2 Competencias desarrolladas o aplicadas

1. Planificar, diseñar y poner en marcha un proyecto avanzado en grupo, colaborando de forma activa en la consecución de objetivos comunes.
2. Capacidad de estudio, síntesis y autonomía suficiente para desarrollar cada uno de los objetivos básicos de investigación.
3. Capacidades asociadas al trabajo en equipo: cooperación, liderazgo, flexibilidad, comunicación, saber escuchar, entusiasmo, etc.
4. Profundizar nuestro conocimiento referente al impacto y las nuevas oportunidades que surgen del desarrollo de las tecnologías big data, aplicando estos conocimientos a nuestro trabajo.
5. Acudir a herramientas como notion para organizar las tareas del equipo en remoto, aplicando una metodología agile.

6. Desarrollo de competencias referentes al análisis estadístico, tal como: analítica descriptiva básica de datos estructurados y la extracción de conocimiento basado en datos.
7. Realización de un análisis exploratorio (EDA) a través de librerías como Pandas y Numpy en notebooks de Python.
8. Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas.
9. Saber aplicar los conocimientos adquiridos referentes al desarrollo de código en Python y SQL, además de ser capaz de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios.
10. Capacidad para proponer soluciones imaginativas a problemas encontrados, así como ser capaz de promover la innovación e identificación de alternativas contrapuestas a los métodos comúnmente utilizados.
11. Entender los distintos problemas de calidad de los datos con los que se enfrenta un científico de datos y las posibilidades de evitarlos o mitigar su impacto.
12. Identificar y utilizar herramientas software especializadas para el tratamiento de grandes volúmenes de datos en distintos contextos.
13. Capacidad para idear un proyecto con perspectiva de negocio.
14. Aplicar diferentes técnicas para la limpieza de los datos manteniendo las “buenas prácticas”.
15. Realizar scraping de datos para extraer información de las webs de supermercados
16. Diseñar e implementar sistema de ETL automatizado mediante una de las herramientas trabajadas en el máster (Pentaho data integration).
17. Comprender y manejar nuestra propia base de datos alojada en un servidor compartido de MySQL.
18. Entender el funcionamiento de los sistemas recomendadores, además diseñar el propio.
19. Diseñar nuestra propia red neuronal a través de la librería Keras, e incorporarlo a nuestro sistema recomendador.
20. Manejar uno de los softwares de analítica y visualización empresarial más utilizados como Power BI para generar informes y presentar resultados.

## Capítulo 7. Sigüientes pasos

El proyecto realizado para el desarrollo de un recomendador de alimentación saludable para productos alimenticios de los principales supermercados españoles es susceptible de posibles mejoras que se describen a continuación.

1. En esta primera iteración no contamos con datos de usuarios ni base de usuarios reales, por lo que hemos añadido al recomendador un filtro colaborativo simulado en forma de demo. Por una parte, el recomendador debe tener la inteligencia necesaria para recomendar productos en función de las preferencias personales del usuario, esto implica recoger el histórico de interacciones que ha realizado la persona con el sistema y aprender de ellas para que así al entrenar con los gustos del usuario se ofrezca un mejor servicio ajustado a sus necesidades. Para su implementación sería necesario un sistema de login para poder crear un id de usuario donde en una base de datos se recoja todo su histórico.
2. Por otro lado, la implementación de un filtro colaborativo proporciona al sistema la inteligencia suficiente, como para inferir gustos de usuarios que todavía no han tenido interacción suficiente con el recomendador, esto lo consigue gracias a la creación de perfiles similares que ha aprendido con el histórico de la base de usuarios existente e intenta aproximar la limitada interacción de una nueva persona con uno de los perfiles.
3. Cold start, para evitar el problema de “inicio en frío”, tras el usuario elija el perfil de uso que prefiere (pérdida de peso, balanceado...), se le ofrece una serie de productos, al elegir los que más prefiere, se puede afinar la recomendación sin necesidad de entrenar previamente al sistema con el histórico.
4. Integración con Google shopping, el servicio está principalmente pensado para integrarlo con herramientas de listas de la compra o en el carrito de la compra de los diferentes supermercados que ofrecen compra online. La opción más factible bajo nuestro estudio es ofrecerlo como extensión de navegador para mejorar el producto de lista de la compra de Google, en la actualidad, la herramienta únicamente ofrece una lista de la compra, pero nuestra extensión haría recomendaciones activas de productos en base a los criterios definidos por el usuario, funcionando como asistente, permitiendo al comprador realizar opciones más saludables y económicas.

5. Servicio de alerta de precios “in-aplicación”, por último, valoramos crear un stand alone aplicación con diferentes features como alertas en tiempo real de los precios de alimentos seleccionados por los usuarios que ofrecerían un valor añadido, ya que esta información actualmente se está recogiendo en la versión actual.



# Referencias

- Alvarez, J. (2020). *Macro y Micronutrientes*. Fundación Diabetes. Retrieved Septiembre 1, 2022, from <https://www.fundaciondiabetes.org/infantil/203/micronutrientes>
- Bar-Ilan, J. (2001). *Data collection methods on the web for infometric purposes – A review and analysis*.
- Breve Ramírez, M. A. (2021, 10 22). *Deep learning based Recommender System for an online retailer* [MASTER THESIS]. <https://upcommons.upc.edu/>. Retrieved 10 9, 2022, from <https://upcommons.upc.edu/bitstream/handle/2117/361554/161078.pdf?sequence=1>
- Burgos, N. (2007). Alimentación y nutrición en edad escolar. *Revista Digital Universitaria, Huelva, España*.
- Deossa Restrepo, G. C., Orozco Soto, D. M., & Urrego Borja, Y. (2020). Alimentación y nutrición durante la pandemia del COVID-19. *Kerwa*.
- González, A. (n.d.). *Sistemas de recomendación de contenido con Machine Learning – Cleverdata*. Cleverdata. Retrieved August 30, 2022, from <https://cleverdata.io/sistemas-recomendacion-machine-learning/>
- Instituto Nacional de Estadística. (n.d.). *Determinantes de salud*. sobrepeso, consumo de fruta y verdura, tipo de lactancia, actividad física). Retrieved Agosto 4, 2022, from [https://www.ine.es/ss/Satellite?c=INESeccion\\_C&cid=1259926457058&p=%5C&pagename=ProductosYServicios%2FPYSLayout&param1=PYSDetalle&param3=125992482288](https://www.ine.es/ss/Satellite?c=INESeccion_C&cid=1259926457058&p=%5C&pagename=ProductosYServicios%2FPYSLayout&param1=PYSDetalle&param3=125992482288)
- Martínez, C. V., Blanco, A. I. D.C., & Nomdedeu, C. L. (2005). *Alimentación y nutrición: manual teórico-práctico*. Ediciones Díaz de Santos.
- Mooney, S. J., Westreich, D. J., & El-Sayed, A. M. (2015). *Epidemiology in the era of big data*. (26(3 ed.). 10.1097/EDE.0000000000000274
- Open Food Facts. (2022). *Open Food Facts*. Open Food Facts - España. Retrieved August 18, 2022, from <https://es.openfoodfacts.org/>
- Sanz De La Torre, A., Martín Cerdeño, V. J., & Fernández Angulo, J. (2021). ALIMENTACIÓN EN ESPAÑA, PRODUCCIÓN, INDUSTRIA, DISTRIBUCIÓN Y CONSUMO. *Mercasa*, (24ª EDICIÓN.), 32-36.

Srinivas kulkarni. (2021, Mar 26). *Jaro winkler vs Levenshtein Distance*. <https://srinivas-kulkarni.medium.com/jaro-winkler-vs-levenshtein-distance-2eab21832fd>

Wehbe, E. N. (2021). *Sistema de información para la recopilación y centralización de formación sobre productos alimenticios*. Escuela Superior de Ingeniería y Tecnología Universidad de La Laguna.  
<https://riull.ull.es/xmlui/bitstream/handle/915/25438/Sistema%20de%20informacion%20para%20la%20recopilacion%20y%20centralizacion%20de%20informacion%20sobre%20productos%20alimenticios.pdf?sequence=1>

Zhao, B. (2017). *Web scraping. Encyclopedia of big data*.

## Anexos

### Anexo 1: Descarga&limpieza\_openfood.ipynb

```
import numpy as np
import pandas as pd
path = 'en.openfoodfacts.org.products.csv'
df = pd.read_table(path, sep='\t')
df.head(15)
df = df[[
    'product_name',
    'brands', 'categories_en', 'countries_en',
    'nutriscore_grade', 'food_groups_en', 'main_category_en',
    'energy-kcal_100g', 'fat_100g',
    'saturated-fat_100g', 'carbohydrates_100g',
    'sugars_100g', 'proteins_100g', 'salt_100g']]

df = df[df['nutriscore_grade'].notna()]
df = df[df['product_name'].notna()]
df = df[df['brands'].notna()]
df =
df[df.countries_en.str.contains('Spain|spain|en:es|España|spanien|Es
pagne', na=False)]
import unidecode

df['product_name'] = df['product_name'].str.lower() #pasando a
minusculas en los duplicados hay menos repetidos
df['brands'] = df['brands'].str.lower()

df = df.replace('ñ', '-&-', regex = True) #Para no sustituir las ñ
por n en el siguiente paso
cols = df.select_dtypes(include=[np.object]).columns
df[cols] = df[cols].aplicaciónonly(lambda x:
x.str.normalize('NFKD').str.encode('ascii',
errors='ignore').str.decode('utf-8'))
df = df.replace('-&-', 'ñ', regex = True )
```

```
df

df_nombre_marca =
df.drop_duplicates(df.columns[df.columns.isin(['product_name',
'brands'])], keep='first')

#df_nombre_marca_cate =
df.drop_duplicates(df.columns[df.columns.isin(['product_name',
'brands', 'categories_en'])], keep='first')
df_nombre_marca
df_nombre_marca.isnull().sum()
df_nombre_marca.to_csv('openfood_no_dup.csv')
```

## Anexo 2: downloader.py

```
import gdown
url='https://drive.google.com/uc?id=1Y1CAna8hFl6Yc_iR6ilJ_ysxsZgP_S5
O'
output = 'supermercados_raw.csv'
gdown.download (url, output, quiet=False)
reader.py
import pandas as pd

def read_file(filename) -> pd.DataFrame:
    """Read csv file and return formatted dataframe

    Args:
        filename (str): Name of file with path

    Return:
        DataFrame
    """
    food = pd.read_csv(filename, encoding="utf-8")
    return food

transform.py
import pandas as pd
```

```
from scipy import stats
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import operator
from datetime import datetime
from datetime import timedelta
import unicodecode

def transform(food) -> pd.DataFrame:
    """Gets pandas dataframe and returns it transformed

    Args:
        food (DataFrame)

    Return:
        DataFrame
    """
    blacklist =
['cocina','basura','detergente','electrodomestico','corporal','higie
nico','limpieza', 'jardineria',
    'casa', 'hogar', 'cuidado', 'cabello', 'bebe',
    'perfumeria', 'mascota', 'parafarmacia', 'farmacia',
'maquillaje','drogueria','soy_solidario','perfumeria_e_higiene_cuida
do_facial_hidratantes_y_nutritivas','perfumeria_e_higiene_cuidado_fa
cial_antiarrugas_y_antiedad',

'perfumeria_e_higiene_cuidado_facial_mascarillas','perfumeria_e_higi
ene_cuidado_corporal_anticeluliticos','perfumeria_e_higiene_cuidado_
corporal_cremas_cuerpo__body_milk',

'bebe_higiene_champu','bebe_higiene_colonia_infantil','bebe_higiene_
cremas_y_lociones',

'perfumeria_e_higiene_botiquin_gel_higienizante_y_mascarillas','perf
umeria_e_higiene_colonias_masculinas',
```

