Desarrollo de Aplicación para la Recolección de Tweets para Proyecto de Agenda Setting

Cristhian Richard, Ramiro Rivera, Esteban Schab,
Lautaro Ramos, Patricia Cristaldo,
Soledad Retamar, Anabella De Battista
Depto. Ingeniería en Sistemas de Información
Fac. Reg. Conc. del Uruguay, Univ. Tecnológica Nacional
Entre Ríos, Argentina
{richardc, riverar, schabe, ramosl, cristaldop, retamars, debattistaa}@frcu.utn.edu.ar

Leticia Cagnina
Norma Edith Herrera
Departamento de Informática
Univ. Nac. de San Luis
San Luis, Argentina
{lcagnina, nherrera}@unsl.edu.ar

Resumen

La Teoría de la Fijación de Agenda postula que los medios de comunicación tienen una gran influencia sobre el público y logran determinar qué asuntos poseen interés informativo y qué relevancia le otorgan los usuarios. En este artículo se presenta un proyecto que, mediante técnicas de minería de textos, pretende determinar si los medios periodísticos argentinos logran o no instalar temáticas en usuarios de redes sociales como Twitter. Como resultado de la primer etapa del proyecto se presentan en este artículo algunos desarrollos como: un script en R para realizar web scraping sobre los sitios web de periódicos digitales de Argentina, obtener las noticias publicadas en un período de tiempo y determinar los tópicos claves que se abordan en dichas noticias; y el desarrollo de una aplicación web que permite realizar la captura de tweets, en base a parámetros definidos previos a la búsqueda, para constrastar si los temas abordados en los periódicos tambien tienen repercusión en redes sociales. Conjuntamente se presenta una novedosa metodología para la gestión de proyectos de ciencias de datos.

1. Introducción

En la actualidad se generan diariamente, desde diversos orígenes, grandes cantidades de datos de diversos tipos (textos, imágenes, audios, videos, entre otros). Esta constante producción de datos pone a disposición nuevas fuentes de información que pueden ser aprovechadas para agregar valor en la toma de decisiones. En este contexto, y gracias a la masividad que han adquirido tanto las redes sociales como las aplicaciones de mensajería instantánea o los sistemas de recomendación generados por usuarios que expresan opiniones sobre productos o noticias, el texto no estructurado se ha convertido en un insumo de gran valor.

La disciplina que permite identificar conocimiento implícito dentro de los textos se conoce como Minería de Textos (Text Mining). Su objetivo se centra en descubrir y extraer conocimiento relevante y no trivial a partir de textos no estructurados. Las técnicas de minería de textos son empleadas en diversas áreas como la medicina para el etiquetado automático de notas clínicas [1] o la definición de cohortes de pacientes utilizando notas clínicas y datos de historias clínicas electrónicas [2]. Estas técnicas también son empleadas por plataformas como Netflix, que analiza los comentarios vertidos por sus usuarios en diferentes medios sociales para mejorar su sistema de recomendación y proveer contenido personalizado [3].

La mayoría de los algoritmos, herramientas y recursos disponibles para Minería de Textos han sido probados y/o desarrollados para el idioma inglés, por lo que presentan dificultades al ser empleados sobre textos escritos en otros idiomas como el español. Es por esta razón que es necesario trabajar en la producción de recursos específicos y en la adaptación de algoritmos y herramientas que contemplen las particularidades del idioma español.

El término *Agenda Setting* hace referencia a la

influencia que tienen los medios de comunicación en la fijación de temas en la opinión pública. En este artículo se presenta el desarrollo de una aplicación, en el marco de un proyecto cuyo objetivo es medir los efectos de la instalación de asuntos en la agenda pública, tomando como base artículos escritos sobre diferentes temáticas en medios digitales de relevancia para determinar los tópicos que tratan y luego analizar su difusión en redes sociales empleando técnicas de minería de textos.

