

# **Escuela de Ingenierías Industrial, Informática y Aeroespacial**

## **GRADO EN INGENIERÍA INFORMÁTICA**

### **SISTEMA RECOMENDACIÓN DE RECETAS DE COCINA**

Autor: Carlos Díez Gutiérrez

Tutor: D. Enrique López González

(Diciembre, 2021)

# ÍNDICE

<b>1. DESCRIPCIÓN DEL PROBLEMA.....</b>	<b>3</b>
<b>2. HERRAMIENTAS UTILIZADAS .....</b>	<b>5</b>
2.1. FRONTEND .....	5
2.2. BACKEND.....	5
2.3. BASE DE DATOS.....	6
<b>3. BASE DE DATOS UTILIZADA .....</b>	<b>7</b>
CREACIÓN BASE DE DATOS.....	7
ESTRUCTURA DEL GRAFO DE CONOCIMIENTO.....	8
<b>4. DESCRIPCIÓN DE LA APLICACIÓN.....</b>	<b>10</b>
<b>5. EXPLICACIÓN DEL ALGORITMO .....</b>	<b>15</b>
5.1. ALGORITMO BASADO EN CONTENIDO .....	15
5.2. ALGORITMO DE FILTRADO COLABORATIVO .....	19
5.3. ALGORITMO HÍBRIDO .....	23
<b>6. ANALISIS DE RESULTADOS.....</b>	<b>25</b>
6.1. ALGORITMO BASADO EN CONTENIDO .....	25
6.2. ALGORITMO DE FILTRADO COLABORATIVO .....	30
6.3. ALGORITMO HÍBRIDO .....	33
<b>7. ANALISIS CRÍTICO.....</b>	<b>36</b>
<b>8. LINEAS DE FUTURO .....</b>	<b>38</b>
<b>9. LECCIONES APRENDIDAS .....</b>	<b>39</b>
<b>10. BIBLIOGRAFÍA.....</b>	<b>40</b>

## 1. DESCRIPCIÓN DEL PROBLEMA

El origen de los sistemas de recomendación surge a mediados de la década del 2000 con la aparición de la web 2.0. Este tipo de web fue un cambio de paradigma en el que los usuarios además de acceder a la información la creaban ya sea comprando determinados productos, dando me gustas, etc. Es decir, todas las acciones que puedan modificar la base de datos de una aplicación.

En general los sistemas de recomendación buscan mejorar la experiencia de usuario en una web o aplicación, proporcionándole contenido o productos personalizados en base a sus gustos o preferencias. De esta forma, el usuario puede encontrar fácilmente lo que está buscando y reducir el tiempo de búsqueda.

El tema del que tratará el proyecto será sobre la recomendación de recetas de cocina en base a una serie de parámetros que se comentarán más abajo. Esta idea surgió ante una necesidad que tenía y no hubiera ninguna herramienta en internet que la resolviera. Como este es mi último año de carrera y probablemente cuando empiece a trabajar tenga que salir fuera, allí no habrá nadie que cocine por mí así que se me ocurrió hacer un sistema de recomendación en base a la dificultad, tiempo de preparación, gustos del usuario y calorías. De manera que si una persona normalmente no suele tener mucho tiempo se le recomienden recetas con poco tiempo de preparación o si está empezando a cocinar que la dificultad sea baja. Además, a todo esto, se tienen en cuenta los gustos de la persona y las calorías diarias que tiene que cumplir dicha persona.

Existen numerosos sistemas de recomendación, pero los tipos más importantes son dos:

- Sistemas basados en contenido: se basan en la información facilitada por el usuario acerca de sus gustos y preferencias.
- Sistemas basados en filtros colaborativos: se basan en la información que el sistema extrae a partir de la actividad de navegación del usuario.

Estos 2 tipos de sistemas tienen muchos tipos de variaciones en las que dependiendo el contexto puede funcionar uno u otro tipo, pero al final de cuentas se encuentran en estos dos grupos.

Ahora que se tiene un conocimiento general de cómo funcionan los sistemas de recomendación mi proyecto plantea resolver 2 problemas en cuanto el tema de la elección de recetas de cocina:

- Cuando vamos a hacer la comida muchas veces nos surgen las dudas de qué comida hacer simplemente porque no se nos ocurren más recetas o porque las recetas que uno ve en internet no sabe si le van a gustar o no. El sistema de recomendación lo que hará será aprender cuáles son las recetas que mejor y peor puntúa el usuario para dar una recomendación precisa en base a sus gustos,

dificultad y tiempo de preparación.

- Otro factor muy importante dentro del mundo gastronómico es la salud ya que la comida influye muy directamente en la salud de una persona. El sistema recomendará recetas de cocina en función de cuanto haya comido la persona durante el día para adecuarse al estándar propuesto por la OMS (Organización mundial de la salud) de 2000 kcal de media.

Aunque el objetivo principal de la aplicación es recomendar unas recetas en base a unos parámetros también podría servir para llevar un registro de la información nutricional (azúcares, fibra, carbohidratos, etc) de un usuario a lo largo del día.

## 2. HERRAMIENTAS UTILIZADAS

En la aplicación existen 3 partes bien diferenciadas, por un lado, está el frontend, por otro el backend y por último la base de datos.

### 2.1. FRONTEND

Es la capa que se encuentra en el lado del cliente, es decir, todo lo que nos encontramos delante de la pantalla. En la aplicación web se usan las tecnologías **HTML, CSS y JavaScript**.

Pero con estas 3 tecnologías no es suficiente para construir una buena página web a nivel de gráficos y de forma sencilla. Por eso, también se utilizaron 2 tecnologías extra:

- **Bootstrap:** Es un framework que sirve para crear interfaces de forma simple y adaptándose a cualquier dispositivo con un estilo responsive.
- **Bootstrap studio:** Es un programa a nivel de escritorio que te permite diseñar aplicaciones web sin casi necesidad de programar utilizando el framework Bootstrap.

### 2.2. BACKEND

Es la capa oculta (no visible al usuario) donde se procesan todas las peticiones hechas por el cliente en el frontend. En el backend se realizan funciones de lógica y la conexión con la base de datos.

- **JavaScript:** Estuve dudando entre javascript y python pero al final me decanté por javascript (NodeJS) ya que es un lenguaje más ligero y menos complejo que python, lo que lo hace más rápido de ejecutar.
- **NodeJS:** Node.js es un entorno de ejecución de código JavaScript que permite la creación de aplicaciones. Se pueden usar las mismas herramientas y librerías que se usan en el navegador, con la ventaja de que se ejecutan en el servidor. Es bastante simple y hay un montón de tutoriales en Internet donde se puede aprender fácilmente, además de un amplio número de paquetes que ayudan a la simplificación del trabajo.
- **Express JS:** Es un framework de desarrollo flexible y simple para Nodejs. Express nos permite crear el mecanismo de autenticación de un usuario, recibir las peticiones http que ha mandado el usuario, establecer motores de plantillas para la generación de páginas dinámicas, etc.

### 2.3. BASE DE DATOS

En esta parte existen muchas tecnologías y algunas con un esquema muy diferente en el funcionamiento. Por un lado, están las típicas bases de datos relaciones (MySQL, SQLServer, Oracle, etc) y por otro están las bases de datos orientadas a grafos (Neo4j, OrientDB, etc). La principal diferencia entre estos 2 esquemas es como tratan las relaciones entre 2 entidades que están almacenadas. En las bases de datos orientadas a grafos (**BDOG**) las relaciones se almacenan en el nivel de registro individual, mientras que, en una base de datos relacional, la estructura se define en una tabla. En consecuencia, las bases de datos orientadas a grafos (BDOG) son mucho más rápidas para datos que están conectados que las relacionales. Además, hacen que la creación de la base de datos y las consultas sean mucho más agradables, lo que significa un desarrollo mucho más rápido.

En mi caso que el tema trata de recetas de cocina existen muchas recetas que contienen diferentes ingredientes y numerosos usuarios que evalúan las recetas. Todo eso provoca que haya muchas relaciones por lo que la tecnología que se usó para la base de datos fue Neo4j, que encaja perfectamente con nuestro tipo de problema.

Nunca había oído hablar de Neo4j por lo que mis conocimientos sobre esta tecnología eran nulos. Para aprender acerca de Neo4j hice el curso Introduction to Neo4j 4.0 el cual me dio los conocimientos necesarios como para poder crear la base de datos.

Neo4j es una BDOG Open Source de las más conocidas. El lenguaje para hacer consultas (insertar, borrar, actualizar y seleccionar) a la base de datos se llama CYPHER y está inspirado en SQL. La plataforma de Neo4j permite a los desarrolladores desplegar la base de datos de 3 formas diferentes:

1. **Neo4j Desktop:** En este apartado el usuario descargará el binario e instalará la base de datos en local.
2. **Neo4j Aura:** Es un servicio de base de datos de gráficos en la nube completamente administrado.
3. **Neo4j Sandbox:** Es un servicio con el que se puede probar neo4j de forma gratuita y remota igual que con Aura, con la salvedad de que contiene herramientas para científicos de datos.

De estas 3 opciones se eligió Neo4j Aura, así uno se asegura de poder ejecutarlo en cualquier momento y desde cualquier ordenador.

### 3. BASE DE DATOS UTILIZADA

Para poder realizar un sistema de recomendación que funcione correctamente es necesario tener una buena base de datos. Como mi tema eran las recetas de cocina estuve buscando bases de datos en Kaggle y diferentes plataformas, pero no encontré nada que me sirviera y si encontraba algo, estaba en inglés. Ante este problema decidí hacer web scraping para la construcción de la base de datos.

#### 3.1. CREACIÓN BASE DE DATOS

La mayor parte del web scraping se hizo en la página web <https://www.kiwilimon.com> con la que obtuve los datos de tiempo de preparación, dificultad, nombre de la receta, instrucciones para elaborar la receta e información nutricional (proteínas, carbohidratos, etc).

A la hora de hacer web scraping me topé con un gran problema y es que los datos de la web se cargaban dinámicamente, por tanto, tenía que esperar a que el navegador los renderizara. Para resolver ese problema utilicé la librería Selenium de python. Código de la aplicación para hacer web scraping ([recetasKiwiLimon.py](#)).

