Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

MINERÍA DE DATOS: LO QUE LAS PÁGINAS DE REALFOODING NO NOS DEJAN VER

LAURA ARGÜELLES PÉREZ

Tutora:

Noelia Rico Pachón

Dpto. de Informática

UNIVERSIDAD DE OVIEDO

Máster Universitario en Análisis de Datos

para la Inteligencia de Negocios

Resumen

El Realfooding, promovido en España por Carlos Ríos, se enfoca en la promoción de una alimentación basada en el consumo de alimentos frescos y mínimamente procesados con el objetivo de fomentar una dieta equilibrada y variada. Este concepto de alimentación se basa en estudios que determinan que el consumo de alimentos sin procesar ayuda a mejorar la salud general y a reducir el riesgo de enfermedades diversas.

En el presente trabajo fin de máster se realiza un proceso de minería de datos centrado en la aplicación de técnicas de *web scraping* y análisis de datos para explorar y evaluar la información disponible en línea sobre las recetas que se alinean con la alimentación Realfooding.

En primer lugar, se recolectan recetas publicadas en la principal página web Realfooding con el fin de analizar su estructura para, a partir de las características del texto, determinar diferentes aspectos de estas. Esto se realiza mediante técnicas de *web scraping*, que permiten la obtención de un gran volumen de datos de manera eficiente.

Una vez extraídos los datos relativos a las recetas se aplican sobre ellos herramientas para el análisis de la información obtenida, centrándose en realización de consultas de información ordenada y en técnicas de visualización de datos eficiente. De este modo se desarrollaron gráficos y un panel de control con visualizaciones dinámicas que permitirán un análisis claro de las recetas. Además, el panel de control permite a los usuarios interactuar con los datos de manera intuitiva y explorar diferentes aspectos de la información recopilada.

Índice

1 Introducción

La cantidad de información publicada en medios digitales es actualmente abrumadora, llegando a generarse cada día aproximadamente 3,5 quintillones de bytes (gizblogs, 2024). Por este motivo, hay muchos ámbitos en los cuales el gran volumen de datos que se genera a diario es demasiado grande para ser sintetizado por los humanos de forma directa. Esto dificulta el acceso a informaciones que surgen de la agregación de los datos disponibles, como, por ejemplo: números totales, estadísticas descriptivas, identificación de patrones y tendencias etc. Este es el caso de las páginas web dedicadas a la elaboración de comidas, donde muchas veces simplemente se proporcionan catálogos de recetas, los cuales son útiles para consultas directas, pero dificultan la búsqueda más sofisticada de información sobre las mismas.

Quienes emplean estas recetas suelen ser personas que buscan inspiración para su cocina, o bien aquellas que buscan cuidar o mejorar su alimentación. Sin embargo, en numerosas ocasiones estas personas carecen del tiempo necesario para explorar estas webs en busca de las recetas que mejor se adapten a sus necesidades. Buscar la receta adecuada en cada momento requiere tiempo y dedicación. La receta debe satisfacer los gustos individuales, ajustarse a una dieta concreta, requerir un cierto tiempo de preparación o poder ser elaborada con los ingredientes disponibles en el momento. Según un estudio elaborado por Hello Fresh en enero de 2023 un 58% de los españoles encuestados cocina en casa todos los días de la semana. De los españoles encuestados el 56% reconoce que tomar la decisión sobre qué cocinar le produce estrés, esto sin tener en cuenta la compra de ingredientes o la propia elaboración del plato. Además, el 34 % de los encuestados considera que no lleva a cabo una dieta equilibrada por falta de tiempo y el 27% porque no encuentra inspiración sobre que cocinar (Hello Fresh, 2023).

Del problema ilustrado en el estudio realizado por Hello Fresh surge la idea de este trabajo, cuyo objetivo es **unificar las recetas disponibles** para poder **conocer datos globales** de las mismas y **facilitar su búsqueda** en base a diferentes aspectos.

Sin duda, hacer esto para todas las recetas disponibles en internet sería inabordable, por lo que el trabajo se acota a las recetas relacionados con el “movimiento Realfooding”. Este modelo de alimentación se ha popularizado en los últimos años bajo la promoción

del nutricionista Carlos Ríos. La influencia de este se refleja en el registro de la marca "Realfooding" por parte del propio nutricionista en 2021 (Oficina Española de Patentes y Marcas, s.f.). El movimiento no se enfoca como una dieta, sino como un estilo de vida a través de la alimentación, tratando de buscar alimentos de temporada, frescos y de calidad (Olalla, 2020), dejando de lado aquellos alimentos ultraprocesados que por su intenso sabor buscan crear una dependencia en las personas y en los que a menudo, durante las diferentes etapas de fabricación del alimento, se utilizan técnicas, materiales o productos químicos que persisten en el producto final y pueden ocasionar problemas de salud de diversa gravedad tanto a corto como a largo plazo.

Con este fin, para realizar este proyecto se decide **extraer** toda la **información** posible la principal web de recetas de alimentación Realfooding, llamada MyRealfood, por medio de técnicas de minería de datos. Entre las numerosas técnicas disponibles, en este trabajo comenzaremos realizando ***web scraping***, proceso mediante el cual se pueden obtener grandes volúmenes de datos automatizando la extracción de información de la web, basándose en el uso de los lenguajes de marcado. Esto permitirá obtener información para realizar un **análisis sobre los datos de recetas**, almacenándolos de forma estructurada para así posteriormente poder realizar consultas sobre ellos y visualizarlos en un panel de control que muestre las distintas recetas obtenidas de forma clara. El panel de control nos permitirá así afinar la búsqueda de recetas mediante el uso de filtros personalizados.

De esta forma el presente trabajo fin de máster pretende simplificar una búsqueda que podría tomar bastante tiempo, conduciendo al abandono del propósito de una dieta equilibrada por la falta de tiempo.

La estructura de este trabajo es la que sigue: En el Capítulo 2 se define la situación actual y los problemas identificados, que motivan la realización de este trabajo. A continuación, en el Capítulo 3 se presenta la propuesta de solución para la gestión de recetas Realfooding, abordando los problemas identificados en el capítulo anterior. Para ello, entre otras cosas, se definen los requisitos de la solución propuesta y la arquitectura definida para abordar la implementación, detallando las tecnologías utilizadas. En el Capítulo 4 se detalla el proceso de recopilación de información, cuyo análisis es explicado en el Capítulo 5. En el Capítulo 6 se presenta la implementación del panel de control en el que se visualiza y permite la interacción con la información más relevante obtenida.

Por último, el Capítulo 7 muestra la planificación del proyecto incluyendo la definición de tareas y el cronograma de trabajo. El trabajo finaliza con las conclusiones obtenidas.

2 Descripción de la situación actual

El acceso a contenido de calidad de forma sencilla, en cuanto a recetas se refiere, es un desafío para aquellos que desean mejorar sus hábitos alimenticios. Es cierto que hoy en día existen múltiples páginas webs para la recopilación y organización de recetas, que buscan facilitar a las personas el acceso a la información de una forma clara. A pesar de ello, habitualmente estas páginas web que aúnan estas recetas disponen de extensos catálogos, pero es común encontrar que la interfaz de las páginas limita el acceso a su información. Esto dificulta los tipos de consultas más refinadas basadas en filtros o el uso de información agregada.

Este es precisamente el caso de las recetas que se basan en el movimiento Realfooding. En la actualidad, la página de referencia de recetas Realfooding es la web Myrealfood , y por eso a lo largo de este trabajo nos centraremos en la extracción de información de esta para su posterior análisis.

2.1 Auditoría de la página web

La página web seleccionada dificulta el acceso a la información filtrada y la obtención de características comunes de las recetas agrupando por categorías flexibles. Un ejemplo de esto es que únicamente muestra la información agrupada por las recetas mejor valoradas, por aquellas más populares o por las que están en tendencia. Pero ¿qué ocurre si queremos agrupar aquellas recetas que tienen el mismo conjunto de ingredientes?, ¿o aquellas de una duración similar? A pesar de que esta información puede obtenerse con los datos de los que dispone de la web, la interfaz limita al usuario el acceso a esta información.

Tal y como está implementada la página, para obtener más información, como los ingredientes o los pasos para elaborarlas, sería necesario ir mirando receta por receta. Esto es excesivamente tedioso si por ejemplo lo que se quiere es un conjunto de recetas que cumplan unos determinados requisitos, por no decir inabordable debido a la gran cantidad de recetas almacenadas en la web.

Para comprender mejor la problemática, a continuación, se examinará la interfaz de la web, desde la página principal a la vista de las recetas de forma individual, lo que nos ayudará a entender cómo está organizada la información actualmente y por qué resulta

insuficiente para los usuarios. Además, esta auditoría nos permitirá conocer la estructura de la web, lo cual será útil para la posterior extracción de información.

PÁGINA DE INICIO

En la Figura 2.1 podemos ver la información que se muestra en la primera página de acceso a las recetas.

En ella encontramos un buscador que permite introducir el nombre de la receta. Al realizar la búsqueda, se muestran debajo del campo del buscador aquellas recetas que coinciden, ordenadas siguiendo un criterio en base a su valoración. Sin embargo, esta búsqueda de coincidencias se hace simplemente en base al título de la receta, no usa otra información adicional sobre el texto contenido en la receta, lo cual no permite por ejemplo buscar por ingredientes.

A continuación, ilustraremos esto mediante un ejemplo. Para ello, realizamos la búsqueda para encontrar aquellas recetas que contienen “huevo”, pero como podemos observar en la Figura 2.2 el resultado que arroja, debido a la implementación del sistema, son aquellas recetas cuyo ingrediente principal es el “huevo” porque lo llevan en el título. En la Figura 2.2 también podemos observar en la parte superior, debajo de la barra de búsqueda, una serie de filtros predefinidos como por ejemplo “desayuno”, “comida” o “postres”, pero estos no son accesibles puesto que para ello es necesaria una suscripción de pago.

Otro ejemplo de búsqueda sería utilizar la palabra “harina”, ingrediente el cual no suele ser principal en las recetas. Por ello, como podemos ver en la Figura 2.3, el resultado contendrá tanto las recetas que tengan harina como ingrediente como las que no, pero que aparecen en la búsqueda porque su nombre incluye “sin harina”, lo cual probablemente no sea el resultado esperado.

Figura 2.1 Página principal de Myrealfood

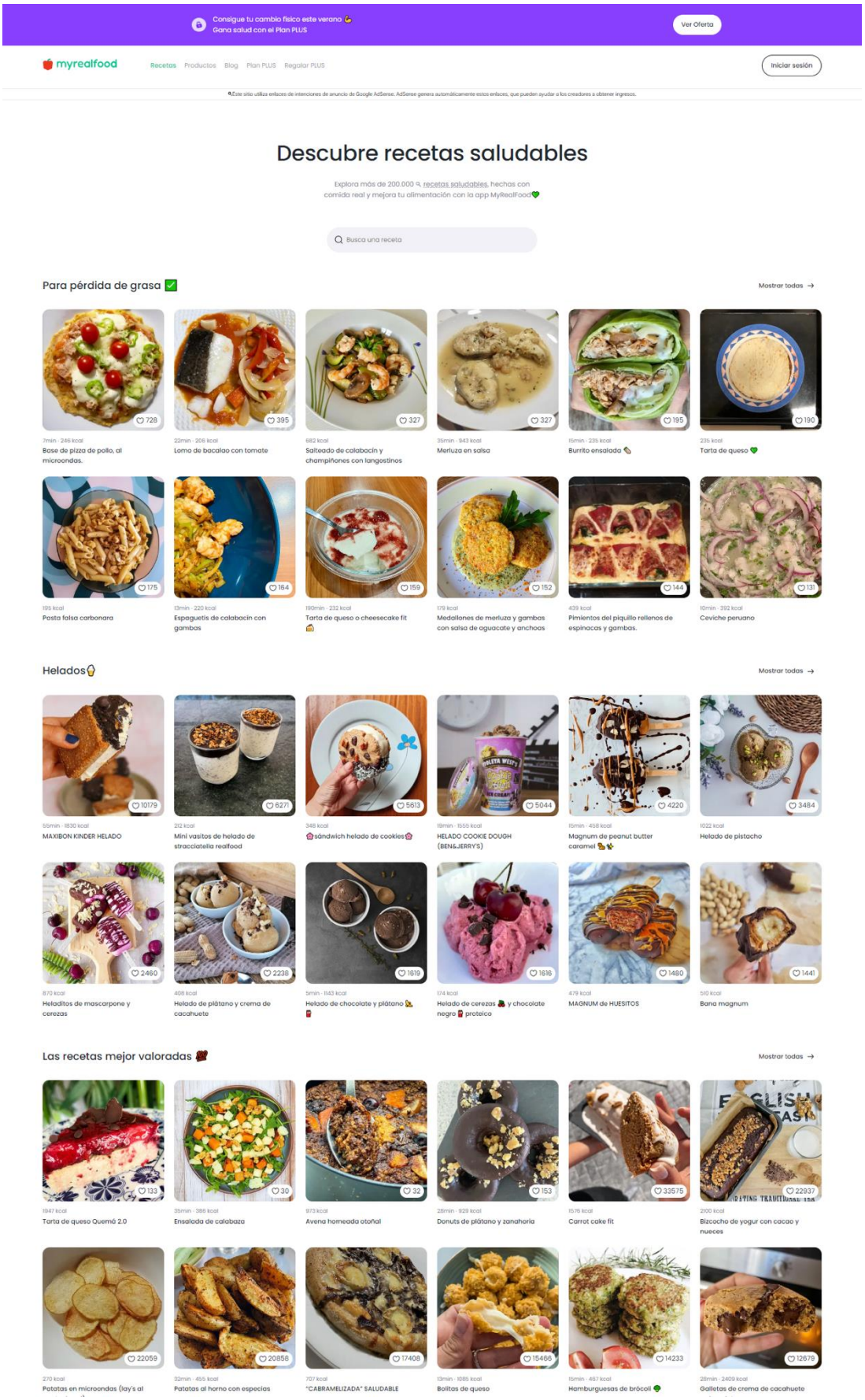


Figura 2.2 Resultado de la búsqueda "huevo"

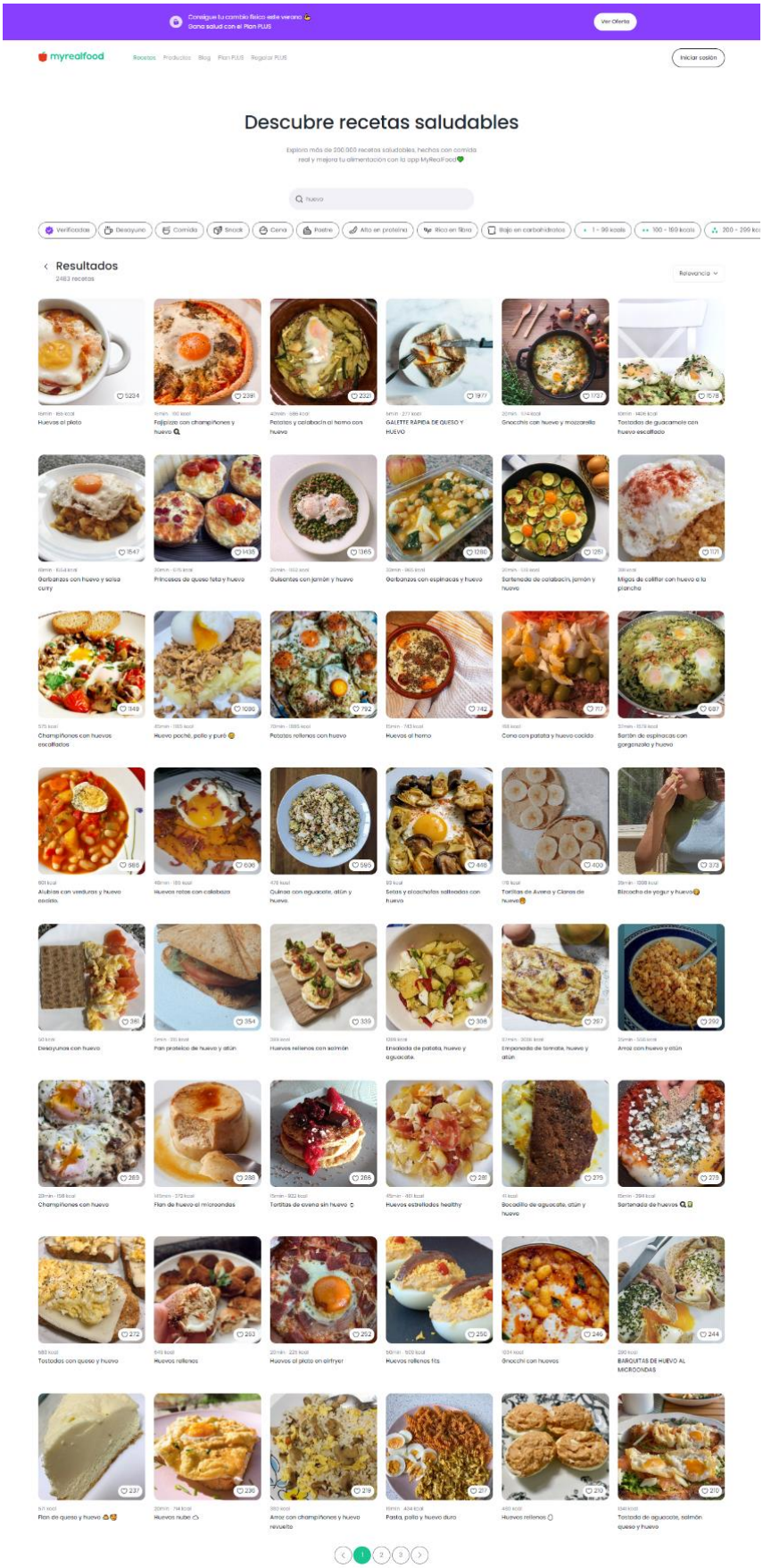
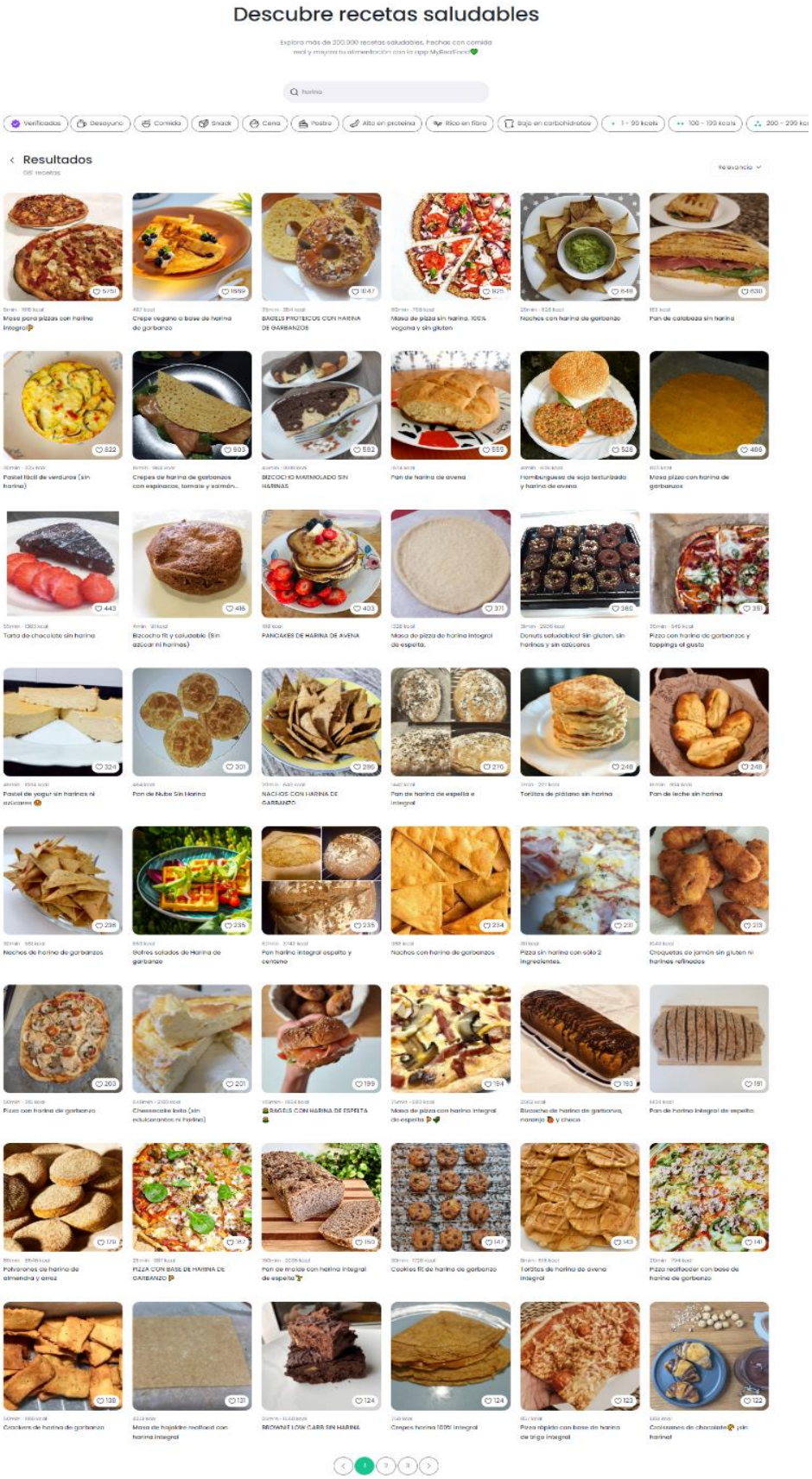


