RECUPERACIÓN DE INFORMACIÓN Y RECOMENDACIONES EN LA WEB

CURSO 2019

GRUPO 6

Ramiro Clavijo 4573465-8
Alvaro Callero 4034747-8

Índice

1. Introducción

El documento que se presenta a continuación, refleja el trabajo realizado para la segunda y final entrega correspondiente a la asignatura Recuperación de Información y Recomendaciones en la Web.
La temática a abordar se lo conoce como “scraping”, el cual tiene como cometido extraer información de determinados sitios web por medio de una entidad denominada bot.

El tema elegido se encuentra dentro de lo que se conoce como Inteligencia Artificial, la cual tiene como implicancia la combinación de algoritmos planteados con el objetivo de crear sistemas que presenten capacidades similares a las del ser humano, automatizando actividades como la toma de decisiones, la resolución de problemas y el aprendizaje.

2. Problema

Tal como se mencionó en la sección correspondiente a la introducción, el Web Scraping es una posible manera de extraer información de la Web, por lo que resulta interesante poder considerar cuales serian las posibles aplicaciones, o determinar que tanta utilidad puede llegar a tener el hecho de usar este tipo de tecnología.
Normalmente este tipo de programas, realizar la simulación de un ser humano navegando por determinados sitios, ya sea por medio del conocido protocolo HTTP de forma manual, o incrustando un navegador en nuestra aplicación.

Entonces, esta técnica se puede aplicar en diferentes sectores, resolviendo problemáticas de las más variadas.
Entre ellas se encuentran:

1. **Comercial y Ventas:**
   Poder cualificar bases de datos de manera automática, lo que permitiría agregar información adicional bases de datos de clientes, prospectos, suscriptores, entre otros.

2. **Monitorizar precios de la competencia:**
   Se podría lograr el mantener un listado actualizado a tiempo real de los precios que tiene la competencia en determinadas referencias, como también de venta de nuestros partners y minoristas.

3. **Marketing e investigación de mercado:**
   Investigar compradores, tendencias, monitorizar nuestra marca, como también permitir el rastreo de la web para buscar de cualquier dato como redes sociales, foros, etc. Hay mucha información de nuestros potenciales consumidores a nuestra disposición.

4. **Detectar influencers:**
   Sería una información muy útil para planificar tu campaña de marketing, y con web scraping podrías conocerlo y organizarlo.

Como se puede apreciar, son variados los usos que se le puede dar a la técnica de scraping: en esta ocasión, la utilizaremos para obtener noticias de deportes.