Para realizar dicho análisis está previsto obtener y procesar por un lado los artículos periodísticos y por otro lado las publicaciones y/o comentarios en redes sociales, para luego realizar un análisis cruzado para obtener conclusiones. Para obtener la información periodística se aplican técnicas de *web scrapping* a varios medios digitales de noticias previamente seleccionados. El *web scrapping* es una técnica que sirve para extraer información de páginas web de forma automatizada. Una vez obtenidas las noticias, se realiza la detección de *keywords* [4]. Utilizando como filtro los términos o palabras claves encontrados en las noticias, se realiza el proceso de recolección de tweets.

Como un subárea específica de la Minería de Datos se puede citar al Data Stream Mining, que es el proceso de extraer conocimiento en estructuras de datos continuas y con rápidas transiciones [5]. Los data streams son flujos continuos de datos que se generan a altas velocidades. Pueden provenir de diversas fuentes, como registros generados por clientes que utilizan aplicaciones móviles, transacciones electrónicas, logs de navegación de una red de datos, información de redes sociales, datos provenientes de dispositivos *wereables*, entre muchos otros ejemplos.

El procesamiento de estos datos debe realizarse de forma secuencial y gradual registro por registro o bien, en ventanas de tiempo graduales. Los resultados de dicho procesamiento se utilizan para una amplia variedad de tipos de análisis, como correlaciones, agregaciones, filtrado y muestreo. Las conclusiones obtenidas a partir de dicho análisis aporta a las empresas visibilidad de numerosos aspectos del negocio y de las actividades de sus clientes, como la tasa de uso de un servicio, la actividad de un servidor, la ubicación geográfica de un móvil, personas o mercadería, la afluencia de determinado tipo de clientes, entre otros aspectos, y les permite responder con rapidez ante cualquier situación que surja. Por ejemplo, un banco podría analizar el incremento de determinada categoría de clientes en un momento dado y responder rápidamente habilitando más puestos de atención al cliente.

En el presente trabajo se realiza una descripción general del proyecto de *Agenda Setting* y se presenta una de las primeras etapas de dicho proyecto, que consiste en la recolección de noticias mediante web scrapping y el diseño e implementación de una aplicación web para la recolección y preprocesamiento de *tweets* de forma automática según determinados parámetros.

2. Marco Teórico

En los últimos años, se ha experimentado un aumento en el volumen de datos en formato textual no estructurado debido principalmente a la expansión de las TIC y las plataformas de redes sociales. Este crecimiento continuo de los datos en formato de texto también se ve en el ámbito de las organizaciones, que han pasado de utilizar sólo fuentes de datos estructurados, organizados en sistemas de gestión de bases de datos relacionales y data warehouses, a resguardar datos semiestructurados y no estructurados, incluidos los textos. Dentro de estos enormes volúmenes de textos hay información valiosa que, si se identifica y extrae correctamente, puede explotarse para la toma de decisiones y para apoyar una amplia gama de actividades empresariales [6].

Como herramienta para aprovechar el conocimiento potencial que puede obtenerse a partir del procesamiento y análisis de los textos, surge una particularización de la Minería de Datos (Data Mining) conocida como Minería de Textos (o Text Mining) que es el proceso de extraer patrones relevantes a partir de un gran conjunto de textos con el propósito de obtener conocimiento [7].

La Minería de Textos comprende una amplia serie de tareas que se enfocan en distintos aspectos del análisis de textos. Algunas de las más relevantes son [8]:

- Recuperación de información (Information Retrieval, IR): es la tarea de encontrar material (usualmente documentos) de naturaleza no estructurada (generalmente textos) proveniente de grandes colecciones que satisfagan determinadas necesidades de información [9]. El objetivo del IR es encontrar documentos relevantes para una necesidad de información, no solo identificar simples coincidencias con patrones léxicos en una consulta, es decir, la relevancia de los resultados es medida en función del requerimiento de información, y no respecto de si los documentos contienen las palabras incluidas en la consulta de búsqueda [10]. Una tarea crucial para un sistema de IR es indexar la colección de documentos para hacer que sus contenidos sean accesibles de manera eficiente. Generalmente la indexación se realiza sobre una representación lógica del documento, que puede consistir en un conjunto de

palabras clave o términos relevantes que aparezcan en el texto [10].