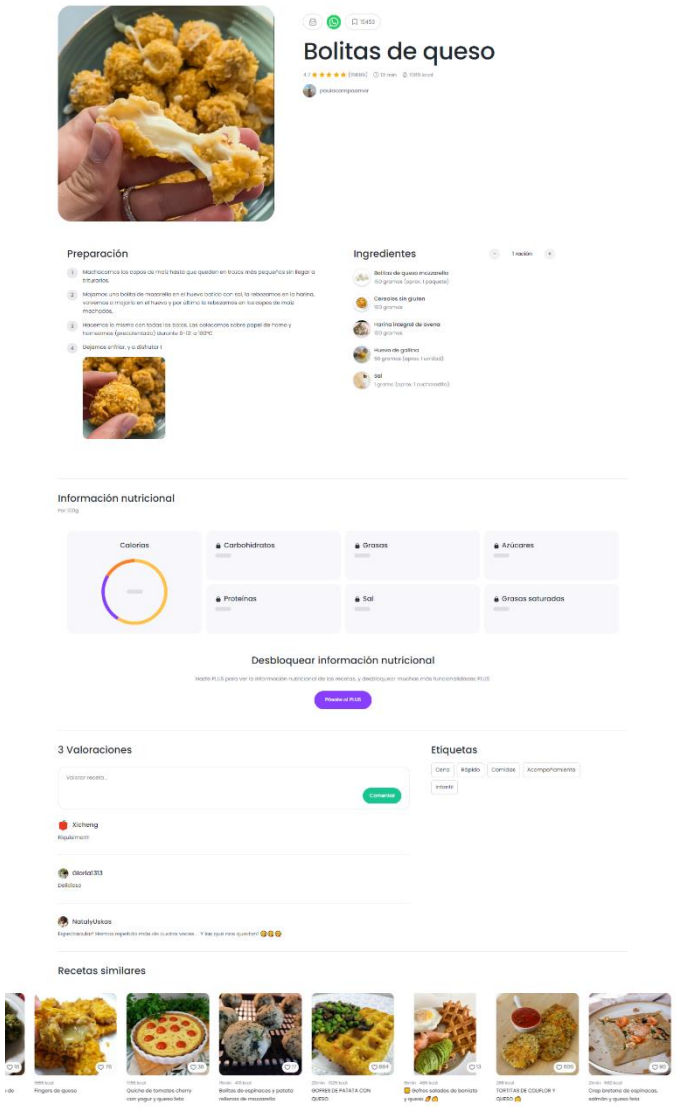
Figura 2.3 Resultado de la búsqueda "harina"



INFORMACIÓN DE LA RECETA

En lo que se refiere a las recetas completas, debemos acceder de forma individual a cada una de ellas. Esto nos proporciona los ingredientes y pasos de la receta, así como sus características. El acceso a la información de una receta puede realizarse mediante la búsqueda del nombre o accediendo a la receta a través del listado que aparece predefinido en la pantalla de inicio. A continuación, veremos como ejemplo una receta de "bolitas de queso". En la Figura 2.4 tenemos una visión general de la página particular de esta receta. La estructura que se muestra en esta imagen se corresponde con la que se muestra para todas las recetas.

Figura 2.4 Visión general de la página "información de la receta"



Como decíamos, la Figura 2.4 muestra la interfaz para una receta concreta, que se compone de los elementos de interacción social, el nombre de la receta, los ingredientes, los pasos para elaborar la receta y una parte donde los usuarios pueden introducir valoraciones en formato textual. Además, una parte de esta interfaz está reservada para la información nutricional de la receta, pero este contenido no es gratuito. Finalmente, aparecen una serie de recetas consideradas similares a la principal. En lo subsiguiente se desglosarán los distintos componentes de esta interfaz. Conocer las distintas partes de la interfaz es indispensable para realizar el posterior proceso de web scraping, a partir del cual se extraerá la información a analizar.

La En lo que respecta al título de la receta, en este caso "Bolitas de queso", está destacado en una fuente grande y clara. Estos aspectos visuales son importantes ya que la codificación de la página web trata con diferentes elementos sus componentes visuales. Entender la distribución de estos resultará útil para identificar las etiquetas del código fuente de las que se extraerá la información. Tal y como se detallará en el siguiente capítulo, esto se hará usando las etiquetas `HTML` que se usan para codificar la web. A modo de avance, en este caso, esta fuente destacada indica que su correspondencia en el código `HTML` será con una etiqueta de formato cabecera. Esto motiva de nuevo la necesidad de este análisis previo a la comprensión del código fuente. Justo debajo de este título se proporciona información como la valoración, en este caso tiene 4.7 estrellas representada por 15699 valoraciones, el tiempo de preparación (13 minutos) y las calorías (1085 kcal). La receta también incluye el nombre de la persona que la ha publicado.

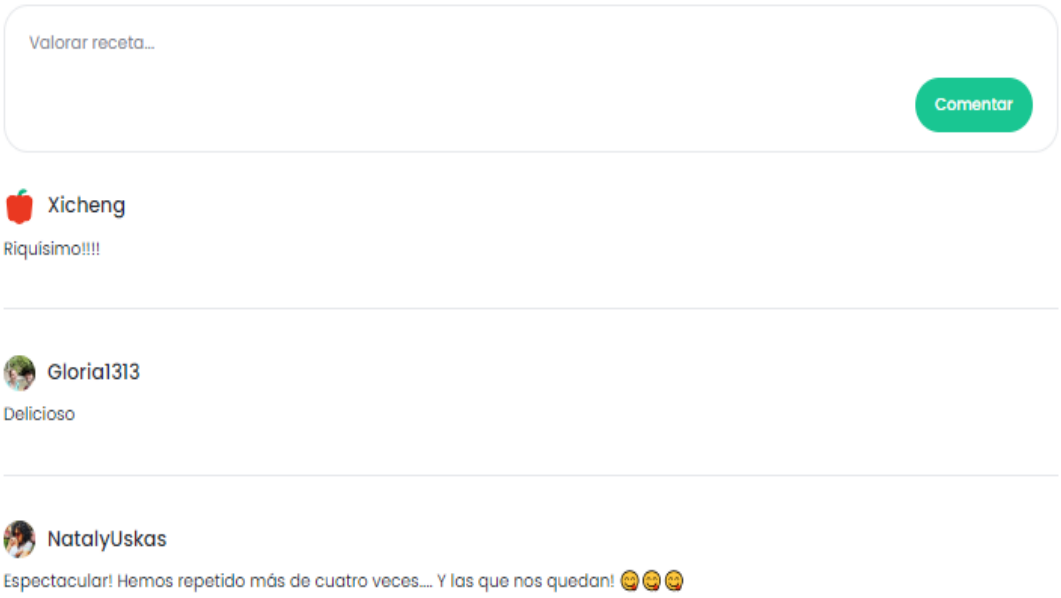
Figura 2.5 hace referencia al **título y parte superior** de la receta. En la parte superior se observan varios iconos de interacción social, la opción de compartir la receta por redes sociales y la de guardar la receta en la lista de favoritos (para esta opción es necesario tener registrada una cuenta de usuario en la página), y además también permite imprimirla. En lo que respecta al título de la receta, en este caso "Bolitas de queso", está destacado en una fuente grande y clara. Estos aspectos visuales son importantes ya que la codificación de la página web trata con diferentes elementos sus componentes visuales. Entender la distribución de estos resultará útil para identificar las etiquetas del código fuente de las que se extraerá la información. Tal y como se detallará en el siguiente capítulo, esto se hará usando las etiquetas `HTML` que se usan para codificar la web. A modo de avance, en este caso, esta fuente destacada indica que su correspondencia en el código

HTML será con una etiqueta de formato cabecera. Esto motiva de nuevo la necesidad de este análisis previo a la comprensión del código fuente. Justo debajo de este título se proporciona información como la valoración, en este caso tiene 4.7 estrellas representada por 15699 valoraciones, el tiempo de preparación (13 minutos) y las calorías (1085 kcal). La receta también incluye el nombre de la persona que la ha publicado.

Figura 2.5 Título, valoración, tiempo, calorías y autor de la receta

Otro elemento de interacción social es la posibilidad de escribir valoraciones como se puede ver en la Figura 2.6.

Figura 2.6 Valoraciones de la receta

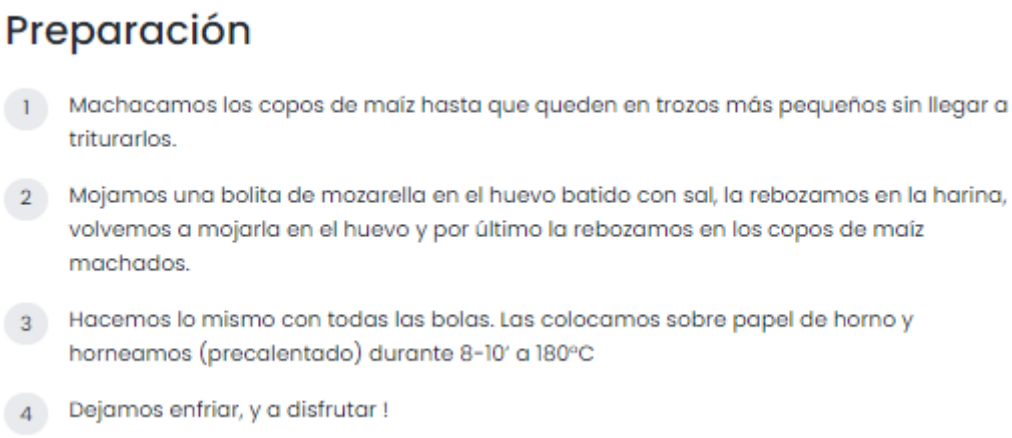


Por otro lado, los **ingredientes** para elaborar la receta se encuentran en la sección "Ingredientes", que se muestra también en formato cabecera, la cual se puede ver reflejada en la Figura 2. 7. En esta sección se enumeran los ingredientes necesarios para llevar a cabo, en este caso, las "bolitas de queso".

Figura 2. 7 Ingredientes

Los **pasos detallados** para elaborar la receta se encuentran en la sección "Preparación", que se ilustra en la Figura 2. 8 .En dicha sección, se enumeran las instrucciones necesarias para preparar la receta.

Figura 2. 8 Preparación



2.2 Problemas identificados

El primer paso para poder plantear soluciones de minería de datos eficientes es conocer los problemas a los que nos enfrentamos. En este caso esos problemas se identifican a través de las carencias del sitio web para ofrecer información filtrada y agregada. A continuación, se listan los problemas detectados en el análisis de la situación actual realizado en la sección anterior:

- No es posible obtener estadísticas descriptivas sobre las recetas agrupadas en función de sus características.
- No es posible conocer qué ingredientes son los más o menos utilizados.
- No es posible buscar aquellas recetas que contienen ciertos ingredientes en la preparación.
- No es posible hacer comparaciones sobre la frecuencia del uso de los ingredientes. Esto impide evaluar el nivel al que las recetas cumplen con los estándares de Realfooding.
- No es posible extraer las recetas filtradas por el tiempo de preparación que requieren.

- Para conocer con qué etiquetas se marca cada tipo de receta, es necesario hacerlo de forma individual por lo que no se tiene un conocimiento conjunto de las palabras que se usan con más frecuencia para categorizar las recetas.
- No es posible realizar ningún tipo de ordenación de las recetas, por ejemplo, por valoración.

Por todo lo expuesto, los problemas mencionados dificultan la ejecución de las consultas de datos más refinadas, lo cual motiva la realización del proceso de minería de datos que se presenta en este proyecto.

3 Solución propuesta

Este proyecto fin de máster tiene como objetivo ofrecer una solución a los problemas de la situación actual identificados en la Sección 2.2 de este documento. Con el fin de resolver estos problemas se realizará un proceso de minería de datos que comienza con la extracción de información para su posterior análisis y visualización. En este capítulo, en el cual se concentra la memoria técnica del proceso seguido, comenzaremos dando una idea global del proceso completo, la cual se detallará en profundidad de la siguiente forma. Tras realizar un esquema general de la solución, en primer lugar, se definirán los requisitos funcionales que debe implementar dicha solución, indispensables en cualquier trabajo de desarrollo, los cuales están basados en las necesidades de problemas a resolver identificados. El siguiente paso para lograr dichos requisitos será identificar las herramientas con las que se desarrollará la implementación de la solución propuesta. Una vez identificados requisitos y herramientas, es necesario detallar la arquitectura que involucre los diferentes componentes a tener en cuenta bajo la que se llevará a cabo el proyecto, la cual define la interacción de los distintos módulos.

3.1 Propuesta general

Para llevar a cabo este proceso, una vez determinado el sitio web del que se extraerán los datos se procede a la automatización de la extracción de la información. Este proceso implica acceder a la página web mediante su código fuente sin requerir el uso de un navegador. Este código fuente está escrito en el Lenguaje de Marcado de Hipertexto, HTML por sus siglas en inglés (HyperText Markup Language) es el estándar para crear y diseñar páginas web. Fue creado por un científico de la computación británico, Tim Berners-Lee, en la década de 1990 (Soto, El mundo, 2019). Con ese código fuente, se puede acceder a distintas partes del contenido HTML para encontrar las estructuras en las que se almacene la información, descargar distintos trozos de datos y extraer la información deseada. En este caso, entre los elementos de interés encontramos por ejemplo los títulos de recetas, los ingredientes o los pasos a seguir para la preparación de estas.

Los datos extraídos se almacenan en una estructura organizada para facilitar su manipulación y posterior análisis. Pero esto no es directo a partir del código fuente, sino que una vez que se dispone de los datos, es necesario llevar a cabo una limpieza de los

mismos puesto que estos pueden estar en formatos inconsistentes. Por ejemplo, la letra ñ no es reconocida de forma correcta como tampoco lo son los caracteres especiales que los usuarios incluyen en los textos de preparación de las recetas.

Una vez realizado el paso anterior los datos se analizan para extraer información útil, como por ejemplo la categorización de recetas por tipo de comida principal (desayuno, almuerzo, cena), o la filtración de recetas por alimentos según las preferencias dietéticas o alérgicas.