'perfumeria\_e\_higiene\_colonias\_femeninas','perfumeia\_e\_higiene\_colonias\_familiar','perfumeria\_e\_higiene\_desodorante\_roll\_on','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_champu',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_acondicionador\_y\_suavizante','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_mascarillas\_cabello',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_laca\_espuma\_y\_fijadores','perfumeria\_e\_higiene\_desodorante\_spray',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_fregasuelos','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_abrillantador\_suelos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_desatascador\_limpia\_tuberias','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_multiusos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_vitroceramica','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_antical',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_banos','drogueria\_y\_limpieza\_lavavajillas\_a\_mano',

'perfumeria\_e\_higiene\_higiene\_bucal\_dentifricos','perfumeria\_e\_higiene\_gel\_de\_bano\_gel\_de\_bano','perfumeria\_e\_higiene\_depilacion\_crema\_gel\_y\_spray',

'perfumeria\_e\_higiene\_cuidado\_manos\_jabon\_de\_manos\_liquido','perfumeria\_e\_higiene\_cuidado\_manos\_crema\_de\_manos',

'perfumeria\_e\_higiene\_limpieza\_facial\_exfoliante','perfumeria\_e\_higiene\_limpieza\_facial\_limpieza\_desmaquilladores','perfumeria\_e\_higiene\_limpieza\_facial\_leche\_y\_tonicos\_limpiadores',

'perfumeria\_e\_higiene\_cuidado\_corporal\_aceite','mascotas\_gatos\_alimento\_humedo','mascotas\_gatos\_alimento\_seco','mascotas\_perros\_alimento\_humedo',

'mascotas\_perros\_alimento\_seco','mascotas\_perros\_snacks','mascotas\_resto\_animales\_alimento\_pajaros',

'mascotas\_resto\_animales\_alimento\_peces\_tortugas','limpieza\_y\_hogar\_lejia\_y\_liquidos\_fuertes','limpieza\_y\_hogar\_limpiacristales',

'limpieza\_y\_hogar\_limpieza\_muebles\_y\_multiusos','limpieza\_y\_hogar\_limpieza\_vajilla','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_y\_calzado',

'cuidado\_facial\_y\_corporal\_afeitado\_y\_cuidado\_para\_hombre','cuidado\_facial\_y\_corporal\_cuidado\_corporal','cuidado\_facial\_y\_corporal\_cuidado\_e\_higiene\_facial',

'cuidado\_facial\_y\_corporal\_depilacion','cuidado\_facial\_y\_corporal\_de\_sodorante','cuidado\_facial\_y\_corporal\_higiene\_bucal',

'cuidado\_facial\_y\_corporal\_higiene\_intima','cuidado\_facial\_y\_corporal\_manicura\_y\_pedicura',

'cuidado\_facial\_y\_corporal\_perfume\_y\_colonia','limpieza\_y\_hogar\_insecticida\_y\_ambientador',

'limpieza\_y\_hogar\_limpieza\_bano\_y\_wc','maquillaje\_bases\_de\_maquillaje\_y\_corrector','maquillaje\_ojos','mascotas\_gato',

'mascotas\_perro','cuidado\_del\_cabello\_acondicionador\_y\_mascarilla','cuidado\_del\_cabello\_champu',

'cuidado\_del\_cabello\_fijacion\_cabello','drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_prendas\_delicadas',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_liquido','drogueria\_y\_limpieza\_cuidado\_ropa\_\_suavizante\_concentrado',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_a\_mano\_y\_jabon\_comun',

'limpieza\_y\_hogar\_detergente\_y\_suavizante\_ropa','fitoterapia\_y\_parafarmacia\_fitoterapia',

'cuidado\_facial\_y\_corporal\_gel\_y\_jabon\_de\_manos','maquillaje\_colorete\_y\_polvos','maquillaje\_labios',

'bebe\_higiene\_puericultura','bebe\_higiene\_toallitas','bebe\_panales\_pequenos\_hasta\_6\_kg','bebe\_panales\_medianos\_410\_kg',

'bebe\_panales\_grandes\_915\_kg','bebe\_panales\_de\_noche\_y\_aprendizaje','bebe\_panales\_banadores',

'perfumeria\_e\_higiene\_afeitado\_maquinillas\_y\_hojas\_de\_afeitar','perfumeria\_e\_higiene\_afeitado\_maquinillas\_desechables',

'perfumeria\_e\_higiene\_botiquin\_tiritas\_protectoras','perfumeria\_e\_higiene\_afeitado\_espuma\_gel\_y\_crema\_afeitar',

'perfumeria\_e\_higiene\_botiquin\_algodon\_bastoncillos','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_tintes\_y\_coloracion',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_accesorios\_cabello','perfumeria\_e\_higiene\_cuidado\_corporal\_protector\_solar',

'perfumeria\_e\_higiene\_cuidado\_pies\_crema','perfumeria\_e\_higiene\_depilacion\_bandas',

'perfumeria\_e\_higiene\_depilacion\_maquinillas\_y\_recambios','perfumeria\_e\_higiene\_desodorante\_crema\_y\_barra',



'perfumeria\_e\_higiene\_gel\_de\_bano\_sal\_y\_espuma\_de\_bano','perfumeria\_e\_higiene\_gel\_de\_bano\_esponja\_y\_accesorios',

'perfumeria\_e\_higiene\_higiene\_bucal\_cepillos\_de\_dientes','perfumeria\_e\_higiene\_higiene\_bucal\_seda\_dental',

'perfumeria\_e\_higiene\_higiene\_bucal\_productos\_protesicos','perfumeria\_e\_higiene\_higiene\_intima\_compresas',

'perfumeria\_e\_higiene\_higiene\_intima\_protege\_slips','perfumeria\_e\_higiene\_higiene\_intima\_aseo\_intimo',

'perfumeria\_e\_higiene\_higiene\_intima\_incontinencia','perfumeria\_e\_higiene\_higiene\_sexual\_lubricantes',

'perfumeria\_e\_higiene\_higiene\_sexual\_preservativos','drogueria\_y\_limpieza\_accesorios\_limpieza\_bayetas\_y\_gamuzas',

'drogueria\_y\_limpieza\_accesorios\_limpieza\_estropajos','drogueria\_y\_limpieza\_accesorios\_limpieza\_fregonas',

'drogueria\_y\_limpieza\_accesorios\_limpieza\_guantes','drogueria\_y\_limpieza\_ambientadores\_electricos\_y\_automaticos',

'drogueria\_y\_limpieza\_ambientadores\_antihumedad','drogueria\_y\_limpieza\_ambientadores\_para\_coche\_y\_espacios\_pequenos',

'drogueria\_y\_limpieza\_cerillas\_y\_mecheros\_cerillas\_y\_mecheros','drogueria\_y\_limpieza\_bolsas\_basura\_y\_reutilizable\_bolsas\_y\_sacos\_de\_basura',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_bolsas\_de\_congelar','drogueria\_y\_limpieza\_celulosa\_papel\_de\_cocina',

'drogueria\_y\_limpieza\_celulosa\_servilletas\_de\_papel','drogueria\_y\_limpieza\_cuidado\_ropa\_complementos\_aditivos\_para\_el\_lavado',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_inodoro\_wc',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_amoniacodesinfecantes\_agua\_destilada',  
    'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_lejia',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_cristales',  
    'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_muebles',  
  
'drogueria\_y\_limpieza\_pilas\_y\_bombillas\_pilas','mascotas\_accesorios\_accesorios',  
  
'soy\_solidario\_soy\_solidario\_soy\_solidario','perfumeria\_e\_higiene\_cabello\_cepillos\_peines\_y\_accesorios',  
  
'limpieza\_y\_hogar\_estropajo\_bayeta\_y\_guantes','limpieza\_y\_hogar\_limpieza\_cocina',  
  
'limpieza\_y\_hogar\_limpiahogar\_y\_friegasuelos','limpieza\_y\_hogar\_manejo\_y\_conservacion\_de\_alimentos',  
    'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_colonias',  
  
'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_esponjas\_manoplas\_y\_cepillos\_de\_bano',  
  
'perfumeria\_e\_higiene\_cabello\_cuidado\_y\_tratamientos\_del\_cabello','perfumeria\_e\_higiene\_cabello\_acondicionadores',  
  
'perfumeria\_e\_higiene\_cabello\_fijadores','perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_piel',  
  
'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_body\_milk\_hidratacion\_bajo\_la\_ducha',  
    'perfumeria\_e\_higiene\_botiquin\_optica',

'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_limpieza\_facial'  
,  
  
'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_productos\_para\_viaje',  
  
'perfumeria\_e\_higiene\_boca\_y\_sonrisa\_cepillos\_recambios\_y\_accesorios',  
,  
    'perfumeria\_e\_higiene\_higiene\_intima\_tampones',  
    'perfumeria\_e\_higiene\_higiene\_intima\_toallitas\_y\_geles\_intimos',  
    'perfumeria\_e\_higiene\_higiene\_intima\_protege\_slip',  
  
'perfumeria\_e\_higiene\_cuidado\_facial\_cremas\_especificas','drogueria\_y\_limpieza\_insecticidas\_hogar\_y\_plantas',  
  
'drogueria\_y\_limpieza\_insecticidas\_antipolillas\_y\_carcoma','drogueria\_y\_limpieza\_insecticidas\_caminantes',  
  
'drogueria\_y\_limpieza\_insecticidas\_voladores','drogueria\_y\_limpieza\_lavavajillas\_complementos\_lavavajillas',  
  
'drogueria\_y\_limpieza\_lavavajillas\_maquina\_liquido\_\_polvo','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_quitagrasas',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_suelo\_madera',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_especifico',  
'bebe\_panales\_y\_toallitas\_panales\_carrefour\_baby',  
  
'bebe\_panales\_y\_toallitas\_toallitas','bebe\_perfumeria\_e\_higiene\_jabon\_liquido','bebe\_puericultura\_accesorios',  
  
'limpieza\_y\_hogar\_menaje\_ollas\_cazos\_y\_accesorios','bebe\_perfumeria\_e\_higiene\_crema\_corporal\_talcos\_y\_antiirritacion',

'bebe\_perfumeria\_e\_higiene\_bastoncillos\_algodon\_y\_sueros','bebe\_perfumeria\_e\_higiene\_colonia',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_detergentes','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_tendido\_y\_planchado',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_agua\_de\_plancha\_y\_apresto',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_toallitas\_atrapacolors','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_tinte\_para\_la\_ropa',

'limpieza\_y\_hogar\_papel\_y\_celulosa\_servilletas','limpieza\_y\_hogar\_papel\_y\_celulosa\_panuelos',

'limpieza\_y\_hogar\_productos\_para\_cocina\_lavavajillas\_a\_mano',

'limpieza\_y\_hogar\_productos\_para\_cocina\_aditivos\_y\_limpiamaquinas','limpieza\_y\_hogar\_productos\_para\_bano\_wc',

'limpieza\_y\_hogar\_productos\_para\_bano\_desatascadores\_y\_limpiar\_tuberias',

'limpieza\_y\_hogar\_productos\_para\_bano\_limpiadores\_antical\_bano','limpieza\_y\_hogar\_productos\_para\_bano\_limpiar\_juntas',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_suelos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiar\_cristales\_y\_multiusos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_insecticidas',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_bayetas\_microfibra\_atrapapollvo','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_estropajos',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_plumeros\_rodillos\_y\_recambios',

'limpieza\_y\_hogar\_conservacion\_de\_alimentos\_papel\_de\_aluminio','limpieza\_y\_hogar\_conservacion\_de\_alimentos\_bolsas',

'limpieza\_y\_hogar\_ambientadores\_electricos',

'limpieza\_y\_hogar\_ambientadores\_automaticos','limpieza\_y\_hogar\_ambientadores\_coche',

'limpieza\_y\_hogar\_ambientadores\_antihumedad','limpieza\_y\_hogar\_ambientadores\_un\_toque',

'limpieza\_y\_hogar\_calzado\_desodorantes\_para\_calzado','limpieza\_y\_hogar\_calzado\_plantillas\_de\_calzado',

'limpieza\_y\_hogar\_calzado\_crema','limpieza\_y\_hogar\_menaje\_utensilios\_de\_cocina',