- Procesamiento del Lenguaje Natural (Natural Language Processing, NLP): es un campo de las ciencias de la computación que combina Inteligencia Artificial y conceptos lingüísticos con el fin de hacer que oraciones o palabras escritas en lenguaje natural puedan ser interpretados por programas de computadoras [8], [11].
- Extracción de Información (Information Extraction, IE): es una subdisciplina de la Inteligencia Artificial que se aboca a la identificación, y consecuente clasificación y estructuración en grupos semánticos, de información específica que se encuentran en fuentes de datos no estructurados, como el texto en lenguaje natural, lo que hace que la información sea más adecuada para las tareas de procesamiento de la información [12]. Dentro del ámbito de la IE se encuentran diferentes tareas como la extracción de entidades nombradas y relaciones entre las mismas, la identificación de eventos e información temporal, y la resolución de coreferencias de dos o más expresiones al mismo término o entidad [13].
- Resumen de textos (Text Summarization): es la tarea de producir un resumen conciso y fluido preservando el contenido clave de la información y el significado general de una colección de textos [14]. Los métodos de resumen de texto se pueden clasificar en resumen abstractivo y extractivo. Un método de resumen extractivo consiste en seleccionar oraciones importantes, párrafos, etc. del documento original y concatenarlos en una forma más corta. Un resumen abstracto expresa de forma concisa y en un lenguaje natural claro los conceptos principales que se encuentran en un documento, luego de realizar una comprensión de los mismos.
- Métodos de Aprendizaje Supervisado y No Supervisado: los métodos de aprendizaje supervisado son técnicas de aprendizaje automático relacionadas con entrenar un modelo de clasificación utilizando un conjunto de datos de entrenamiento para realizar predicciones sobre datos desconocidos de antemano. Existe una amplia gama de métodos supervisados, como clasificadores de vecinos más cercanos, árboles de decisión, clasificadores basados en reglas y clasificadores probabilísticos [8]. En cuanto a la clasificación de texto, los aspectos más relevantes radican en la construcción de una estructura de datos que pueda representar los documentos y la construcción de un clasificador que pueda usarse para predecir la clase de un documento con alta precisión [15]. Los métodos de aprendizaje no supervisados son técnicas que intentan encontrar una estructura oculta a partir de datos no etiquetados. No necesitan ninguna fase de entrenamiento, por lo tanto se pueden aplicar a cualquier información de texto sin esfuerzo manual. La agrupación en clústeres y el modelado de temas son los dos algoritmos de aprendizaje no supervisados utilizados comúnmente en el contexto de los datos de texto. La agrupación es la tarea de segmentar una colección de documentos en particiones donde los documentos en el mismo grupo (clúster) son más similares entre sí que los de otros clústeres. En el modelado de temas, se usa un modelo probabilístico para determinar un clúster suave, en el que cada documento tiene una distribución de probabilidad en todos los grupos, en oposición a la agrupación en clúster de documentos. En los modelos de tema, cada tema se puede representar como una distribución de probabilidad sobre las palabras y cada documento se expresa como distribución de probabilidad sobre los temas. Por lo tanto, un tema es similar a un clúster y la pertenencia de un documento a un tema es probabilística.

Los trabajos sobre minería de textos en español que se presentan en la actualidad se enfocan principalmente en Análisis de Sentimientos o Minería de Opinión, en los cuales se evidencian dos enfoques, uno basado en el empleo de lexemas y otro en técnicas de Machine Learning, para identificar los sentimientos expresados en los textos. En la gran mayoría de estos trabajos se utilizan recursos traducidos de forma automática generados para otros idiomas, como el inglés, lo cual manifiesta una escasez de recursos genuinos para el lenguaje español [16].

Existen también trabajos sobre perfilado de autor, en los que se menciona la dificultad de encontrar colecciones de textos adecuadamente etiquetados y con poco ruido [17]. A partir de eso se han producido trabajos tendientes al desarrollo de conjuntos de textos en español específicos para esta tarea [18].