Una vez extraída la información, es muy importante la forma de presentarla para que sea fácilmente comprensible y usable por los usuarios. Se pueden crear gráficos que permitan a los usuarios explorar recetas según diferentes criterios, como calorías, tiempo de preparación o ingredientes. Por ejemplo, los gráficos de barras pueden ser útiles para comparar la cantidad de recetas disponibles en diferentes categorías o los ingredientes más frecuentes en cada una de ellas. Una técnica de visualización más avanzadas y por tanto más útil para los usuarios el desarrollo de un panel de control. Este panel debe permitir el uso de filtros dinámicos para refinar la búsqueda de recetas. Estos filtros pueden incluir diversa información contenida en los datos extraídos, además de poder elaborarse filtros más avanzados como por ejemplo las barras de búsqueda en las que el usuario pueda introducir los ingredientes que desea en sus recetas.

De esta forma visual se busca reducir el tiempo que las personas tienen que emplear en buscar recetas que se adapten a sus necesidades, o simplemente buscar inspiración para elaborar sus propias recetas y así poder mantener una dieta equilibrada sin emplear demasiado tiempo buscando en la web.

Otro aspecto interesante es el uso de ingredientes en las recetas. Al disponer de información almacenada acerca de todos los ingredientes que se emplean en las recetas, esto nos permitirá poder obtener aquellas que contengan ciertos ingredientes, esto nos facilitará optimizar el uso de alimentos en el hogar. Adicionalmente se podrá conocer información sobre que alimentos son utilizados con mayor frecuencia lo que puede ayudar a los usuarios a organizar sus listas de la compra o simplemente como inspiración para futuras recetas.

Además de los aspectos mencionados, el proyecto también busca sintetizar información relacionada con la complejidad y aporte nutricional de las recetas. Al analizar los tiempos

de preparación para diferentes platos, así como sus calorías, se puede ofrecer a los usuarios la capacidad de establecer límites para dichos valores, lo que mejorará la experiencia del usuario al proporcionarle una mayor capacidad de selección.

3.2 Requisitos

Los requisitos en este proyecto se refieren a las especificaciones y funcionalidades que deben cumplir tanto los datos extraídos como el análisis y visualizaciones posteriores para satisfacer las necesidades de los usuarios. El alcance de estos requisitos permitirá evaluar de una forma objetiva si el proyecto cumple con los resultados esperados.

Con el fin de desglosar estos requisitos se dividirán en dos partes. Por un lado, se definen aquellos requisitos relativos al proceso de extracción de datos. Por otro lado, se definen aquellos requisitos relacionados con la visualización de la información. Esto se debe a que, para abordar el desarrollo del trabajo, se ha decidido estructurar el proyecto en dos niveles diferenciados. En primer lugar, los requisitos orientados a los desarrolladores se centrarán en la interacción con los datos y la implementación técnica del sistema. Este nivel detalla las funcionalidades permitidas para un usuario del sistema con conocimientos técnicos. En segundo lugar, los requisitos del panel de control están dirigidos a los hipotéticos usuarios finales, quienes pueden no tener conocimientos técnicos previos. El objetivo aquí es crear una interfaz intuitiva y accesible que permita a los usuarios interactuar con el sistema de manera sencilla y eficiente.

Respecto del análisis de los datos a nivel desarrollador, los requisitos se enfocarán en el planteamiento de diferentes consultas sobre ellos para obtener información concisa. A continuación, se detallan las condiciones establecidas para la realización de estas:

- REQUISITO DE.1. El desarrollador tiene que ser capaz de calcular las calorías promedio según la categoría temporal de la receta (desayuno, comida, cena).
- REQUISITO DE.2. El desarrollador tiene que ser capaz de calcular el tiempo medio de preparación según la categoría temporal de la receta (desayuno, comida, cena).
- REQUISITO DE.3. El desarrollador tiene que ser capaz de conocer los ingredientes utilizados con mayor o menor frecuencia para cada categoría temporal de recetas (desayuno, comida, cena).

- REQUISITO DE.4. El desarrollador tiene que ser capaz de conocer los ingredientes utilizados con mayor o menor frecuencia en una lista de recetas filtrada en función de palabras claves.
- REQUISITO DE.5. El desarrollador tiene que ser capaz de conocer los ingredientes utilizados con mayor o menor frecuencia en una lista de recetas filtrada en función de palabras claves para cada categoría temporal de recetas (desayuno, comida, cena).
- REQUISITO DE.6. El desarrollador tiene que ser capaz de consultar las recetas que contienen todos los ingredientes de un subconjunto de ingredientes.
- REQUISITO DE.7. El desarrollador tiene que ser capaz de consultar las recetas que contienen al menos uno de los ingredientes de un subconjunto de ingredientes.
- REQUISITO DE.8. El desarrollador tiene que ser capaz de consultar las recetas que requieren un cierto tiempo de preparación, definido por un intervalo de tiempo mínimo y máximo, siendo cada uno de estos valores opcionales.
- REQUISITO DE.9. El desarrollador tiene que ser capaz de consultar las recetas que tienen una valoración concreta, definida por un intervalo de valoración mínimo y máximo, siendo cada uno de estos valores opcionales.
- REQUISITO DE.10. El desarrollador tiene que ser capaz de consultar las recetas que tienen un número de calorías concreto, definido por un intervalo de valores mínimo y máximo, siendo cada uno de estos valores opcionales.
- REQUISITO DE.11. El desarrollador tiene que ser capaz de realizar la comparación en la frecuencia de usos de ingredientes.
- REQUISITO DE.12. El desarrollador tiene que ser capaz de consultar el cálculo del número de pasos de cada receta.
- REQUISITO DE.13. El desarrollador tiene que ser capaz de consultar el cálculo de ingredientes utilizados en cada receta.

En lo que respecta al panel en control (habitualmente referido como *dashboard* en ambientes técnicos) se detallan a continuación los requisitos considerados:

- REQUISITO DA.1. El *dashboard* debe mostrar tres cajas de información en las que se resuman el tiempo y las calorías medias de las tres comidas principales (desayuno, comida y cena).

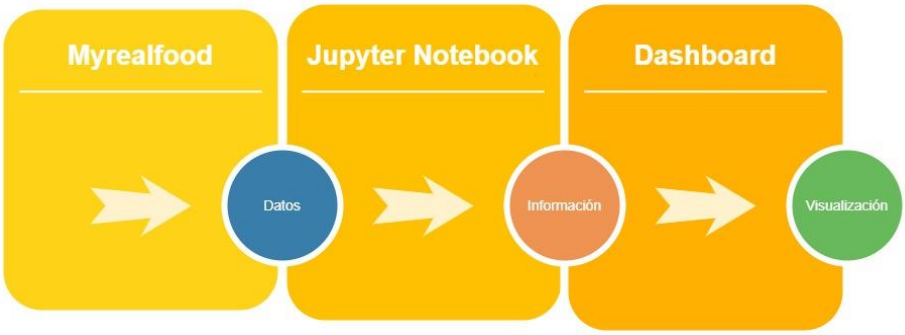
- REQUISITO DA.2. El *dashboard* debe mostrar dos botones de selección, uno para el tipo de comida y otro para elegir la palabra clave que hace referencia a las categorías y con ello filtrar un gráfico de barras.
- REQUISITO DA.3. El *dashboard* debe mostrar dos botones para filtrar la tabla en función de la lista de ingredientes que se quieran incluir.
- REQUISITO DA.4. El *dashboard* debe mostrar un filtro deslizante para filtrar la tabla en función del tiempo que queremos emplear para elaborar el plato.
- REQUISITO DA.5. El *dashboard* debe mostrar un filtro deslizante para filtrar la tabla en función de las calorías que queremos que proporcione la receta.
- REQUISITO DA.6. El *dashboard* debe mostrar un filtro deslizante para filtrar la tabla en función de la valoración que queremos que haya recibido la receta.

3.3 Arquitectura

Por resumir lo explicado hasta ahora, el proceso seguido es el siguiente. En primer lugar, se recopilará mediante la información disponible, mediante la cual se creará un `DataFrame` con el objetivo de almacenar todos los datos procedentes de la página web. Los `DataFrame` son estructuras de datos tabulares que permiten almacenar datos de diferentes tipos (numéricos, cadenas de texto, fechas, etc.) en filas y columnas. Cada columna representa una variable y cada fila representa una observación. Tener toda la información almacenada en un `DataFrame` permitirá tener una fuente de datos estructurada sobre la que posteriormente se podrán realizar consultas. De esta manera se busca poder sintetizar diversos aspectos que no están disponibles en la página web, así como visualizarlos, para este último paso se elaborará un panel de control (*dashboard*), es una interfaz que permite a los usuarios gestionar y controlar diferentes aspectos de las recetas.

Como se puede ver en la Figura 3.1, el proyecto parte de la extracción de datos de la página web Myrealfood. Estos se extraen, procesan y analizan en una `Jupyter Notebook`, tal y como se detallará en más profundidad en la siguiente sección y capítulos. Una vez los datos han sido obtenidos y procesados, se pasa al desarrollo de un *dashboard* independiente.

Figura 3.1 Diagrama de arquitectura



3.3.1 Web pública (Myrealfood)

La elección de la página web Myrealfood como fuente de datos se debe a su enfoque claro hacia las recetas promovidas por el movimiento Realfooding, el cual como se ha comentado anteriormente está registrado como marca, además de estar gestionada por el propio equipo de Carlos Ríos, promotor principal del movimiento. Los usuarios pueden crear perfiles personales y publicar sus propias recetas, cada una incluye información como ingredientes y sus cantidades, instrucciones detalladas de preparación paso a paso, y a menudo información adicional como el tiempo de preparación. Los usuarios también pueden explorar recetas utilizando búsquedas por tipo de cocina (por ejemplo, vegetariana, vegana, sin gluten).

3.3.2 Notebook de *Web Scraping* y análisis

Para recopilar y procesar datos de Myrealfood de manera eficiente, se ha empleado una notebook, en la que se han utilizado técnicas de *web scraping* mediante el lenguaje de programación Python, el cual se trata de un software de código abierto que fue creado a finales de los años ochenta por Guido van Rossum (Toledo, 2023) .

Python utiliza una sintaxis clara y concisa que se asemeja al lenguaje natural, por ejemplo, la estructura de un bloque de código en Python se define por la indentación en lugar del uso de llaves o palabras clave como ocurre en otros lenguajes. Además, es un lenguaje interpretado, lo que significa que el código se ejecuta línea por línea por un intérprete, sin necesidad de una compilación previa. Esto permite una mayor flexibilidad y rapidez en el desarrollo y prueba de código. Este *software* ha crecido hasta convertirse en uno de los lenguajes de programación más populares y utilizados en el mundo.

El *web scraping* automatiza la extracción de datos de múltiples páginas de la web. Posteriormente, se realiza un análisis inicial para limpiar y estructurar los datos extraídos, una vez realizado esto se almacena la información de forma estructurada en un `DataFrame`.

Finalmente, se proceden a realizar diversas consultas sobre la información recopilada como por ejemplo el tiempo medio de preparación de las recetas en función de la categoría a la que pertenecen.

3.3.3 *Dashboard* de visualización

Una vez que los datos han sido extraídos, procesados y estructurados, se procede a presentar la información obtenida de ellos de manera visual en un *dashboard* interactivo el cual cuenta con una estructura de archivo independiente.

El paquete `Shiny` del lenguaje Python permite crear aplicaciones web interactivas y dinámicas directamente desde Python, sin necesidad de conocimientos avanzados de programación web y sus lenguajes. `Shiny` simplifica el proceso de desarrollo de aplicaciones, ya que proporciona una interfaz de usuario basada en Python y se encarga de la comunicación entre el servidor y el cliente de la aplicación.

La vista se encarga de la presentación visual de los datos y la interacción con el usuario. En el contexto de `Shiny`, un `dashboard` se define en el componente `ui` y muestra los componentes visuales de la interfaz gráfica. La vista incluye elementos visuales como gráficos, tablas, botones, cuadros de texto y cualquier otro componente que sea necesario para mostrar los datos y permitir al usuario interactuar con ellos.

Sin embargo, la vista no incluye las funcionalidades de esos componentes, cuya gestión pertenece al controlador. El controlador actúa como un intermediario entre el modelo y la vista. Es responsable de manejar las interacciones del usuario y de realizar las acciones necesarias para procesar y presentar los datos en la vista. En un `dashboard Shiny`, se implementa con el componente `server`. El controlador recibe las interacciones del usuario desde la vista, como hacer clic en un botón o seleccionar opciones en un menú desplegable. Esto puede incluir filtrar los datos, realizar cálculos o consultas a la base de datos, y cualquier otra operación necesaria para generar la respuesta deseada. Una vez que el controlador ha procesado los datos, los envía de vuelta a la vista para su

visualización. La vista se actualiza dinámicamente, de forma reactiva, con los datos procesados y el usuario puede ver los resultados de sus interacciones.

Este panel de control permite a los usuarios explorar y filtrar recetas según diversos criterios, un ejemplo de ello es la capacidad para introducir una serie de ingredientes y que se muestre como resultado únicamente aquellas recetas que los contienen.

3.4 Herramientas utilizadas

En esta sección se detallarán las soluciones técnicas elegidas para dar solución a la implementación propuesta. En concreto se explicará el lenguaje de programación elegido, así como el entorno de desarrollo y las funcionalidades de las librerías específicas utilizadas tanto para el análisis de datos estático del desarrollador como para la creación de gráficos interactivos y filtros dinámicos a partir del *dashboard*.

3.4.1 Lenguaje de programación

Python es un lenguaje de programación que se ha ganado un lugar destacado en la comunidad tecnológica gracias a su simplicidad en el tipado, pero quizás especialmente gracias a su versatilidad, la cual permite un desarrollo *full stack* que hace que sea mucho más adaptable para el desarrollo de tareas que requieren una integración fluida de múltiples procesos. Esta creciente popularidad ha llevado también a una expansión constante de recursos y mejora de soporte de aquellos recursos ya disponibles, lo cual lo hace una buena opción para su uso en el desarrollo de este proyecto.

Además, cuando se trata de *web scraping*, Python ofrece una gran robustez para interactuar con páginas web, debido a su carácter más generalista que permite la fácil integración con librerías que manejan Javascript (lenguaje habitualmente incrustado en el `HTML` de las páginas web), lo cual lo hace una alternativa más potente respecto a otros lenguajes de análisis de datos como R.

Para la visualización de datos y la creación de *dashboards*, en Python es tremendamente popular la realización de estas interfaces con Dash, pero recientemente se ha popularizado la adaptación de Shiny (originalmente para R) mejorada para Python basada en las funcionalidades reactivas del propio lenguaje, más fluidas que las del lenguaje de programación estadístico.

Respecto al tratamiento de datos, en Python se puede realizar una manipulación y análisis de datos de una manera estructura y eficiente gracias a la biblioteca `pandas`, la cual nos permite construir unas estructuras tabulares eficientes denominadas `DataFrames`. Una de las ventajas clave de estos `DataFrames` proporcionados en `pandas` es su capacidad para realizar operaciones vectorizadas, lo que significa que puedes aplicar funciones y operaciones a columnas enteras de datos de manera eficiente, sin necesidad de bucles explícitos. Esto mejora el rendimiento y la legibilidad del código. Además, a esto hay que añadirle que ofrece múltiples opciones para almacenar datos en diferentes formatos de archivo, como `CSV`, `Excel, json`… entre otros. Esto garantiza que los datos puedan ser almacenados y compartidos de manera consistente entre diferentes sesiones de trabajo.

Por otro lado, los `DataFrames` de `pandas` se integran fácilmente con otras bibliotecas de Python utilizadas en el análisis de datos, como `NumPy` (para operaciones numéricas) y `Matplotlib` o `Seaborn` (para visualización de datos).

Todo esto ha motivado la elección de Python como lenguaje de desarrollo para este proyecto.

3.4.2 Librerías

Para el desarrollo del proyecto se han utilizado librerías complementarias para Python, las cuales se detallan a continuación.

- `Shiny` :

Es una biblioteca de desarrollo de aplicaciones web interactivas para Python que proporciona una forma de construir paneles de control y conectarlos con el código de Python subyacente. Las aplicaciones web interactivas se actualizan automáticamente en función de los cambios en las entradas que realiza el usuario. Es decir, `Shiny` utiliza programación reactiva, donde los cambios en los *inputs* (como deslizadores o botones) desencadenan actualizaciones automáticas en los *outputs* (como gráficos o tablas), las cuales se gestionan a través de la lógica de negocio implementada en Python.

- `Matplotlib` :

Se trata de una biblioteca en Python la cual es una herramienta esencial para la visualización de datos. Permite a los usuarios crear una amplia variedad de gráficos y

visualizaciones, desde simples gráficos de líneas y dispersión hasta complejas visualizaciones en 2D y 3D. Con esta biblioteca es posible personalizar prácticamente todos los aspectos de un gráfico, incluidos los ejes, las etiquetas, los colores y los estilos de línea. Esta capacidad de personalización y su integración con otras bibliotecas populares de Python, como `NumPy` y `pandas`, la hacen ideal para el análisis de datos, la presentación de resultados mediante la creación de visualizaciones interactivas en aplicaciones.

- `Seaborn` :

Es una biblioteca de visualización de datos en Python basada en `Matplotlib`. Proporciona una interfaz de alto nivel para crear gráficos atractivos e informativos. Se integra bien con `pandas`, la biblioteca estándar de Python para análisis de datos, pudiendo aceptar directamente objetos de tipo `DataFrame` como entrada para sus funciones de trazado, lo que facilita la visualización de datos tabulares.

- `Pandas` :

Es una biblioteca que proporciona estructuras y herramientas para el análisis de datos de manera eficiente. Es una de las herramientas más utilizadas en el mundo del análisis de datos y la ciencia de datos. La estructura central en `pandas` es el `DataFrame`. Además de esto, `pandas` también proporciona la estructura de datos `series`, que es una matriz unidimensional de datos etiquetados. Son utilizadas para representar una sola columna o fila de datos en un `DataFrame.` Finalmente, aunque `pandas` no es una biblioteca de visualización de datos en sí misma, se integra bien con bibliotecas de visualización mencionadas anteriormente.

