

WebIR – grupo 5

Informe del trabajo obligatorio

Ignacio Camiruaga

Andrés González

Joaquín Lezama

Melisa Techera

Índice

1. Introducción	3
1. Problema	3
2. Enfoque de la Solución	3
3. Diseño	4
4. Implementación	6
5. Funcionalidad y uso	6
6. Evaluación y resultados	7
7. Conclusiones	7
8. Trabajo futuro	8
9. Referencias	8

1. Introducción

Nuestra intención al realizar este trabajo obligatorio fue utilizar los conocimientos adquiridos en la materia WebIR, para estudiar la factibilidad de una aplicación que combine información de la web en diferentes formatos. En particular, estudiamos el caso de vídeos de cocina de YouTube, para devolverle a los usuarios recetas de cocina que incluyan ingredientes de una lista de interés proporcionada por esos usuarios. Fue necesario estudiar la forma de manejar información en diferentes formatos como p.ej. de video, de audio y de texto, procesar estos datos y devolver a los usuarios la información solicitada que cumpla con los parámetros de la búsqueda.

2. Problema

El problema que intentamos resolver fue el de la búsqueda en Internet de vídeos de recetas de cocina que incluyan los ingredientes de una lista proporcionada por los usuarios.

Al utilizar la aplicación, un usuario ingresa una lista de ingredientes de su preferencia. Tomando los elementos de esa lista como parámetros, el sistema realiza una búsqueda de vídeos de cocina de YouTube previamente agregados al sistema, y devuelve al usuario una lista de videos de cocina que incluyan al menos esos ingredientes.

El problema tiene varios aspectos que debimos considerar. En primer lugar y punto central en el contexto de este obligatorio, cómo extraer información de un video de cocina de forma que nos resultará útil para saber los ingredientes utilizados en la receta.

3. Enfoque de la solución

Para resolver el problema planteado, optamos por utilizar los subtítulos de los videos, que son generados en forma automática. Si bien existen algunos errores en la transcripción de los diálogos de un video, los subtítulos son una herramienta que nos facilitaron la realización del obligatorio.

En líneas generales, en primer lugar obtenemos los subtítulos de un video mediante el package *Youtube Captions scraper* [1]. Luego procesamos cada archivo de subtítulo, buscando en el texto las coincidencias con los elementos de una lista de ingredientes que ya tenemos creada. De esta forma, guardamos en una BD la receta de cocina y los ingredientes que se utilizan en su preparación. Las consultas de los usuarios se resuelven a partir de esta BBDD, buscando las recetas que contengan al menos los ingredientes deseados.

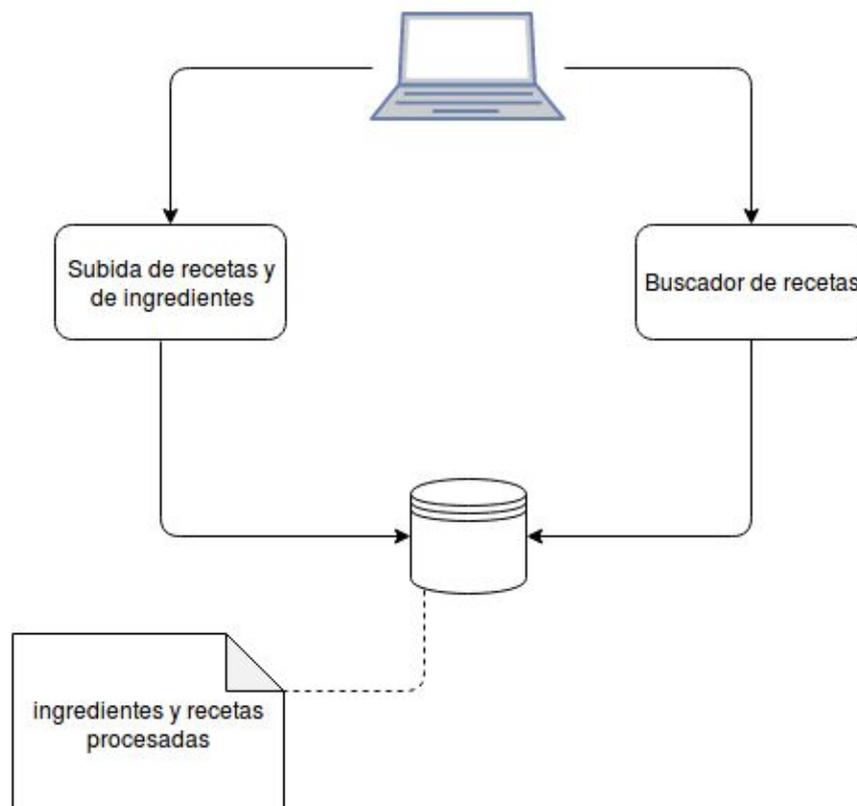
En nuestro caso, nos limitamos al idioma español.

4. Diseño

Nuestro diseño consiste en una API en NodeJS, una página web en ReactJs y una base de datos con MongoDB.

La API ofrece las siguientes funcionalidades:

- Listar recetas: recibe como parámetros una lista de ingredientes. Devuelve una lista de recetas que incluyan al menos todos esos ingredientes.
- Ingresar receta: a partir de una URL pasada como parámetro, busca los subtítulos en esa URL y los guarda en la BD utilizando la estructura (I) que se describe más adelante.
- Listar ingredientes: devuelve una lista con todos los ingredientes de la base.
- Ingresar ingrediente: recibe como parámetro un ingrediente. Agrega ese ingrediente a la BD usando el modelo (III).
- Procesar data: se utiliza luego de agregar nuevos ingredientes a la lista. Actualiza toda la BD de recetas.