    'limpieza\_y\_hogar\_menaje\_jarras\_y\_filtros\_de\_agua',

'limpieza\_y\_hogar\_menaje\_sartenes\_paelleras\_y\_wok\_fondue\_parrillas\_grill\_accesorios',

'limpieza\_y\_hogar\_menaje\_vajillas\_y\_vasos','limpieza\_y\_hogar\_menaje\_cuberteria',

'limpieza\_y\_hogar\_papeleria\_cartuchos\_de\_tinta','limpieza\_y\_hogar\_bazar\_barbacoas\_y\_accesorios',

'mascotas\_perros\_pienso\_para\_perros','limpieza\_y\_hogar\_papeleria\_pequeno\_accesorio',

'limpieza\_y\_hogar\_papeleria\_cuadernos\_y\_carpetas','limpieza\_y\_hogar\_papeleria\_accesorios\_manualidades',

'limpieza\_y\_hogar\_papeleria\_archivadores','limpieza\_y\_hogar\_papeleria\_dibujo\_artistico',

'limpieza\_y\_hogar\_papeleria\_dibujo\_tecnico','limpieza\_y\_hogar\_bazar\_jardineria',

'limpieza\_y\_hogar\_bazar\_pequeno\_electrodomestico','limpieza\_y\_hogar\_bazar\_pegamentos\_y\_siliconas',

'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_jabon\_de\_manos','perfumeria\_e\_higiene\_boca\_y\_sonrisa\_dentifricos',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_afeitado','perfumeria\_e\_higiene\_cosmetica\_unas',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_after\_shave','perfumeria\_e\_higiene\_cosmetica\_ojos',

'perfumeria\_e\_higiene\_cosmetica\_accesorios\_de\_maquillaje\_y\_manicura\_y\_pedicura','perfumeria\_e\_higiene\_cosmetica\_labios',

'mascotas\_gatos\_arena','mascotas\_perros\_champus\_para\_perro','mascotas\_perros\_comederos','mascotas\_gatos\_pienso\_para\_gatos',

'mascotas\_conejos\_y\_roedores\_pienso\_para\_conejos\_y\_roedores','mascotas\_conejos\_y\_roedores\_accesorios\_e\_higiene',

'mascotas\_pajaros\_pienso\_para\_pajaros','parafarmacia\_higiene\_bucal\_colutorio',

'parafarmacia\_higiene\_bucal\_frescor\_y\_aliento','parafarmacia\_higiene\_bucal\_cuidado\_y\_fijacion\_protesis\_dentales',

'parafarmacia\_higiene\_bucal\_ortodoncia','parafarmacia\_higiene\_bucal\_cepillos\_y\_seda',

'parafarmacia\_botiquin\_mascarillas','parafarmacia\_botiquin\_higiene\_y\_tiras\_nasales',

'parafarmacia\_botiquin\_antisepticos\_y\_talcos','parafarmacia\_botiquin\_tos\_y\_garganta',

'parafarmacia\_botiquin\_alivio\_del\_dolor','parafarmacia\_botiquin\_oido\_y\_protectores',

'parafarmacia\_botiquin\_termometro\_y\_tensiometros','parafarmacia\_botiquin\_antimosquitos',

'parafarmacia\_cuidado\_corporal\_cremas\_y\_lociones','parafarmacia\_cuidado\_corporal\_cuidado\_intimo',

'parafarmacia\_cuidado\_e\_higiene\_facial\_cremas\_faciales','parafarmacia\_cuidado\_e\_higiene\_facial\_cuidado\_labial',

'parafarmacia\_cuidado\_e\_higiene\_facial\_exfoliantes\_y\_mascarillas','parafarmacia\_cuidado\_e\_higiene\_facial\_tonicos\_y\_lociones',

'parafarmacia\_cuidado\_e\_higiene\_facial\_maquillaje','parafarmacia\_cabello\_champus\_anticaida',

'parafarmacia\_cabello\_otros\_champus\_de\_tratamiento','parafarmacia\_cabello\_antiparasitarios',

'parafarmacia\_nutricion\_y\_dietetica\_tratamientos\_naturales','parafarmacia\_nutricion\_y\_dietetica\_control\_de\_peso',

'bebe\_higiene\_gel\_y\_jabon','perfumeria\_e\_higiene\_higiene\_bucal\_enjuagues\_y\_antisepticos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_alfombras\_tapicerias','cuidado\_facial\_y\_corporal\_protector\_solar\_y\_aftersun',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_papel\_aluminio','drogueria\_y\_limpieza\_conservacion\_alimentos\_film\_transparente',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_tabletas','drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_polvo',

'drogueria\_y\_limpieza\_pilas\_y\_bombillas\_bombillas','drogueria\_y\_limpieza\_otros\_articulos\_bazar\_filtros\_cafe',

'drogueria\_y\_limpieza\_otros\_articulos\_bazar\_cubiertos\_vasos\_y\_platos\_desechables','drogueria\_y\_limpieza\_otros\_articulos\_bazar\_otros\_articulos\_bazar',

'mascotas\_gatos\_snacks','mascotas\_gatos\_arena\_higiene','perfumeria\_e\_higiene\_cosmetica\_cosmetica',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_bolsas\_de\_conservacion',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_papel\_horno','drogueria\_y\_limpieza\_celulosa\_paneles\_y\_tissues',

'limpieza\_y\_hogar\_papel\_higienico\_y\_celulosa','limpieza\_y\_hogar\_pilas\_y\_bolsas\_de\_basura','perfumeria\_e\_higiene\_cabello\_champus','perfumeria\_e\_higiene\_cabello\_tinte','perfumeria\_e\_higiene\_botiquin\_alcohol\_agua\_oxigenada\_y\_otros',

'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_pies','perfumeria\_e\_higiene\_cuidado\_facial\_protector\_labial','perfumeria\_e\_higiene\_cuidado\_facial\_contorno\_de\_ojos',

'perfumeria\_e\_higiene\_limpieza\_facial\_limpieza\_especificos','drogueria\_y\_limpieza\_ambientadores\_aerosol\_spray','drogueria\_y\_limpieza\_ambientadores\_continuos\_y\_decorativos',

'drogueria\_y\_limpieza\_bolsas\_basura\_y\_reutilizable\_bolsas\_reutilizables','parafarmacia\_nutricion\_y\_dietetica\_complementos\_vitaminicos',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_fregonas','bebe\_panales\_y\_toallitas\_baberos\_protegecamas\_y\_bolsas\_para\_panales','bebe\_panales\_y\_toallitas\_panales\_huggies','bebe\_perfumeria\_e\_higiene\_champu',



'bebe\_puericultura\_chupetes\_biberones\_y\_tetinas','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_suavizantes','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_aditivos\_y\_quitamanchas',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_limpiadores\_y\_antical\_para\_lavadora','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_lejias\_lavadora','limpieza\_y\_hogar\_papel\_y\_celulosa\_papel\_cocina\_y\_multiusos',

'limpieza\_y\_hogar\_papel\_y\_celulosa\_toallitas\_gafas','limpieza\_y\_hogar\_productos\_para\_cocina\_lavavajillas\_a\_maquina','limpieza\_y\_hogar\_productos\_para\_cocina\_quitagrasas',

'limpieza\_y\_hogar\_productos\_para\_cocina\_vitroceramicas\_e\_induccion','limpieza\_y\_hogar\_productos\_para\_cocina\_limpiadores\_electrodomesticos\_cocina','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_lejias\_y\_amoniacos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpia\_muebles','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiador\_de\_alfombras\_y\_tapicerias','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiametales',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_bolsas\_de\_basura','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_escobas\_mopas\_y\_recogedores','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_guantes',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_cubos\_de\_basura','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_cubos\_de\_fregar\_y\_barrenos','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_otros\_utiles',

'limpieza\_y\_hogar\_conservacion\_de\_alimentos\_film\_transparente','limpieza\_y\_hogar\_conservacion\_de\_alimentos\_papel\_y\_moldes\_para\_horno','limpieza\_y\_hogar\_ambientadores\_decorativos',

'limpieza\_y\_hogar\_ambientadores\_aerosol\_o\_pistola','limpieza\_y\_hogar\_ambientadores\_absorbeolores','limpieza\_y\_hogar\_menaje\_menaje\_desechable','limpieza\_y\_hogar\_papeleria\_colorear',

'limpieza\_y\_hogar\_papeleria\_maquinaria\_de\_oficina','limpieza\_y\_hogar\_menaje\_ordenacion','limpieza\_y\_hogar\_menaje\_hermeticos','limpieza\_y\_hogar\_papeleria\_boligrafos\_y\_correctores',

'limpieza\_y\_hogar\_papeleria\_lapices\_y\_accesorios','limpieza\_y\_hogar\_papeleria\_marcadores','limpieza\_y\_hogar\_papeleria\_forralibros','limpieza\_y\_hogar\_bazar\_pilas',

'limpieza\_y\_hogar\_bazar\_bombillas\_y\_tubos','limpieza\_y\_hogar\_bazar\_automovil','perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_geles\_de\_bano','perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_desodorantes',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_maquinillas\_y\_recambios','perfumeria\_e\_higiene\_cosmetica\_rostro','perfumeria\_e\_higiene\_cosmetica\_estuches\_de\_bano\_y\_cosmetica',

'perfumeria\_e\_higiene\_bienestar\_sexual\_preservativos','mascotas\_perr os\_premios\_snacks\_y\_huesos','mascotas\_perros\_confort','mascotas\_perr os\_higiene',

'mascotas\_gatos\_accesorios\_e\_higiene','mascotas\_pajaros\_accesorios\_e\_higiene','mascotas\_peces\_y\_tortugas\_tortugas','mascotas\_peces\_y\_tortugas\_peces','mascotas\_peces\_y\_tortugas\_accesorios\_peces\_y\_tortugas',

'parafarmacia\_bebe\_anti\_irritacion','parafarmacia\_bebe\_hidratantes\_y\_aceites\_corporales','parafarmacia\_bebe\_toallitas\_bebe','parafarmacia\_higiene\_bucal\_pasta\_de\_dientes','parafarmacia\_botiquin\_geles\_hidroalcoholicos',

```
'parafarmacia_botiquin_apositos_y_gasas','parafarmacia_cuidado_corpo  
ral_jabones_y_geles','parafarmacia_cuidado_e_higiene_facial_desmaqui  
llantes','parafarmacia_cuidado_e_higiene_facial_cuidado_acne',
```

```
'parafarmacia_cuidado_de_manos_y_pies_crema_de_manos','parafarmacia_  
cuidado_de_manos_y_pies_desodorante_pies','parafarmacia_cuidado_de_m  
anos_y_pies_apositos_y_plantillas','charcuteria_y quesos_pates_foie_  
y_untables_foie','charcuteria_y quesos_pates_foie_y_untables_sobrasa  
da'] #categorias que no son alimentos y queremos eliminar de nuestro  
dataset
```

```
food = food[~food.category.str.contains('|'.join(blacklist))]  
food = food.drop(columns = ['product_id'])  
food = food.rename(columns = {'insert_date' : 'date'})  
food['date'] = food['date'].astype('datetime64[ns]')  
food.loc[food.description=="Granel", "price"]=  
food.reference_price  
#select todays date and substract one day from it  
food[food.date == (pd.to_datetime('today') -  
timedelta(days=1)).strftime('%Y-%m-%d')]
```

```
food=food.replace('ñ','-&-', regex=True)  
cols = food.select_dtypes(include=[np.object]).columns  
food[cols] = food[cols].aplicaciónonly(lambda x:  
x.str.normalize('NFKD').str.encode('ascii',  
errors='ignore').str.decode('utf-8'))  
food = food.replace('-&-','ñ', regex=True)  
  
return food
```