3.4.3 Entorno de desarrollo

Un entorno de desarrollo (ED) es un conjunto de herramientas y funcionalidades diseñadas para facilitar la escritura, depuración y ejecución de programas o aplicaciones de software. Proporciona un entorno integrado que incluye un editor de código, una consola interactiva, herramientas de depuración, administración de proyectos y otras características que mejoran la productividad del desarrollador.

En este proyecto se utilizan dos entornos de desarrollo distintos, uno para la fase de extracción y análisis de datos y otro para el desarrollo del *dashboard*.

Uno de los editores es Visual Studio Code (VSCode), que es un editor de código fuente desarrollado por Microsoft, popular entre los desarrolladores por su extensibilidad, y muy popular actualmente tanto en procesos desarrollo software como de análisis de datos ya que es *open source*. Aunque no es un ED completo por sí mismo, sus extensiones le permiten funcionar como uno, haciéndolo totalmente adaptable a diferentes lenguajes de programación y tipos de proyectos. VSCode cuenta con una extensa colección de extensiones que pueden añadirse fácilmente, destacando en este proyecto la extensión de Python de Microsoft que proporciona soporte avanzado, incluyendo IntelliSense (autocompletado de código), *linting*, depuración y herramientas de *testing*. Su interfaz de usuario es moderna y diseñada para facilitar la navegación entre archivos y proyectos, además de contar con un terminal integrado para la ejecución de comandos sin salir del editor, lo cual justifique también su creciente popularidad. Además, VSCode es multiplataforma, disponible para Windows, macOS y Linux, lo que lo hace accesible para desarrolladores en diferentes sistemas operativos.

VSCode ha sido el editor utilizado para la implementación del *dashboard* en Python utilizando Shiny, ya que facilita el lanzamiento del proyecto en un servidor local (localhost). Mediante la extensión de Shiny para VSCode se proporciona además una integración visual para previsualizar en el ED el desarrollo de estas aplicaciones.

Además, para el desarrollo de este proyecto se ha utilizado una `Jupyter Notebook.` Una `Jupyter Notebook` es un entorno de desarrollo interactivo que permite a los usuarios escribir y ejecutar código, ver los resultados y crear contenido explicativo en un mismo documento. Debido a estas características, las `Jupyter Notebook` son útiles para el desarrollo de software y el análisis de datos. Además, admite extensiones que permiten ampliar su funcionalidad y personalizar su entorno según las necesidades específicas del usuario.

A continuación, se describen algunas de las características principales de una `Jupyter Notebook` por las cuales se ha elegido como entorno de desarrollo frente a otras alternativas como podría haber sido `VSCode`.

- **Documentación Integrada**:

Permite combinar código con texto, imágenes y ecuaciones LaTeX, lo que facilita la creación de informes.

- **Entornos aislados**:

Jupyter Notebook es compatible con entornos virtuales (por ejemplo, conda) que permiten aislar las dependencias de un proyecto específico. Esto ayuda a evitar conflictos entre diferentes versiones de bibliotecas instaladas en el sistema.

- **Control de versiones de bibliotecas**:

Puedes especificar las versiones de las bibliotecas que deseas utilizar en cada cuaderno de Jupyter. Esto asegura que el código funcione de manera consistente, incluso si se comparte o ejecuta en otro entorno.

4 Extracción de la información

Como se comentaba en capítulos anteriores, para realizar un proceso de extracción de información mediante *web scraping* en primer lugar, se debe identificar la página web de la que se quiere obtener información, de la cual se analizará su código fuente. Como ya hemos dicho, en este caso será https://new.myrealfood.app/ y las páginas a las que se puede acceder a través de esta que contengan información relevante.

Como se ha detallado en el capítulo anterior, para realizar este trabajo se ha seleccionado la librería de Python llamada `BeautifulSoup`, la cual permite analizar documentos `HTML` con un enfoque orientado a objetos. Esto agiliza el proceso de extracción, ya que la librería se encarga de limpiar y normalizar el código `HTML` en una estructura jerárquica en base a las anidaciones de las etiquetas, lo que facilita el trabajo.

Para poder acceder a la información que se quiere analizar, es necesario extraerla de la web, las cuales están identificadas por URLs, (del inglés Uniform Resource Locator). Una URL es la dirección que se utiliza para acceder a recursos específicos en Internet. En otras palabras, es la dirección que habitualmente se introduce en la barra de direcciones de un navegador web para visitar una página web o recurso en particular. Procesar esta información manualmente resultaría imposible, con lo cual se requiere de algún proceso técnico que automatice la preparación de estos datos y pueda ser tratado desde Python. Con este fin, se realiza una petición a la página web seleccionada usando el protocolo `HTTP` (de las siglas en inglés de `Hypertext Transfer Protocol`), lo cual permitirá extraer los datos. Una petición `HTTP` es un mensaje que un cliente (generalmente un navegador web) envía a un servidor web para solicitar la recuperación de un recurso. Este recurso puede ser un archivo, una imagen, o cualquier otro tipo de contenido que esté disponible en el servidor. Sin embargo, para realizar el proceso de *web scraping* no es el navegador quien debe realizar esta petición, si no que será necesario tener la información disponible desde el código.

Las peticiones `HTTP` se pueden realizar en Python sin requerir un navegador utilizando la librería `Requests`, que envía dicha petición a la página seleccionada, la cual responde devolviendo el contenido `HTML`.

Una vez que se dispone del contenido, este se analiza utilizando la biblioteca `BeautifulSoup` en combinación con las librerías estándar de Python, lo cual permitirá la preparación de los datos para su posterior análisis. Para utilizar `BeautifulSoup` es necesario conocer los elementos que resultan de interés, a partir del análisis del interfaz previamente realizado. Para esta tarea se utilizan las herramientas para desarrolladores del navegador, que permiten identificar la equivalencia entre los componentes gráficos que se visualizan en el navegador y su código fuente.

HTML5 es la última versión del lenguaje con capacidades como por ejemplo reproducir contenido multimedia sin complementos adicionales, o la posibilidad de trabajar sin conexión. Basado en etiquetas, HTML5 utiliza una serie de elementos que rodean y definen el contenido. Cada etiqueta tiene una función específica, como `<h1>` para encabezados de primer nivel, `<p>` para párrafos, `<a>` para enlaces y `<img>` para imágenes. Las etiquetas suelen venir en pares de apertura y cierre, como `<div>` y `</div>`, aunque algunas, como `<br>` para saltos de línea, son auto-cerradas. Estas etiquetas organizan y determinan cómo se mostrará el contenido en un navegador, proporcionando la estructura básica y permitiendo la inclusión de textos, imágenes, enlaces y otros recursos multimedia. Además, se pueden añadir atributos a las etiquetas, como `class`, `id` o `style`, para mejorar la funcionalidad y el estilo del contenido. Cada una de estas etiquetas es comúnmente particularizada a una clase CSS (Cascading Style Sheets), que hace que su aspecto en la interfaz gráfica cambie, lo cual permite la identificación de los componentes de interés. CSS permite definir estilos para elementos específicos, como colores, tamaños, márgenes y fuentes, proporcionando una apariencia uniforme y estéticamente agradable en todo el sitio web. Así, HTML y CSS trabajan juntos: HTML estructura el contenido y CSS lo estiliza, creando una experiencia de usuario coherente y visualmente atractiva.

En general, las etiquetas principales de HTML a considerar son las siguientes:

- Documento: Se inicia con la declaración `<!DOCTYPE html>` que indica al navegador que se trata de un documento `HTML5`.
- Elemento `HTML`: Todo el contenido de la página web está comprendido dentro de un elemento `<html>`, que a su vez contiene dos secciones principales: `<head>` y `<body>.`
- Elemento `head`: Contiene elementos descriptivos de la cabecera como el título de la página (`<title>`) y la inserción de dependencias.

- Elemento `body`: Aquí se encuentra el contenido visible de la página. Incluye encabezados, navegación, contenido principal y pie de página.
- Contenido principal (`<main>`): Abarca el contenido principal de la página.
- Encabezados (`<h1>, <h2>, <h3>,` etc): Definen títulos y subtítulos.
- Párrafos (`<p>`): Contienen texto.
- Enlaces (`<a>`): Permiten la navegación a otras páginas web o dentro de la misma página.
- Imágenes (`<img>`): Muestra imágenes en la página.
- Videos (`<video>`): Reproduce contenido de video.

Cada una de estas etiquetas es comúnmente particularizada a una clase CSS, que hace que su aspecto en la interfaz gráfica cambie, lo cual permite la identificación de los componentes de interés.

El proceso de extracción consta de dos partes. En primer lugar, se analizará la interfaz en la que se listan las recetas, y luego se utilizarán las URL a las páginas individuales de recetas detectadas desde esta interfaz para extraer la información particular.

Figura 4.1 Inicio código HTML


```
'<!DOCTYPE html><html lang="es"><head><meta charSet="utf-8"/><meta name="viewport" content="width=device-width"/><title>Myrealf
ood: 0 Resultados para recetas</title><meta name="next-head-count" content="3"/><meta name="description" content="La app para m
ejorar tu estilo de vida saludable. Encuentra recetas, planes de alimentación y mucho más."/><meta name="title" content="MyReal
Food: recetas y planes nutricionales"/><meta name="apple-itunes-app" content="app-id=1458031749"/><meta name="google-play-app"
content="app-id=es.myrealfood.myrealfood"/><link rel="icon" href="/favicon.png"/><link rel="preload" href="/_next/static/media/
4b4ebe20759bdbf2-s.p.ttf" as="font" type="font/ttf" crossorigin="anonymous" data-next-font="size-adjust"/><link rel="preload" h
ref="/_next/static/media/a52d0bf095c248da-s.p.ttf" as="font" type="font/ttf" crossorigin="anonymous" data-next-font="size-adjus
t"/><link rel="preload" href="/_next/static/css/c36a2820a96d555d.css" as="style"/><link rel="stylesheet" href="/_next/static/cs
s/c36a2820a96d555d.css" data-n-g=""/><link rel="preload" href="/_next/static/css/c74192b1c20e3730.css" as="style"/><link rel="s
tylesheet" href="/_next/static/css/c74192b1c20e3730.css" data-n-p=""/><noscript data-n-css=""></noscript><script defer="" nomod
ule="" src="/_next/static/chunks/polyfills-78c92fac7aa8fdd8.js"></script><script src="/_next/static/chunks/webpack-a707e99c6936
1791.js" defer=""></script><script src="/_next/static/chunks/framework-56eb74ff06128874.js" defer=""></script><script src="/_ne
xt/static/chunks/main-0e1af0e19b31c7e7.js" defer=""></script><script src="/_next/static/chunks/pages/_app-efb01a94c2299acc.js"
defer=""></script><script src="/_next/static/chunks/2141-7dfcbf3315b5d591.js" defer=""></script><script src="/_next/static/chun
ks/939-9583c36c893822ae.js" defer=""></script><script src="/_next/static/chunks/2362-98ec84e805d6114c.js" defer=""></script><sc
ript src="/_next/static/chunks/pages/%5Blang%5D/recipes-e2e1e581827f34b7.js" defer=""></script><script src="/_next/static/RLQ7L
ey0xPlHL0mmQoGNq/_buildManifest.js" defer=""></script><script src="/_next/static/RLQ7Ley0xPlHL0mmQoGNq/_ssgManifest.js" defer
=""></script></head><div id="__next"><main class="__className_95cf1c"><div style="position:relative"><div class="banner_banner_
_CDOgt"><div class="banner_mobileText__VPGAd"><img alt="BlackFriday Label" loading="lazy" width="400" height="400" decoding="as
```


Comenzando con el código fuente que nos atañe en este proyecto, veremos en primer lugar la estructura de la página que contiene un recetario divido en grupos en función de la comida a la que se asocia la receta respecto a su franja horaria (desayuno, comida, cena). En este caso, el objetivo es extraer todas las recetas que se muestran en las

secciones desayuno, comida y cena de la página web. Para ello, como se puede ver en la Figura 4.1 y la Figura 4.2, se procede a inspeccionar el código `HTML` de la página y a observar en que elementos se encuentra la información que queremos obtener.

El primer paso es obtener las URL de cada receta individual, para así poder acceder a todas ellas de forma directa. Puesto que para obtener la información completa de cada receta es necesario mirarlas de forma individual, esto también afecta para obtener la dirección URL de cada una. En el momento de la extracción de información la forma de cargar más recetas era pulsando sobre el botón “cargar más”, recientemente la página web ha sufrido una actualización y ahora el formato es paginado. Puesto que para obtener todas las recetas deseadas era necesario hacer *click* sobre el botón esté proceso se automatizó en el siguiente código.

Esto significa simular darle al botón siguiente. Es necesario controlar la franja de tiempo con la que se hace esto, ya que, en caso de ser demasiado rápida, la petición devolverá una respuesta negativa con información incorrecta. Para ello, deben importarse las librerías necesarias, en este caso `Selenium` para `webdriver`, `times` para establecer pausas y `BeautifulSoup` para analizar el contenido `HTML`.

Figura 4.2 Final código HTML


```
v0/b/realfooding-app.appspot.com/o/ocr%2Fproducts%2F8436039435421%2Fcover%2F1637332493634%2F1637332493634.jpg?alt=media\\u0026t
oken=d9deae92-1e39-40b8-b9c1-121e32c03d82"},"name":{"_":"Queso fresco batido 0%","es":"Queso fresco batido 0%"},"objectID":"843
6039435421"},{"score":9.9,"color":"#19C692","images":{"ingredients":"https://cdn3.myrealfood.app/s3-myrealfood/ocr/products/840
2001012310/ingredients_1691246998123","nutritional":"https://cdn3.myrealfood.app/s3-myrealfood/ocr/products/8402001012310/nutri
tional_1691246850764","product":"https://cdn3.myrealfood.app/s3-myrealfood/ocr/products/8402001012310/cover_1691246800775_BaGYd
Dbx1RTyK0nESUUIn9necAu2.jpg"},"name":{"_":"Crema de calabacin","es":"Crema de calabacin"},"objectID":"8402001012310"},{"score":
9,"color":"#19C692","images":{"ingredients":"https://cdn3.myrealfood.app/s3-myrealfood/ocr/products/8480000864970/ingredients/1
665392568838.jpg","nutritional":"https://cdn3.myrealfood.app/s3-myrealfood/ocr/products/8480000864970/nutritional/166539249234
4.jpg","product":"https://cdn3.myrealfood.app/s3-myrealfood/products/8480000864970/Aw3wWqeFz4WO3v4zExfxgJFwZqp2_image.jpg/16939
99344523.jpg"},"name":{"_":"Caldo de pollo","es":"Caldo de pollo"},"objectID":"8480000864970"},{"score":9.9,"color":"#19C69
2","images":{"ingredients":"https://firebasestorage.googleapis.com/v0/b/realfooding-app.appspot.com/o/products%2F8032804430822%
2Fimage_ingredients_url.jpeg?alt=media\\u0026token=f8043c2b-c6a4-4151-a5d8-40f4baa0556b","nutritional":"https://firebasestorag
e.googleapis.com/v0/b/realfooding-app.appspot.com/o/products%2F8032804430822%2Fimage_nutrition_url.jpeg?alt=media\\u0026token=a
99f9314-d025-4068-8e1c-f22117bfbf39","product":"https://firebasestorage.googleapis.com/v0/b/realfooding-app.appspot.com/o/produ
cts%2F8032804430822%2Fimage_url.jpeg?alt=media\\u0026token=02e4051f-c397-4ec3-b3ed-0bcdb86e754f"},"name":{"_":"Fusilli lentejas
rojas","es":"Fusilli lentejas rojas"},"objectID":"8032804430822"},{"score":10,"color":"#19C692","images":{"ingredients":"http
s://firebasestorage.googleapis.com/v0/b/realfooding-app.appspot.com/o/products%2F5600499526040%2Fimage_ingredients_url.jpeg?alt
=media\\u0026token=b94dd502-0c15-4dc5-9a86-e09a22edce70","nutritional":"https://firebasestorage.googleapis.com/v0/b/realfooding
-app.appspot.com/o/products%2F5600499526040%2Fimage_nutrition_url.jpeg?alt=media\\u0026token=701aeef6-c365-4209-839a-0930bcd688
4d","product":"https://firebasestorage.googleapis.com/v0/b/realfooding-app.appspot.com/o/products%2F5600499526040%2Fimage_url.j
peg?alt=media\\u0026token=1ece5f02-84b0-479c-906a-231053cddb59"},"name":{"_":"Tortitas de maíz Krispees","es":"Tortitas de maíz
Krispees"},"objectID":"5600499526040"}]}],"recommended":[{"en":"Healthy","it":"Sano","fr":"En bonne santé","es":"Saludabl
e","_":"Saludable"},{"en":"Ice cream","it":"Gelato","fr":"Glaces","es":"Helado","_":"Helado"},{"en":"Oatmeal pancakes","it":"Fr
ittelle di farina d\'avena","fr":"Crêpes à l\'avoine","es":"Tortitas de avena","_":"Tortitas de avena"},{"en":"Salad to go","i
t":"Insalata da portare via","fr":"Salade à emporter","es":"Ensalada para llevar","_":"Ensalada para llevar"},{"en":"Gazpach
o","it":"Gazpacho","fr":"Gazpacho","es":"Gazpacho","_":"Gazpacho"},{"en":"Vegetables","it":"Verdure","fr":"Légumes","es":"Verdu
ras","_":"Verduras"}]}},"__N_SSG":true},"page":"/[lang]/recipes","query":{"lang":"es"},"buildId":"RLQ7Ley0xPlHL0mmQoGNq","isFal
lback":false,"gsp":true,"scriptLoader":[{"async":true,"src":"https://pagead2.googlesyndication.com/pagead/js/adsbygoogle.js?cli
ent=ca-pub-5243124893640381","strategy":"lazyOnload","crossOrigin":"anonymous"}]}</script></html></body></html>'
```

```
options = webdriver.ChromeOptions()
options.add_argument('--headless')
driver = webdriver.Chrome(options=options)
```

En el código mostrado se muestra esta obtención. Para ello se instancia un objeto `WebDriver` (controlador de navegador) en este caso para Chrome al que se le pasan las opciones de configuración. En este caso se configura la opción para ejecutarlo en modo `headless` (sin interfaz gráfica) de esta forma el navegador funciona en segundo plano sin abrir una ventana visible en el escritorio del usuario. En este caso se está simulando que la petición se hace desde un navegador con el que podemos hacer *click* para obtener más recetas.

```
driver.get('https://www.myrealfood.app/recipes/category/mnJxw7vP6760YBwuMOYt')
```

Se procede a cargar la dirección URL de la página web en el navegador.

```
urls_desayuno = []
click_counter = 0
```

Se inicializan las variables `urls_desayuno` para almacenar las URLs correspondientes a esa categoría y `click_counter` para contar el número de veces que se hace *click* en el botón “cargar más”.

```
while len(urls_desayuno) < 5100:
    try:
boton_cargar_mas=WebDriverWait(driver,3).until(EC.element_to_be_clickable((By
.XPATH, "//button[contains(., 'Cargar más')]")))
driver.execute_script("arguments[0].click();", boton_cargar_mas)
time.sleep(2)
click_counter += 1
        contenido = driver.page_source
        soup = BeautifulSoup(contenido, 'html.parser')
        patron = r'recipes/[\w\d]+\d+[\w\d]*'
        url_receta_desayuno = re.findall(patron, contenido)
        for url in url_receta_desayuno:
            urls_desayuno.append(url)
    except:
        break
```

Finalmente se quieren extraer 5000 recetas de cada tipo, para ello se inicia un bucle de tipo `while` que se ejecutará mientras el número de URLs recolectadas sea menor o igual a 5100 y dentro del bucle:

- Se intenta localizar el botón "cargar más" dejando un margen de espera de tres segundos para que este sea *clickable*.
- Si se encuentra el botón se hace *click* sobre él.
- Se espera dos segundos para que la página cargue los nuevos elementos.
- Se incrementa el contador de *clicks*.
- Se actualiza el contenido de la página y se analiza con `BeautifulSoup` para extraer las URLs utilizando expresiones regulares, en este caso se busca mediante `re.findall()` un texto que comience por la palabra "`recipes`" y esté seguida por caracteres alfanuméricos. Estas URLs se agregan a la lista `urls_desayuno`
- En caso de producirse algún error se cierra el bucle.