Dado que una receta de cocina puede contener diferentes tipos de ingredientes, decidimos usar una BD con MongoDB ya que nos brinda mayor flexibilidad que una base relacional. También tuvimos en cuenta que un mismo ingrediente puede aparecer con más de un nombre (regionalismos y sinónimos) o estar en singular/plural.

Modelos de datos utilizados:

I - Raw files con subtítulos: para almacenar los subtítulos descargados

```
1  {
2    "_id": "5dcf3347ea25ca2a60bc5cf2",
3    "url": "https://www.youtube.com/watch?v=f3vQx5pJFtc",
4    "captions": [
5      {
6        "start": "0.52",
7        "dur": "3.139",
8        "text": "[Aplausos]"
9      },
10     {
11       "start": "6.92",
12       "dur": "11.79",
13       "text": "hoy helado de mango ay qué rico"
14     },
15     {
16       "start": "14.77",
17       "dur": "3.94",
18       "text": "[Música]"
19     }
20   ]
21 }
22
```

II – Receta: para almacenar una receta de cocina. Se guarda la URL del video de la receta y una lista con los identificadores de los ingredientes utilizados.

```
1  {
2    "_id": "5dcf3e83d79e544ad4ae12f7",
3    "videoUrl": "https://www.youtube.com/watch?v=WesvOv8UkLI",
4    "ingredients": [ 1, 3, 4 ]
5  }
6
```

III – Ingrediente: para almacenar el identificador de un ingrediente y sus variantes p. ej. singular/plural

```
1  {
2    "id": 1,
3    "names": ["papa", "papas"]
4  }
5
```

5. Implementación

La implementación fue hecha en lenguaje Javascript, tanto para la API como para la interfaz de usuario. Utilizamos Node JS [2] y ReactJS [3] para crear la página web. Como ya se mencionó anteriormente, el manejador de base de datos elegido fue MongoDB [4]. Para la etapa de pruebas y testing usamos Postman [5]. Se adjunta a este informe el código desarrollado.

6. Funcionalidades y uso

Las funcionalidades implementadas son las estrictamente necesarias para obtener los resultados de las búsquedas y para almacenar videos y una lista con ingredientes en una base de datos. Dado que implementamos solo un prototipo, aún queda trabajo por hacer al que nos referimos en el apartado correspondiente.

Nuestro prototipo es una aplicación web con una página de inicio desde la cual pueden utilizarse sus funcionalidades con un simple click del botón correspondiente.

- Agregar ingrediente: agrega un nuevo ingrediente a la lista, se debe ingresar uno o varios separados por coma.
- Agregar receta: a partir de la URL de un video de Youtube pasada como parámetro, agrega una nueva receta a la BD.
- Actualizar contenido: actualiza la BD de recetas luego de agregar nuevos ingredientes.
- Buscar: buscador de recetas, debe seleccionarse de una lista los ingredientes que tendrán las recetas.

Para utilizar la aplicación primero debe iniciarse el servidor y el cliente de MongoDB. Luego se debe ejecutar el servidor de Nodejs y por último la interfaz web a través de la cual los usuarios podrán utilizar la aplicación.

7. Resultados

A continuación mostramos una captura de pantalla mostrando el buscador. Al abrir la aplicación se muestra el listado de ingredientes de los cuales el usuario selecciona los que utilizará, y a continuación al presionar el botón de buscar, se muestran todos los videos agregados que contienen estos ingredientes. Los mismos pueden ser vistos desde la aplicación, o en YouTube.

En la imagen vemos el resultado de la consulta con el ingrediente “mango” que devuelve dos videos de recetas que incluyen este ingrediente.



8. Conclusiones

Youtube es una herramienta que va creciendo día a día, y por esto cada vez nos encontramos con nuevos videos y nuevas recetas, en una aplicación utilizada por millones de usuarios diariamente. Muchas veces suele suceder que tenemos algún ingrediente en nuestras casas que no se nos ocurre que podríamos cocinar con él, por esto, mediante la utilización de la aplicación planteada, se podrían encontrar fácilmente muchas recetas en las que se lo utiliza, aún no formando parte de los ingredientes principales.

Tras las pruebas realizadas a nuestro prototipo de buscador, entendemos que es factible su implementación. Es posible implementar una aplicación del tipo descrito en este informe y lograr buenos resultados y la satisfacción de sus usuarios. Consideramos que sería un aporte al tema de la materia WebIR y además, podría ser útil para el desarrollo de aplicaciones similares en otros campos de interés como en el área de educación.

9. Trabajo Futuro

Por tratarse de un prototipo este trabajo no está concluido y aún quedan funcionalidades y optimizaciones por estudiar y desarrollar. Al respecto destacamos los siguientes:

- Mejorar búsquedas: buscar una palabra exacta y que no sea substring de otra, búsqueda de sinónimos de ingredientes, nombres de ingredientes usados en singular/plural, palabras usadas en el video pero que no se corresponden con ingredientes utilizados en la receta.
- Agregar otros ingredientes a la lista para aumentar las posibilidades de encontrar coincidencias con los deseos de los usuarios.
- Buscar/desarrollar otras herramientas para extracción de subtítulos.
- Buscar/desarrollar/mejorar otras herramientas para la generación de subtítulos (YouTube).
- Nuevas funcionalidades para la API: borrar receta, borrar ingrediente, buscar recetas que excluyan algunos ingredientes no deseados por un usuario pero que incluyan otros.
- Buscar por ingredientes en las descripciones de los videos.
- Agregar un usuario administrador que sea capaz de utilizar las funcionalidades de agregar videos, ingredientes.

Referencias:

[1] <https://www.npmjs.com/package/youtube-captions-scraper>

[2] <https://nodejs.org/es/>

[3] <https://es.reactjs.org/>

[4] <https://www.mongodb.com/es>

[5] <https://www.getpostman.com/>