## Anexo 3: writer.py

```
import pandas as pd
from scipy import stats
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import operator
from datetime import datetime
from datetime import timedelta
import unidecode

def transform(food) -> pd.DataFrame:
    """Gets pandas dataframe and returns it transformed

    Args:
        food (DataFrame)

    Return:
        DataFrame
    """
    blacklist =
['cocina','basura','detergente','electrodomestico','corporal','higie
nico','limpieza', 'jardineria',
    'casa', 'hogar', 'cuidado', 'cabello', 'bebe',
    'perfumeria', 'mascota', 'parafarmacia', 'farmacia',
'maquillaje','drogueria','soy_solidario','perfumeria_e_higiene_cuida
do_facial_hidratantes_y_nutritivas','perfumeria_e_higiene_cuidado_fa
cial_antiarrugas_y_antiedad',

'perfumeria_e_higiene_cuidado_facial_mascarillas','perfumeria_e_higi
ene_cuidado_corporal_anticeluliticos','perfumeria_e_higiene_cuidado_
corporal_cremas_cuerpo__body_milk',

'bebe_higiene_champu','bebe_higiene_colonia_infantil','bebe_higiene_
cremas_y_lociones',
```

'perfumeria\_e\_higiene\_botiquin\_gel\_higienizante\_y\_mascarillas','perfumeria\_e\_higiene\_colonias\_masculinas',

'perfumeria\_e\_higiene\_colonias\_femeninas','perfumeria\_e\_higiene\_colonias\_familiar','perfumeria\_e\_higiene\_desodorante\_roll\_on','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_champu',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_acondicionador\_y\_suavizante','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_mascarillas\_cabello',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_laca\_espuma\_y\_fijadores','perfumeria\_e\_higiene\_desodorante\_spray',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_fregasuelos','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_abrillantador\_suelos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_desatascador\_limpia\_tuberias','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_multiusos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_vitroceramica','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_antical',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_banos','drogueria\_y\_limpieza\_lavavajillas\_a\_mano',

'perfumeria\_e\_higiene\_higiene\_bucal\_dentifricos','perfumeria\_e\_higiene\_gel\_de\_bano\_gel\_de\_bano','perfumeria\_e\_higiene\_depilacion\_crema\_gel\_y\_spray',

'perfumeria\_e\_higiene\_cuidado\_manos\_jabon\_de\_manos\_liquido','perfumeria\_e\_higiene\_cuidado\_manos\_crema\_de\_manos',

'perfumeria\_e\_higiene\_limpieza\_facial\_exfoliante','perfumeria\_e\_higiene\_limpieza\_facial\_limpieza\_desmaquilladores','perfumeria\_e\_higiene\_limpieza\_facial\_leche\_y\_tonicos\_limpiadores',

'perfumeria\_e\_higiene\_cuidado\_corporal\_aceite','mascotas\_gatos\_alimento\_humedo','mascotas\_gatos\_alimento\_seco','mascotas\_perros\_alimento\_humedo',

'mascotas\_perros\_alimento\_seco','mascotas\_perros\_snacks','mascotas\_resto\_animales\_alimento\_pajaros',

'mascotas\_resto\_animales\_alimento\_peces\_tortugas','limpieza\_y\_hogar\_lejia\_y\_liquidos\_fuertes','limpieza\_y\_hogar\_limpiacristales',

'limpieza\_y\_hogar\_limpieza\_muebles\_y\_multiusos','limpieza\_y\_hogar\_limpieza\_vajilla','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_y\_calzado',

'cuidado\_facial\_y\_corporal\_afeitado\_y\_cuidado\_para\_hombre','cuidado\_facial\_y\_corporal\_cuidado\_corporal','cuidado\_facial\_y\_corporal\_cuidado\_e\_higiene\_facial',

'cuidado\_facial\_y\_corporal\_depilacion','cuidado\_facial\_y\_corporal\_de\_sodorante','cuidado\_facial\_y\_corporal\_higiene\_bucal',

'cuidado\_facial\_y\_corporal\_higiene\_intima','cuidado\_facial\_y\_corporal\_manicura\_y\_pedicura',

'cuidado\_facial\_y\_corporal\_perfume\_y\_colonia','limpieza\_y\_hogar\_insecticida\_y\_ambientador',

'limpieza\_y\_hogar\_limpieza\_bano\_y\_wc','maquillaje\_bases\_de\_maquillaje\_y\_corrector','maquillaje\_ojos','mascotas\_gato',

'mascotas\_perro','cuidado\_del\_cabello\_acondicionador\_y\_mascarilla','cuidado\_del\_cabello\_champu',

'cuidado\_del\_cabello\_fijacion\_cabello','drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_prendas\_delicadas',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_liquido','drogueria\_y\_limpieza\_cuidado\_ropa\_\_suavizante\_concentrado',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_a\_mano\_y\_jabon\_comun',

'limpieza\_y\_hogar\_detergente\_y\_suavizante\_ropa','fitoterapia\_y\_parafarmacia\_fitoterapia',

'cuidado\_facial\_y\_corporal\_gel\_y\_jabon\_de\_manos','maquillaje\_colorete\_y\_polvos','maquillaje\_labios',

'bebe\_higiene\_puericultura','bebe\_higiene\_toallitas','bebe\_panales\_pequenos\_hasta\_6\_kg','bebe\_panales\_medianos\_410\_kg',

'bebe\_panales\_grandes\_915\_kg','bebe\_panales\_de\_noche\_y\_aprendizaje','bebe\_panales\_banadores',

'perfumeria\_e\_higiene\_afeitado\_maquinillas\_y\_hojas\_de\_afeitar','perfumeria\_e\_higiene\_afeitado\_maquinillas\_desechables',

'perfumeria\_e\_higiene\_botiquin\_tiritas\_protectoras','perfumeria\_e\_higiene\_afeitado\_espuma\_gel\_y\_crema\_afeitar',

'perfumeria\_e\_higiene\_botiquin\_algodon\_bastoncillos','perfumeria\_e\_higiene\_cuidado\_del\_cabello\_tintes\_y\_coloracion',

'perfumeria\_e\_higiene\_cuidado\_del\_cabello\_accesorios\_cabello','perfumeria\_e\_higiene\_cuidado\_corporal\_protector\_solar',

'perfumeria\_e\_higiene\_cuidado\_pies\_crema','perfumeria\_e\_higiene\_depilacion\_bandas',

'perfumeria\_e\_higiene\_depilacion\_maquinillas\_y\_recambios','perfumeria\_e\_higiene\_desodorante\_crema\_y\_barra',

'perfumeria\_e\_higiene\_gel\_de\_bano\_sal\_y\_espuma\_de\_bano','perfumeria\_e\_higiene\_gel\_de\_bano\_esponja\_y\_accesorios',

'perfumeria\_e\_higiene\_higiene\_bucal\_cepillos\_de\_dientes','perfumeria\_e\_higiene\_higiene\_bucal\_seda\_dental',

'perfumeria\_e\_higiene\_higiene\_bucal\_productos\_protesicos','perfumeria\_e\_higiene\_higiene\_intima\_compresas',

'perfumeria\_e\_higiene\_higiene\_intima\_protege\_slips','perfumeria\_e\_higiene\_higiene\_intima\_aseo\_intimo',

'perfumeria\_e\_higiene\_higiene\_intima\_incontinencia','perfumeria\_e\_higiene\_higiene\_sexual\_lubricantes',

'perfumeria\_e\_higiene\_higiene\_sexual\_preservativos','drogueria\_y\_limpieza\_accesorios\_limpieza\_bayetas\_y\_gamuzas',

'drogueria\_y\_limpieza\_accesorios\_limpieza\_estropajos','drogueria\_y\_limpieza\_accesorios\_limpieza\_fregonas',

'drogueria\_y\_limpieza\_accesorios\_limpieza\_guantes','drogueria\_y\_limpieza\_ambientadores\_electricos\_y\_automaticos',

'drogueria\_y\_limpieza\_ambientadores\_antihumedad','drogueria\_y\_limpieza\_ambientadores\_para\_coche\_y\_espacios\_pequenos',

'drogueria\_y\_limpieza\_cerillas\_y\_mecheros\_cerillas\_y\_mecheros','drogueria\_y\_limpieza\_bolsas\_basura\_y\_reutilizable\_bolsas\_y\_sacos\_de\_basura',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_bolsas\_de\_congelar','drogueria\_y\_limpieza\_celulosa\_papel\_de\_cocina',

'drogueria\_y\_limpieza\_celulosa\_servilletas\_de\_papel','drogueria\_y\_limpieza\_cuidado\_ropa\_complementos\_aditivos\_para\_el\_lavado',



'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_inodoro\_wc',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_amoniacodesinfectantes\_agua\_destilada',  
    'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_lejia',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_cristales',  
    'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_muebles',  
  
'drogueria\_y\_limpieza\_pilas\_y\_bombillas\_pilas','mascotas\_accesorios\_accesorios',  
  
'soy\_solidario\_soy\_solidario\_soy\_solidario','perfumeria\_e\_higiene\_cabello\_cepillos\_peines\_y\_accesorios',  
  
'limpieza\_y\_hogar\_estropajo\_bayeta\_y\_guantes','limpieza\_y\_hogar\_limpieza\_cocina',  
  
'limpieza\_y\_hogar\_limpiahogar\_y\_friegasuelos','limpieza\_y\_hogar\_manejo\_y\_conservacion\_de\_alimentos',  
    'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_colonias',  
  
'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_esponjas\_manoplas\_y\_cepillos\_de\_bano',  
  
'perfumeria\_e\_higiene\_cabello\_cuidado\_y\_tratamientos\_del\_cabello','perfumeria\_e\_higiene\_cabello\_acondicionadores',  
  
'perfumeria\_e\_higiene\_cabello\_fijadores','perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_piel',  
  
'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_body\_milk\_hidratacion\_bajo\_la\_ducha',  
    'perfumeria\_e\_higiene\_botiquin\_optica',

'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_limpieza\_facial'  
,  
  
'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_productos\_para\_viaje',  
  
'perfumeria\_e\_higiene\_boca\_y\_sonrisa\_cepillos\_recambios\_y\_accesorios',  
,  
    'perfumeria\_e\_higiene\_higiene\_intima\_tampones',  
    'perfumeria\_e\_higiene\_higiene\_intima\_toallitas\_y\_geles\_intimos',  
    'perfumeria\_e\_higiene\_higiene\_intima\_protege\_slip',  
  
'perfumeria\_e\_higiene\_cuidado\_facial\_cremas\_especificas','drogueria\_y\_limpieza\_insecticidas\_hogar\_y\_plantas',  
  
'drogueria\_y\_limpieza\_insecticidas\_antipolillas\_y\_carcoma','drogueria\_y\_limpieza\_insecticidas\_caminantes',  
  
'drogueria\_y\_limpieza\_insecticidas\_voladores','drogueria\_y\_limpieza\_lavavajillas\_complementos\_lavavajillas',  
  
'drogueria\_y\_limpieza\_lavavajillas\_maquina\_liquido\_\_polvo','drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_quitagrasas',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_suelo\_madera',  
  
'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpiador\_especifico',  
'bebe\_panales\_y\_toallitas\_panales\_carrefour\_baby',  
  
'bebe\_panales\_y\_toallitas\_toallitas','bebe\_perfumeria\_e\_higiene\_jabon\_liquido','bebe\_puericultura\_accesorios',  
  
'limpieza\_y\_hogar\_menaje\_ollas\_cazos\_y\_accesorios','bebe\_perfumeria\_e\_higiene\_crema\_corporal\_talcos\_y\_antiirritacion',

'bebe\_perfumeria\_e\_higiene\_bastoncillos\_algodon\_y\_sueros','bebe\_perfumeria\_e\_higiene\_colonia',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_detergentes','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_tendido\_y\_planchado',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_agua\_de\_plancha\_y\_apresto',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_toallitas\_atrapacolors','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_tinte\_para\_la\_ropa',

'limpieza\_y\_hogar\_papel\_y\_celulosa\_servilletas','limpieza\_y\_hogar\_papel\_y\_celulosa\_panuelos',

'limpieza\_y\_hogar\_productos\_para\_cocina\_lavavajillas\_a\_mano',

'limpieza\_y\_hogar\_productos\_para\_cocina\_aditivos\_y\_limpiamaquinas','limpieza\_y\_hogar\_productos\_para\_bano\_wc',