Una vez se tienen todos los datos era el momento de sacar los ingredientes porque en la web no venían tal cual, sino que estaban dentro de las instrucciones de la receta. Para hacer esto encontré una base de datos <https://bedca.net/bdpub/index.php> en la que están todos los alimentos en español que existen. Así que por un lado descargué la base de datos ([Archivo para descargar base de datos](#)) y por otro dividí las instrucciones de la receta en palabras para poder compararla con la base de datos ([obtencionIngredientes.py](#)). Ahora que ya se tienen los ingredientes aparecieron algunos que no existían y otros que eran más de una palabra y por tanto estaban separados (“clara” y “huevo” → “clara de huevo”). Todo ese proceso se hizo en este [archivo](#).

Ya hecho el procesado de datos se exporta en formato CSV para que lo pueda importar neo4j. En este caso como se está utilizando Neo4j Aura es necesario alojarlo en un repositorio de GitHub para que neo4j acceda al CSV de manera remota.

Para la construcción del grafo se utilizó el lenguaje Cypher y cada una de las sentencias ejecutadas se encuentran en el archivo [export.cyp](#). Este es un ejemplo de una sentencia de Cypher para crear las recetas:

```
LOAD CSV WITH HEADERS FROM
'https://raw.githubusercontent.com/cdiezg07/SIBI-FOODRATE/main/nuevoooooMemoria.csv?token=ALKVWTVXB3AZZJCD72ECMBY2NBW' as row
MERGE (r:Receta {id: row.name, link: row.link, foto: row.fotografia, calorías: toFloat(row.calorías), carbohidratos: toFloat(row.carbohidratos), proteina:
toFloat(row.proteina), grasa: toFloat(row.grasa), fibra: toFloat(row.fibra), azúcares: toFloat(row.azúcares), colesterol: toFloat(row.colesterol)})
```

Figura 3.1. – Creación nodo Receta

En la creación se emplea el uso de MERGE en vez de CREATE ya que permite crear objetos asegurándose de que no haya repetidos.

Durante la construcción del grafo noté que se demoraba bastante al crear las relaciones de los ingredientes con las recetas, así que decidí introducir un index en el “id” de las recetas de la siguiente manera:

```
CREATE INDEX FOR (r:Receta) ON (r.id)
```

Figura 3.2. – Creación de index en el nodo Receta

Esto mejoró notablemente el rendimiento porque lo que hace el index es cambiar la forma de búsqueda de un objeto. Antes del index el objeto “receta” se buscaba mirando todas las propiedades de este, calorías, tiempo, dificultad, etc. En cambio, cuando se crea el index en la característica “id” solo buscará las recetas teniendo en cuenta el nombre de la receta e ignorando todo lo demás.

```
LOAD CSV WITH HEADERS
FROM "https://raw.githubusercontent.com/cdiezg07/SIBI-FOODRATE/main/nuevooooMemoria.csv?token=ALKVWTVXB3AZZJCD72ECMDBYZNBW" AS row
MERGE (r:Receta {id: row.name})
MERGE (i:Ingredientes {value: row.ingredientes})
CREATE (r)-[:CONTIENE_INGREDIENTES]->(i)
```

Figura 3.3. – Creación de relación entre el nodo Receta e Ingredientes

## ESTRUCTURA DEL GRAFO DE CONOCIMIENTO

Después de cargar todos los datos el tamaño del grafo ha quedado con 4229 nodos y 48966 relaciones.

El grafo de conocimiento está compuesto de 5 nodos y 5 relaciones entre ellos como se puede ver a continuación:

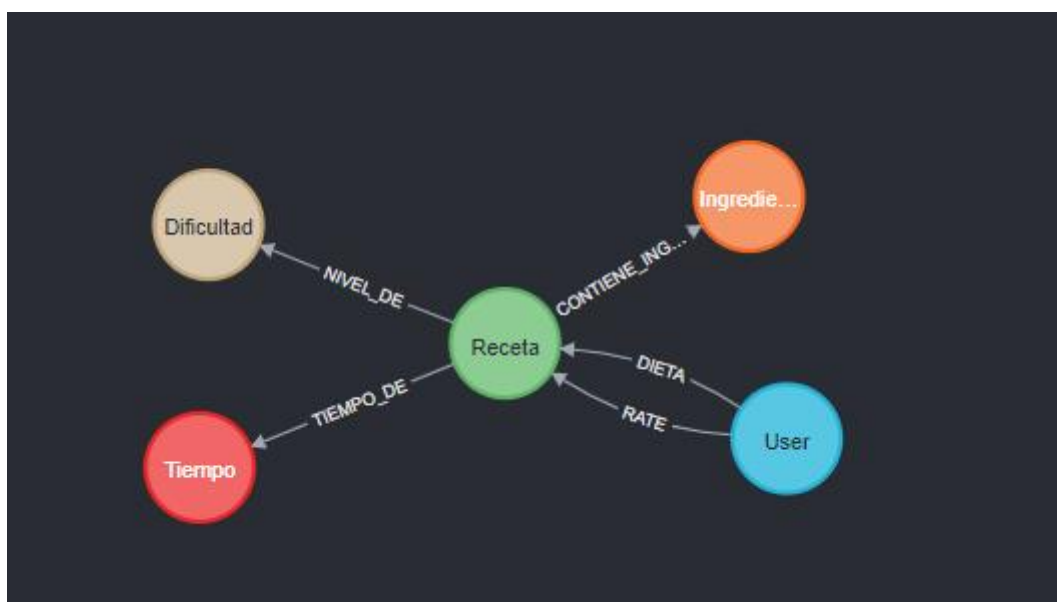


Figura 3.4. – Visualización grafo de conocimiento



Los nodos en el grafo están definidos de la siguiente manera:

- **Receta.** Es el nodo principal y por el cual entran y salen todas las relaciones. Las propiedades del nodo son: nombre, información nutricional (calorías, proteínas, carbohidratos, grasa, fibra, azúcares, colesterol, fotografía, link de la receta).
- **Dificultad.** Es la dificultad que puede tener cada receta (Baja, Media y Alta).
- **Ingrediente.** Nodo que presenta la información del nombre de los ingredientes (aceite, salmón, etc)
- **Tiempo.** Es el tiempo de preparación que tiene una receta y viene dado en número.
- **User.** Este nodo tiene 2 propiedades, una el usuario que es el identificador del nodo y otra la contraseña. Este nodo se encarga de almacenar todos los usuarios de la aplicación.

Las relaciones en el grafo están definidas de la siguiente manera:

- **CONTIENE\_INGREDIENTES.** Esta relación se encargará de unir cada receta con los ingredientes correspondientes. Como es lógico cada receta puede tener relaciones con bastantes ingredientes.
- **NIVEL\_DE.** Relaciona las recetas con su nivel de dificultad. Cada receta solo puede estar relacionada con una dificultad.
- **TIEMPO\_DE.** Relaciona las recetas con su tiempo de preparación. Cada receta solo puede estar relacionada con un nodo Tiempo.
- **RATE.** Consiste en relacionar el usuario con la receta. Además, la relación tiene una propiedad “star” que tendrá el valor que el usuario haya asignado a cada receta a la hora de la puntuación. Un usuario puede puntuar las recetas que le de la gana así que habrá tantas relaciones como el usuario decida.
- **DIETA.** Relaciona al usuario con las recetas que este haya consumido. Esta relación servirá para llevar un cálculo de las calorías, proteínas, carbohidratos, etc, que el usuario haya ingerido durante el día. Un usuario puede comer cualquier receta así que habrá tantas relaciones como recetas haya comido.

## 4. DESCRIPCIÓN DE LA APLICACIÓN

La aplicación es una aplicación web por lo que presenta la arquitectura cliente-servidor. Está constituida de 3 partes, servidor web, base de datos y servidor de la aplicación.

- **Servidor web (Frontend):** Es la parte de la aplicación que el usuario ve, se encarga de mostrar el diseño, la información y los botones además de detectar los clics de los usuarios y de enviar esta información al servidor de la aplicación (backend).
- **Servidor de la aplicación (backend):** El backend es el conjunto de servidores y programas que se encarga de almacenar y procesar la información de una aplicación. Además, el backend es responsable de la interacción entre la aplicación y otros servicios, como una base de datos. La conexión con la base de datos (neo4j) se realiza mediante un driver. Este driver permite abrir la conexión desde el backend para realizar consultas a la base de datos de una forma fácil. Una vez se cierra la conexión con neo4j y se ha procesado la solicitud se devuelve una respuesta al servidor web para que el cliente pueda visualizar los cambios.
- **Base de datos:** Es la parte encargada de guardar toda la información haciéndola persistente. El objetivo es acceder a los datos de una manera sencilla ya sea para insertar nuevos datos, borrar algunos registros o actualizarlos.

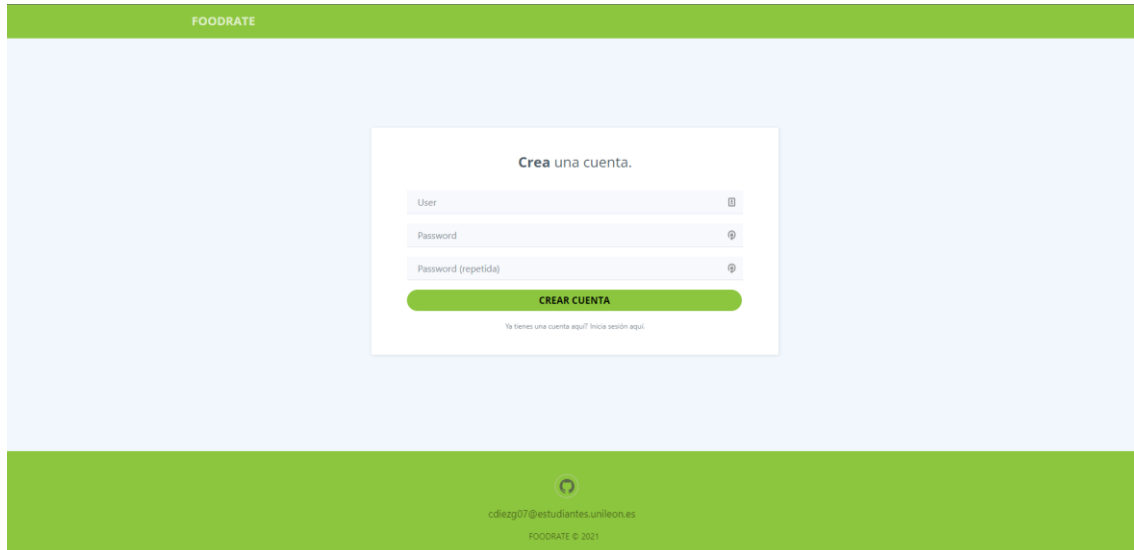
Ahora que ya se sabe cómo es la estructura de la aplicación y como interactúan cada parte entre ellas, se va a explicar el funcionamiento a nivel de usuario.

