Recuperación de Información y Recomendaciones en la Web 2016

Grupo 5

Verónica Martínez - CI: 4.462.695-1

Nicolás Martínez - CI: 4.857.362-7

Índice

1. Definición del problema

Como entusiastas de la cocina, en más de una oportunidad nos encontramos ante la necesidad de obtener, de forma práctica, posibles recetas a preparar a partir de los ingredientes que dispongamos en el momento. De esta forma surgió la idea de poder obtener, a través de las técnicas detalladas en el transcurso del presente informe, una gran base de datos de recetas que permita obtener una colección de las mismas que satisfagan los requerimientos de búsqueda deseados por el usuario. De esta forma, un usuario podría consultar recetas en una base de datos de gran tamaño que contenga determinados ingredientes, asi como tambien que no contengan otros ingredientes.

Esta última funcionalidad se vio motivada por los distintos tipos de enfermedades que aquejan a la sociedad, tales como la enfermedad celíaca que restringe ampliamente las posibilidades alimenticias de las personas. De esta forma, un usuario podría obtener una colección de recetas que no contengan aquellos ingredientes perjudiciales para su salud.
Otra funcionalidad que resultó de interés para implementar en el sistema, fue brindarle al usuario la posibilidad de realizar búsquedas multi idioma. De esta forma, el usuario podría ingresar los ingredientes que desea (y los que no) en el idioma en el que le son familiares para obtener así las recetas candidatas a elaborar.

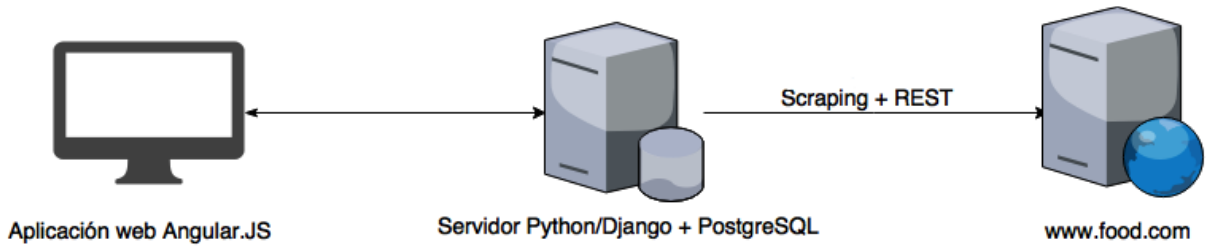
2. Objetivos

Los objetivos principales del desarrollo de esta aplicación se resumen en familiarizarse con herramientas de recuperación de información en la web tales como el scraping, así como también ofrecer una solución a una problemática particular.
De esta forma, al aplicar las herramientas de extracción de datos disponibles, se enfrentaron problemáticas relacionadas tanto a la extracción de los datos, como de los datos mismos y sus traducciones.

3. Arquitectura

La aplicación seguirá una arquitectura cliente-servidor. Este último será construido utilizando el lenguaje de programación Python, junto con el framework Django. Alojará una base de datos PostgreSQL donde residirán una gran cantidad de ingredientes y recetas extraídas de la web.
El servidor también es el encargado de las tareas de extracción de información del sitio www.food.com, a través de técnicas de scraping y consultas a una API REST, para luego almacenar esta información recuperada en la base de datos para su posterior acceso.
Finalmente, ofrecerá una serie de Web Services al frontend a través de una API REST, donde se podrá consultar un listado paginado de ingredientes y recetas bajo una serie de restricciones y parámetros.

El frontend (cliente) consistirá de una aplicación web construida con el framework AngularJS, HTML5 y CSS3. Este será el encargado de consultarle al backend los ingredientes y recetas solicitados por el usuario a partir de las restricciones indicadas por el mismo; así como también de presentar los resultados en un formato amigable para el usuario.



4. Origen de los datos fuente

Tanto el listado de ingredientes como las recetas fueron obtenidos del sitio www.food.com.
Este sitio ofrece una API del tipo REST desde donde se pudo obtener todo el listado de ingredientes, incluyendo una descripción e imagen del mismo.
A continuación se utilizaron las herramientas `requests`[1] y `lxml`[2] para obtener las páginas de las recetas a procesar y scrapearlas, respectivamente.
Dada la correcta correlación que posee el sitio web entre los ingredientes incluidos en una receta y las recetas mismas, los resultados de extracción de la información fueron sumamente satisfactorios. Esto se vió favorecido por el hecho de que cada receta contenía dos tipos de ingredientes que se clasificaron como mapeados y no mapeados. Los mapeados son aquellos donde era posible acceder al detalle del mismo desde la receta y contienen un identificador que posibilita el mapeo del mismo con un ingrediente obtenido de la API REST. Por otro lado, los ingredientes no mapeados se corresponden con aquellos que no son recuperables a través de la api REST.