A partir de este código hemos obtenido la parte variable de cada URL para las recetas seleccionadas. Esta parte está precedida por un texto fijo que en este caso es "https://new.myrealfood.app/".

Una vez que disponemos de las direcciones a cada receta analizamos el contenido HTML, que se puede ver en Figura 4.4 y la Figura 4.3, individual de cada una de ellas.

Figura 4.4 Inicio código HTML receta

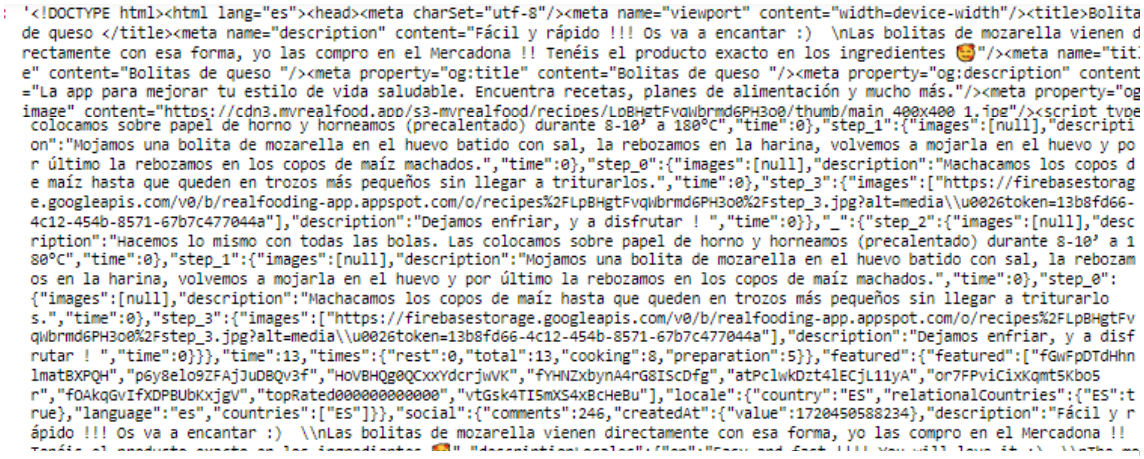
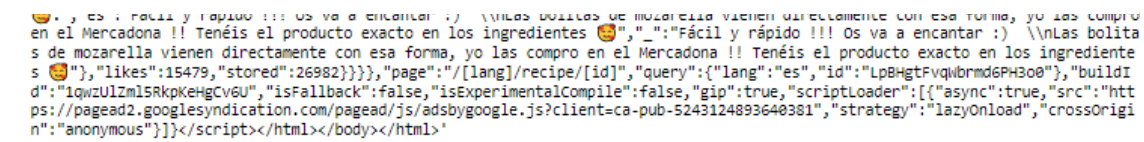


Figura 4.3 Final código HTML receta



El primer paso es obtener el nombre de cada una de las recetas, para ello empleamos el siguiente código. En primer lugar, definimos una función para una mayor eficiencia:

```
def process_url(url):
    try:
        r = requests.get(url)
        r.raise_for_status()

        contenido = r.text
        soup = BeautifulSoup(contenido, 'html.parser')
        nombres= soup.find_all(class_='recipeInfo_basic__GpfDG')
        nombres_recetas = []
        for nombre in nombres:
            nombre_receta=nombre.find('h1').get_text(strip=True)
            nombre_receta = unidecode(nombre_receta).lower()
            nombres_recetas.append(nombre_receta)
        return nombres_recetas
    except requests.RequestException as e:
        print(f"Error al acceder a la URL {url}: {e}")
        return ["URL no accesible"]
```

La función `process_url` toma como argumento de entrada una URL, accede a la página web correspondiente, extrae el nombre de las recetas y los devuelve en una lista. Para ello se utiliza la biblioteca `requests` que permite realizar una solicitud `HTTP` a la URL proporcionada. Se verifica si la respuesta ha sido exitosa y en caso de no serlo se genera una excepción. En caso de solicitud exitosa se obtiene el contenido de la página web y mediante `BeautifulSoup` se analiza.

Se indica que los nombre se encuentran en el elemento `HTML` "`recipeInfo_basic_GpfDG`" y sobre estos elementos extrae el texto del elemento `<h1>.` Posteriormente normalizamos el texto para convertir caracteres especiales como puede ser la "ñ" a su representación ASCII, que es un estándar de codificación de caracteres que define cómo los caracteres alfanuméricos, símbolos y otros especiales deben ser representados, y también pasamos el texto a minúsculas. Finalmente agregamos el texto a la lista "`nombres_recetas`".

Ahora procedemos a aplicar la anterior función a todas las direcciones web, debido al gran número de URLs de las que disponemos será necesario implementar una ejecución de múltiples direcciones en paralelo.

```
url_base = "https://new.myrealfood.app/"
```

Establecemos la base de la URL que posteriormente completaremos con la información obtenida en el primer paso.

```
with ThreadPoolExecutor(max_workers=20) as executor:
    resultados = list(executor.map(process_url, [url_base + url for url in
urls_desayuno]))
desayuno_nombre = [nombre for sublist in resultados for nombre in sublist]
```

Se utiliza `ThreadPoolExecutor,` que pertenece al paquete `concurrent.futures` para crear un conjunto de hasta veinte hilos de ejecución. Luego se utiliza `executor.map()` para aplicar la función `proces_url()` a cada URL de la lista en paralelo. Los resultados se almacenan en una lista llamada "`desayuno_nombre`".

El tercer paso es obtener los ingredientes necesarios para elaborar cada una de las recetas, para ello creamos una función que llamaremos "`obtener_ingredientes`".

```
def obtener_ingredientes(url):
    url_base = "https://new.myrealfood.app/"
    url_completa= url_base+url
    r = requests.get(url_completa)
    contenido = r.text
    soup = BeautifulSoup(contenido, 'html.parser')
      ingredientes=soup.find_all(class_='recipeInfo_ingredient__SpoEM')
    ingredientes_unicos = set()
    for ingrediente in ingredientes:
        nombre = ingrediente.find('h6').get_text(strip=True)
cantidad=ingrediente.find('div',class_="recipeInfo_subtitle__ID_hK").get_text
(strip=True)
        ingrediente_completo = f"{nombre} {cantidad}"
        ingredientes_unicos.add(ingrediente_completo)
    return list(ingredientes_unicos)
```

Dicha función toma nuevamente como argumento una dirección web.

En primer lugar, se establece la base de la URL para posteriormente completarla con la parte variable personal de cada receta. Se realiza la solicitud `HTTP` utilizando `requests.get()` y de ese resultado obtenemos el contenido, nuevamente mediante `BeautifulSoup`.

El siguiente paso es encontrar todos los elementos `HTML` de clase "`recipeInfo_ingredient_SpoEM`", inicializamos un conjunto `ingredientes_unicos` para almacenar los resultados de forma única y así evitar duplicados. Se procede a iterar sobre cada resultado encontrado y se extrae el nombre que se encuentra en el elemento `<h6>` y la cantidad disponible en el elemento `<div>` con la clase específica "`recipeInfo_subtitle__ID_hK`" y se construye una cadena con toda la información anterior de forma conjunta.

Con el objetivo de obtener una información más legible y con más provecho para el análisis, separaremos la cantidad del nombre del ingrediente contenido en la cadena de texto anterior, para ello utilizaremos dos expresiones regulares.

Para conseguir únicamente el nombre del ingrediente:

```
patron = r'([^0-9]+)'
nombre_ingredientes_receta_desayuno = []

for lista in ingredientes_desayuno:
    nombres_ingrediente = []
    for ingrediente in lista:
        nombre_ingrediente = re.search(patron, ingrediente)
        if nombre_ingrediente:
            nombres_ingrediente.append(nombre_ingrediente.group(1).strip())
    nombre_ingredientes_receta_desayuno.append(nombres_ingrediente)
```

La expresión regular, es una secuencia de caracteres que define un patrón de búsqueda, `r'([^0-9]+)'`, se corresponde a los ingredientes puesto que nos permite extraer todo aquel texto que no contiene números y, por último, para extraer las cantidades de cada uno de los anteriores ingredientes aplicaremos lo siguiente:

```
patron = r'(\d+.*?)$'
cantidad_ingredientes_receta_desayuno = []

for lista in ingredientes_desayuno:
    cantidad_ingredientes_receta = []
    for ingrediente in lista:
        match = re.search(patron, ingrediente)
        if match:
            cantidad_ingredientes_receta.append(match.group(1).strip())

cantidad_ingredientes_receta_desayuno.append(cantidad_ingredientes_receta)
```

En este caso el patrón `r'(\d+.*?)$'` nos permite buscar uno o más dígitos seguidos por cualquier carácter hasta el final de la línea.

Con el objetivo de seguir recopilando información se procede a extraer los pasos a seguir para elaborar cada una de las recetas y el autor que las ha publicado. Para ello construimos dos funciones.

```
def obtener_pasos(url):
    r = requests.get(url)
    contenido = r.text
    soup = BeautifulSoup(contenido, 'html.parser')
    pasos = soup.find_all(class_='recipeSteps_step__piG9x')
    pasos_completos = []
    for paso in pasos:
numero=paso.find(class_='recipeSteps_stepNumber__igpaz').get_text(strip=True)

explicacion=paso.find('p',class_='typography_typography__2U0Nqtypography_body
1__UfbFitypography_left__zd4Id').get_text(strip=True)
        paso_completo = f"Paso {numero}: {explicacion}"
        pasos_completos.append(paso_completo)
    return pasos_completos
```

En primer lugar, realizamos una solicitud `GET` a la URL especificada. Para obtener su contenido y utilizamos `BeautifulSoup` para analizar su contenido. Le indicamos que encuentre todos los elementos de la clase `'recipeSteps_step__piG9x'` y sobre cada uno de ellos extraemos el elemento cuya clase es `'recipeSteps_stepNumber__igpaz'` que contiene el número del paso y la clase

'typography_typography__2U0Nqtypography_body1__UfbFitypography_left__zd4Id' que contiene las instrucciones. Agrupamos toda esta información en una cadena de texto llamada pasos_completo.

Construimos otra función para extraer los autores:

```
def obtener_autor(url):
    r = requests.get(url)
    contenido = r.text
    soup = BeautifulSoup(contenido, 'html.parser')
    autores = soup.find_all(class_='recipeInfo_author__gpTf8')
    autores_completos = []
    for autor in autores:
nombre_elemento=autor.find('h6',class_='typography_typography__2U0Nq')
        nombre = nombre_elemento.get_text(strip=True)
        autores_completos.append(nombre)
 return autores_completos
```

Realizamos una solicitud GET a la URL especificada para obtener su contenido y utilizamos BeautifulSoup para analizar su contenido. Le indicamos que encuentre todos los elementos de la clase 'recipeInfo_author__gpTf8' sobre cada uno de ellos extraemos el encabezado <h6> y el elemento cuya clase es 'typography_typography__2U0Nq' que contiene el nombre del autor. Extraemos la cadena de texto y la almacenados en la lista autores_completos.

Para terminar de completar el DataFrame, extraemos la información relativa al tiempo de elaboración, las calorías por receta, los puntos y la valoración dada por los usuarios que oscila entre cero y cinco. Esta información la extraeremos del contenido HTML mediante expresiones regulares que incluiremos en una función.

```
def info(url):
    try:
        r = requests.get(url)
        contenido = r.text
        regex_total_time = r'"totalTime":"([a-zA-Z]+\d+[a-zA-Z]+)"'
        regex_calories = r'"calories":"(\d+(\.\d+)?)'
        regex_rating_value = r'"ratingValue":(\d+\.*\d+)'
        regex_rating_count = r'"ratingCount":(\d+\.*\d+)}'
        total_time_match = re.search(regex_total_time, contenido)
```

```
        calories_match = re.search(regex_calories, contenido)
       rating_value_match=re.search(regex_rating_value,contenido)
        rating_count_match=re.search(regex_rating_count,contenido)
tiempo_total=total_time_match.group(1)
if total_time_match else np.nan
        calorias = calories_match.group(1)
if calories_match else np.nan
        valoracion = rating_value_match.group(1)
 if rating_value_match else np.nan
        puntos= rating_count_match.group(1)
if rating_count_match else np.nan
        return tiempo_total, calorias, valoracion, puntos
    except: return np.nan, np.nan, np.nan, np.nan
```

La función `info` toma como argumento una dirección web y realiza una petición `GET` a la misma y extraemos el contenido en formato texto, en el buscamos la información anterior mediante cuatro expresiones regulares:

- Tiempo total: `r'"totalTime":"([a-zA-Z]+\d+[a-zA-Z]+)"'`. Esta expresión busca aquella cadena de texto que comienza por "`totalTime:`" seguida de una o más letras mayúsculas o minúsculas, uno o más dígitos y finalmente una o más letras tanto mayúsculas como minúscula.
- Calorías: `r'"calories":"(\d+(\.\d+)?)'`. Busca uno o más dígitos seguidos de un punto y uno o más dígitos, es decir, extrae texto como por ejemplo 123.56 .
- Valoración: `r'"ratingValue":(\d+\.*\d+)'`. Extrae cadenas que comienzan por "`ratingValue:`" además contienen uno o más dígitos, tiene o no un punto que es seguido de uno o más dígitos.
- Puntuación: `r'"ratingCount":(\d+\.*\d+)}'`. Extrae cadenas que comienzan por "`ratingCount:`" y que además contienen uno o más dígitos, y tiene o no un punto que es seguido de uno o más dígitos y finaliza con el carácter }.

Por otro lado, cuando encuentra las correspondientes coincidencias extrae la información y en caso contrario asigna un valor de tipo nulo (`NaN`). Además, si se produce algún error durante la búsqueda también se asigna también este valor.

Otro aspecto interesante son las palabras clave con las que se etiqueta a cada receta, para extraerlas del código `HTML` emplearemos el siguiente código:

Primero creamos la función “buscar_keywords”.

```
def buscar_keywords(url):
    r = requests.get(url)
    contenido = r.text
    regex_keywords = r'"keywords":"(.*?)"'
    palabras = re.search(regex_keywords, contenido)
    palabras_clave = palabras.group(1) if palabras else np.nan
    return palabras_clave
```

La función toma una URL como entrada y devuelve las palabras clave encontradas en el contenido de la página web correspondiente a una determinada URL. Dichas palabras se extraen con la expresión regular “"keywords":"(.*?)"” que busca los términos que comienzan por "keywords":" y luego captura cualquier contenido (incluso vacío) hasta que encuentre las comillas dobles (“) y con re.search(), que pertenece a la librería re se busca la primera coincidencia en el contenido HTML.

Posteriormente aplicamos la anterior función a todas las URL de la siguiente forma:

```
urls = df["url"]
url_base = "https://new.myrealfood.app/"
with ThreadPoolExecutor(max_workers=20) as executor:
    claves = list(executor.map(buscar_keywords, [url_base + url for url in
urls])
```

4.1 Estructura de la información obtenida

En el apartado anterior hemos procedido a extraer toda la información relevante de la página web, ahora es el momento de almacenar todo el contenido en un DataFrame sobre el cual realizaremos las consultas.

Como se puede ver en la Figura 3.2 el conjunto final de datos consta de la URL de cada receta, el tipo de comida, el nombre, los ingredientes, las cantidades correspondientes a cada ingrediente, los pasos para elaborar, el autor que la ha publicado, el tiempo, las calorías, las valoraciones y puntos recibidos y las palabras clave.