'limpieza\_y\_hogar\_productos\_para\_bano\_desatascadores\_y\_limpiar\_tuberias',

'limpieza\_y\_hogar\_productos\_para\_bano\_limpiadores\_antical\_bano','limpieza\_y\_hogar\_productos\_para\_bano\_limpiar\_juntas',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_suelos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiar\_cristales\_y\_multiusos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_insecticidas',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_bayetas\_microfibra\_atrapapollvo','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_estropajos',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_plumeros\_rodillos\_y\_recambios',

'limpieza\_y\_hogar\_conservacion\_de\_alimentos\_papel\_de\_aluminio','limpieza\_y\_hogar\_conservacion\_de\_alimentos\_bolsas',

'limpieza\_y\_hogar\_ambientadores\_electricos',

'limpieza\_y\_hogar\_ambientadores\_automaticos','limpieza\_y\_hogar\_ambientadores\_coche',

'limpieza\_y\_hogar\_ambientadores\_antihumedad','limpieza\_y\_hogar\_ambientadores\_un\_toque',

'limpieza\_y\_hogar\_calzado\_desodorantes\_para\_calzado','limpieza\_y\_hogar\_calzado\_plantillas\_de\_calzado',

'limpieza\_y\_hogar\_calzado\_crema','limpieza\_y\_hogar\_menaje\_utensilios\_de\_cocina',

    'limpieza\_y\_hogar\_menaje\_jarras\_y\_filtros\_de\_agua',

'limpieza\_y\_hogar\_menaje\_sartenes\_paelleras\_y\_wok\_fondue\_parrillas\_grill\_accesorios',

'limpieza\_y\_hogar\_menaje\_vajillas\_y\_vasos','limpieza\_y\_hogar\_menaje\_cuberteria',

'limpieza\_y\_hogar\_papeleria\_cartuchos\_de\_tinta','limpieza\_y\_hogar\_bazar\_barbacoas\_y\_accesorios',

'mascotas\_perros\_pienso\_para\_perros','limpieza\_y\_hogar\_papeleria\_pequeno\_accesorio',

'limpieza\_y\_hogar\_papeleria\_cuadernos\_y\_carpetas','limpieza\_y\_hogar\_papeleria\_accesorios\_manualidades',

'limpieza\_y\_hogar\_papeleria\_archivadores','limpieza\_y\_hogar\_papeleria\_dibujo\_artistico',

'limpieza\_y\_hogar\_papeleria\_dibujo\_tecnico','limpieza\_y\_hogar\_bazar\_jardineria',

'limpieza\_y\_hogar\_bazar\_pequeno\_electrodomestico','limpieza\_y\_hogar\_bazar\_pegamentos\_y\_siliconas',

'perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_jabon\_de\_manos','perfumeria\_e\_higiene\_boca\_y\_sonrisa\_dentifricos',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_afeitado','perfumeria\_e\_higiene\_cosmetica\_unas',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_after\_shave','perfumeria\_e\_higiene\_cosmetica\_ojos',

'perfumeria\_e\_higiene\_cosmetica\_accesorios\_de\_maquillaje\_y\_manicura\_y\_pedicura','perfumeria\_e\_higiene\_cosmetica\_labios',

'mascotas\_gatos\_arena','mascotas\_perros\_champus\_para\_perro','mascotas\_perros\_comederos','mascotas\_gatos\_pienso\_para\_gatos',

'mascotas\_conejos\_y\_roedores\_pienso\_para\_conejos\_y\_roedores','mascotas\_conejos\_y\_roedores\_accesorios\_e\_higiene',

'mascotas\_pajaros\_pienso\_para\_pajaros','parafarmacia\_higiene\_bucal\_colutorio',

'parafarmacia\_higiene\_bucal\_frescor\_y\_aliento','parafarmacia\_higiene\_bucal\_cuidado\_y\_fijacion\_protesis\_dentales',

'parafarmacia\_higiene\_bucal\_ortodoncia','parafarmacia\_higiene\_bucal\_cepillos\_y\_seda',

'parafarmacia\_botiquin\_mascarillas','parafarmacia\_botiquin\_higiene\_y\_tiras\_nasales',

'parafarmacia\_botiquin\_antisepticos\_y\_talcos','parafarmacia\_botiquin\_tos\_y\_garganta',

'parafarmacia\_botiquin\_alivio\_del\_dolor','parafarmacia\_botiquin\_oido\_y\_protectores',

'parafarmacia\_botiquin\_termometro\_y\_tensiometros','parafarmacia\_botiquin\_antimosquitos',

'parafarmacia\_cuidado\_corporal\_cremas\_y\_lociones','parafarmacia\_cuidado\_corporal\_cuidado\_intimo',

'parafarmacia\_cuidado\_e\_higiene\_facial\_cremas\_faciales','parafarmacia\_cuidado\_e\_higiene\_facial\_cuidado\_labial',

'parafarmacia\_cuidado\_e\_higiene\_facial\_exfoliantes\_y\_mascarillas','parafarmacia\_cuidado\_e\_higiene\_facial\_tonicos\_y\_lociones',

'parafarmacia\_cuidado\_e\_higiene\_facial\_maquillaje','parafarmacia\_cabello\_champus\_anticaida',

'parafarmacia\_cabello\_otros\_champus\_de\_tratamiento','parafarmacia\_cabello\_antiparasitarios',

'parafarmacia\_nutricion\_y\_dietetica\_tratamientos\_naturales','parafarmacia\_nutricion\_y\_dietetica\_control\_de\_peso',

'bebe\_higiene\_gel\_y\_jabon','perfumeria\_e\_higiene\_higiene\_bucal\_enjuagues\_y\_antisepticos',

'drogueria\_y\_limpieza\_limpiadores\_para\_el\_hogar\_limpia\_alfombras\_tapicerias','cuidado\_facial\_y\_corporal\_protector\_solar\_y\_aftersun',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_papel\_aluminio','drogueria\_y\_limpieza\_conservacion\_alimentos\_film\_transparente',

'drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_tabletas','drogueria\_y\_limpieza\_cuidado\_ropa\_\_detergente\_maquina\_polvo',

'drogueria\_y\_limpieza\_pilas\_y\_bombillas\_bombillas','drogueria\_y\_limpieza\_otros\_articulos\_bazar\_filtros\_cafe',

'drogueria\_y\_limpieza\_otros\_articulos\_bazar\_cubiertos\_vasos\_y\_platos\_desechables','drogueria\_y\_limpieza\_otros\_articulos\_bazar\_otros\_articulos\_bazar',

'mascotas\_gatos\_snacks','mascotas\_gatos\_arena\_higiene','perfumeria\_e\_higiene\_cosmetica\_cosmetica',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_bolsas\_de\_conservacion',

'drogueria\_y\_limpieza\_conservacion\_alimentos\_papel\_horno','drogueria\_y\_limpieza\_celulosa\_paneles\_y\_tissues',

'limpieza\_y\_hogar\_papel\_higienico\_y\_celulosa','limpieza\_y\_hogar\_pilas\_y\_bolsas\_de\_basura','perfumeria\_e\_higiene\_cabello\_champus','perfumeria\_e\_higiene\_cabello\_tinte','perfumeria\_e\_higiene\_botiquin\_alcohol\_agua\_oxigenada\_y\_otros',

'perfumeria\_e\_higiene\_cuidado\_y\_proteccion\_corporal\_pies','perfumeria\_e\_higiene\_cuidado\_facial\_protector\_labial','perfumeria\_e\_higiene\_cuidado\_facial\_contorno\_de\_ojos',

'perfumeria\_e\_higiene\_limpieza\_facial\_limpieza\_especificos','drogueria\_y\_limpieza\_ambientadores\_aerosol\_spray','drogueria\_y\_limpieza\_ambientadores\_continuos\_y\_decorativos',

'drogueria\_y\_limpieza\_bolsas\_basura\_y\_reutilizable\_bolsas\_reutilizables','parafarmacia\_nutricion\_y\_dietetica\_complementos\_vitaminicos',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_fregonas','bebe\_panales\_y\_toallitas\_baberos\_protegecamas\_y\_bolsas\_para\_panales','bebe\_panales\_y\_toallitas\_panales\_huggies','bebe\_perfumeria\_e\_higiene\_champu',

'bebe\_puericultura\_chupetes\_biberones\_y\_tetinas','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_suavizantes','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_aditivos\_y\_quitamanchas',

'limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_limpiadores\_y\_antical\_para\_lavadora','limpieza\_y\_hogar\_cuidado\_de\_la\_ropa\_lejias\_lavadora','limpieza\_y\_hogar\_papel\_y\_celulosa\_papel\_cocina\_y\_multiusos',

'limpieza\_y\_hogar\_papel\_y\_celulosa\_toallitas\_gafas','limpieza\_y\_hogar\_productos\_para\_cocina\_lavavajillas\_a\_maquina','limpieza\_y\_hogar\_productos\_para\_cocina\_quitagrasas',

'limpieza\_y\_hogar\_productos\_para\_cocina\_vitroceramicas\_e\_induccion','limpieza\_y\_hogar\_productos\_para\_cocina\_limpiadores\_electrodomesticos\_cocina','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_lejias\_y\_amoniacos',

'limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiar\_muebles','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiador\_de\_alfombras\_y\_tapicerias','limpieza\_y\_hogar\_productos\_para\_toda\_la\_casa\_limpiar\_metales',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_bolsas\_de\_basura','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_escobas\_mopas\_y\_recogedores','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_guantes',

'limpieza\_y\_hogar\_utensilios\_de\_limpieza\_cubos\_de\_basura','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_cubos\_de\_fregar\_y\_barrenos','limpieza\_y\_hogar\_utensilios\_de\_limpieza\_otros\_utiles',

'limpieza\_y\_hogar\_conservacion\_de\_alimentos\_film\_transparente','limpieza\_y\_hogar\_conservacion\_de\_alimentos\_papel\_y\_moldes\_para\_horno','limpieza\_y\_hogar\_ambientadores\_decorativos',



'limpieza\_y\_hogar\_ambientadores\_aerosol\_o\_pistola','limpieza\_y\_hogar\_ambientadores\_absorbeolores','limpieza\_y\_hogar\_menaje\_menaje\_desechable','limpieza\_y\_hogar\_papeleria\_colorear',

'limpieza\_y\_hogar\_papeleria\_maquinaria\_de\_oficina','limpieza\_y\_hogar\_menaje\_ordenacion','limpieza\_y\_hogar\_menaje\_hermeticos','limpieza\_y\_hogar\_papeleria\_boligrafos\_y\_correctores',

'limpieza\_y\_hogar\_papeleria\_lapices\_y\_accesorios','limpieza\_y\_hogar\_papeleria\_marcadores','limpieza\_y\_hogar\_papeleria\_forralibros','limpieza\_y\_hogar\_bazar\_pilas',

'limpieza\_y\_hogar\_bazar\_bombillas\_y\_tubos','limpieza\_y\_hogar\_bazar\_automovil','perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_geles\_de\_bano','perfumeria\_e\_higiene\_bano\_e\_higiene\_corporal\_desodorantes',

'perfumeria\_e\_higiene\_depilacion\_y\_afeitado\_maquinillas\_y\_recambios','perfumeria\_e\_higiene\_cosmetica\_rostro','perfumeria\_e\_higiene\_cosmetica\_estuches\_de\_bano\_y\_cosmetica',

'perfumeria\_e\_higiene\_bienestar\_sexual\_preservativos','mascotas\_perr os\_premios\_snacks\_y\_huesos','mascotas\_perros\_confort','mascotas\_perr os\_higiene',

'mascotas\_gatos\_accesorios\_e\_higiene','mascotas\_pajaros\_accesorios\_e\_higiene','mascotas\_peces\_y\_tortugas\_tortugas','mascotas\_peces\_y\_tortugas\_peces','mascotas\_peces\_y\_tortugas\_accesorios\_peces\_y\_tortugas',