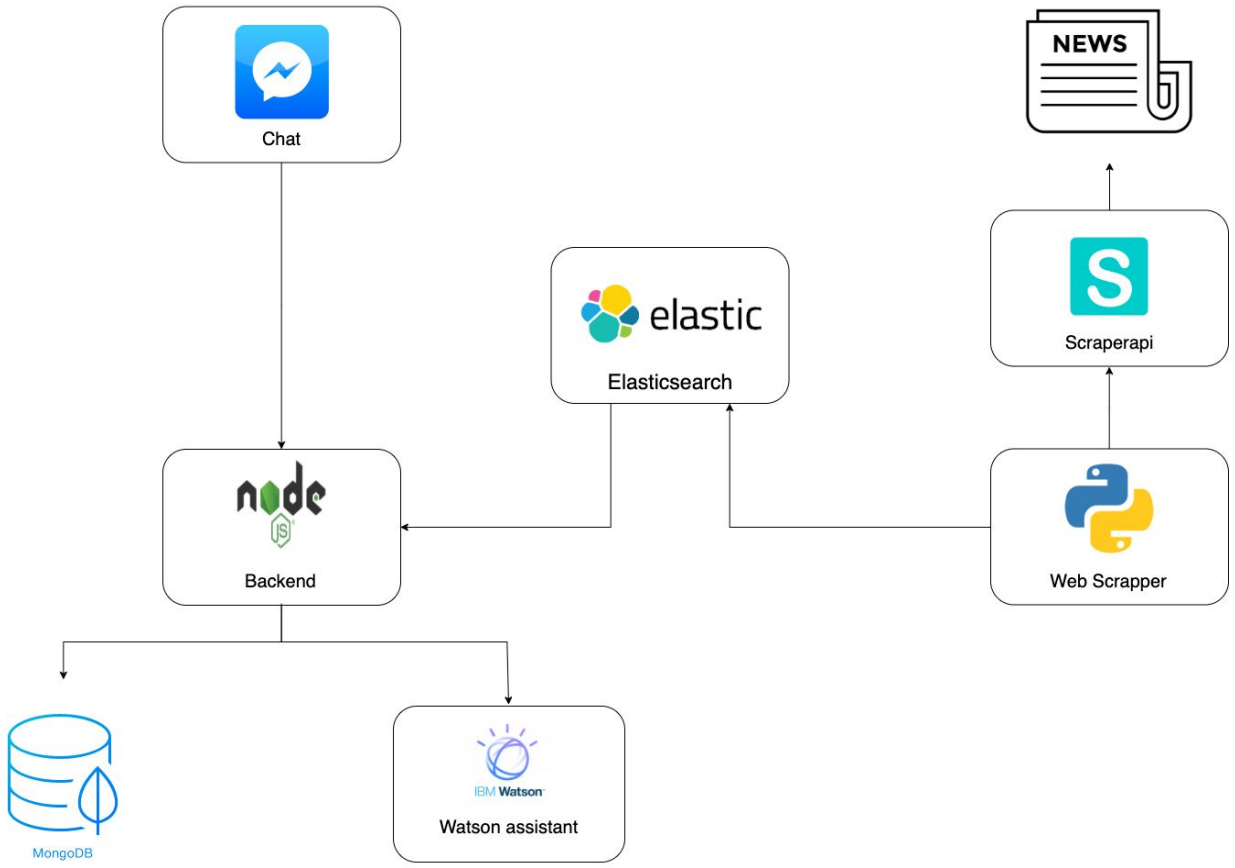
3. Enfoque de la solución

La solución que se plantea, es la de construir un ChatBot que obtiene noticias deportivas, el cual en principio va a utilizar como sitio para aplicar el scraping a la página Referi, de El Observador (https://www.elobservador.com.uy/referi), con el objetivo de que los usuarios puedan recibir las noticias deportivas en el chat, obteniendo así un resumen de la misma y el link si desea ver la noticia completa. Dicho ChatBot está integrado a Facebook, con el cual se puede realizar la interacción desde Messenger o Facebook.
La solución tiene un enfoque tal que consta de 5 componentes, los que se mencionan a continuación:

- Backend (se usará la tecnologîa NodeJs)
- Base de datos (no relacional siendo la misma MongoDB)
- Web Scraper (realizado utilizando Python)
- Watson Assistant (herramienta de IBM para el entrenamiento, Watson)
- Interfaz de usuario (chatbot de Facebook Messenger)

4. Diseño e implementación

En lo que refiere al diseño de la solución, se siguió la siguiente arquitectura:



Se pasará a explicar cada uno de los componente en mayor detalle, para lograr así un entendimiento global del camino que se siguió para dar con la solución:

- **Backend:**
  Se optó por la utilización de la tecnología Node.js, la cual es un entorno de ejecución para JavaScript, construido con el motor de JavaScript V8 de Chrome.
  Ahondando más en detalles técnicos, Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

También se utiliza el framework Express.js, es llamado el framework standard para desarrollo de backend con js, permite crear APIs y aplicaciones web fácilmente, provee un conjunto de características como manejo de rutas (direccionamiento), archivos estáticos, uso de motor de plantillas, integración con bases de datos, manejo de errores, middlewares entre otras.

- **Base de Datos:**
  Para lograr la persistencia de datos, se utilizó la base de datos no relacional MongoDB, junto con el ORM Mongoose, en lugar de guardar los datos en tablas, tal y como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos JSON, haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida. NodeJS y MongoDB son buenos socios, la estructura estándar usada en JavaScript es Json, con lo cual almacenar los datos y leerlos es muy simple y rápido. Dado que no se tendrán Queries complejas ni muchos datos almacenados, la aplicación de esta base de datos es beneficiosa.