Figura 3.2 Estructura del `DataFrame`

url	tipo_comida	Nombre_receta	ingredientes	cantidades	pasos_receta	autor	tiempo	kcal	valoracion	puntos	keywords
recipe/IepYxJholbTpQgUQFF20	desayuno	carrot cake fit	Queso crema, Aceite de oliva virgen extra, Leche, Frutos secos, Huevo de gallina, Dátiles, Harina integral de trigo, Zanahoria, Canela en polvo, Levadura fresca	125 gramos, 15 gramos (aprox. 1 cucharada), 200 gramos, 60 gramos (aprox. 12 unidades), 20 gramos (aprox. 2 unidades), 110 gramos (aprox. 11 unidades), 100 gramos, 80 gramos (aprox. 2 unidades), 10 gramos (aprox. 1 cucharadita), 15 gramos (aprox. 1 cucharadita)	Paso 1: Rallamos y trituramos las dos zanahorias. Trituramos 8 nueces y las juntamos con la zanahoria., Paso 2: Preparamos la pasta de dátiles deshuesando 7 dátiles y batimos junto con 125 ml de leche, hasta lograr una mezcla homogénea., Paso 3: En un bol mezclamos las nueces trituradas junto con las zanahorias, la pasta de dátiles, dos huevos, la harina, una cucharada de AOVE, una cucharadita de canela y otra de levadura. Removemos hasta que se integren bien todos los ingredientes., Paso 4: Vertemos la mezcla en un molde apto para horno y horneamos durante 25-30 minutos a 180°C (con el horno previamente calentado)., Paso 5: Mientras se hace el bizcocho, preparamos la cobertura con 75ml de leche + 4 dátiles + 150g de crema de queso (medio paquete). Batimos todos los ingredientes y lo metemos en el frigorífico para que se solidifique., Paso 6: Cuando el bizcocho se enfríe, untamos la cobertura, añadimos unas 5 nueces mas por encima y espolvoreamos un poco de canela al gusto. Dejamos enfriar y a disfrutar !	paulacampoamor	0.0	1575.52	4.7	33875.0	Consumo ocasional, Snack, Postres, Desayuno

5 Análisis de la información

Una vez recopilada toda la información posible procedemos a realizar los siguientes análisis sobre ella. Por un lado, realizaremos diversas consultas sobre los datos almacenados cuyos resultados veremos a continuación. Por otro lado, plantaremos respuesta en el *dashboard* a preguntas como los tres ingredientes más usados en cada categoría, duración media de la elaboración para las tres comidas principales del día o calorías medias aportadas por las recetas de las categorías principales.

5.1.1 Veinte ingredientes más y menos utilizados en las recetas

Conocer los ingredientes que más se utilizan en las recetas publicadas por los internautas nos ayuda a conocer si dichas preparaciones cumplen realmente con los principios con los que predica el movimiento Realfooding.

En primer lugar, podemos ver en la Figura 5.1 los veinte ingredientes que más se han utilizado en todas las recetas recopiladas. Podemos observar que los ingredientes más frecuentes pertenecen al grupo de comida real y buenos procesados. En el caso contrario, los veinte ingredientes menos utilizados los podemos ver en la Figura 5. y se corresponden con alimentos como los cereales o azúcares, los cuales pertenecen al grupo de comida

real en el caso de cereales integrales y el resto se clasificarían como ultraprocesados (azúcar blanco) o buenos procesados.

Figura 5.1 Veinte ingredientes más comunes

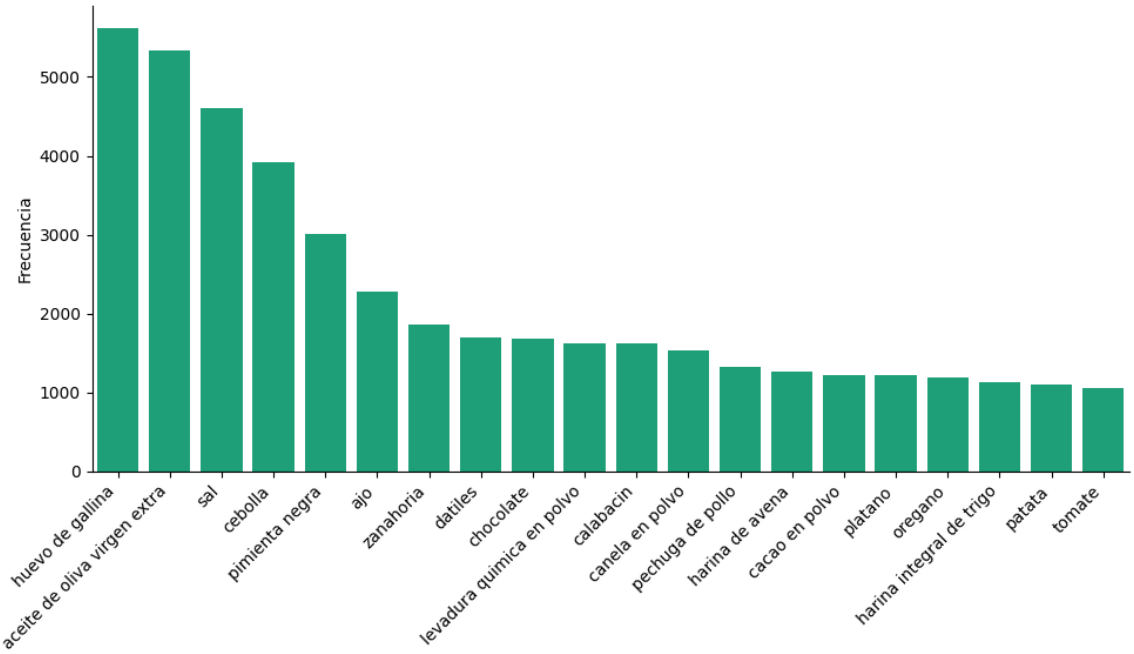
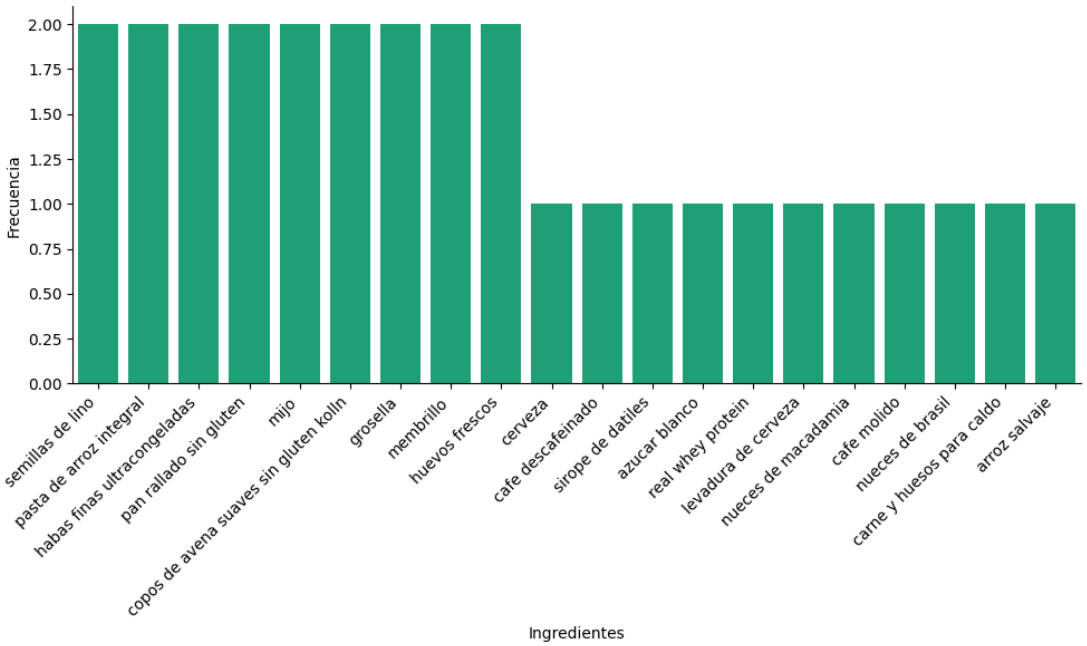


Figura 5.2 Veinte ingredientes menos comunes

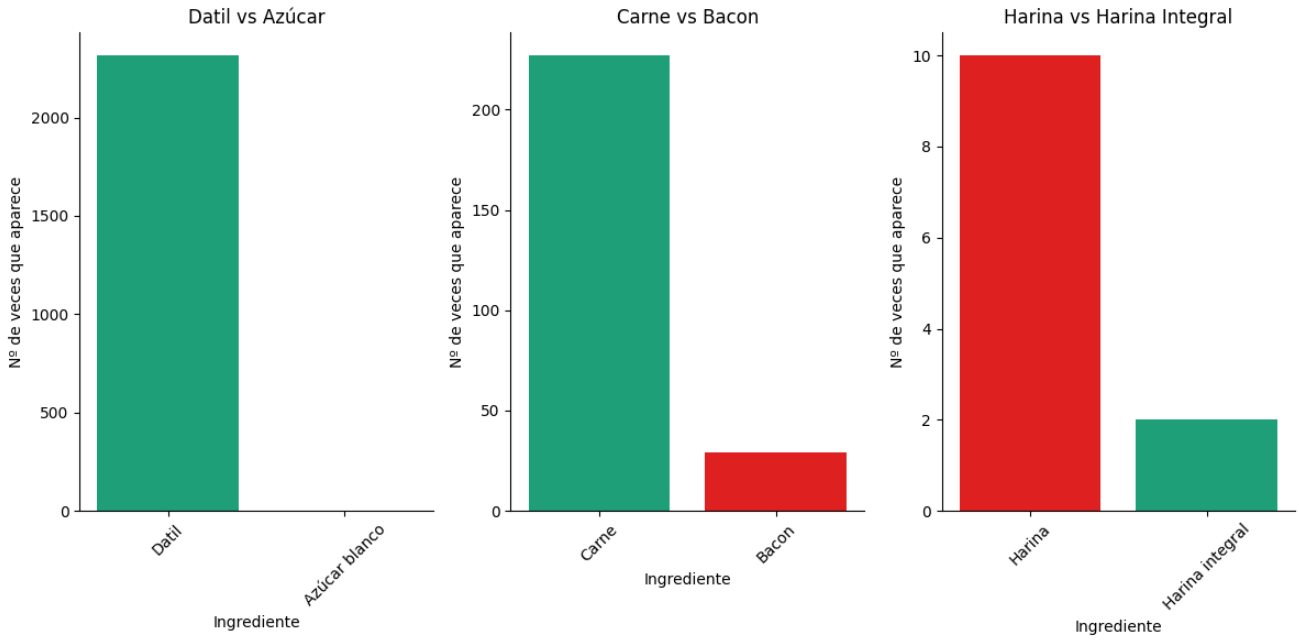


5.1.2 Comparación de ingredientes Realfooding / no Realfooding

Una de las recomendaciones del movimiento Realfooding es el consumo de harinas integrales frente a las refinadas, además de evitar ultraprocesados como el azúcar o el *bacon* en favor de otros ingredientes como los dátiles o la carne sin procesar. Por ello buscaremos el número de veces que aparecen los ingredientes anteriores en el total de las recetas.

En este caso, a tenor de los resultados mostrados en la Figura 5., en las recetas publicadas por los usuarios predominan el dátil y sus derivados, como la pasta de dátil, frente al azúcar refinada. En lo que respecta a la carne sin procesar como la de cerdo o pollo frente al *bacon* que, si está procesado, el uso de las primeras es mucho más amplio. Por otro lado, las harinas refinadas son más frecuentes que las integrales.

Figura 5.3 Comparación de ingredientes (dátil, azúcar, carne, bacon y harinas)



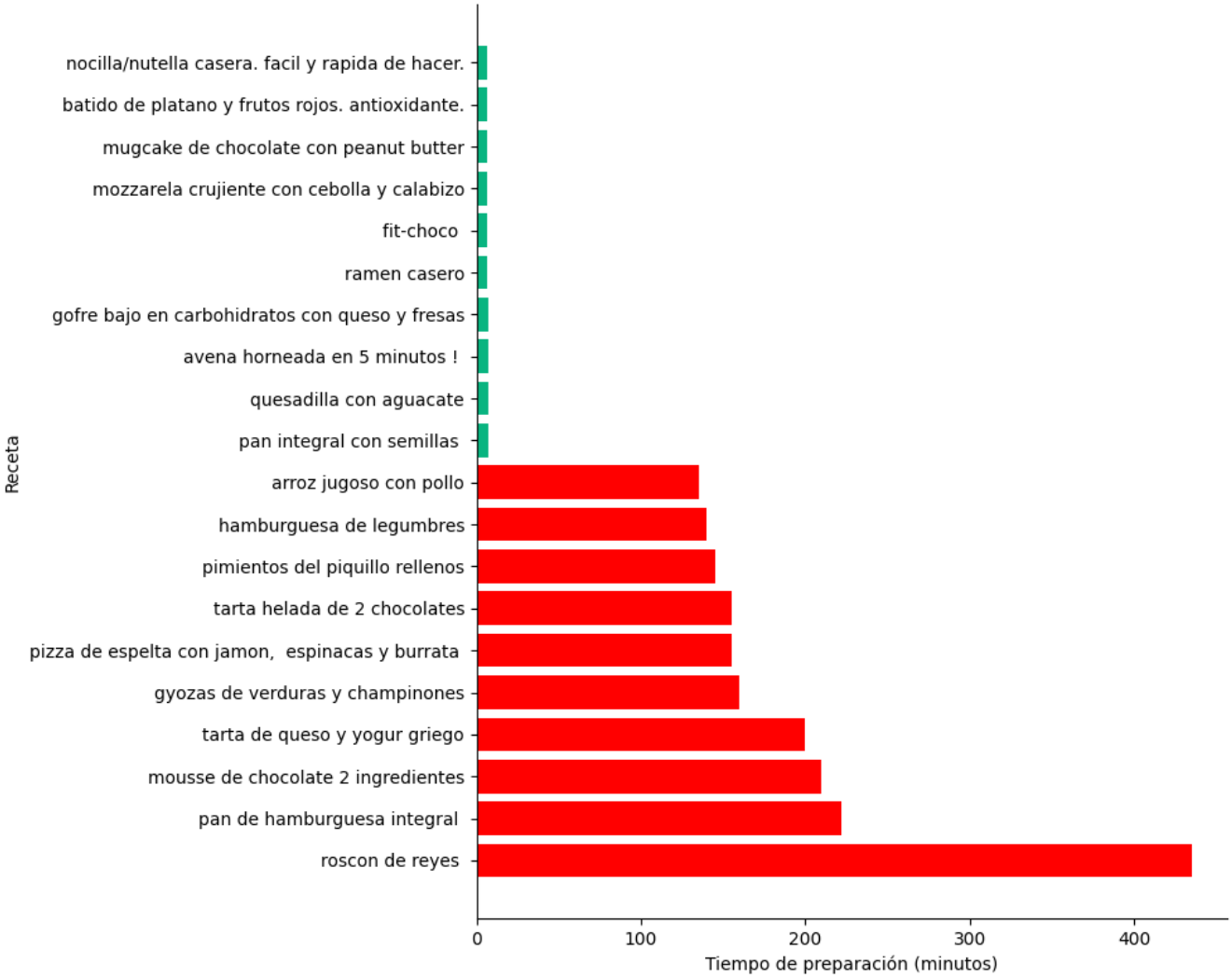
5.1.3 Recetas en función del tiempo de preparación

Otro aspecto importante a la hora de elaborar recetas en casa es el tiempo de preparación puesto que según un estudio realizado por Hello Fresh en 2023 el 34% de los encuestados reconocer que no sigue una dieta equilibrada por la falta de tiempo (Hello Fresh, 2023).

De este modo para elaborar recetas en casa la eficiencia en el tiempo de preparación se vuelve fundamental. La Figura 5.4 nos muestra que recetas se pueden preparar en un menor tiempo frente a las que requieren de una mayor dedicación.

Podemos observar que las recetas más rápidas por lo general se podrían considerar como *snacks,* mientras que las que requieren una mayor elaboración implican mayoritariamente técnicas como el amasado o el guiso de componentes extra para el plato.

Figura 5.4 TOP 10 recetas más y menos rápidas



5.1.4 Correlación entre el tiempo de preparación, calorías, valoración, puntos, cantidad de pasos y cantidad de ingredientes

En un contexto de falta de tiempo como se ha comentado anteriormente, surge la necesidad de comprender como influye el tiempo en la elaboración de recetas caseras. Exploraremos la correlación entre el tiempo de preparación, el número de ingredientes, los pasos necesarios para elaborar cada una de las recetas y la puntuación que reciben entre otros factores. Para calcular la correlación se ha aplicado el método de Spearman puesto que los datos no siguen una distribución normal y este método permite encontrar relaciones no lineales. Calcula la correlación utilizando los rangos de los datos en lugar de los valores originales, lo que la hace más robusta frente a distribuciones no lineales (Documentación pandas.DataFrame.corr, s.f.).

A continuación, se explicará el resultado de las correlaciones que se pueden observar en la Figura 5.5.

- Correlación entre el tiempo y las calorías (0,47).

Esta correlación positiva moderada alta indica que hay una relación moderadamente significativa entre la cantidad de calorías en una receta y el tiempo que se utiliza en ella. En general, las recetas que requieren más tiempo tienden a tener más calorías, y viceversa. Esto podría deberse a que los platos con mayor proceso de elaboración tienen más ingredientes los cuales aportan calorías, como grasas, proteínas o carbohidratos.

- Correlación entre cantidad de pasos y calorías (0,43).