La primera página que se va a encontrar el usuario será la pantalla principal. En la parte de arriba existen 3 enlaces, el primero llamado FOODRATE (nombre de la aplicación) siempre enlazará con la página principal. Por otro lado, están los enlaces de inicio de sesión y registro para que el usuario pueda iniciar sesión si ya tiene cuenta o registrarse si todavía no se ha creado una cuenta en la plataforma respectivamente.



Figura 4.1 Pantalla principal

En la pantalla de registro el usuario tendrá que introducir su usuario que tendrá que ser único y la contraseña 2 veces para que verifique que se ha introducido bien. El nombre de usuario introducido tiene que ser único por lo que no puede haber 2 usuarios con el mismo nombre. Debajo del botón “CREAR CUENTA” se tiene la opción directamente de ir a la pantalla de inicio de sesión después de haberse creado la cuenta.



*Figura 4.2 Pantalla de registro*

Una vez creada la cuenta se irá a la ventana de inicio de sesión donde se introducirán el usuario y la contraseña puestos en el registro. Si el inicio de sesión ha sido exitoso le aparecerá la pantalla de búsqueda de recetas. Además de iniciar sesión también hay un enlace en el que el usuario puede ir a la página de registro por si no está registrado.



*Figura 4.3 Pantalla de inicio de sesión*

En la pantalla de búsqueda de recetas aparece en la cabecera el nombre del usuario que ha iniciado sesión junto con su foto de perfil. La foto de perfil es un enlace el cual lleva

a la ventana de perfil del usuario. También existe otro enlace llamado “MIS GUSTOS” que redirigirá al usuario a otra página donde podrá visualizar las recetas que ha puntuado.

En el cuerpo de la página se encuentran las características por las que se puede buscar una receta de cocina. Las recetas pueden filtrarse por tiempo de preparación, dificultad y la información nutricional que uno quiera, además de que la búsqueda puede ser por nombre o ingrediente. Todos estos filtros puestos a la vez hacen que la búsqueda de las recetas sea muy precisa y adecuándose a lo que el usuario desea. Luego está el sistema de recomendación que en este caso para ejecutarlo solo hace falta hacer click a uno de los botones dependiendo de que algoritmo quiera utilizar el usuario.



Figura 4.4 Pantalla de inicio de sesión

A continuación, se visualizan las recetas las cuales tienen las características de dificultad y tiempo de preparación. También hay un checkbox de comida diaria el cual si el usuario lo activa indicará que esa receta la ha comido y los datos de información nutricional se actualizarán en su perfil para que sepa cuantas calorías ha ingerido en el día. Si el usuario quiere quitar una receta por cualquier razón tan solo tendrá que desmarcar el checkbox y automáticamente se actualizará en la base de datos. Para evaluar una receta el usuario solo tiene que elegir entre 1 y 5 estrellas y en ese mismo momento de realizar una puntuación a la receta ese valor se actualizará en la base de datos. Si se da el caso de que el usuario quiere cambiar de puntuación tan solo tendrá que cambiar la selección de las estrellas y se actualizará automáticamente en la base de datos.



Figura 4.5 Ejemplo de receta

Cuando el usuario pincha en el enlace “MIS GUSTOS” de la pagina de buscador de recetas le lleva a una página donde se ven las recetas que ha valorado.

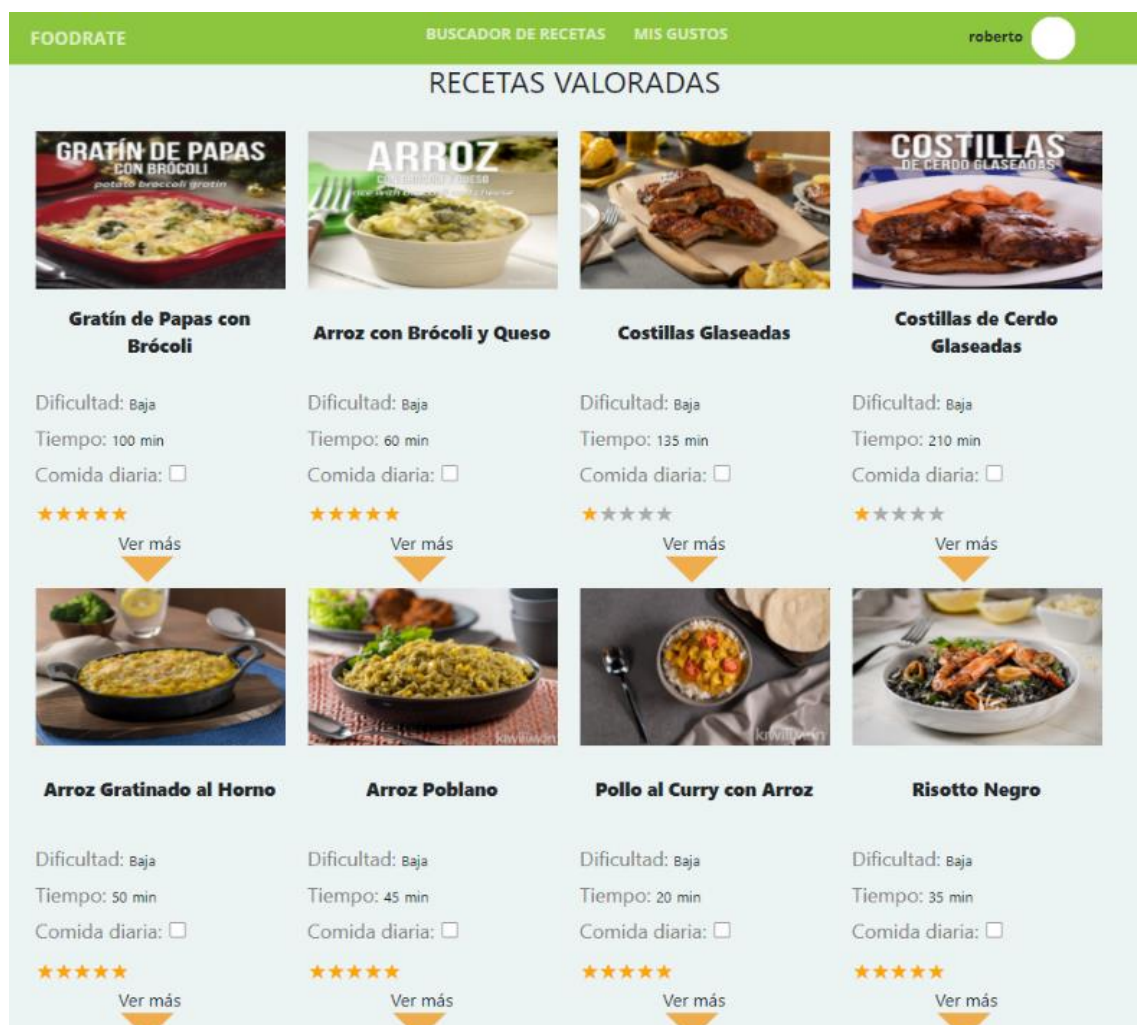


Figura 4.6 Página de recetas valoradas

En la cabecera se encuentra el enlace “BUSCADOR DE RECETAS” que sirve para retroceder y volver a la página de búsqueda. En el cuerpo de la página se pueden ver todas las recetas que el usuario puntuó y además de que puntuación le dio a cada una.

La última opción que queda por cubrir es la posibilidad que tiene el usuario de ver su perfil. Para acceder a esta página lo único que tiene que hacer es click en la foto de perfil y directamente se le llevará al perfil.



Figura 4.7 Página perfil de usuario

En esta ventana se tiene toda la información nutricional que el usuario haya ingerido. Con una barra de porcentaje que si supera el 100% quiere decir que se ha pasado del límite recomendable para un usuario promedio. Los valores promedios se han puesto según lo que recomienda la OMS (Organización Mundial de la Salud).



## 5. EXPLICACIÓN DEL ALGORITMO

En el trabajo se utilizan 3 algoritmos distintos para realizar las recomendaciones de recetas de cocina al usuario.

Los algoritmos que se van a explicar a continuación van a ser basado en contenido, filtrado colaborativo e híbrido que es la mezcla de los 2 algoritmos anteriores.

### 5.1. ALGORITMO BASADO EN CONTENIDO

El sistema de recomendación basado en contenido intenta adivinar las características o el comportamiento de un usuario dadas las características del elemento, al que reacciona positivamente.

El algoritmo presenta 3 partes bien diferenciadas:

- **Extracción de características y procesamiento**

Se van a obtener todas las recetas junto con sus características las cuales serán el tiempo de preparación de la receta, ingredientes, dificultad y calorías.

En la característica del tiempo de preparación como los datos que había eran números lo que se hizo fue transformarla en 3 clases, “TiempoBajo”, “TiempoMedio” o “TiempoAlto” dependiendo del valor que tenga y según mis propios parámetros que he establecido:

- Menos de 25 minutos “TiempoBajo”.
- Entre 25 y 60 minutos “TiempoMedio”.
- Más de 60 minutos “TiempoAlto”.

Por otro lado, la característica de calorías funciona de la siguiente manera. Un usuario anota las recetas que ha comido durante el día por lo tanto las calorías van subiendo. Entonces cuanto más cerca esté de cantidad recomendable de calorías la columna de calorías tendrá más valor poniendo más valor a recetas que contengan menos calorías para no sobrepasar el límite diario. Esta columna no tendrá en cuenta la puntuación del usuario en la valoración de la receta.

Por cada receta se generará un vector que tendrá tantas columnas como características distintas haya en el dataset. Así que habrá 3 columnas para el tiempo de preparación (“TiempoBajo”, “TiempoMedio” o “TiempoAlto”), 3 columnas para la dificultad (“Baja”, “Media”, “Alta”), 253 para los diferentes ingredientes y 1 columna para las calorías, en total  $3 + 3 + 253 + 1 = 260$  características. Además, tendrá una columna más llamada “Valoración” que será la puntuación que haya puesto el usuario a la receta.