'parafarmacia\_bebe\_anti\_irritacion','parafarmacia\_bebe\_hidratantes\_y\_aceites\_corporales','parafarmacia\_bebe\_toallitas\_bebe','parafarmacia\_higiene\_bucal\_pasta\_de\_dientes','parafarmacia\_botiquin\_geles\_hidroalcoholicos',

```
'parafarmacia_botiquin_apositos_y_gasas','parafarmacia_cuidado_corpo  
ral_jabones_y_geles','parafarmacia_cuidado_e_higiene_facial_desmaqui  
llantes','parafarmacia_cuidado_e_higiene_facial_cuidado_acne',
```

```
'parafarmacia_cuidado_de_manos_y_pies_crema_de_manos','parafarmacia_  
cuidado_de_manos_y_pies_desodorante_pies','parafarmacia_cuidado_de_m  
anos_y_pies_apositos_y_plantillas','charcuteria_y quesos_pates_foie_  
y_untables_foie','charcuteria_y quesos_pates_foie_y_untables_sobrasa  
da'] #categorias que no son alimentos y queremos eliminar de nuestro  
dataset
```

```
food = food[~food.category.str.contains('|'.join(blacklist))]  
food = food.drop(columns = ['product_id'])  
food = food.rename(columns = {'insert_date' : 'date'})  
food['date'] = food['date'].astype('datetime64[ns]')  
food.loc[food.description=="Granel", "price"]=  
food.reference_price  
#select todays date and substract one day from it  
food[food.date == (pd.to_datetime('today') -  
timedelta(days=1)).strftime('%Y-%m-%d')]
```

```
food=food.replace('ñ','-&-', regex=True)  
cols = food.select_dtypes(include=[np.object]).columns  
food[cols] = food[cols].aplicaciónonly(lambda x:  
x.str.normalize('NFKD').str.encode('ascii',  
errors='ignore').str.decode('utf-8'))  
food = food.replace('-&-','ñ', regex=True)  
  
return food
```

## Anexo 4: process.py

```
import logging
import argparse
from reader import read_file
from transform import transform
from writer import output_data
import datetime

def main(filename):

    logging.info(f"Reading filename: {filename}")
    food = read_file(filename)

    logging.info("Processing data...")
    transform_data = transform(food)

    output =
'output/data{}.csv'.format(datetime.datetime.now().strftime("%Y%m%d%H%M%S"))
    logging.info(f"Writing data in {output}")
    output_data(transform_data, output)

if __name__ == '__main__':

    logging.basicConfig(
        format='[%(name)s] [%(levelname)s] %(message)s',
        level=logging.DEBUG)

    parser = argparse.ArgumentParser()
    parser.add_argument("-f", "--filename", help="File to process")
    args = parser.parse_args()
    main(args.filename)
```

## Anexo 5: Script creación base de datos SQL Text File

```
-- MySQL Workbench Forward Engineering

SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_
ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';

-- -----
-- Schema mydb
-- -----
-- -----
-- Schema alimentos
-- -----

-- -----
-- Schema alimentos
-- -----

CREATE SCHEMA IF NOT EXISTS `alimentos` DEFAULT CHARACTER SET utf8 ;
USE `alimentos` ;

-- -----
-- Table `alimentos`.`fuzzy_match`
-- -----

CREATE TABLE IF NOT EXISTS `alimentos`.`fuzzy_match` (
  `url` VARCHAR(300) NULL DEFAULT NULL,
  `supermarket` VARCHAR(100) NULL DEFAULT NULL,
  `category` VARCHAR(100) NULL DEFAULT NULL,
  `name` LONGTEXT NULL DEFAULT NULL,
  `description` LONGTEXT NULL DEFAULT NULL,
  `price` FLOAT NULL DEFAULT NULL,
  `reference_price` FLOAT NULL DEFAULT NULL,
  `reference_unit` VARCHAR(10) NULL DEFAULT NULL,
  `product_name` TINYTEXT NULL DEFAULT NULL,
```

```
`medida_similitud` DOUBLE NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb3;

-- -----
-- Table `alimentos`.`merge`
-- -----
CREATE TABLE IF NOT EXISTS `alimentos`.`merge` (
  `id_food` INT NULL DEFAULT NULL,
  `product_name` VARCHAR(200) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `brands` VARCHAR(250) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `categories_en` VARCHAR(1000) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `countries_en` LONGTEXT CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `nutriscore_grade` VARCHAR(1) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `food_groups_en` VARCHAR(70) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `main_category_en` LONGTEXT CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `energy_kcal_100g` INT NULL DEFAULT NULL,
  `fat_100g` INT NULL DEFAULT NULL,
  `saturated_fat_100g` INT NULL DEFAULT NULL,
  `carbohydrates_100g` INT NULL DEFAULT NULL,
  `sugars_100g` INT NULL DEFAULT NULL,
  `proteins_100g` INT NULL DEFAULT NULL,
  `salt_100g` INT NULL DEFAULT NULL,
  `url` VARCHAR(300) NULL DEFAULT NULL,
  `supermarket` VARCHAR(100) NULL DEFAULT NULL,
  `category` VARCHAR(100) NULL DEFAULT NULL,
  `name` LONGTEXT NULL DEFAULT NULL,
  `description` LONGTEXT NULL DEFAULT NULL,
  `price` FLOAT NULL DEFAULT NULL,
  `reference_price` FLOAT NULL DEFAULT NULL,
  `reference_unit` VARCHAR(10) NULL DEFAULT NULL,
```

```
`product_name_copy1` VARCHAR(300) NULL DEFAULT NULL,
`medida_similitud` FLOAT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb3;

-- -----
-- Table `alimentos`.`open_food`
-- -----
CREATE TABLE IF NOT EXISTS `alimentos`.`open_food` (
  `id_food` INT NOT NULL,
  `product_name` VARCHAR(200) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `brands` VARCHAR(250) CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `categories_en` VARCHAR(1000) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `countries_en` LONGTEXT CHARACTER SET 'utf8' NULL DEFAULT NULL,
  `nutriscore_grade` VARCHAR(1) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `food_groups_en` VARCHAR(70) CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `main_category_en` LONGTEXT CHARACTER SET 'utf8' NULL DEFAULT
NULL,
  `energy-kcal_100g` INT NULL DEFAULT NULL,
  `fat_100g` INT NULL DEFAULT NULL,
  `saturated-fat_100g` INT NULL DEFAULT NULL,
  `carbohydrates_100g` INT NULL DEFAULT NULL,
  `sugars_100g` INT NULL DEFAULT NULL,
  `proteins_100g` INT NULL DEFAULT NULL,
  `salt_100g` INT NULL DEFAULT NULL)
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8mb4
COLLATE = utf8mb4_0900_ai_ci;

-- -----
-- Table `alimentos`.`supermercado`
```

```
-- -----  
CREATE TABLE IF NOT EXISTS `alimentos`.`supermercado` (  
  `url` VARCHAR(300) NULL DEFAULT NULL,  
  `supermarket` VARCHAR(100) NULL DEFAULT NULL,  
  `category` VARCHAR(100) NULL DEFAULT NULL,  
  `name` LONGTEXT NULL DEFAULT NULL,  
  `description` LONGTEXT NULL DEFAULT NULL,  
  `price` FLOAT NULL DEFAULT NULL,  
  `reference_price` FLOAT NULL DEFAULT NULL,  
  `reference_unit` VARCHAR(10) NULL DEFAULT NULL,  
  `date` DATE NULL DEFAULT curdate())  
ENGINE = InnoDB  
DEFAULT CHARACTER SET = utf8mb4  
COLLATE = utf8mb4_0900_ai_ci;
```

```
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## Anexo 6: Perfiles alimentación.ipynb

```
import pandas as pd  
import numpy as np  
  
merge = pd.read_csv("/content/merge.csv", index_col=False)  
merge = merge.drop(merge.columns[0], axis=1)  
merge  
perfil = str(input("Ingrese el perfil deseado entre: A) baja en  
calorias, B) Volumen, C) Alto rendimiento, D) Balanceado, E) Ahorro  
"))  
  
Pérdida de peso: Baja en calorias  
Nutriscore: A, B  
Kcal: < 130  
Fat: < 1  
saturated fat: 0
```

carbohydrates: < 6  
sugar: < 5  
proteins: no aplica  
salt: < 1  
price: a gusto del usuario

Volumen: alto en proteina y hidratos  
Nutriscore: a, b, c, d  
Kcal: no aplica  
Fat: < 3  
saturated fat: <1  
carbohydrates: no aplica  
sugar: < 10  
proteins: max  
salt: < 5  
price: no determinante

Alto rendimiento: hidratos y azucar  
Nutriscore: a, b, c  
Kcal: max  
Fat: < 3  
saturated fat: < 1  
carbohydrates: max  
sugar: max  
proteins: no aplica  
salt: no aplica  
price: no determinante

Balanceado: Economica, nutriscore ABC  
Nutriscore: a, b, c, d  
Kcal: < 86  
Fat: < 1  
saturated fat: 0  
carbohydrates: < 6



sugar: <1  
proteins: <4  
salt: < 0.5  
price: ~€2

```
Ahorro:
Nutriscore: no aplica
Kcal: < 245
Fat: < 3
saturated fat: < 1
carbohydrates: < 28
sugar: < 4
proteins: < 12
salt: < 1
price: min (asc)
if perfil == "a":
    #Perdida de peso
    productos = merge.loc[(merge.nutriscore_grade.isin(["a","b"])) &
    (merge.energy_kcal_100g <130) & (merge.fat_100g<1) &
    (merge.saturated_fat_100g==0)
    & (merge.carbohydrates_100g<6) & (merge.sugars_100g <5) &
    (merge.salt_100g<=1)]
elif perfil == "b":
    #Perdida de peso
    productos =
merge.loc[(merge.nutriscore_grade.isin(["a","b","c","d"])) &
    (merge.fat_100g<3) & (merge.saturated_fat_100g<=1)
    & (merge.sugars_100g <10) &
    (merge.salt_100g<=1)].sort_values(by='proteins_100g',
ascending=False)

elif perfil == "c":
    #Alto Rendimiento
    productos = merge.loc[(merge.nutriscore_grade.isin(["a","b","c"]))
    & (merge.fat_100g<3) & (merge.saturated_fat_100g<=1)
```

```
].sort_values(by=
['energy_kcal_100g','sugars_100g','carbohydrates_100g'],ascending=False)

elif perfil == "d":
    #Balanceado
    productos =
merge.loc[(merge.nutriscore_grade.isin(["a","b","c","d"])) &
(merge.energy_kcal_100g <86) & (merge.fat_100g<1) &
(merge.saturated_fat_100g==0)
          & (merge.carbohydrates_100g<6) & (merge.proteins_100g <4)
& (merge.salt_100g<=1) & (merge.price
<=2)].sort_values(by=['price'])

else:
    #Ahorro
    productos = merge.loc[(merge.energy_kcal_100g <245) &
(merge.fat_100g<3) & (merge.saturated_fat_100g<=1)
          & (merge.carbohydrates_100g<28) & (merge.proteins_100g
<12) & (merge.salt_100g<=1)].sort_values(by=['price'])

productos.head(10)
#Perdida de peso
merge.loc[(merge.nutriscore_grade.isin(["a","b"])) &
(merge.energy_kcal_100g <130) & (merge.fat_100g<1) &
(merge.saturated_fat_100g==0)
          & (merge.carbohydrates_100g<6) & (merge.sugars_100g <5) &
(merge.salt_100g<=1)].sort_values(by=['price'])

#Volumen
merge.loc[(merge.nutriscore_grade.isin(["a","b","c","d"])) &
(merge.fat_100g<3) & (merge.saturated_fat_100g<=1)
          & (merge.sugars_100g <10) &
(merge.salt_100g<=1)].sort_values(by='proteins_100g',
ascending=False)
```

```
#Alto Rendimiento
merge.loc[(merge.nutriscore_grade.isin(["a","b","c"])) &
(merge.fat_100g<3) & (merge.saturated_fat_100g<=1)
].sort_values(by=
['energy_kcal_100g','sugars_100g','carbohydrates_100g'],ascending=False)