De los ingredientes mapeados se logró recuperar el nombre, una breve descripción, el identificador dentro del sitio www.food.com y una URL de una imagen del ingrediente.
Respecto a los ingredientes no mapeados simplemente se logró recuperar el nombre de los mismos.
Finalmente, de las recetas se logró obtener bastante información: nombre, descripción, identificador dentro del sitio web, URL de una imagen, URL a la receta en sí en el sitio original, ratings de los usuarios, tiempo de preparación y cocción, porciones, y los ingredientes en sí.
De cada ingrediente en cada receta, se obtuvo el orden en que deben ser agregados a la misma, cantidad e instrucciones especiales en caso ser necesarias.

5. Procesamiento de los datos fuente

Una vez recuperados los datos de los ingredientes del sitio, fue necesario obtener la traducción de los mismos en diferentes idiomas con el fin de soportar la funcionalidad antes mencionada.
No caben dudas que el mejor servicio de traducción existente a la fecha le pertenece a Google con su *Google Translate.* En un principio se optó por utilizar la api que ofrece dicho servicio, pero dado que Google no lo ofrece más como un servicio gratuito, se debió investigar otras alternativa.
Como forma de realizar la prueba de concepto de la búsqueda multi-idioma, se recurrió a la librería Python `nltk` (Natural Language Toolkit), una librería dedicada a tareas de procesamiento del lenguaje natural. Con la misma se descargó el corpus WordNet. El mismo es una base de datos léxica en inglés, que categoriza y agrupa verbos, adjetivos, sustantivos, etc. según su significado. Es posible además obtener el lema o raíz de una palabra. Este es el mecanismo que se explotó para obtener las traducciones de los ingredientes.
Dado que el WordNet utilizado por `nltk` cuenta con subconjuntos en otros idiomas, se seleccionaron cuatro de los más tradicionales para este proyecto, estos fueron Español, Italiano, Francés y Portugués.
De esta forma y durante la recuperación de los ingrediente, se prosiguió a obtener el equivalente de la lematización de los nombres en los idiomas seleccionados, los cuales no devolvieron los resultados esperados. Una de las desventajas del método, es que al requerir la lematización de los términos, los cuales ya son sustantivos, no se encuentran en la base aquellos formados por más de una palabra y generando traducciones discutibles en otros casos. Por ejemplo, para el ingrediente *Almond* se obtuvieron traducciones correctas tanto para francés (*amande*), italiano (*mandorlo*) y portugués (*amendoeira*), pero en el caso del español se obtuvo el nombre “*prunus dulcis*”, el cual corresponde al nombre científico del árbol, en lugar de *almendra*, que es la traducción esperada del ingrediente en cuestión.
De todas formas, se decidió mantener esta funcionalidad en la aplicación web como forma de demostrar que es posible satisfacer este requerimiento con la herramienta de traducción adecuada, por ejemplo, el antes mencionado *Google Translate*.

6. Tecnologías utilizadas

Como se explicó anteriormente, durante el transcurso del desarrollo del sistema se utilizó el lenguaje de programación Python[3] junto con el framework Django[4] para el desarrollo del servidor, PostgreSQL[5] para el almacenamiento de los datos, NTLK[6] para la traducción de los ingredientes.
Para el desarrollo de la aplicación frontend se utilizó el framework javascript Angular.js[8], junto con HTML[9] y CSS3[10] para el diseño de la interfaz de usuario.

7. Evaluación de la aplicación

La tablas creadas por el framework Django para almacenar los modelos de datos definidos generan índices sobre las claves de las mismas. Se evalúa a continuación el rendimiento y tiempos de respuesta de la aplicación sin agregar más índices a la misma. Las pruebas se realizan utilizando la aplicación web donde tanto el servidor web como el backend corren localmente.