- **Web Scraper:**
  El scraping o también conocido como Web scraping, es una técnica utilizada mediante programas de software para extraer información de sitios web. Usualmente, estos programas simulan la navegación de un humano en sitios de la web, ya sea utilizando el protocolo HTTP manualmente, o incrustando un navegador en una aplicación.
  En la presente resolución, se utilizará la librería de Python *elasticsearch* , y un programa en este lenguaje para obtener los datos y exponer APIs para consultar. También se usa una librería de Python llamada *BeautifulSoup*

- **Watson Assistant:**
  La misma es una herramienta cloud de IBM, la cual permite entrenar un modelo de inteligencia artificial capaz de identificar intenciones y entidades para de esta forma responder a lo que ingrese un usuario, posterior a ser entrenado. Los conceptos más importantes que maneja Watson son las intenciones, las entidades y los nodos. Las intenciones como lo dice su nombre, son acciones que el usuario intenta realizar, las entidades por su parte son conceptos que se desean identificar en la entrada del usuario, por ejemplo deportes (futbol, tenis, natacion), por último los nodos son básicamente el conjunto formado por entidades, intenciones y la respuesta deseada. Las entidades e intenciones se comportan como una condición lógica, cuando se identifica cierta intención y entidad en una entrada del usuario, se ejecuta lo que contenga el nodo.

- **Interfaz de usuario:**
  Para poder concebir la interacción con el usuario, se llevo a cabo una integración del chatbot con Facebook Messenger, el cual permite una fácil en intuitiva interacción con el usuario, para poder consultar las noticias.

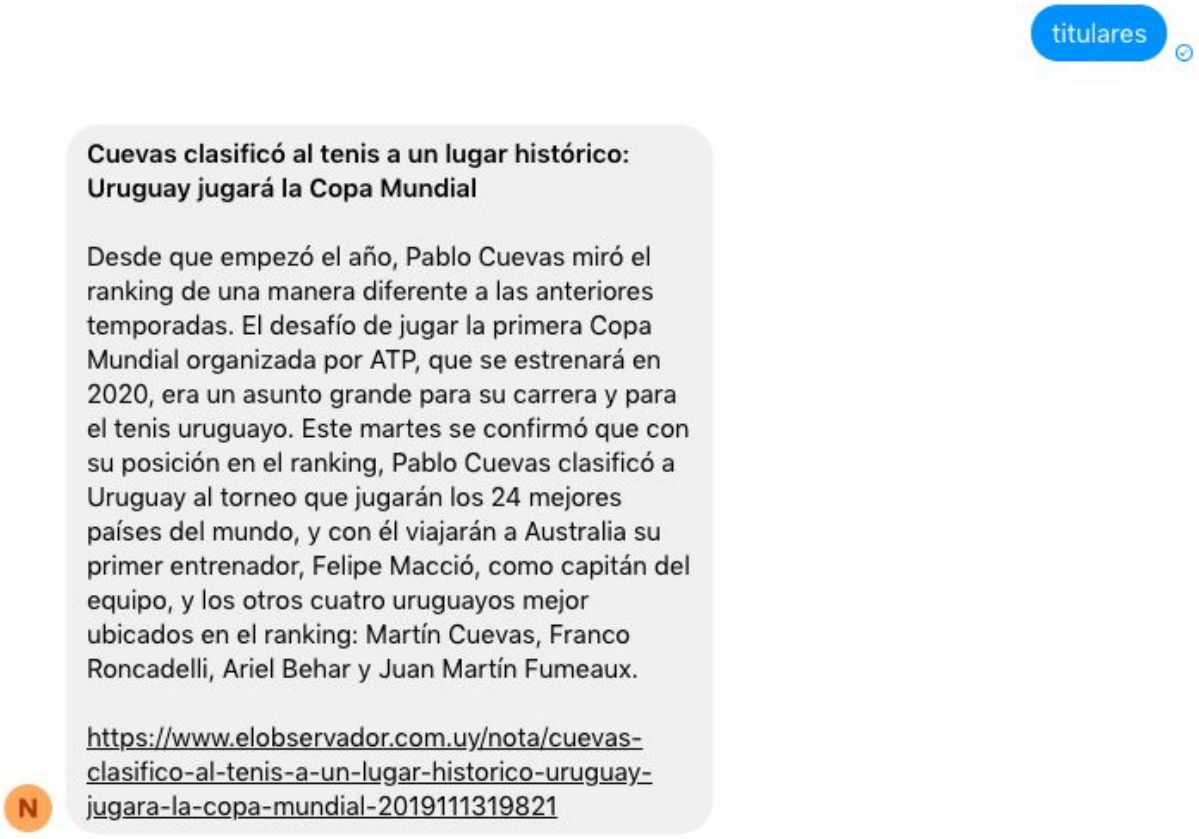
- **Elasticsearch:**
  Es un motor de búsqueda y análisis. Es distribuible y fácilmente escalable y a través de interfaces web HTTP y documentos JSON, permite interactuar de forma sencilla con su núcleo y realizar búsquedas de texto completo muy eficaces. Se utiliza para consultar las noticias previamente indexadas en el proceso Python.