#Balanceado
merge.loc[(merge.nutriscore_grade.isin(["a","b","c","d"])) &
(merge.energy_kcal_100g <86) & (merge.fat_100g<1) &
(merge.saturated_fat_100g==0)
& (merge.carbohydrates_100g<6) & (merge.proteins_100g <4)
& (merge.salt_100g<=1) & (merge.price
<=2)].sort_values(by=['price'])

#Ahorro
merge.loc[(merge.energy_kcal_100g <245) & (merge.fat_100g<3) &
(merge.saturated_fat_100g<=1)
& (merge.carbohydrates_100g<28) & (merge.proteins_100g
<12) & (merge.salt_100g<=1)].sort_values(by=['price'])
```

## **Anexo 7: Reglas\_Perfiles.txt**

Perdida de peso: Baja en calorías

Nutriscore: A, B

Kcal: < 130

Fat: < 1

saturated fat: 0

carbohydrates: < 6

sugar: < 5

proteins: no aplica

salt: < 1

price: a gusto del usuario

Volumen: alto en proteína y hidratos

Nutriscore: a, b, c, d

Kcal: no aplica

Fat: < 3

saturated fat: <1  
carbohydrates: no aplica  
sugar: < 10  
proteins: max  
salt: < 5  
price: no determinante

Alto rendimiento: hidratos y azucar  
Nutriscore: a, b, c  
Kcal: max  
Fat: < 3  
saturated fat: < 1  
carbohydrates: max  
sugar: max  
proteins: no aplica  
salt: no aplica  
price: no determinante

Balanceado: Economica, nutriscore ABC  
Nutriscore: a, b, c, d  
Kcal: < 86  
Fat: < 1  
saturated fat: 0  
carbohydrates: < 6  
sugar: <1  
proteins: <4  
salt: < 0.5  
price: ~€2

Ahorro:  
Nutriscore: no aplica  
Kcal: < 245  
Fat: < 3  
saturated fat: < 1  
carbohydrates: < 28  
sugar: < 4  
proteins: < 12

```
salt: < 1  
price: min (asc)
```

## Anexo 8: Sistema recomendador red neuronal.ipynb

Librerias

```
import pandas as pd  
import tensorflow as tf  
from tensorflow import keras  
from tensorflow.keras import layers  
import numpy as np  
from math import sqrt  
from sklearn.metrics import mean_squared_error  
from sklearn.neighbors import NearestNeighbors  
from matplotlib import pyplot as plt
```

Data

```
alimentos = pd.read_csv('productos_recomendados.csv').drop('Unnamed:  
0', axis=1)
```

```
alimentos.drop_duplicates(inplace=True, subset='id_food')
```

```
usuarios = pd.read_csv('users.csv').drop('Unnamed: 0', axis=1)
```

usuarios

alimentos

Numero de usuarios : 9

Rating = 1-4

1. Malo

2. Regular

3. Bueno

4. Excelente

productos\_recomendados de los dos dataframes

```
df = pd.merge(usuarios, alimentos, on= 'id_food')
```

df

Groupby rating & product\_name

```
df.groupby('product_name')['random_users_ratings'].mean().head()
```

```
df.groupby('product_name')['random_users_ratings'].mean().sort_value  
s(ascending=False)
```

```
print(df.keys())
```

```
df.rename(columns = {'random_users_id':'userID'}, inplace = True)
```

```
df.rename(columns = {'random_users_ratings':'rating'}, inplace =
True)
print(df.keys())
user_ids = df["userID"].unique().tolist()
user2user_encoded = {x: i for i, x in enumerate(user_ids)}
userencoded2user = {i: x for i, x in enumerate(user_ids)}
food_ids = df["id_food"].unique().tolist()
food2food_encoded = {x: i for i, x in enumerate(food_ids)}
food_encoded2food = {i: x for i, x in enumerate(food_ids)}
df["user"] = df["userID"].map(user2user_encoded)
df["food"] = df["id_food"].map(food2food_encoded)

num_users = len(user2user_encoded)
num_food = len(food_encoded2food)
df["rating"] = df["rating"].values.astype(np.float32)
# min and max ratings will be used to normalize the ratings later
min_rating = min(df["rating"])
max_rating = max(df["rating"])
print(
"Number of users: {}, Number of foods: {}, Min rating: {}, Max rating:
{}".format(
                                num_users, num_food, min_rating, max_rating
                                )
)
df = df.sample(frac=1, random_state=42)

x = df[["user", "food"]].values
y = df["rating"].aplicaciónly(lambda x: (x - min_rating) / (max_rating
- min_rating)).values

train_indices = int(0.9 * df.shape[0])
x_train, x_val, y_train, y_val = (
                                x[:train_indices],
                                x[train_indices:],
                                y[:train_indices],
                                y[train_indices:],
)
```

```
EMBEDDING_SIZE = 50
```

```
class RecommenderNet(keras.Model):
    def __init__(self, num_users, num_food, embedding_size, **kwargs):
        super(RecommenderNet, self).__init__(**kwargs)
        self.num_users = num_users

        self.num_food = num_food
        self.embedding_size = embedding_size
        self.user_embedding = layers.Embedding(
            num_users,
            embedding_size,
            embeddings_initializer="he_normal",
            embeddings_regularizer=keras.regularizers.l2(1e-6),
        )
        self.user_bias = layers.Embedding(num_users, 1)
        self.food_embedding = layers.Embedding(
            num_food,
            embedding_size,
            embeddings_initializer="he_normal",
            embeddings_regularizer=keras.regularizers.l2(1e-6),
        )
        self.food_bias = layers.Embedding(num_food, 1)

        def call(self, inputs):
            user_vector = self.user_embedding(inputs[:, 0])
            user_bias = self.user_bias(inputs[:, 0])
            food_vector = self.food_embedding(inputs[:, 1])
            food_bias = self.food_bias(inputs[:, 1])
            dot_user_food = tf.tensordot(user_vector, food_vector, 2)
            # Add all the components (including bias)
            x = dot_user_food + user_bias + food_bias
            # The sigmoid activation forces the rating to between 0 and 1
            return tf.nn.sigmoid(x)

model = RecommenderNet(num_users, num_food, EMBEDDING_SIZE)
```

```
model.compile(
    loss=tf.keras.losses.BinaryCrossentropy(),
    optimizer=keras.optimizers.Adam(learning_rate=0.001),
)
history = model.fit(
    x=x_train,
    y=y_train,
    batch_size=64,
    epochs=5,
    verbose=1,
    validation_data=(x_val, y_val),
)
plt.plot(history.history["loss"])
plt.plot(history.history["val_loss"])
plt.title("model loss")
plt.ylabel("loss")
plt.xlabel("epoch")
plt.legend(["train", "test"], loc="upper left")
plt.show()
# Let us get a user and see the top recommendations.
user_id = df.userID.sample(1).iloc[0]
foods Rated by user = df[df.userID == user_id]
food_not Rated = alimentos[
    ~alimentos["id_food"].isin(foods Rated by user.id_food.values)
]["id_food"]
food_not Rated = list(
    set(food_not Rated).intersection(set(food2food_encoded.keys()))
)

food_not Rated = [[food2food_encoded.get(x)] for x in food_not Rated]
user_encoder = user2user_encoded.get(user_id)
user_food_array = np.hstack(
    ([[user_encoder]] * len(food_not Rated), food_not Rated)
)

ratings = model.predict(user_food_array).flatten()
top_ratings_indices = ratings.argsort()[-100:][::-1]
```



```
recommended_food_ids = [
    food_encoded2food.get(food_notRated[x][0]) for x in
    top_ratings_indices
]

print("Showing recommendations for user: {}".format(user_id))
print("====" * 9)
print("Food with high ratings from user")
print("----" * 8)
top_food_user = (
    foods_rated_by_user.sort_values(by="rating", ascending=False)
                                .head(5)
                                .id_food.values
)
food_df_rows = alimentos[alimentos["id_food"].isin(top_food_user)]
for row in food_df_rows.itertuples():
    print(row.product_name, ":", row.nutriscore_grade)

print("----" * 8)
print("Top food recommendations")
print("----" * 8)
recommended_food =
alimentos[alimentos["id_food"].isin(recommended_food_ids)]

rec=[]
for row in recommended_food.itertuples():
    rec.aplicaciónend({'id_food': row.id_food})

rec = pd.DataFrame(rec)
rec
productos_recomendados = pd.merge(rec, alimentos, 'inner',
on='id_food')

perfil = str(input("Ingrese el perfil deseado entre: A) baja en
calorias, B) Volumen, C) Alto rendimiento, D) Balanceado, E) Ahorro
"))
perfil = perfil.lower()
```

```
if perfil == "a":
    #Perdida de peso
    productos =
productos_recomendados.loc[(productos_recomendados.nutriscore_grade.
isin(["a","b"])) & (productos_recomendados.energy_kcal_100g <130) &
(productos_recomendados.fat_100g<1) &
(productos_recomendados.saturated_fat_100g==0)
& (productos_recomendados.carbohydrates_100g<6) &
(productos_recomendados.sugars_100g <5) &
(productos_recomendados.salt_100g<=1)]
elif perfil == "b":
    #Perdida de peso
    productos =
productos_recomendados.loc[(productos_recomendados.nutriscore_grade.
isin(["a","b","c","d"])) & (productos_recomendados.fat_100g<3) &
(productos_recomendados.saturated_fat_100g<=1)
& (productos_recomendados.sugars_100g <10) &
(productos_recomendados.salt_100g<=1)].sort_values(by='proteins_100g',
ascending=False)

elif perfil == "c":
    #Alto Rendimiento
    productos =
productos_recomendados.loc[(productos_recomendados.nutriscore_grade.
isin(["a","b","c"])) & (productos_recomendados.fat_100g<3) &
(productos_recomendados.saturated_fat_100g<=1)
].sort_values(by=
['energy_kcal_100g','sugars_100g','carbohydrates_100g'],ascending=False)

elif perfil == "d":
    #Balanceado
    productos =
productos_recomendados.loc[(productos_recomendados.nutriscore_grade.
isin(["a","b","c","d"])) & (productos_recomendados.energy_kcal_100g
<86) & (productos_recomendados.fat_100g<1) &
(productos_recomendados.saturated_fat_100g==0)
```

```
        &      (productos_recomendados.carbohydrates_100g<6)      &
(productos_recomendados.proteins_100g      <4)      &
(productos_recomendados.salt_100g<=1)      &
(productos_recomendados.price <=2)].sort_values(by=['price'])

else:
    #Ahorro
    productos      =
productos_recomendados.loc[(productos_recomendados.energy_kcal_100g
<245)      &      (productos_recomendados.fat_100g<3)      &
(productos_recomendados.saturated_fat_100g<=1)
      &      (productos_recomendados.carbohydrates_100g<28)      &
(productos_recomendados.proteins_100g      <12)      &
(productos_recomendados.salt_100g<=1)].sort_values(by=['price'])

productos
```

## Anexo 9: Get\_data\_usuarios

```
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np
from math import sqrt
from sklearn.metrics import mean_squared_error
from sklearn.neighbors import NearestNeighbors
from matplotlib import pyplot as plt
GET DATA
df = pd.read_csv('merge.csv').drop('Unnamed: 0', axis=1)
df.drop_duplicates(inplace=True, subset='id_food')
print(len(df))
print(df.keys())
user_rating = df[['id_food', 'nutriscore_grade']]
print(len(user_rating))
sample_df_usuario1 = df.sample(2525, random_state = 6)
sample_df_usuario1
Condicion usuario 1 basado en nutriscore "a" y "b"
def f(row):
    if row['nutriscore_grade'] == 'a':
        val = 4
    elif row['nutriscore_grade'] == 'b':
```