Por tanto, tendremos una matriz en la que las columnas son las características y las filas las recetas. Las características que no estén en las recetas serán rellenadas con un 0 y si

están presentes habrá distintas opciones dependiendo el tipo de característica:

- En las características de Tiempo y Dificultad si están presentes en la receta se rellenará con un 1.
- En las columnas de los ingredientes si están presentes en las recetas se rellenará con un 3, a excepción de algunos ingredientes como pueden ser las especias (“laurel”, “comino”, etc) las cuales tendrán un valor de 0 debido a que no aportan un sabor a la receta y pueden ser reemplazados por cualquier otro tipo en función de lo que le guste al usuario. Esto hará que el algoritmo mejore ya que sabrá generalizar mejor.
- En las calorías se pondrá un 1,2 si la receta no supera en calorías a las calorías recomendadas del usuario, en caso contrario se le aplicará un valor de 0,8.
- La columna “Valoración” puede estar entre el rango [1, 5] dependiendo de cómo haya valorado el usuario la receta. Sino se valora estará por defecto a 0.

Este algoritmo le va a dar mucha más importancia a las recetas que tengan los ingredientes más parecidos. Esto ocurre debido a que el tiempo y la dificultad solo se les da el valor de 1, frente a los muchos ingredientes que tiene una receta por tanto hace que esas 2 características sean relevantes en caso de empate.

En las columnas de ingredientes y calorías se ha puesto un 3 debido a que normalmente a la hora de elegir una receta casi siempre se tiene en cuenta los ingredientes que lleva más que el tiempo y la dificultad. La razón por la que en la columna de calorías también se puso un 3 es porque la aplicación da mucha importancia a comer sano y no sobrepasar el límite de calorías diario.

- **Creación del perfil de usuario**

Una vez que se encuentra la matriz rellena con todas las premisas del apartado anterior ahora toca multiplicar cada característica por la valoración que haya dado el usuario. Si por ejemplo se ha evaluado una receta con un 3, todas las características se multiplicarán por 3. Después se tendrá que calcular la media de todas las características si contar los valores tienen 0. Es decir, se ira recorriendo cada columna y a la vez calculando su media para posteriormente generar un vector con las mismas dimensiones que los de las recetas (1x261) y se añadirá al final de la matriz. Después se aplicará una normalización a cada característica del vector del perfil haciendo que las columnas que se encuentran en el rango [0, 5] pasen a [0, 1] y las que están en [0, 15] pasen a [0, 3].

Por último, habría que multiplicar cada característica por su frecuencia de aparición. Esto se hace para evitar que solo una característica que se valore de manera muy positiva tenga una gran repercusión en el algoritmo debido a que no se trata de una



generalidad. Por ejemplo, si el usuario ha valorado 5 recetas y hay una característica con un 4 habría que multiplicar esa característica por la frecuencia de aparición que si en este caso solo aparece una vez quedaría como  $(1/5) \cdot 4 = 0,8$  disminuyendo sustancialmente su valor.

- **Filtrado y recomendación**

En este último paso utilizando la formula del coseno se calculará el valor de similitud entre el perfil del usuario y las distintas recetas.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

Figura 1.1 Función similitud coseno

Una vez se tienen todos los valores calculados, se ordenan de mayor a menor y se mostrarán las 20 mejores recetas.

Por si no ha quedado claro se pondrá un ejemplo práctico con 4 recetas 2 valoradas por el usuario y 2 sin valorar. También se disminuirán los ingredientes para que pueda entrar en el Excel.

- **Extracción de características y procesamiento**

Se han quitado varios ingredientes de las recetas para que pudiera entrar en el ejemplo. Se supone que el usuario ha comido unas 1850 kcal y por tanto le quedan 150 kcal para no sobrepasar el límite diario recomendado.

	Baja	Media	Alta	TiempoBajo	TiempoMedio	TiempoAlto	mantequilla	mostaza	calabaza	garbanzo	chile	limon	cebolla	harina	Calorias	valoracion	
Costillas glaseadas	1	0	0	0	0	1	3	3	0	0	0	0	0	0	0	5	222 kcal
Caldo Tlalpeño	1	0	0	0	1	0	0	0	3	3	3	0	0	0	0	4	76 kcal
Ensalada de Pollo	1	0	0	1	0	0	0	0	0	0	0	3	3	0	1,2		110 kcal
Pastel de Zanahoria	1	0	0	0	1	0	3	0	0	0	0	3	0	3	0,8		313 kcal

Figura 5.1 Matriz de características

Como se puede ver, en las columnas de los ingredientes se pondrá un 3, en las de tiempo y dificultad un 1 y la columna de calorías dependerá de cuanto le quede para sobrepasar el límite de calorías. Por ejemplo, la ensalada de pollo como tiene 110 kcal y no supera las 150 kcal restantes se le pondrá un 1,2 incentivando así su consumición. En cambio, el pastel de zanahoria que presenta 313 kcal superaría el límite diario de calorías del usuario por tanto se introducirá un 0,8 para reducir la probabilidad de ser recomendado.

- **Creación del perfil de usuario**

Una vez se ha completado la matriz, en este paso solo se necesitan las filas valoradas por el usuario que en este ejemplo son las 2 primeras.

	Baja	Media	Alta	TiempoBajo	TiempoMedio	TiempoAlto	mantequilla	mostaza	calabaza	garbanzo	chile	limon	cebolla	harina	Calorias	valoracion
Costillas glaseadas	1	0	0	0	0	1	3	3	0	0	0	0	0	0	0	5
Caldo Tlalpeño	1	0	0	0	1	0	0	0	3	3	3	0	0	0	0	4

Figura 5.2 Matriz de recetas con características

Después de elegir las dos primeras filas se multiplica cada característica por la columna valoración del usuario, quedando de la siguiente manera.

	Baja	Media	Alta	TiempoBajo	TiempoMedio	TiempoAlto	mantequilla	mostaza	calabaza	garbanzo	chile	limon	cebolla	harina	Calorias	valoracion
Costillas glaseadas	5	0	0	0	0	5	15	15	0	0	0	0	0	0	0	5
Caldo Tlalpeño	4	0	0	0	4	0	0	0	12	12	12	0	0	0	0	4

Figura 5.3 Matriz de recetas con características

A continuación, se hará la media aritmética de las características sin contar los valores que son 0 y se normalizarán los valores dividiéndolos entre 5 que es la máxima valoración que puede dar el usuario. Por último, se multiplicará el valor resultante de cada columna por la frecuencia de aparición.

	Baja	Media	Alta	TiempoBajo	TiempoMedio	TiempoAlto	mantequilla	mostaza	calabaza	garbanzo	chile	limon	cebolla	harina	Calorias	valoracion
Perfil	0,9	0	0	0	0,4	0,5	1,5	1,5	1,2	1,2	1,2	0	0	0	0	No hay

Figura 5.4 Perfil del usuario.

- **Filtrado y recomendación**

Para hacer la predicción de las 2 ultimas recetas se utilizará la fórmula de similitud del coseno.

	Baja	Media	Alta	TiempoBajo	TiempoMedio	TiempoAlto	mantequilla	mostaza	calabaza	garbanzo	chile	limon	cebolla	harina	Calorias	valoracion
Perfil	0,9	0	0	0	0,4	0,5	1,5	1,5	1,2	1,2	1,2	0	0	0	No hay	
Ensalada de Pollo	1	0	0	1	0	0	0	0	0	0	0	3	3	0	1,2	0,1413477
Pastel de Zanahoria	1	0	0	0	1	0	3	3	3	3	3	0	0	0	0,8	1,6839164

Figura 5.5 Predicción de nuevas recetas

El resultado que nos reporta está entre [0, 3,6] por tanto habrá que normalizarlo dividiéndolo entre 3,6 para que se quede entre [0, 1].

valoracion	Normalizado
0,1413477	0,039263255
0,9841989	0,273388587

Figura 5.6 Salida normalizada

Como se puede ver el pastel de zanahoria sale mejor recomendado a pesar de sobrepasar las calorías recomendadas debido a que tiene 2 ingredientes que de acuerdo con su perfil están bien valorados y la ensalada de pollo no tiene ningún ingrediente en común.

## 5.2. ALGORITMO DE FILTRADO COLABORATIVO

El algoritmo de filtrado colaborativo se divide en 4 partes, de la misma manera que el algoritmo anterior.

- **Construcción de matriz de usuarios x recetas**

Primeramente, se tiene que hacer una solicitud a la base de datos para que nos devuelva todos los usuarios de la aplicación y otra solicitud para obtener todas las recetas. Con estos datos ya se pueden construir las columnas en las irían las recetas y en las filas los usuarios, quedando una matriz de usuarios x recetas.

Una vez se tiene la estructura de la matriz se tiene que hacer otra consulta que involucre todos los usuarios de la aplicación que hayan valorado alguna receta, obteniendo así dicha puntuación para posteriormente colocarlo en la matriz. Las recetas que no hayan sido puntuadas por el usuario se les pondrá un 0 en la matriz.

Entonces para obtener la valoración que haya hecho una persona se localizará en la posición (i, j) teniendo en cuenta que i es para las filas y j para las columnas.

El tipo de matriz que casi siempre se generará será una enorme matriz en la que la mayor parte de sus elementos son cero, a esto se le llama matriz dispersa.

- **Cálculo de similitud entre usuarios.**

Para calcular que personas son más similares al usuario al cual se le quiere aplicar la recomendación se puede utilizar diferentes métricas (MSD, la correlación de Pearson o el coseno). En mi caso utilizaré la correlación de Pearson que es la siguiente:

$$sim(x, y) = \frac{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x) \cdot (r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x)^2 \cdot \sum_{i \in B_{x,y}} (r_{y,i} - \bar{r}_y)^2}} \in [-1, 1]$$

Figura 5.7 Correlación de Pearson

La variable  $B_{x,y}$  son todos los items comunes entre los 2 usuarios. La  $r_{x,i}$  e  $r_{y,i}$  son las valoraciones que los 2 usuarios han dado a las recetas y la  $\bar{r}_x$  es la media de votos del usuario.