5. Funcionalidades y uso

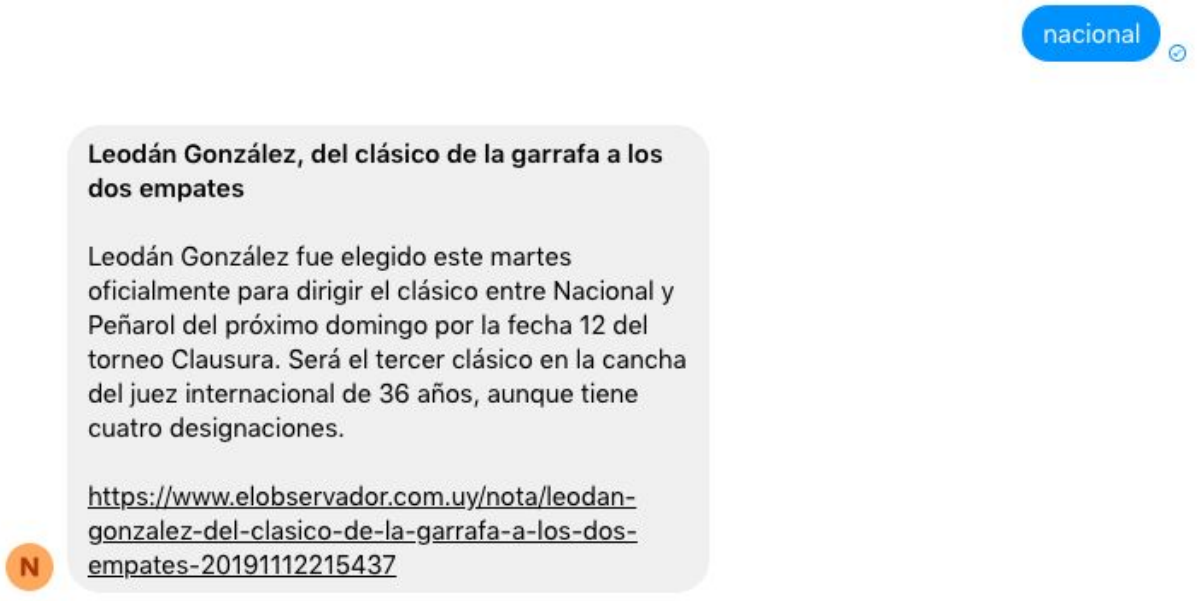
Se implementaron 2 funcionalidades, consultar los titulares del día, y consulta sobre algún deporte o equipo deportivo.

El chatbot lista estas opciones para que el usuario pueda saberlo, y luego no es necesario seguir ningún flujo ni apretar botones.