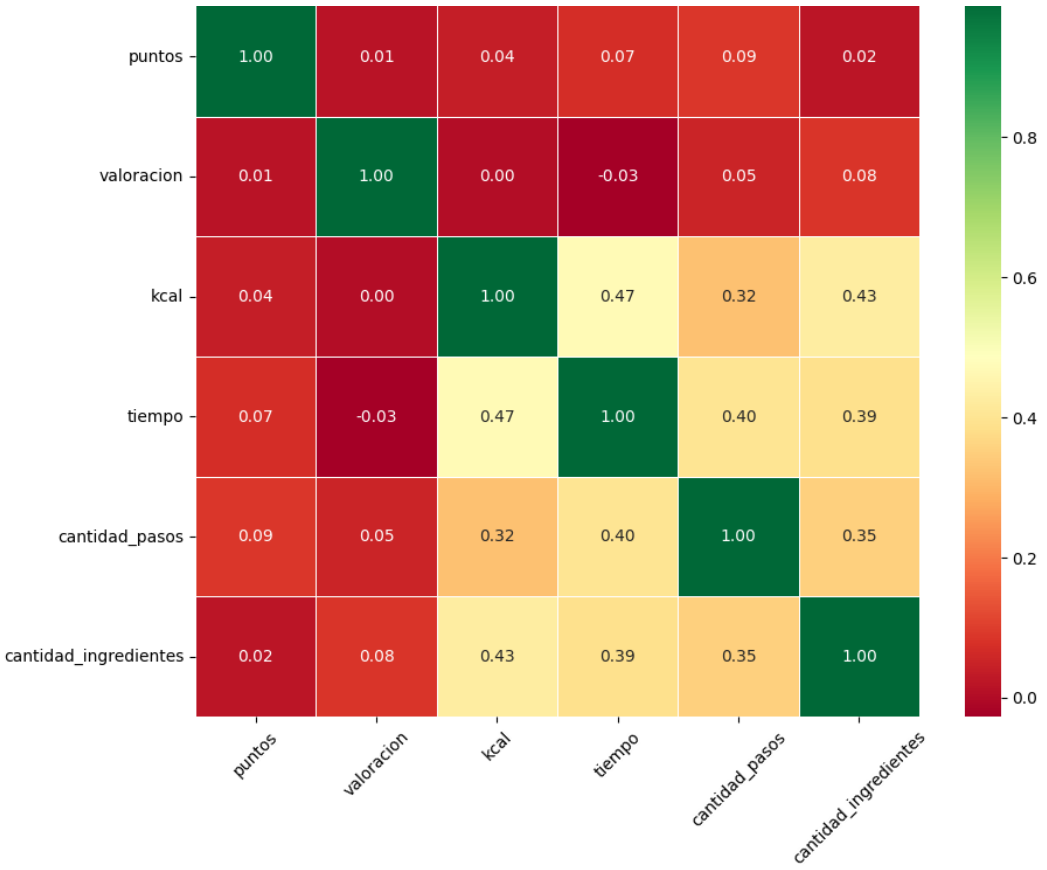
La correlación positiva moderada indica que hay una relación significativa entre la cantidad de pasos en una receta y las calorías que aporta. En general, las recetas con más pasos tienden a ser más calóricas. Esto podría deberse a que cada paso se corresponda con la preparación de un ingrediente que puede requerir su propio paso de elaboración, como picar, mezclar o cocinar.

- Correlación entre tiempo de preparación y la cantidad de pasos (0,40).

En este caso la correlación también es positiva moderada sugiriendo que hay una relación significativa entre el tiempo de preparación de una receta y la cantidad de pasos

necesarios para completarla. En general, las recetas que requieren más pasos también tienden a tomar más tiempo para prepararse.

Figura 5.5 Mapa de correlación

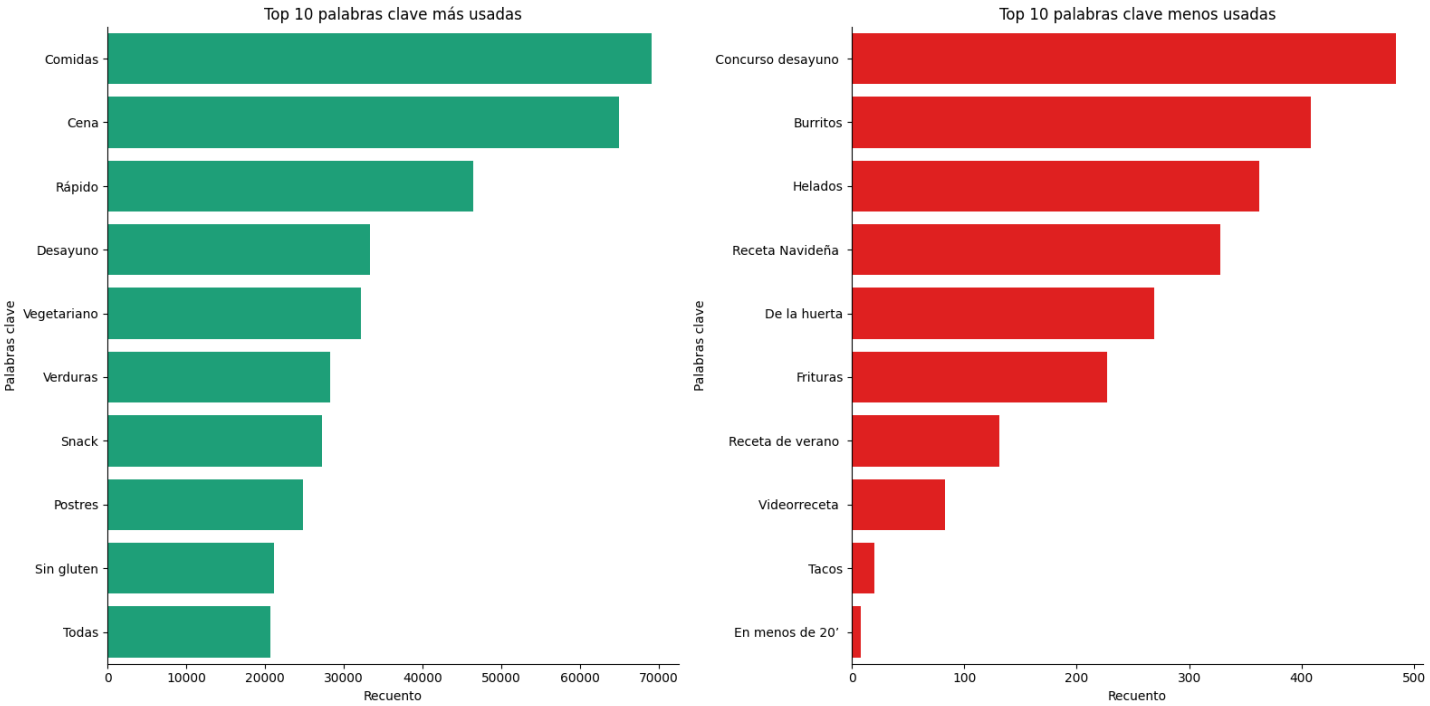


5.1.5 Palabras clave más utilizadas

Para concluir estudiaremos las palabras clave que se les asignan a cada una de las recetas. Esto nos permite agrupar las recetas por categorías, por ejemplo, comida, cena, desayuno, vegetariano etc.

En la Figura 5. podemos observar que las palabras clave que más se repiten son comida, cena y desayuno, las cuales se corresponden con las tres comidas principales del día. También se destaca la gran frecuencia con la que se utiliza la etiqueta de rápido, esto hace referencia a lo ya comentado anteriormente sobre la valoración del tiempo. Por otro lado, las palabras menos usadas son “en menos de 20” y tacos.

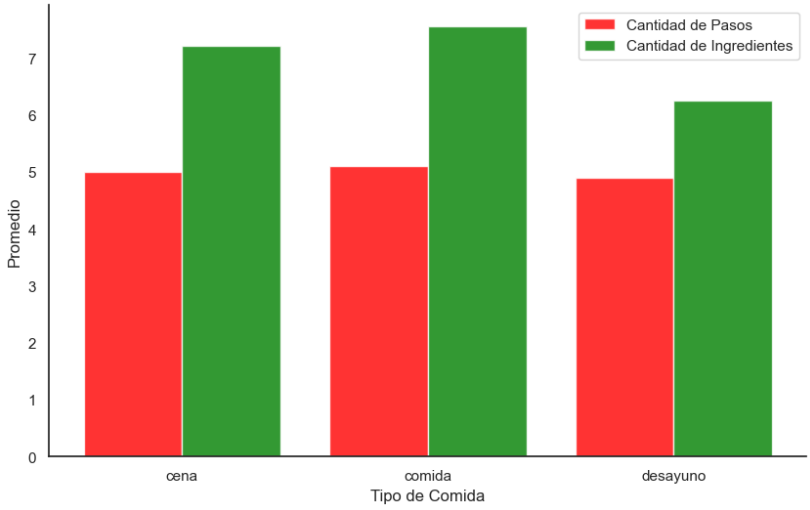
Figura 5.6 TOP 10 Palabras más y menos utilizadas



5.1.6. Número medio de pasos e ingredientes en cada tipo de comida

Se han calculado los valores medios del número de pasos necesarios para completar cada receta y el número medio de ingredientes utilizados. Estos datos proporcionan una visión general de la complejidad y la diversidad de ingredientes involucrados en la preparación de las tres categorías principales de tipos de comida: desayuno, comida y cena.

Figura 5.7 Promedio de pasos e ingredientes por tipo de comida



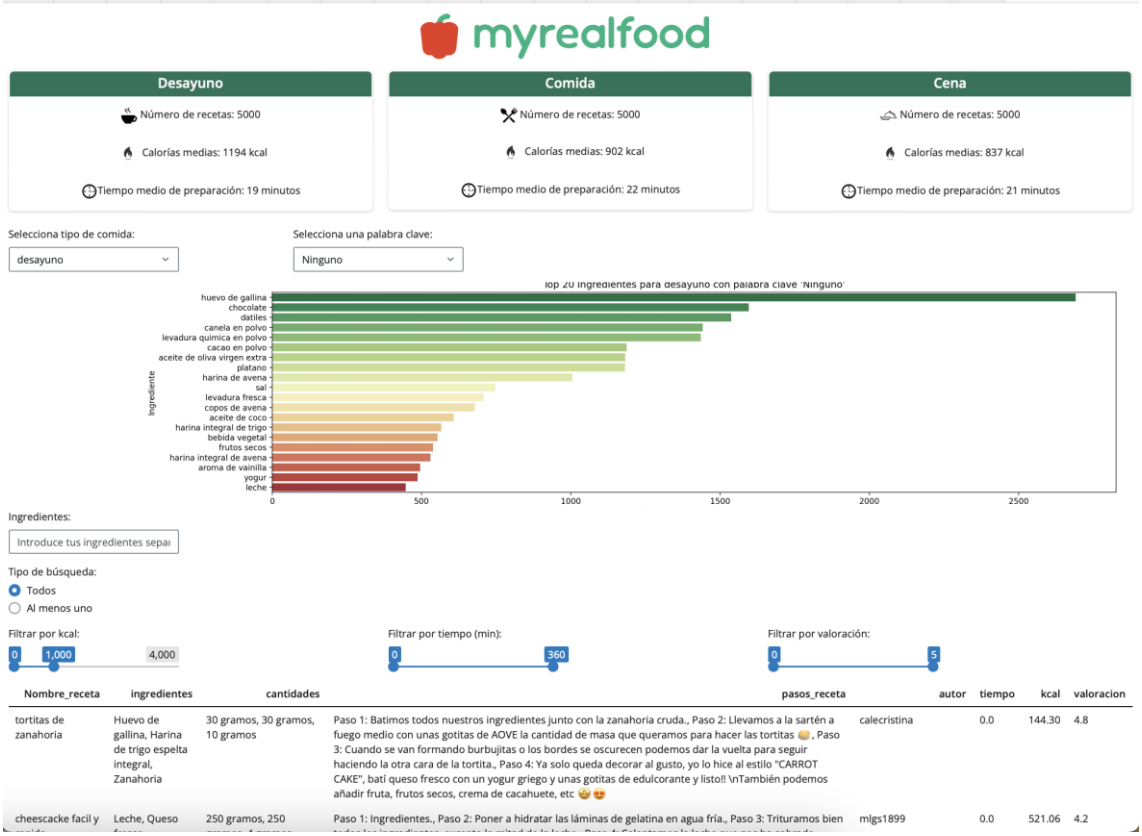
Como se puede ver en Figura 5.7 parece haber una leve variación en la cantidad promedio de pasos entre los diferentes tipos de comidas. Por ejemplo, las comidas podrían ser ligeramente más elaboradas que las cenas, mientras que los desayunos tienden a ser más simples. En lo que se refiere a la cantidad promedio de ingredientes varía más, podría deberse a que las comidas tienden a incluir más platos principales, guarniciones y posiblemente postres en comparación con las cenas y desayunos, que generalmente son comidas más simples.

6 Implementación del *dashboard*

En este capítulo se describe el resultado final del panel de control (*dashboard*) el cual busca ayudar a los usuarios a tomar decisiones en el día a día.

6.1 Resultado del panel de control

Para mejorar la experiencia de los usuarios en la organización y visualización de recetas, se ha desarrollado un panel de control interactivo, está diseñado para ofrecer una plataforma intuitiva y de fácil uso, permitiendo a los usuarios filtrar y gestionar recetas de manera eficiente y personalizada. La interfaz principal se muestra en la siguiente imagen.



Una de las principales ventajas de este panel de control es su capacidad para realizar un filtrado eficiente de recetas. Los usuarios pueden explorar las recetas disponibles según diferentes categorías de comida, como desayuno, comida o cena, lo que facilita la búsqueda específica de acuerdo con sus necesidades. Además, la funcionalidad de búsqueda mediante palabras clave permite localizar rápidamente recetas que se ajusten a

ciertas preferencias dietéticas, por ejemplo, dietas sin gluten. Esto, combinado con la opción de filtrar por tiempos de preparación, ofrece a los usuarios la posibilidad de elegir recetas que se ajusten a su disponibilidad de tiempo.

Otra característica destacada del panel de control es su capacidad para la visualización avanzada de ingredientes a través de gráficos interactivos. Estos gráficos muestran, de manera clara y detallada los ingredientes más usados en las recetas lo que puede dar inspiración a los usuarios para experimentar con ingredientes comunes en sus propias creaciones culinarias.

El diseño intuitivo y las funcionalidades avanzadas de este panel de control no solo mejoran la accesibilidad y el manejo de la información de las recetas, sino que también juegan un papel importante en la reducción del desperdicio alimentario y la promoción de hábitos de cocina más sostenibles al permitir a los usuarios buscar recetas basadas en los ingredientes que ya tienen en sus casas.

A continuación, se detalla cada uno de los componentes principales de la interfaz, explicando su función y composición.

6.1.1 Título y encabezado principal

Como se puede ver en la Figura 6.2 en la parte superior del panel de control se encuentra el logo de la página web. Debajo del título, se dispone de tres cajas informativas cada una de ellas dedicada a un tipo de comida principal: desayuno, comida y cena. Estas cajas presentan datos relevantes para ayudar a los usuarios a obtener una visión rápida de la información por categorías de comida principales.

Para cada caja de información, se muestra el número total de recetas disponibles, las calorías promedio de las recetas correspondientes y el tiempo medio de preparación.

Figura 6.2 Título y encabezado

Estas cajas de información son útiles para proporcionar a los usuarios un resumen inmediato de los datos más relevantes, pudiendo conocer de un vistazo que el desayuno es el tipo de comida con más calorías medias y que en término medio lleva menos tiempo preparar. Dichas cajas se crean a través del componente `ui.card` que es un contenedor que permite mostrar información de manera estructurada y atractiva. Puede contener texto, imágenes u otros elementos.

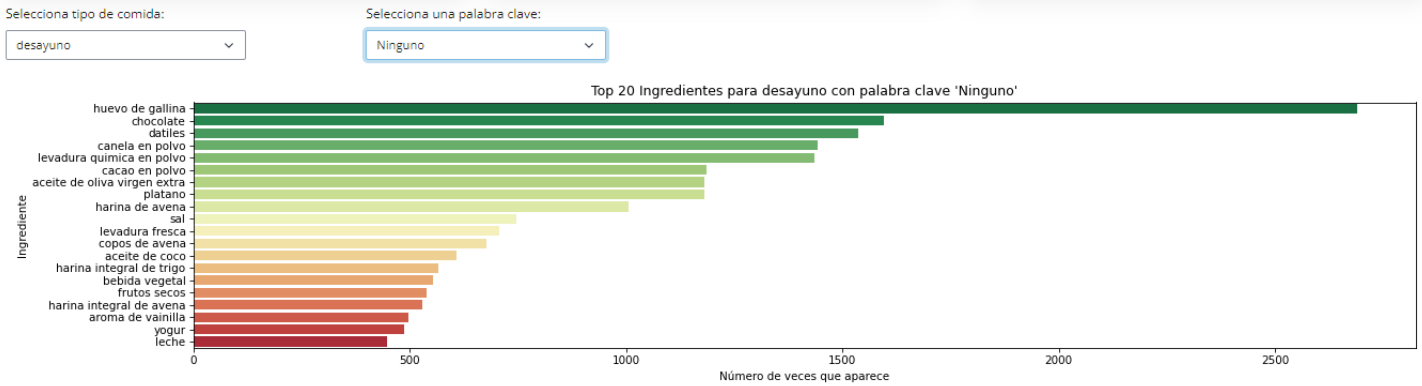
6.1.2 Gráfico de barras de ingredientes Comunes

Bajo las cajas de información, podemos ver el contenido de la Figura 5. 1, el gráfico de barras interactivo que muestra los veinte ingredientes más comunes utilizados en las recetas disponibles en la aplicación. Este gráfico es una herramienta visual que permite a los usuarios ver de un vistazo qué ingredientes son frecuentemente utilizados.

Una de las características destacas de este gráfico es su capacidad de filtrado avanzado. Los usuarios pueden personalizar la visualización de los ingredientes más comunes aplicando filtros por comida principal (desayuno, comida o cena) y de forma opcional mediante la introducción de palabras clave que hacen referencia a distintas variedades dietéticas como por ejemplo dietas sin gluten, pero también incluyen categorías como postres.