Al aplicar la correlación de Pearson a todos los usuarios comparándolo con el usuario activo, se obtendrá una lista con los valores de similitud entre  $[-1, 1]$ .

- **Predicción de items**

En la aplicación se tendrán en cuenta los 5 usuarios con mayor similitud respecto al usuario actual descartando los que hayan dado un -1. Si da -1 quiere decir que no hay ninguna receta en común y por consecuencia no se podrá hacer ninguna recomendación.

Para hacer la recomendación hay que normalizar la salida del algoritmo de Pearson a [0, 1] para que el resultado no sea mayor 5. La normalización se hará de la siguiente manera:

$$\begin{aligned} OldRange &= (OldMax - OldMin) \\ NewRange &= (NewMax - NewMin) \\ NewValue &= (((OldValue - OldMin) * NewRange) / OldRange) + NewMin \end{aligned}$$

Después de haber elegido los usuarios con la mayor similitud, las recomendaciones de las recetas se calcularán si existen 2 o más usuarios que hayan puntuado esa receta en concreto. Es decir, si se le va a recomendar una receta a un usuario, pero solo una persona ha evaluado esa receta, la recomendación no se producirá. Esto hace que sea mucho más fiable la predicción debido a que se tienen más valoraciones de la receta.

La fórmula para hacer la predicción será la media ponderada de las recetas valoradas por los usuarios multiplicado por el resultado del cálculo de la similitud respecto al usuario actual (Correlación de Pearson) y dividido por la suma de resultados de la correlación de Pearson.

$$p_{u,i} = \frac{\sum_{n \in G_{u,i}} sim(u, n) \cdot r_{n,i}}{\sum_{n \in G_{u,i}} sim(u, n)}$$

Figura 5.7 Media ponderada

Por último, después de aplicar la media ponderada los resultados que aparecerán estarán entre el rango de [1,5] y las recetas se mostrarán en la aplicación de mayor a menor puntuación.

Para que este algoritmo funcione el usuario previamente ha tenido que evaluar al menos 5 recetas.

He hecho en Excel un ejemplo para que se entienda de una manera más fácil cómo funciona el algoritmo colaborativo.

- **Construcción de matriz de usuarios x valoraciones**

El ejemplo consistirá en 4 usuarios (Carlos, Fede, Pablo e Iván) los cuales han valorado una serie de recetas y en total hay 11 recetas distintas. Al usuario al que se le aplicará la recomendación será a Iván.

	Costillas glaseadas	Caldo Tlapeño	Ensalada de pollo	Pastel de zanahoria	Pizza al pastor	Mini buñuelos	Gelatina	Souffle	Ensalada de atún	Pepinos rellenos	Crema de quesos
carlos	5	4	3	2	3	0	0	0	0	0	1
fede	5	0	0	0	5	4	5	4	0	0	0
pablo	2	0	0	2	5	5	0	0	5	5	0
iván	0	0	0	0	4	4	0	3	4	0	5

Figura 5.8 Matriz usuarios x valoraciones

- **Cálculo de similitud entre usuarios.**

Para calcular la similitud entre Iván y todos los demás usuarios estuve dudando entre utilizar la fórmula MSD (Minimum Square Difference) o la correlación de Pearson. Al final me decanté con la correlación de Pearson por la siguiente razón:

$$sim(x, y) = 1 - \frac{1}{\#B_{x,y}} \sum_{i \in I_u} \left( \frac{r_{x,i} - r_{y,i}}{max - min} \right)^2 \in [0, 1]$$

Figura 5.9 MSD (Minimum Square Difference)

$$sim(x, y) = \frac{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x) \cdot (r_{y,i} - \bar{r}_y)}{\sqrt{\sum_{i \in B_{x,y}} (r_{x,i} - \bar{r}_x)^2 \cdot \sum_{i \in B_{x,y}} (r_{y,i} - \bar{r}_y)^2}} \in [-1, 1]$$

Figura 5.10 Correlación de Pearson

	Costillas glaseadas	Caldo Tlapeño	Ensalada de pollo	Pastel de zanahoria	Pizza al pastor	Mini buñuelos	Gelatina	Souffle	Ensalada de atun	Pepinos rellenos	Crema de quesos	Pearson	MSD	Pearson
carlos	5	4	3	2	3	0	0	0	0	0	1	0	0,469	0
fedede	5	0	0	0	5	4	5	4	0	0	0	0,36	0,833	0,36
pablo	2	0	0	2	5	5	0	0	5	5	0	1	0,889	1
iván	0	0	0	0	4	4	0	3	4	0	3			

Figura 5.11 Calculo de similitudes usuarios

Si hay un usuario que tiene unas valoraciones con las mismas diferencias de puntuación entre los 2 usuarios la correlación de Pearson dará 1 a pesar de que sean distintos. En el Excel está el caso de Pablo e Iván los cuales tienen la misma diferencia de votos y por tanto Pearson da 1 y el MSD 0,889. El resultado de Pearson podría resultar un poco ilógico ya que como no son iguales el resultado no debería ser 1, pero Pearson lo trata más el comportamiento de las personas y en este ejemplo estas 2 personas se comportan de la misma manera ante las mismas recetas por tanto el resultado es 1.

Al aplicarla nos queda que Pablo y Fede tienen unos gustos muy similares a Iván, en cambio, Carlos no se parece en nada. Como se puede ver cuanto más cerca de 1 sea el resultado más se parecerá al usuario actual. Por tanto, ahora que ya se saben cuáles son los usuarios más similares, es el momento de hacer la predicción de nuevas recetas.

- **Predicción de ítems**

Como vimos en el apartado anterior Pablo y Fede son los usuarios más similares a Iván así que se le aplicará la fórmula de predicción a estos 2 usuarios.

Para hacer la predicción de recetas opté por un método el cual te recomienda menos recetas, pero la recomendación es más robusta ya que hay más usuarios que respaldan la valoración de la receta. Aplicando esta estrategia a este ejemplo, vemos que solo se va a poder valorar una receta porque las demás recetas solo están valoradas por un usuario de los 2 posibles.

Antes de hacer de aplicar la fórmula de predicción es necesario normalizar el resultado dado por la correlación de Pearson.

$$OldRange = (1 - (-1))$$

$$NewRange = (1 - 0)$$

$$NewValue = (((OldValue - (-1)) * NewRange) / OldRange) + 0$$

La fórmula para hacer la predicción será la media ponderada:

$$p_{u,i} = \frac{\sum_{n \in G_{u,i}} sim(u, n) \cdot r_{n,i}}{\sum_{n \in G_{u,i}} sim(u, n)}$$

Figura 5.12 Media ponderada

	Costillas glaseadas	Caldo Tlapeño	Ensalada de pollo	Pastel de zanahoria	Pizza al pastor	Mini buñuelos	Gelatina	Souffle	Ensalada de atun	Pepinos rellenos	Crema de quesos	Pearson	MSD	Normalización
carlos	5	4	3	2	3	0	0	0	0	0	1	0	0,469	0,5
fedé	5	0	0	0	5	4	5	4	0	0	0	0,36	0,833	0,68
pablo	2	0	0	2	5	5	0	0	5	5	0	1	0,889	1
iván	3,215327508	0	0	0	4	4	0	3	4	0	3			

Figura 5.13 Recomendación costillas glaseadas

Los resultados que dará la media ponderada siempre estarán entre [1,5].

### 5.3. ALGORITMO HÍBRIDO

Este algoritmo consiste en la unión de los 2 algoritmos anteriores (algoritmo basado en contenido y de filtrado colaborativo) para conseguir una recomendación más precisa.

Existen distintos tipos de hibridación ponderación, conmutación, en cascada, multinivel, etc. En mi caso utilizaré la hibridación por ponderación.

La hibridación por ponderación utiliza las salidas de los 2 algoritmos (colaborativo y contenido) y mediante una suma ponderada se obtiene un resultado que será el grado de recomendación para el usuario. Este algoritmo se divide en 2 partes:

- **Obtención de resultados de ambos algoritmos**

Se obtienen los resultados de los 2 algoritmos y la salida del algoritmo colaborativo se normaliza dividiéndola entre 5 de manera que el resultado esté entre [0, 1]. Con esto ya tenemos el mismo rango de salida que en el algoritmo de contenido.

- **Predicción de items**

Una vez tenemos las 2 salidas con los mismos rangos, realizamos una matriz en la cual la primera columna sea el nombre de la receta, la segunda columna la salida del algoritmo basado en contenido y la tercera la salida del colaborativo. Se computa la suma ponderada y se colocan los resultados en una cuarta columna.

$$\hat{r}_{uj} = \sum_{i=1}^q \alpha_i \hat{r}_{uj}^i$$

Figura 5.14 Suma ponderada

En mi caso he asignado el 80% de peso al algoritmo basado en contenido y 20% al colaborativo debido a que los datos de los usuarios en los que se basa el colaborativo no son reales. Habrá algunos casos en los que solo haya recetas recomendadas por un solo algoritmo así que el algoritmo que no tenga la receta se le pondrá un 0 en la columna correspondiente.

Veamos un ejemplo para poder entender mejor cómo funciona este algoritmo.

- **Obtención de resultados de ambos algoritmos**

Primeramente, se obtiene el resultado dado por el algoritmo basado en contenido entre [0,1].

Recetas	Alg. Basado en Contenido
Caldo Tlapeño	0,888
Ensalada de pollo	0,708
Pastel de zanahoria	1
Pizza al pastor	0,44
Mini buñuelos	0,32

Figura 5.15 Resultado algoritmo basado en contenido

Por otro lado, se obtienen los resultados del algoritmo de filtrado colaborativo que se encuentran entre [1,5] así que hay que hacerles una normalización dividiéndolos entre 5 para que coincidan con el algoritmo de contenido.

Recetas	Alg. Filtrado colaborativo	Normalización
Caldo Tlapeño	3,6	0,72
Ensalada de pollo	4,5	0,9
Pastel de zanahoria	4,6	0,92

Figura 5.16 Resultado alg. Filtrado colaborativo

Una vez se tienen los resultados de ambos algoritmos estos se unen en una matriz conjunta. Si hay alguna receta la cual está evaluado por un algoritmo, pero no por otro entonces se pondrá un 0 en la casilla del algoritmo que no haya evaluado la receta. Esto se hace para que se puedan hacer la suma ponderada sin que de error.