```

                                                    val = 3
else:
                                                    val = 2

    return val

sample_df_usuario1['rating'] = sample_df_usuario1.aplicaciónly(f,
axis=1)
sample_df_usuario1.insert(0, 'Usuario', 1)
sample_df_usuario1
Condicion usuario 2  nutriscore "b","c","d"
sample_df_usuario2 = df.sample(2525, random_state = 4656)
def f(row):
    if row['nutriscore_grade'] == 'b':
                                                    val = 4
    elif row['nutriscore_grade'] == 'c':
                                                    val = 3
    elif row['nutriscore_grade'] == 'd':
                                                    val = 3
    else:
                                                    val = 1
    return val
sample_df_usuario2['rating'] = sample_df_usuario2.aplicaciónly(f,
axis=1)
sample_df_usuario2.insert(0, 'Usuario', 2)
sample_df_usuario2
Condicion usuario 3  nutriscore "a","b","c"
sample_df_usuario3 = df.sample(2525, random_state = 5659)
def f(row):
    if row['nutriscore_grade'] == 'a':
                                                    val = 4
    elif row['nutriscore_grade'] == 'b':
                                                    val = 4
    elif row['nutriscore_grade'] == 'c':
                                                    val = 3
    else:
                                                    val = 2
    return val
sample_df_usuario3['rating'] = sample_df_usuario3.aplicaciónly(f,
axis=1)
sample_df_usuario3.insert(0, 'Usuario', 3)
sample_df_usuario3
Condicion usuario 4
sample_df_usuario4 = df.sample(2525, random_state = 9897)
def f(row):
    if row['nutriscore_grade'] == 'a':
                                                    val = 1
    elif row['nutriscore_grade'] == 'b':
                                                    val = 1
    elif row['nutriscore_grade'] == 'c':
                                                    val = 4

```

```

else:
    val = 3

    return val
sample_df_usuario4['rating'] = sample_df_usuario4.aplicaciónly(f,
axis=1)
sample_df_usuario4.insert(0, 'Usuario', 4)
sample_df_usuario4
Condicion usuario 5
sample_df_usuario5 = df.sample(2525, random_state = 68874)
def f(row):
    if row['nutriscore_grade'] == 'a':
        val = 4

    elif row['nutriscore_grade'] == 'b':
        val = 4

    elif row['nutriscore_grade'] == 'c':
        val = 4

    else:
        val = 3

    return val
sample_df_usuario5['rating'] = sample_df_usuario5.aplicaciónly(f,
axis=1)
sample_df_usuario5.insert(0, 'Usuario', 5)
Aplicaciónend
sample_df_usuario1 =
sample_df_usuario1.aplicaciónend(sample_df_usuario2, ignore_index =
True)
sample_df_usuario1 =
sample_df_usuario1.aplicaciónend(sample_df_usuario3, ignore_index =
True)
sample_df_usuario1 =
sample_df_usuario1.aplicaciónend(sample_df_usuario4, ignore_index =
True)
sample_df_usuario1 =
sample_df_usuario1.aplicaciónend(sample_df_usuario5, ignore_index =
True)
sample_df_usuario1
user_rating =
sample_df_usuario1[['Usuario','rating','id_food','nutriscore_grade']]
]
user_rating
user_rating.to_csv('Usuarios_reglas.csv')

```

## Anexo 10: Get\_data

```

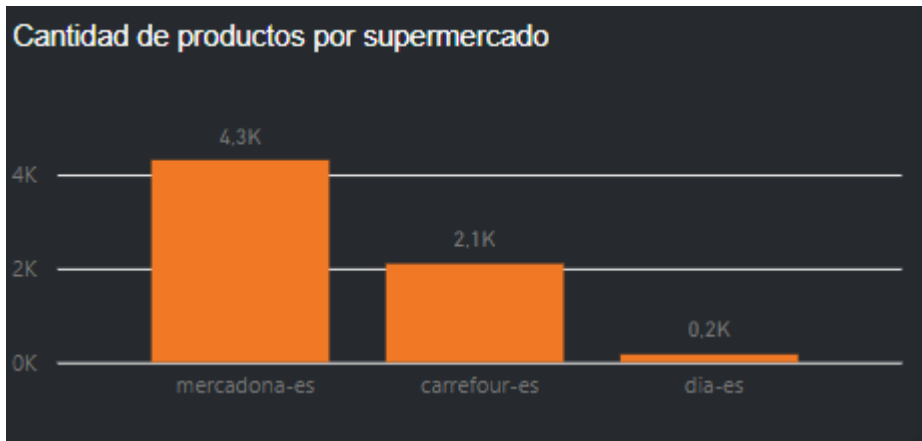
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import numpy as np

```

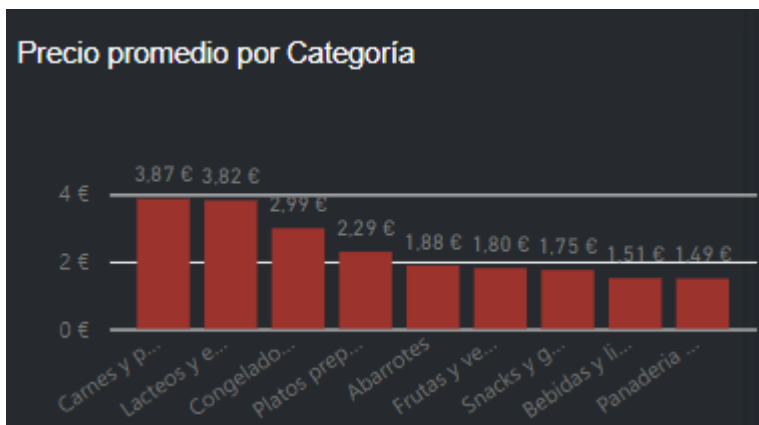
```
from math import sqrt
from sklearn.metrics import mean_squared_error
from sklearn.neighbors import NearestNeighbors
from matplotlib import pyplot as plt
GET DATA
df = pd.read_csv('merge.csv').drop('Unnamed: 0', axis=1)
df.drop_duplicates(inplace=True, subset='id_food')
print(len(df))
print(df.keys())
user_rating = df[['id_food', 'nutriscore_grade']]
print(len(user_rating))
random_users_id = np.random.randint(1, 10, size=7575)
random_users_ratings = np.random.randint(1, 5, size=7575)
user_rating['random_users_id'] = random_users_id
user_rating['random_users_ratings'] = random_users_ratings
user_rating
user_rating.to_csv('user_rating3.csv')
user1 = pd.read_csv('user_rating.csv').drop('Unnamed: 0', axis=1)
user2= pd.read_csv('user_rating2.csv').drop('Unnamed: 0', axis=1)
user3 = pd.read_csv('user_rating3.csv').drop('Unnamed: 0', axis=1)
user1
user2
user3
user1 = user1.aplicaciónend(user2, ignore_index = True)
user1
user1 = user1.aplicaciónend(user3, ignore_index = True)
user1
user1.to_csv('users.csv')
```

## Anexo 11: Tablero de visualización general

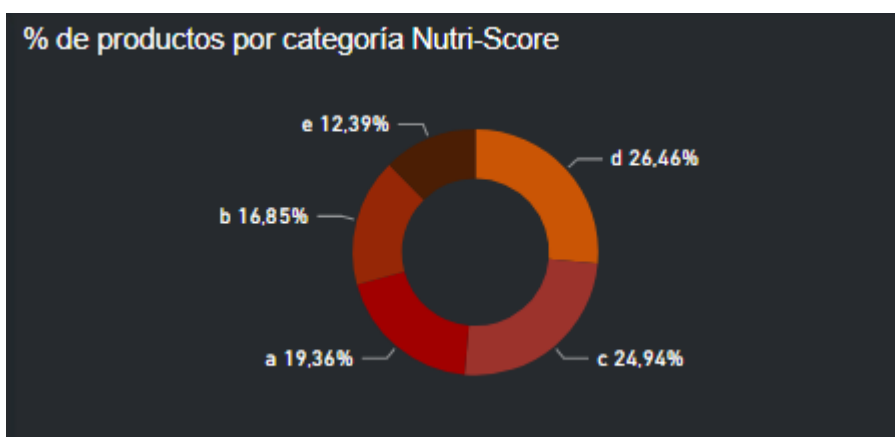
### Anexo 11.1: Cantidad de productos por supermercado



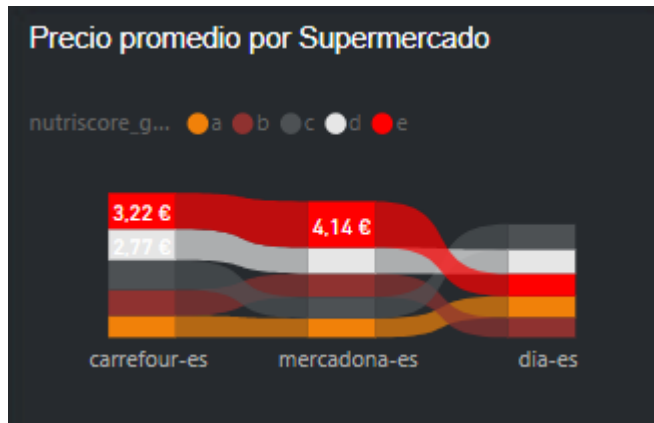
### Anexo 11.2: Precio promedio por categoría



### Anexo 11.3: Porcentaje de productos por categoría Nutri-Score



## Anexo 11.4: Precio promedio por supermercado



## Anexo 12: PyDiaScraper

```
from Categories import Categories
import json
```

```
def jsonParser(data):
    return json.dumps(data, indent=4, sort_keys=True,
ensure_ascii=False).encode('utf8').decode('utf8')
```

```
def init():
    logo = ""
```

D I A S C R A P P E R



[illegible]

```

"""
print(logo)
print("\n")
print("Avoid entering more than 10 categories :)")
print("\n")
print("=====
=")
print("\n")
print("\n")
cat_number = int(input("Enter number of categories (Default 5):
") or "5")
filename = (input("Enter file name: ") or "products")+(".json")
print("\n")
print("Retrieving categories...")
sl = slice(cat_number)
categories = Categories.getCategoriesURLs()[sl]
scrapped = []
for category in categories:
    products = Categories.scrapCategory(category, scrapped)
    scrapped.extend(products)
open(filename, "w").write(jsonParser(scrapped))
print("\n")
print("File saved as " + filename)

init()
-----
-----
import re
import requests
import json
from Sitemap import Sitemap
from bs4 import BeautifulSoup

```

```

class Categories:
    def getCategoriesURLs():
        sitemap = Sitemap.fetchSitemap()
        regexp = "<loc>(https:\\\\.*?\\/cf)<\\/loc>"
        return re.findall(regexp, sitemap)

        scrappedProducts = []
    def scrapCategory(categoryURL, scrappedProducts = []):
        print("\n")
        print("Retrieving products from " + categoryURL)

        print("=====
=")
        response = requests.get(categoryURL)
        soup = BeautifulSoup(response.content, 'html.parser')
        products = soup.select('.product-list__item')
        for product in products:
            anchor = product.select('a')[0]
            price_container
            product.select('.price_container')[0]
            title = anchor.get('title')
            print("Retrieving product: " + title)
            href = anchor.get('href')
            price
            price_container.select('p')[0].text.replace('\n',
            '').replace('\t', '').replace('\r', '').replace(u'\xa0', u' ')
            priceAvg
            price_container.select('p')[1].text.replace('\n',
            '').replace('\t', '').replace('\r', '').replace(u'\xa0', u' ')
            scrappedProducts.append({'title': title, 'href':
            href, 'price': price, 'priceAvg': priceAvg})

        print("=====
=")
        print("\n")
        return scrappedProducts
-----
-----
import re
from Sitemap import Sitemap

class Products:
    def getProductsURLs():
        sitemap = Sitemap.fetchSitemap()
        regexp = "<loc>(https:\\\\.*?\\/p\\/.*?)<\\/loc>"
        return re.findall(regexp, sitemap)

-----
-----
import requests

```

```
from Url import URL
```

```
class Sitemap:  
    def fetchSitemap():  
        response = requests.get(URL.SITEMAP).text  
        return response
```

```
-----  
----
```

```
class URL:  
    SITEMAP = 'https://www.dia.es/compra-online/sitemap.xml'
```