Esta funcionalidad es particularmente útil para aquellos que siguen dietas específicas o con restricciones de alimentos.

Figura 5. 1 Veinte ingredientes más comunes por tipo de comida y palabra clave



Para obtener el gráfico visualmente se define utilizando la función `server`, donde se utilizan decoradores como `@output` (para indicar que una función va a producir un resultado que será renderizado en la interfaz de usuario) y en este caso `@render.plot`

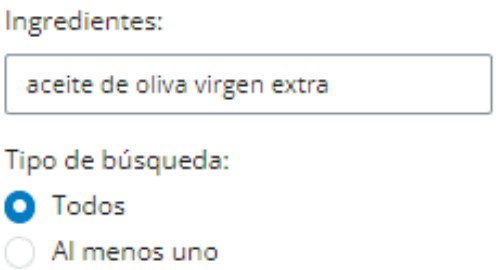
que se utiliza en combinación con `@output` para indicar que la función va a generar un gráfico dinámico.

Los filtros del gráfico se crean a través del elemento `ui.input_select`, este elemento crea un menú desplegable que permite al usuario seleccionar una opción de una lista predefinida, en este caso tipo de comida y palabras clave.

6.1.3 Filtros interactivos

Búsqueda de Ingredientes: Una de las funcionalidades más útiles es la barra de búsqueda por ingredientes. Como se puede ver en la Figura 6.2 los usuarios pueden introducir uno o más ingredientes que tienen disponibles en su despensa. En este caso se están buscando recetas que contengan aceite de oliva virgen extra, y la tabla a la que está asociado mostrará todas las recetas que contienen dicho ingrediente. Esta funcionalidad permite especificar si las recetas deben incluir todos los ingredientes introducidos o solo algunos de ellos, lo que ofrece una gran flexibilidad y precisión en la búsqueda de recetas.

Figura 6.2 Filtro para buscar recetas mediante los ingredientes



Este filtro se genera con el componente `ui.input_text` que proporciona un campo de texto donde los usuarios pueden introducir datos de texto. Además, con `ui.input_radio_buttons` se crea la lista de botones de opción, en este caso para seleccionador todos o algunos de los ingredientes.

Filtros deslizables: Por otra parte, los filtros de las Figura 6.3, Figura 6.4 y la Figura 5 permiten fijar el intervalo de calorías, tiempo o valoración que se desea que tengan las recetas que aparecerán como resultado. En este caso se están seleccionando aquellas recetas que tienen entre 285 y 1694 calorías, cuyo tiempo de preparación oscila entre los 15 y 124 minutos y por última, la valoración debe estar entre 2 y 5.

Figura 6.3 Filtro para las kcal

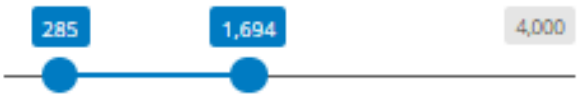


Figura 6.4 Filtro para el tiempo

Figura 5.6 Filtro para la valoración

Los tres filtros anteriores se crean a partir del componente `ui.input_slider`, el cual permite crear un filtro deslizante para que el usuario pueda seleccionar un valor numérico dentro de un rango predefinido.

6.1.4 Tabla de recetas filtrable

Los anteriores filtros dan lugar a la tabla de resultados que se encuentra en la parte inferior del panel de control y que se puede observar en las Figura 5.7 y la Figura 6.6.

Dicha tabla contiene todas las recetas disponibles y permite a los usuarios ver de forma ordena aquellas que se correspondan con los filtros aplicados en los pasos anteriores lo que facilita la búsqueda y selección de la receta ideal según sus necesidades específicas.

Se compone de todos los datos almacenados en el `DataFrame` y como ya se ha comentado anteriormente, se visualiza utilizando los decoradores `@output` y en este caso concreto `@render.table` que se utiliza en combinación con `@output` para indicar que la función va a generar una tabla dinámica.

Figura 5.7 Primera parte de la tabla de resultados

Nombre_receta	ingredientes	cantidades
avena con crema de cacahuete al horno	Levadura fresca, Bebida vegetal, Crema de cacahuete, Banana, Harina de avena	5 gramos, 70 gramos, 20 gramos, 50 gramos, 40 gramos
brownie de platano	Canela en polvo, Harina de trigo, Huevo de gallina, Cacao en polvo, Plátano, Levadura química en polvo	5 gramos, 65 gramos, 134 gramos (aprox. 2 unidades), 10 gramos, 376 gramos (aprox. 2 unidades), 3 gramos (aprox. 1 cucharadita)

Figura 6.6 Segunda parte de la tabla de resultados

pasos_receta	autor	tiempo	kcal	valoracion
Paso 1: Machacamos el plátano y lo mezclamos con los demás ingredientes. No lo puse arriba pero se puede añadir tb 1cda de proteína en polvo para ese extra proteico (yo usé una sin sabor de myprotein), Paso 2: La bebida vegetal la cantidad de arriba es aproximada, en realidad hay que ir añadiendo poco a poco hasta conseguir una textura no muy densa. Opcional añadir chips de chocolate , Paso 3: Al horno a 180 grados 15min más o menos. Opcional y recomendable decorar, una vez se vaya a consumir, con más crema de cacahuete!!	foodievs	25.0	327.72	4.9
Paso 1: Mezclamos todos los ingredientes hasta que quede una pasta., Paso 2: Lo pondremos en el horno 15 minutos a 180 grados.	yadamid	25.0	922.89	4.4

Para que todos estos componentes sean visibles se utiliza el componente `ui.page_fluid` el cual permite crear páginas web que se ajustan dinámicamente. Además, para organizar los elementos que componen la página se han utilizado los componentes `ui.row` (para agrupar elementos en una fila horizontal en la página) y `ui.column` (para dividir una fila en columnas).

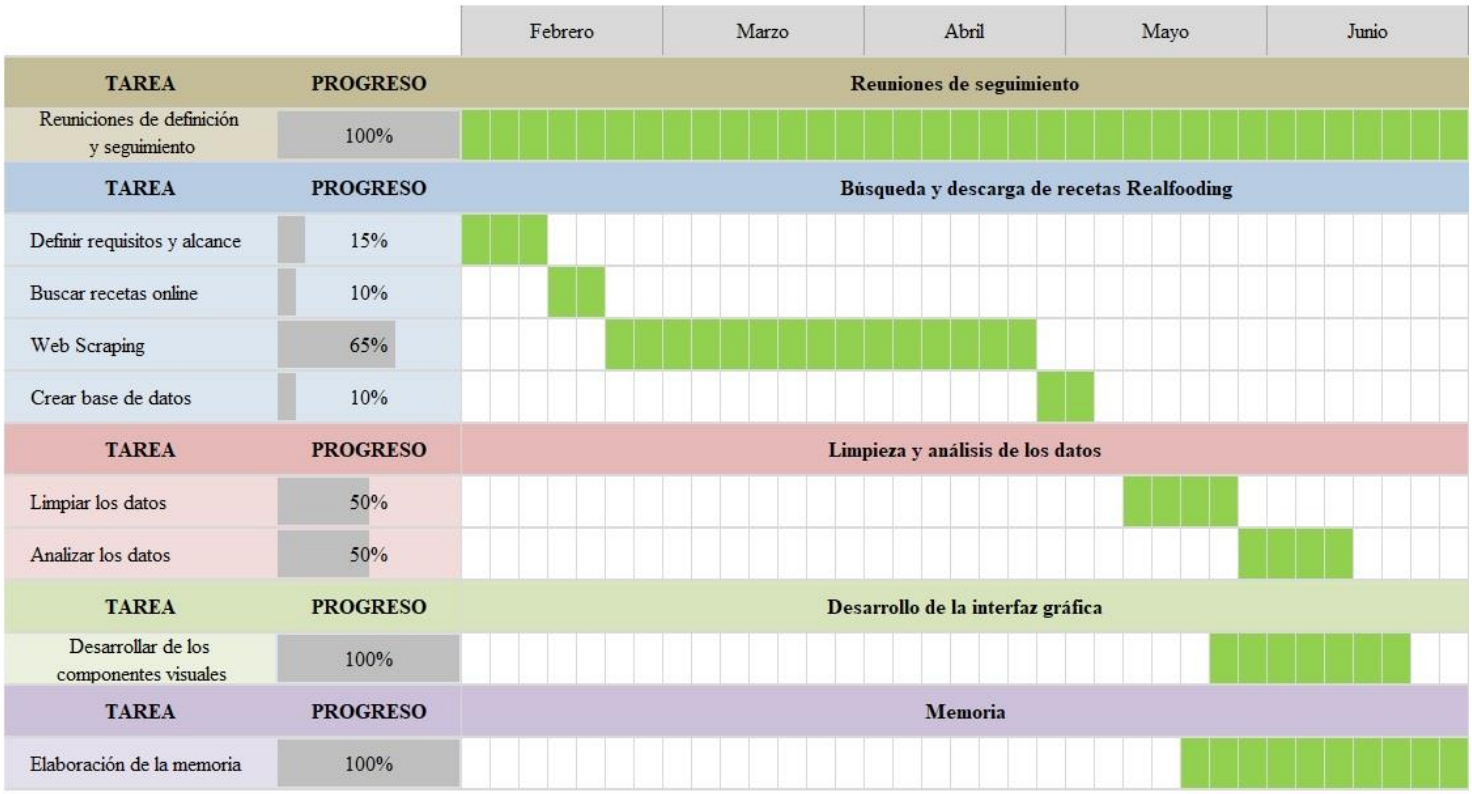
7 Planificación del proyecto

En la gestión de proyectos, la planificación desempeña un papel fundamental para el éxito de cualquier iniciativa. Desde el inicio hasta la finalización, una planificación efectiva garantiza que los recursos se asignen de manera eficiente, los plazos se cumplan y los objetivos se alcancen de manera oportuna. Una herramienta valiosa en este proceso es el diagrama de Gantt, que ofrece una representación visual y detallada del cronograma del proyecto.

Una de las ventajas principales del diagrama de Gantt es su capacidad para identificar las actividades críticas, es decir, aquellas que tienen un impacto directo en la duración total del proyecto. Esto permite dar importancia a las tareas más relevantes y asegurarse de que se completen a tiempo para así evitar retrasos en el proyecto en su conjunto.

A continuación, en el diagrama de Gantt representando en la Figura 7.1, se especifican las tareas que se han realizado en cada una de las actividades que se detallan en el diagrama.

Figura 7.1 Diagrama de Gantt



Fase 1: Reuniones de definición y seguimiento

La primera fase consta de las reuniones mantenidas para definir los requerimientos y un posterior seguimiento.

Se realizan reuniones periódicas para revisar el progreso del proyecto, discutir problemas y tomar decisiones sobre ajustes necesarios. Estas reuniones aseguraron una adecuada gestión del proyecto.

Fase 2: Búsqueda y descarga de recetas Realfooding

En esta fase se realizan las siguientes tareas:

- Definir requisitos y alcance

En esta tarea se establecen los objetivos del proyecto y se definen los requisitos técnicos necesarios para la ejecución de este. Además, se determina el alcance de las actividades a realizar para asegurar una correcta planificación.

- Buscar recetas online

La tarea consiste en la búsqueda exhaustiva de recetas de Realfooding en la web. El objetivo es recopilar la mayor cantidad posible de recetas que cumplan con los criterios definidos previamente.

- *Web Scraping*

Se lleva a cabo la extracción automatizada de recetas desde el sitio web seleccionado. Esta tarea implica el uso de herramientas de *web scraping* para capturar la información necesaria de manera eficiente.

- Crear una fuente de datos sobre la que consultar la información

Se crea y un `DataFrame` para almacenar las recetas recopiladas. Esta tarea es de vital importancia para organizar y gestionar los datos de manera eficiente, permitiendo su fácil acceso y manipulación en fases posteriores.

Fase 3: Limpieza y análisis de los datos

Durante esta fase, se abordan las siguientes tareas:

- Limpiar los datos

En esta tarea se realiza la depuración y limpieza de los datos obtenidos para asegurar su calidad. Se realizan tareas como eliminar duplicados, eliminar acentos, símbolos o carácter especiales y ajustar los formatos de los datos.

- Analizar los datos

Se lleva a cabo un análisis detallado de los datos de las recetas para extraer información valiosa. Esto incluye la identificación de correlaciones o estadísticas de uso de los diversos ingredientes.

Fase 4: Desarrollo del panel de control

En esta fase, se enfoca en la creación del panel de control interactivo con la siguiente tarea:

- Desarrollar los componentes visuales

Esta tarea implica el diseño y la programación de los elementos visuales de la interfaz de usuario. Se desarrollan componentes gráficos que permitirán a los usuarios interactuar con la base de datos de manera intuitiva, visual y eficiente.

Fase 5: Memoria

Finalmente, esta fase abarca el proceso de documentación:

- Elaboración de la memoria

En esta tarea se documenta todo el proyecto en una memoria final que incluye detalles sobre cada fase, los resultados obtenidos y las conclusiones. La memoria sirve como una referencia completa del proyecto y sus descubrimientos.

8 Conclusiones

Una de las actividades para llevar a cabo el estilo de vida que promueve el Realfooding es la elaboración de las comidas en el hogar, que permite a las personas reducir en gran medida el consumo de comida precocinada u otro tipo de ultraprocesados. En la mayoría de los casos encontrar información agrupada y filtrada sobre este tipo de recetas no es una tarea sencilla. El uso de técnicas de *web scraping* en este trabajo ha sido fundamental para recopilar un gran volumen de recetas de la página web de referencia Myrealfood de forma automatizada. Esta metodología permitió obtener información detallada sobre ingredientes, calorías, tiempo de preparación y otra información que se utilizó como fuente para un posterior análisis de datos.

A lo largo del trabajo se presentaron y superaron varios retos técnicos asociados con el proceso de *web scraping*. Se puede destacar la variabilidad en la estructura de las páginas web por las continuas actualizaciones o la implementación de medidas *anti-scraping* por parte del sitio web.

En lo que respecta al análisis realizado posteriormente sobre las recetas obtenidas podemos ver en los gráficos generados una visión clara y concisa sobre la composición de las recetas recogidas. Posteriormente se desarrolló un panel de control que facilita la interacción con los datos. Esta herramienta permite a los usuarios seleccionar y visualizar gráficos personalizados según sus intereses específicos, como por ejemplo acotar el contenido calórico de diferentes recetas o analizar la frecuencia de ciertos ingredientes. Una acción interesante del panel es la posibilidad de introducir los ingredientes que deseamos que contengan las recetas, de modo que se puede hacer una búsqueda más personalizada y concreta. El panel de control proporciona una plataforma intuitiva y accesible para explorar y comprender los datos de manera más efectiva, mejorando la capacidad de los usuarios para tomar decisiones rápidas.

En lo personal, considero que este trabajo ha sido de gran valor ya que me ha ayudado a familiarizarme con el proceso de minería de datos mediante la implementación de un proyecto real de un tema que me interesa.

9 Bibliografía

Documentación pandas.DataFrame.corr. (s.f.). Obtenido de https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.corr.html

gizblogs. (02 de 21 de 2024). Recuperado el 03 de 07 de 2024, de https://gizblogs.com/cuantos-datos-se-crean-cada-dia-en-2024-estadisticas-completas/#%C2%BFCuantos_datos_se_crean_cada_dia_en_2024

Hello Fresh. (01 de 2023). Recuperado el 23 de 06 de 2024, de https://www.hellofresh.es/kits-de-recetas/habitos-en-la-cocina-espanoles

Herrero, J. (23 de 11 de 2021). *La Razón*. Recuperado el 04 de 06 de 2024, de https://www.larazon.es/salud/20211123/ueeugglpnzf2fa5ibu4rl26rom.html

jcweb. (03 de 07 de 2024). Obtenido de https://jcweb.es/la-historia-de-html-el-creador-y-origen-del-lenguaje-web/

Oficina Española de Patentes y Marcas. (s.f.). *Consulta de expedientes OEPM*. Recuperado el 04 de 07 de 2024, de https://consultas2.oepm.es/ceo/jsp/busqueda/consultaExterna.xhtml?numExp=1932hvgq2jayj6iltnfuah1i2y6vgjmqn1yf6huw0gz0y8luci51npmlefbgxj9yvkxjmkin24v31pr5nlbj8e9c0p8xor9zs61j

Olalla, F. (22 de 03 de 2020). *Deusto Salud*. Recuperado el 25 de 06 de 2024, de https://www.deustosalud.com/blog/dietetica-nutricion/que-es-movimiento-real-food-realfooding

Soto, E. (20 de 11 de 2019). *El mundo*. Recuperado el 03 de 07 de 2024, de https://www.elmundo.es/baleares/2019/11/20/5dd514b421efa00d088b45d3.html

Toledo, J. (28 de 07 de 2023). *Scriptología*. Recuperado el 27 de 06 de 2024, de https://scriptologia.com/origen-y-evolucion-de-python-la-historia-del-lenguaje-de-programacion/

10 Declaración de trabajo original

De acuerdo con lo expresado en el *artículo 8.3 del Reglamento para la elaboración y defensa del Trabajo Fin de Máster de la Universidad de Oviedo*, aprobado por su Consejo de Gobierno el 17 de julio de 2020 (BOPA de 7 de agosto de 2020), quiero expresar lo siguiente:

Yo, Laura Argüelles Pérez, con DNI 21090046N, en relación a la memoria que presento ante el Tribunal, para su valoración como ***Trabajo Final en el Máster Universitario en Análisis de Datos para la Inteligencia de Negocios (MANADINE),*** quiero ***DECLARAR*** que ***s***oy el autor de la misma, habiendo citado debidamente las fuentes utilizadas en su desarrollo.

Para que conste, firmo el presente documento.

Oviedo, 15 de julio de 2024

Fdo.-