- **Predicción de items**

Como ya se ha construido la matriz en la que están los dos algoritmos es el momento de hacer la predicción. La predicción se hará mediante la fórmula de suma ponderada:

$$\hat{r}_{uj} = \sum_{i=1}^q \alpha_i \hat{r}_{uj}^i$$

Figura 5.17 Suma ponderada

Después de calcular el resultado como en los anteriores algoritmos se ordena de mayor a menor y se muestra al usuario los que hayan obtenido un mayor resultado.

Recetas	Alg. Basado en Contenido	Alg. Filtrado colaborativo	Resultado
Caldo Tlapeño	0,888	0,72	0,8544
Ensalada de pollo	0,708	0,9	0,7464
Pastel de zanahoria	1	0,92	0,984
Pizza al pastor	0,44	0	0,352
Mini buñuelos	0,32	0	0,256

Figura 5.18 Resultado después de la suma ponderada



## 6. ANALISIS DE RESULTADOS

Para el análisis de los resultados dividiré este apartado en 3, uno para cada tipo de algoritmo.

### 6.1.ALGORITMO BASADO EN CONTENIDO

Para ver cuál es el rendimiento de este algoritmo he creado una serie de pruebas viendo así sus puntos fuertes y débiles.

Primeramente, se empieza con un usuario que no tiene anotada ninguna comida por tanto se deja a un lado la columna de calorías ya que no será relevante por ahora. En este ejemplo el usuario va a valorar de forma positiva recetas que contengan el ingrediente arroz porque le gusta mucho.

#### 1º Ejemplo

### RECETAS VALORADAS

 <p><b>Arroz Gratinado al Horno</b></p> <p>Dificultad: Baja              Tiempo: 50 min              Comida diaria: <input type="checkbox"/></p> <p>★★★★★</p> <p>Ver más</p>	 <p><b>Arroz Poblano</b></p> <p>Dificultad: Baja              Tiempo: 45 min              Comida diaria: <input type="checkbox"/></p> <p>★★★★★</p> <p>Ver más</p>	 <p><b>Bowl de Salmón Teriyaki</b></p> <p>Dificultad: Baja              Tiempo: 65 min              Comida diaria: <input type="checkbox"/></p> <p>★★★★★</p> <p>Ver más</p>
--	---	--

*Figura 6.1 Recetas evaluadas que contienen*

Indirectamente también ha elegido las 3 recetas con dificultad baja y en cuanto al tiempo de preparación han sido 2 recetas de “TiempoMedio” y una de “TiempoAlto” superando los 60 minutos.

Entonces con estas recetas valoradas por el usuario se utiliza el algoritmo basado en contenido para que le haga una recomendación al usuario.

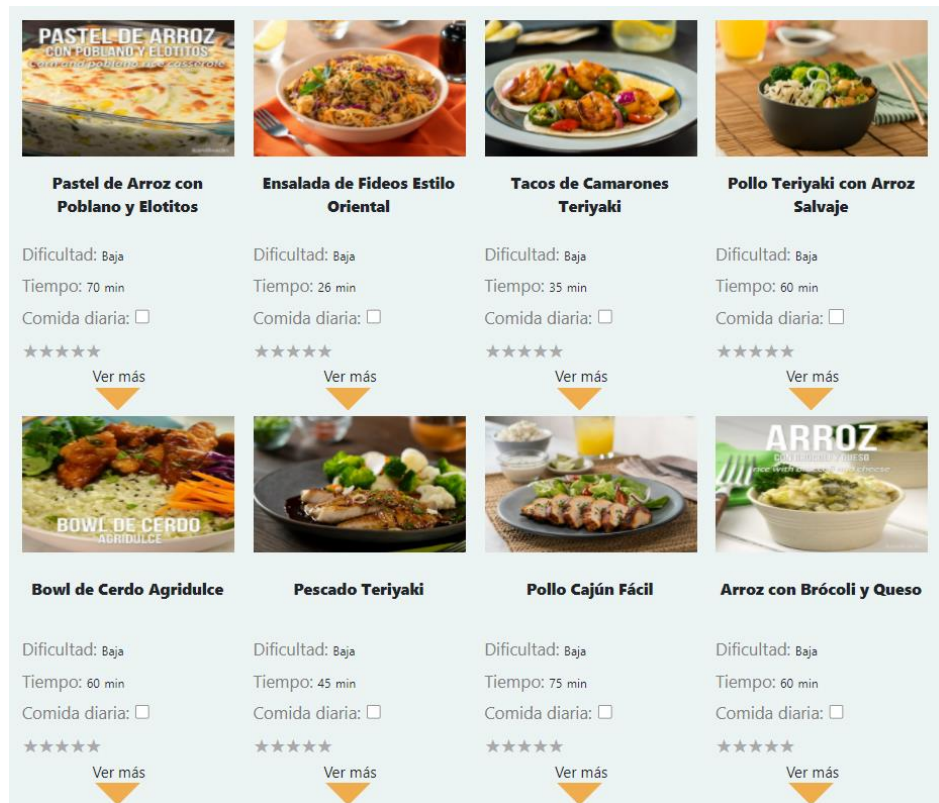


Figura 6.2 Recetas recomendadas

En las recetas recomendadas se puede ver como todas las recetas son de dificultad baja, el tiempo de preparación de la mayoría de las recetas son de “tiempoMedio” y todas las recetas contienen el ingrediente arroz.

## 2º Ejemplo

Por ejemplo, otra comida que me gusta mucho son las costillas por tanto voy a elegir unas cuantas recetas que vayan sobre costillas.

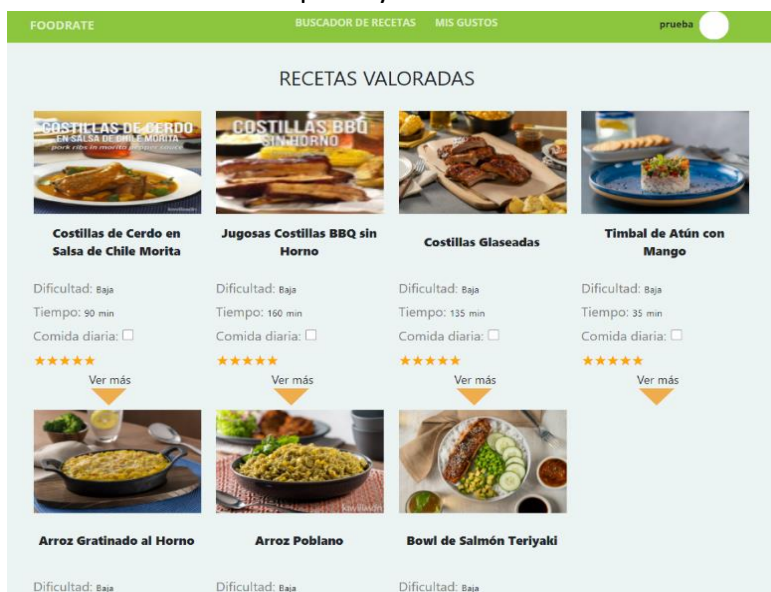


Figura 6.3 Recetas valoradas

Ahora se tienen 3 recetas valorando las costillas y 4 recetas con 5 estrellas de puntuación las que contienen arroz, al ejecutar el algoritmo pasa lo siguiente.









 <p><b>Bowl de Cerdo Agridulce</b></p> <p>Dificultad: Baja Tiempo: 60 min Comida diaria: <input type="checkbox"/> ★★★★★ <a href="#">Ver más</a></p>	 <p><b>Tacos de Camarones Teriyaki</b></p> <p>Dificultad: Baja Tiempo: 35 min Comida diaria: <input type="checkbox"/> ★★★★★ <a href="#">Ver más</a></p>	 <p><b>Pollo Glaseado con Miel y Chipotle</b></p> <p>Dificultad: Baja Tiempo: 40 min Comida diaria: <input type="checkbox"/> ★★★★★ <a href="#">Ver más</a></p>	 <p><b>Pollo a la Naranja Glaseado</b></p> <p>Dificultad: Baja Tiempo: 45 min Comida diaria: <input type="checkbox"/> ★★★★★ <a href="#">Ver más</a></p>
 <p><b>Pollo Cajún Fácil</b></p> <p>Dificultad: Baja Tiempo: 75 min Comida diaria: <input type="checkbox"/> ★★★★★</p>	 <p><b>Costillitas de Cerdo Picosas</b></p> <p>Dificultad: Baja Tiempo: 75 min Comida diaria: <input type="checkbox"/> ★★★★★</p>	 <p><b>Pollo Teriyaki con Arroz Salvaje</b></p> <p>Dificultad: Baja Tiempo: 60 min Comida diaria: <input type="checkbox"/> ★★★★★</p>	 <p><b>Pescado Teriyaki</b></p> <p>Dificultad: Baja Tiempo: 45 min Comida diaria: <input type="checkbox"/> ★★★★★</p>

Figura 6.4 Recomendación de recetas

En las recetas recomendadas se puede visualizar que aparecen las mismas recetas o parecidas que en la primera prueba recomendando comida que contenga arroz. Con la salvedad de que ahora también hay una receta de costillas.

Una de las partes negativas de este algoritmo es que se enfoca mucho en lo que valora positivamente una persona, pero ignora lo que valora negativamente (sobre todo al principio). Se va a poner un ejemplo para poder entender esto de una manera fácil.

### 3º Ejemplo

En el primer ejemplo nos recomendó una receta con brócoli, pero a mí eso no me gusta, por tanto, voy a valorar algunas recetas con brócoli negativamente.