Cada noticia se muestra con el título, un resumen y el link para acceder a la misma, como se muestra en la siguiente imagen, donde tambien se ve una de las funcionalidades, la consulta de titulares:



Otra de las funcionalidades es consultar información sobre un equipo, por ejemplo en la siguiente imagen:



El flujo completo de ejecución desde que se scrapea una noticia, hasta que el usuario recibe la respuesta consiste en lo siguiente:

1. Un proceso implementado en Python, que se ejecuta cada 4 horas, busca nuevas noticias utilizando Scraperapi, las indexa y almacena en Elasticsearch
2. Cuando un usuario escribe en el chat de Facebook, el mensaje viaja hasta el backend Node
3. El backend, envía el texto para ser analizado en Watson Assistant, el cual retorna entidades e intenciones analizadas
4. Nuevamente el backend, dependiendo de lo que pregunto el usuario, consulta al Elasticsearch y busca noticias
5. Se envía la respuesta al chat de Facebook con noticias, o indicando que no se encontró ninguna si es el caso

6. Evaluación y resultados

En lo que refiere a los resultados obtenidos, y a la evaluación de los mismos, podemos decir que a lo largo del desarrollo del ChatBot, se fueron realizando varias pruebas para poder comprobar la correctitud de la funcionalidad del scraping: para esto, si bien se pudo haber usado alguna herramienta del estilo Postman o similar, en el caso en cuestión, solo se optó por hacer pruebas manuales, para corroborar que luego de realizar determinada acción, el resultado era el esperado.

Otro aspecto a tener en cuenta que es de vital importancia, es que tan extensa y precisa es la documentación acerca de las herramientas utilizadas para realizar el scraping, ya que es muy normal que la documentación existente acerca de una determinada API no esté actualizada, o la forma de interacción con la misma, no este correctamente especificada en la documentación.

7. Conclusiones

Al finalizar el proyecto en cuestión, se pueden destacar varios aspectos positivos: por un lado, se realizó un desarrollo que si bien no fue demasiado extenso ni complejo, sirve para poder interactuar con tecnologías que probablemente uno no esté familiarizado, ya que quizá no sea algo cotidiano para alguien que se dedique

al rubro del desarrollo de software, tener que lidiar con técnicas relacionadas al scraping.

Por otro lado, se pudo tener contacto con tecnologías de las mas variadas, ya que el desarrollo del ChatBot implicó el uso de Python, NodeJs, MongoDB, entre otros.

A modo de cierre, se puede decir que fue una experiencia positiva, la cual nos dejó como aprendizaje, la facilidad con la que se puede interactuar con diferentes aplicaciones, que si bien puede llegar a ser un tanto complejo al principio, se pueden lograr buenos resultados a un costo relativamente bajo, en cuanto a tiempo invertido en el mismo.

8. Trabajo a futuro

En el presente proyecto, se optó por realizar la interacción con el usuario por medio de la integración del chat de Facebook Messenger: quizá sería bueno realizar una aplicación desde cero, en donde dicho chat esté construido en alguna otra tecnologîa, pudiendo así customizar ciertos aspectos que quizá, utilizando la integración con Facebook Messenger no se podría.

Otra posible mejora, sería la de poder no solo recuperar noticias de todo tipo y sitios, no solo limitarse a las que forman parte del ámbito deportivo, si no que se logre recuperar dichas noticias de una lista posible y extensa de diferentes sitios, y a su vez en diferentes idiomas.

También se puede pensar la idea de, poder obtener noticias de todo tipo y no solo limitarse a las que forman parte del ámbito deportivo, pudiendo así captar un público de usuarios mucho mayor.

Otra idea considerada fue la de poder manejar suscripciones de parte de usuarios a distintos temas, equipos, deportes. Por ejemplo recibir todas las noticias sobre fútbol el lunes a cierta hora.

9. Referencias

- Elasticsearch - https://www.elastic.co/
- Wikipedia - https://en.wikipedia.org/wiki/Web_scraping
- Python - https://www.python.org/
- Beautifulsoup - https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- MongoDB - https://www.mongodb.com/
- NodeJS - https://nodejs.org/en/
- Watson Assistant - https://www.ibm.com/cloud/watson-assistant/
- Scraperapi - https://www.scraperapi.com/