7.1. Búsqueda de ingredientes

El filtrado promedio de ingredientes se realiza en un promedio de 17 a 21 ms, mientras que el promedio en la búsqueda multi idioma agrega 4 ms más a la búsqueda en promedio. No se observa una diferencia significativa en los tiempos de respuesta de los dos tipos de búsqueda. Cabe aclarar que la base de ingredientes cuenta con 984 filas, por lo que no es una tabla particularmente grande, pero se realizan búsquedas sobre columnas sin índice.

7.2. Búsqueda de recetas

Estas búsquedas son más complejas que las anteriores, ya que no solo se buscará por nombre, sino también por relaciones entre recetas ingredientes, lo que requiere atravesar varias tablas de datos. Para estas pruebas se agregaran recetas de manera gradual, aumentando así el número de filas de la misma y observando los tiempos de respuesta en varios casos descritos a continuación.

- **Caso 1**: Búsqueda sin parámetros, la cual retornará todas las recetas de la base en páginas de 20 registros.
- **Caso 2**: Búsqueda por nombre, filtrando por el término *bread*
- **Caso 3**: Búsqueda por nombre, filtrando por el término *cake*
- **Caso 4**: Búsqueda por ingrediente *flour*
- **Caso 5**: Búsqueda por ingrediente *sugar*
- **Caso 6**: Búsqueda por ingredientes *flour*, *sugar*, *egg*
- **Caso 7**: Búsqueda por nombre *bread* e ingredientes *flour*, *sugar*, *egg*
- **Caso 8**: Búsqueda por nombre *cake* e ingredientes *flour*, *sugar*, *egg*
- **Caso 9**: Búsqueda por nombre *cake* e ingredientes *sugar*, *egg*, sin ingrediente *flour*

En la siguiente tabla se detallan los resultados del tiempo de búsqueda y cantidad de coincidencias de estos casos a medida que crece la tabla de recetas. Tanto los nombres a buscar como los ingredientes se seleccionaron por considerarse que generarían la mayor cantidad de resultados.

Caso #	50 recetas	503 recetas	1003 recetas	5002 recetas	10001 recetas
1	23 ms 50 recetas	18 ms 503 recetas	17 ms 1003 recetas	24 ms 5002 recetas	44 ms 10001 recetas
2	45 ms 3 recetas	29 ms 34 recetas	19 ms 65 recetas	34 ms 293 recetas	40 ms 560 recetas
3	19 ms 5 recetas	20 ms 32 recetas	22 ms 71 recetas	28 ms 322 recetas	41 ms 635 recetas
4	23 ms 16 recetas	21 ms 185 recetas	22 ms 361 recetas	28 ms 1570 recetas	93 ms 3127 recetas
5	18 ms 13 recetas	19 ms 179 recetas	20 ms 365 recetas	25 ms 1639 recetas	31 ms 3140 recetas
6	28 ms 9 recetas	29 ms 99 recetas	30 ms 199 recetas	33 ms 825 recetas	45 ms 1560 recetas
7	25 ms 1 receta	33 ms 20 recetas	27 ms 35 recetas	45 ms 153 recetas	54 ms 259 recetas
8	20 ms 4 recetas	30 ms 24 recetas	41 ms 62 recetas	50 ms 232 recetas	55 ms 434 recetas
9	17 ms 0 recetas	22 ms 3 recetas	25 ms 6 recetas	41 ms 32 recetas	71 ms 69 recetas

8. Objetivos alcanzados y trabajo a futuro

El objetivo principal del proyecto consistía en crear una base de datos de recetas e ingredientes a partir de información recuperada de la web, para luego poder realizar consultas sobre recetas que tanto incluyan como excluyan determinados ingredientes; por lo que es posible afirmar que el resultado final es satisfactorio.
Por otro lado, el objetivo secundario que se definió relacionado a la búsqueda multi idioma no arrojó los resultados esperados a causa de la herramienta utilizada.
Queda como posible trabajo a futuro la integración del sistema con una herramienta comercial de traducción más potente.

9. Referencias

[1] Librería requests - http://docs.python-requests.org/en/master/
[2] Librería lxml - http://lxml.de
[3] Python - https://www.python.org
[4] Django - https://www.djangoproject.com/
[5] PostgreSQL - http://www.postgresql.org.es
[6] NLTK - http://www.nltk.org
[7] WordNet - https://wordnet.princeton.edu/
[8] AngularJS -https://angularjs.org/
[9] HTML5 - https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5
[10] CSS3 - https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3