### RECETAS VALORADAS

<b>Papilla de Pescado con Brócoli y Papa</b>	<b>Papilla de Pollo y Brócoli</b>	<b>Arroz Gratinado al Horno</b>	<b>Arroz Poblano</b>
Dificultad: Baja	Dificultad: Baja	Dificultad: Baja	Dificultad: Baja
Tiempo: 25 min	Tiempo: 20 min	Tiempo: 50 min	Tiempo: 45 min
Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>
★★★★★	★★★★★	★★★★★	★★★★★
<a href="#">Ver más</a>	<a href="#">Ver más</a>	<a href="#">Ver más</a>	<a href="#">Ver más</a>

Figura 6.5 Recetas valoradas

<b>Pastel de Arroz con Poblano y Elotitos</b>	<b>Arroz con Brócoli y Queso</b>	<b>Pollo Teriyaki con Arroz Salvaje</b>	<b>Ensalada de Fideos Estilo Oriental</b>
Dificultad: Baja	Dificultad: Baja	Dificultad: Baja	Dificultad: Baja
Tiempo: 70 min	Tiempo: 60 min	Tiempo: 60 min	Tiempo: 26 min
Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>
★★★★★	★★★★★	★★★★★	★★★★★
<a href="#">Ver más</a>	<a href="#">Ver más</a>	<a href="#">Ver más</a>	<a href="#">Ver más</a>
<b>Pollo Cajún Fácil</b>	<b>Arroz con Pollo</b>	<b>Delicioso Arroz al Chipotle</b>	<b>Pollo Agridulce a la Naranja</b>
Dificultad: Baja	Dificultad: Baja	Dificultad: Baja	Dificultad: Baja
Tiempo: 75 min	Tiempo: 40 min	Tiempo: 60 min	Tiempo: 65 min

Figura 6.6 Recetas recomendadas

¿Cómo es posible que ahora el arroz con brócoli este recomendado más arriba? Bien pues esto ocurre por lo siguiente. La receta de arroz con brócoli en la primera recomendación se colocó en octava posición, eso quiere decir que, aunque el brócoli no estuviese valorado había muchos ingredientes de esa receta que si lo estaban y por tanto se recomendó. Después al valorar negativamente recetas con brócoli lo que provoca es que el perfil del usuario que antes tenía un 0 se rellenó con un valor un poco más alto (0,2 por ejemplo).

La solución a esto sería que el usuario tuviera muchas recetas valoradas de tal manera que la mayoría de las características estuvieran cubiertas entonces no habría ese problema con los 0's. Esto no sería muy complicado debido a que muchas recetas tienen una gran cantidad de ingredientes.

#### 4º Ejemplo

Ahora se va a suponer que el usuario ya ha hecho la mayoría de las comidas del día y por tanto las calorías restantes que le quedan son pocas. Aquí jugará un papel fundamental la característica de las calorías, recomendando las recetas que no sobrepasen el límite calórico diario.



Figura 6.7 Información nutricional usuario



Figura 6.8 Recetas recomendadas

El algoritmo debería recomendar recetas que tengan menos 153 kcal. Como se puede apreciar en la imagen funciona bastante bien ya que aparecen recetas con aptas para el usuario además de respetar sus gustos que en este caso era el arroz.

En general el algoritmo es bastante preciso, pero una de las desventajas es que no puede recomendar recetas con ingredientes que el usuario no haya valorado. Esto no es un fallo de implementación, sino que el problema va intrínseco por la estructura del algoritmo.

## 6.2.ALGORITMO DE FILTRADO COLABORATIVO

Como mi base de datos está hecho a mano me fue imposible encontrar datos de usuarios que hubieran valorado las recetas. Una alternativa que se me ocurrió fue crear unas valoraciones artificiales mediante un script de Python que dejo a continuación.

Se hará el ejemplo con 2 usuarios distintos para ver las diferencias. Al primer usuario llamado “colaborativo1” le gustan las siguientes recetas:



Figura 6.9 Recetas valoradas “colaborativo1”

```
[ 'Mole de Olla con Epazote', 5]
[ 'Pollo a la Diabla con Papas', 5]
[ 'Enchiladas Queretanas de Pollo', 5]
[ 'Deliciosas Croquetas de Jamón Serrano', 5]
[ 'Salmon en Salsa Cremosa ', 5]
[ 'Pollo Frito Picante', 5]
[ 'Costillitas a la Coca Cola', 5]
[ 'Pasta con Ensalada Rusa y Pechuga de Pavo ', 5]
[ 'Lomo de Cerdo a las Hierbas Provenzales', 5]
```

Figura 6.10 Resultado recomendación “colaborativo1”

Muchas recetas tienen un 5 de valoración porque en el script de Python la mayoría de las recetas las puse con una alta puntuación. El resultado en la aplicación se visualizaría de esta manera:

FOODRATE
MIS GUSTOS colaborativo1

<p><b>Mole de Olla con Epazote</b></p> <p>Dificultad: Baja Tiempo: 120 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más</p>	<p><b>Pollo a la Diabla con Papas</b></p> <p>Dificultad: Baja Tiempo: 95 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más</p>	<p><b>Enchiladas Queretanas de Pollo</b></p> <p>Dificultad: Baja Tiempo: 45 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más</p>	<p><b>Deliciosas Croquetas de Jamón Serrano</b></p> <p>Dificultad: Baja Tiempo: 280 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más</p>
<p><b>Salmon en Salsa Cremosa</b></p> <p>Dificultad: Baja Tiempo: 50 min Comida diaria: <input type="checkbox"/></p>	<p><b>Pollo Frito Picante</b></p> <p>Dificultad: Baja Tiempo: 48 min Comida diaria: <input type="checkbox"/></p>	<p><b>Costillitas a la Coca Cola</b></p> <p>Dificultad: Baja Tiempo: 270 min Comida diaria: <input type="checkbox"/></p>	<p><b>Pasta con Ensalada Rusa y Pechuga de Pavo</b></p> <p>Dificultad: Baja Tiempo: 28 min Comida diaria: <input type="checkbox"/></p>

Figura 6.11 Recetas recomendadas “colaborativo1”

Ahora se probará con el usuario “colaborativo2” para ver si la salida del algoritmo es distinta a la del anterior usuario. Las recetas que le gustan son las siguientes:



FOODRATE BUSCADOR DE RECETAS MIS GUSTOS colaborativo2

### RECETAS VALORADAS

<p><b>Costillas de Cerdo en Salsa de Chipotle</b></p> <p>Dificultad: Baja Tiempo: 100 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Flan de Mango y Caramelo</b></p> <p>Dificultad: Baja Tiempo: 235 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Flan de Vainilla con Queso Panela y Jalea de Arándano</b></p> <p>Dificultad: Baja Tiempo: 70 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Rica Hamburguesa de Lentejas</b></p> <p>Dificultad: Baja Tiempo: 50 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>
<p><b>Lentejas con Chorizo Fácil</b></p> <p>Dificultad: Baja Tiempo: 55 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★</p>			

Figura 6.12 Recetas valoradas “colaborativo2”

Se pueden ver que al usuario en general le gustan las lentejas y algún dulce.

FOODRATE MIS GUSTOS colaborativo2

<p><b>Pan de Queso Brasileño Fácil</b></p> <p>Dificultad: Baja Tiempo: 40 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Hot Cakes de Roles de Canela</b></p> <p>Dificultad: Baja Tiempo: 25 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Capirotada de Chocolate</b></p> <p>Dificultad: Baja Tiempo: 85 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>	<p><b>Capirotada de Tres Leches</b></p> <p>Dificultad: Baja Tiempo: 60 min Comida diaria: <input type="checkbox"/></p> <p>★★★★★ Ver más</p>
<p><b>Pastel Imposible de Zanahoria</b></p> <p>Dificultad: Baja Tiempo: 138 min Comida diaria: <input type="checkbox"/></p>	<p><b>Pastel de Elote en Olla de Presión</b></p> <p>Dificultad: Baja Tiempo: 100 min Comida diaria: <input type="checkbox"/></p>	<p><b>Cheesecake de Limón con Mango sin Horno</b></p> <p>Dificultad: Baja Tiempo: 150 min Comida diaria: <input type="checkbox"/></p>	<p><b>Pastel Red Velvet con Cheesecake</b></p> <p>Dificultad: Baja Tiempo: 185 min Comida diaria: <input type="checkbox"/></p>

Figura 6.13 Recetas recomendadas “colaborativo2”

Podemos que ver que la recomendación está muy relacionada con postres, claramente hay una diferencia con el usuario anterior, pero esto no se tiene porque ajustar a la



realidad ya que los usuarios con los que se ha ejecutado el algoritmo no son suficientes para que de unos resultados precisos.

Uno de los grandes problemas con este tipo de algoritmo y que tuve que lidiar es la falta de datos de usuarios (cold-start), ya que este tipo de algoritmo se basa en las interacciones de los usuarios con las recetas.

### 6.3.ALGORITMO HÍBRIDO

Igual que en los anteriores algoritmos crearemos 2 usuarios y se verán las diferencias de resultados en base a diferentes gustos.

El usuario “hibrido1” le gustan las croquetas, brócoli y el pollo como se puede ver a continuación.



FOODRATE BUSCADOR DE RECETAS MIS GUSTOS hibrido1

RECETAS VALORADAS


Nuggets de Pollo con Quinoa	Croquetas de Brócoli y Zanahoria	Pollo Alfredo con Brócoli	Nuggets de Pollo Caseros
Dificultad: Baja	Dificultad: Baja	Dificultad: Baja	Dificultad: Baja
Tiempo: 50 min	Tiempo: 25 min	Tiempo: 20 min	Tiempo: 30 min
Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>	Comida diaria: <input type="checkbox"/>
★★★★★	★★★★★	★★★★★	★★★★★
Ver más	Ver más	Ver más	Ver más

Figura 6.14 Recetas valoradas “hibrido1”

Es bueno recordar que los porcentajes que puse para la suma ponderada fueron del 80% al basado en contenido y del 20% al colaborativo. De esta manera se asegura que la recomendación será robusta y si recomienda una receta que se sale de los gustos del usuario será probablemente también le guste porque se ha comparado con recetas de sus mismos gustos.


FOODRATE

MIS GUSTOS híbrido1




**Pollo Empanizado de Verduras**

Dificultad: Baja  
Tiempo: 35 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Pastel de Pollo con Verduras y Queso**

Dificultad: Baja  
Tiempo: 45 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Nopales Capeados Rellenos de Queso**

Dificultad: Baja  
Tiempo: 95 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Crujientes Croquetas de Pollo**

Dificultad: Baja  
Tiempo: 40 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Lasaña Poblana de Crepas**

Dificultad: Baja  
Tiempo: 50 min  
Comida diaria: ☐  
★★★★★




**Pizza de Pollo Frito**

Dificultad: Baja  
Tiempo: 70 min  
Comida diaria: ☐  
★★★★★



**Chiles Rellenos de Queso en Caldillo de Tomate**

Dificultad: Baja  
Tiempo: 20 min  
Comida diaria: ☐  
★★★★★



**Arroz con Brócoli y Queso**

Dificultad: Baja  
Tiempo: 60 min  
Comida diaria: ☐  
★★★★★

Figura 6.9 Recetas recomendadas “híbrido1”


En este caso el algoritmo ha recomendado croquetas, brócoli y pollo, ingredientes muy parecidos a los que ha valorado el usuario.

Ahora se va a probar con el usuario “híbrido2” para ver si hay algún cambio en la recomendación. A “híbrido2” le gusta el aguacate, el atún, la lechuga más o menos los ingredientes que puede tener una ensalada.

FOODRATE


BUSCADOR DE RECETAS MIS GUSTOS híbrido2

RECETAS VALORADAS




**Ensalada de Malvaiscos**

Dificultad: Baja  
Tiempo: 20 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Ensalada César**

Dificultad: Baja  
Tiempo: 10 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)




**Pasta con Ensalada Rusa y Pechuga de Pavo**

Dificultad: Baja  
Tiempo: 28 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)



**Ensalada de Atún con Aderezo de Naranja**

Dificultad: Baja  
Tiempo: 30 min  
Comida diaria: ☐  
★★★★★  
[Ver más](#)



**Donas de Sushi con Salmón, Atún, Pulpo y Camarón**

Dificultad: Baja  
Tiempo: 55 min  
Comida diaria: ☐  
★★★★★

Figura 6.15 Recetas valoradas “híbrido2”

La primera recomendación que aparece es una ensalada con aguacate, también la hamburguesa hawaiana contiene aguacate y lechuga. En la segunda fila vemos también sushi, aguacate y atún, así que la recomendación se asemeja mucho con lo que le ha gustado al usuario 2.

FOODRATE
MIS GUSTOS
híbrido2

<b>Ensalada Fresca de Aguacate, Fresas y Pechuga de Pollo</b> Dificultad: Baja Tiempo: 25 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Hamburguesa Hawaiana</b> Dificultad: Baja Tiempo: 50 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Tostadas de Salpicon de Pollo</b> Dificultad: Baja Tiempo: 45 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Hamburguesa con Salsa Bbq Casera</b> Dificultad: Baja Tiempo: 50 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más
<b>4 Wraps para Llevar</b> Dificultad: Baja Tiempo: 50 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Timbalito de Sushi con Aguacate y Atún</b> Dificultad: Baja Tiempo: 40 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Ensalada Ranch con Pechuga de Pavo</b> Dificultad: Baja Tiempo: 30 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más	<b>Pollo a la Naranja Glaseado</b> Dificultad: Baja Tiempo: 45 min Comida diaria: <input type="checkbox"/> ★★★★★ Ver más

Figura 6.9 Recetas recomendadas "híbrido2"

Lo que hace el algoritmo híbrido es coger "lo mejor de los dos mundos", es decir coge la robustez del algoritmo basado en contenido y la ventaja del algoritmo filtrado colaborativo de recomendar recetas más allá de lo valorado por el usuario. La principal desventaja es que el tiempo de ejecución es el más largo de todos debido a que tiene que ejecutar los 2 algoritmos y si alguno de estos dura bastante se puede hacer muy tedioso para el usuario.

## 7. ANALISIS CRÍTICO

Una parte fundamental de un proyecto es hacer un análisis crítico, viendo las fortalezas y debilidades del proyecto. Hay distintas herramientas con las que se puede hacer esto, pero una de las más populares es el análisis DAFO.

<p><b>DEBILIDADES</b></p> <ul style="list-style-type: none"> <li>-Precisión del algoritmo colaborativo.</li> <li>-Base de datos desbalanceadas</li> </ul>	<p><b>AMENAZAS</b></p> <ul style="list-style-type: none"> <li>-Dependencia de otro sitio web.</li> <li>-Con una gran cantidad de datos por parte de los usuarios se ralentizaría bastante la ejecución del algoritmo colaborativo</li> </ul>
<p><b>FORTALEZAS</b></p> <ul style="list-style-type: none"> <li>-Base de datos muy grande con información muy exclusiva.</li> <li>-Interfaz bonita y accesible</li> <li>-Implementación de algoritmos de recomendación en este ámbito no hay nada parecido.</li> </ul>	<p><b>OPORTUNIDADES</b></p> <ul style="list-style-type: none"> <li>-Nicho de mercado sin explotar</li> <li>-Ampliación para personas con distintos problemas de salud</li> </ul>

### Debilidades

- Para poder ver la ejecución del algoritmo de filtrado colaborativo los datos se generaron de forma artificial ya que se hizo a través de un script en Python. Por tanto, la salida del algoritmo probablemente no de unos resultados esperados para el usuario.
- La base de datos puede estar un poco desbalanceada porque la mayoría de los datos se han sacado de una página web de México así que hay muchas recetas mexicanas en comparación con las españolas. Esto supondría un problema si la aplicación estuviera destinada al público español.

### Amenazas

- Hay alguna información de la base de datos que se obtiene en tiempo real de una web, así que si esa web está caída la aplicación no funcionará correctamente.
- Si la aplicación tiene éxito y se registran muchos usuarios probablemente haya problemas para que se pueda completar la ejecución del algoritmo de filtrado colaborativo ya que tardará demasiado.

### **Fortalezas**

- La base de datos construida es de gran tamaño para el tema que se está tratando, además de que la información de las recetas es muy precisa y difícil de encontrar.
- Interfaz bonita y bastante accesible, pensada para que cualquier persona sin necesidad de saber informática puede navegar por la aplicación y que esta le haga recomendaciones.
- La implementación de los algoritmos de recomendación en este tópico hace que sea única en el mercado.

### **Oportunidades**

- A partir de lo que está hecho se podría ampliar y hacer que dependiendo el tipo de problema de salud que tenga cada usuario se recomiende unas recetas u otras. Esto se haría tan solo con alguna pequeña modificación de los algoritmos y añadiendo una propiedad más al nodo “user” para saber qué tipo de salud tienen los usuarios.
- Junto con la información de la base de datos y la implementación de los algoritmos de recomendación hace que sea esta aplicación sea única o muy poco común en el mercado.

## 8. LINEAS DE FUTURO

Durante un cuatrimestre esto ha sido lo que me ha dado tiempo a hacer. Me he ido dando cuenta que mientras más cosas hacía más cosas podía mejorar de la aplicación además de que el tema de recetas de cocina es muy amplio y da juego para hacer muchas cosas. Las ideas que se me han ocurrido para poder aplicar en un futuro son:

- **Recomendación de recetas de cocina a personas con problemas de salud.** Durante el registro podría decir que problema de salud tiene la persona (diabetes, obesidad, celiaquía, etc) y regular los parámetros del perfil del usuario de la aplicación en base a su enfermedad. Es decir, si una persona ya ha comido mucho respecto a los parámetros recomendables establecidos entonces el sistema de recomendación se adaptará y los resultados que aparecerán serán recetas con pocas calorías en caso de sobrepeso o poco azúcar en caso de una diabetes.
- **Mejora de la base de datos,** implementando nueva información sobre recetas españolas ya que la base de datos como se mencionó en el apartado anterior está un poco desbalanceada. También podría dividirse las recetas por países en las que se haga recomendaciones en base a los. Por otro lado, estaría bien ampliar la información nutricional indicando que vitaminas aporta la receta.
- **Aplicación móvil.** Para que el sistema de recomendación fuera más popular y accesible habría que hacer una aplicación móvil. Porque cuando uno se quiere poner a cocinar normalmente siempre tiene el móvil a mano y al usuario le es mucho más fácil abrir una app de móvil que ir al navegador, escribir la URL del sitio web e interactuar con este.
- **Integración con frigorífico inteligente.** Esta idea es un poco más futurista y mucho más difícil de hacer ya que muy poca gente tiene un frigorífico que le diga que alimentos tiene. Para que el sistema funcione el frigorífico tendría que comunicar a la aplicación que alimentos tiene la persona y entonces en base a esa información recomendar recetas con esos alimentos.



## 9. LECCIONES APRENDIDAS

Al matricularme en esta asignatura venía con las mismas expectativas como la mayoría de las asignaturas 1 o 2 exámenes y un trabajo. Pero no, el tener que hacer un trabajo para todo el cuatrimestre creo que es una buena dinámica para el aprendizaje.

Normalmente en un examen te aprendes las cosas, pero a los 2 días ya se te ha olvidado la mitad, en cambio con esta forma de trabajo implica ser autodidacta (factor importante en nuestra profesión) y las cosas que uno aprende las aplica en el trabajo por tanto es más difícil de que se olviden. Esta metodología también incita a la curiosidad que es una característica importante a la hora de aprender.

Gracias al trabajo aprendí como funcionaban los grafos de conocimiento cosa que nunca había visto antes. Después tuve que aprender el funcionamiento de las bases de datos orientadas a grafos, en este caso de Neo4j. El tener que aprender sobre neo4j implícitamente tuve que aprender un nuevo lenguaje de consultas llamado Cypher para poder construir una base de datos orientada a grafos. Una de las ventajas que tiene Neo4j es la facilidad para construir una base de datos como la de este proyecto de una manera sencilla. Si esto tuviera que hacerse con una base de datos relacional sería un dolor de cabeza.

Ahora ya entiendo como funcionan la mayoría de los sistemas de recomendación como los que usa Netflix. Se las ventajas y desventajas que tienen cada uno de ellos y cual podría servir mejor dependiendo de la situación. Por ejemplo, si no se tienen muchos datos de los usuarios es mejor utilizar un algoritmo de recomendación basado en contenido antes que el colaborativo porque como tiene pocos datos no hará buenas recomendaciones (cold-Start). Durante el aprendizaje de los algoritmos de recomendación me ayudó mucho el libro recommend systems: The textbook. En el se explica todos los tipos de sistemas de recomendación que hay y sus ventajas y desventajas frente unos a otros.

## 10. BIBLIOGRAFÍA

- <https://neo4j.com/>
- [http://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal\\_c\\_c\\_recommender\\_systems\\_the\\_textbook.pdf](http://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf)
- <https://pandas.pydata.org>
- <https://neo4j.com/graphacademy/>
- <https://expressjs.com/es>
- <https://neo4j.com/javascript-driver/>
- <https://github.com/agonzi02/SIBI>
- <https://bedca.net/bdpub/index.php>
- <https://www.kiwilimon.com>
- <https://selenium-python.readthedocs.io/>
-