Al mismo tiempo pueden encontrarse algunos trabajos que proponen adaptaciones o desarrollos de nuevos algoritmos de *stemming* [19] específicos para tratar textos en español. En la mayoría de estos trabajos se resalta la necesidad de contar con algoritmos que tengan en cuenta las particularidades que pueden presentarse en el lenguaje español como pueden ser, una mayor complejidad gramatical y los cambios que pueden producirse en la raíz de las palabras, y no solo en las terminaciones como suele suceder en el inglés.

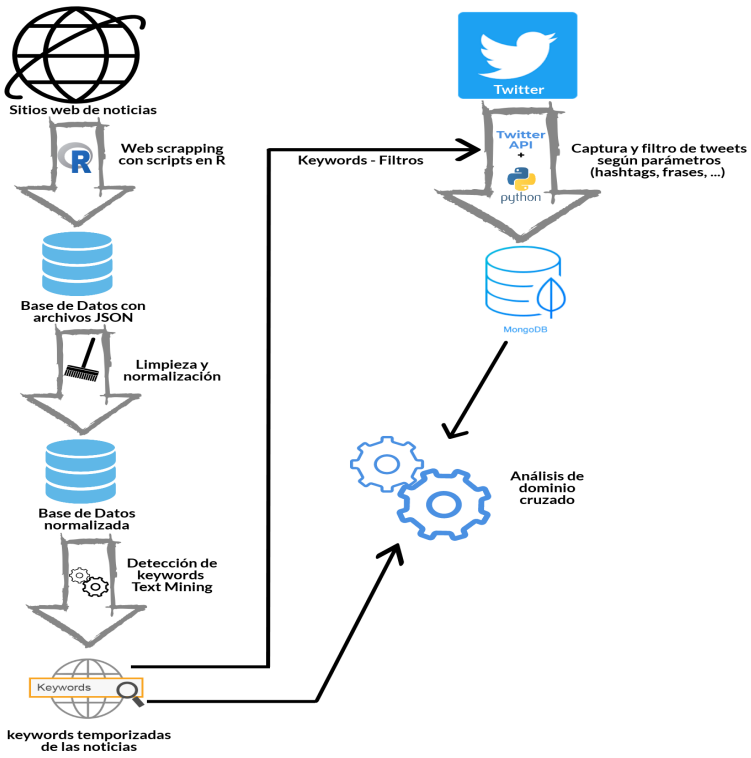

Figura 1. Esquema Diarios-Twitter

3. Estructura del proyecto

El estudio de establecimiento de agenda (del inglés Agenda Setting) se orienta a determinar los efectos de las noticias o temáticas que abordan los medios de comunicación en la opinión pública, mediante el análisis de fuentes alternativas de información (redes sociales, periódicos digitales, portales de noticias). La metodología a seguir en este proyecto consiste en realizar la extracción de palabras o conceptos clave de noticias publicadas en diarios electrónicos y comparar con publicaciones en redes sociales (en particular Twitter). La comparación intenta determinar si para cierto período de tiempo existen coincidencias en las temáticas que se abordan en ambos sectores, o si los medios periodísticos logran instalar temas en la sociedad, cuestión que puede visualizarse con la generación de *tweets* del mismo tema.

3.1. Recolección de Datos

Una tarea clave en todos los proyectos de ciencia de datos es la obtención o recolección de los datos a estudiar. Previo a la realización de dicha tarea se deben determinar las fuentes de datos y los datos o atributos relevantes a recolectar, y las estrategias para recolectar, verificar, filtrar y almacenar dichos datos. En este proyecto se plantean dos esquemas de análisis:

- Esquema Diarios-Twitter: consiste en obtener en primer lugar noticias de medios periodísticos digitales de Argentina, para un período de tiempo determinado, y determinar las palabras o conceptos claves que aparecen en dichas noticias. Posteriormente se toman dichos conceptos clave y se utilizan como filtros en la recolección de tweets, para medir si existe un impacto o instalación de temas por parte de los medios periodísticos en los ciudadanos y, en caso afirmativo, realizar distintos análisis como: por cuánto tiempo permanece activo, en qué grupos de usuarios, entre otros (ver Figura 1).
- Esquema Twitter-Diarios: se plantea en el sentido inverso, es decir, se recolectan en primer lugar *tweets* durante un período de tiempo determinado, se identifican las tendencias o términos clave que aparecen en los *tweets* recolectados, para luego utilizarlos como filtros en la recolección de noticias periodísticas en un período de tiempo igual o mayor al empleado en la recolección de *tweets*. En este enfoque se busca determinar si existe coincidencia en las opiniones de los ciudadanos y las

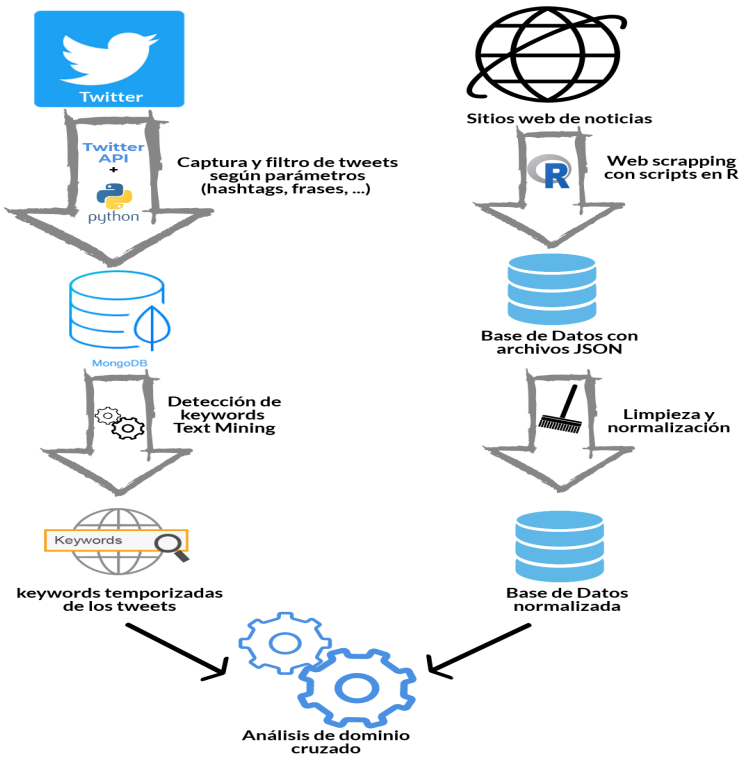

Figura 2. Esquema Twitter-Diarios

noticias publicadas en los periódicos, y además, determinar si las noticias publicadas en los medios digitales son en alguna ocasión generadas a partir de una temática instalada originalmente en las redes sociales (ver Figura 2).

Para la recolección de noticias de periódicos se utiliza la técnica de web scrapping. Se diseñó un script en R que descarga las noticias de los periódicos argentinos La Nación, Clarín e Infobae, de las secciones Espectáculos, Sociedad, Tecnología, Economía, Deportes y Política (Figura 3). Por cada sección de cada diario se genera un archivo JSON que contiene todas las noticias publicadas en dicha sección para una fecha determinada.

3.2. Preprocesamiento de los datos

Luego de la recolección y almacenamiento de los datos, se realizan dos procesos de limpieza de los archivos JSON:

- Eliminación de *stopwords*: artículos, preposiciones u otros componentes del texto que no tiene sentido mantener para el análisis.
- Detección de palabras clave: identificación automática del conjunto de términos que mejor describen el documento. En esta instancia se selecciona un método de extracción de palabras clave y se aplica sobre el conjunto de archivos JSON que contienen las noticias.

4. Aplicación Tweets Harvester

En el Esquema Diarios-Twitter, la lista de palabras o términos clave confeccionada en la primer parte del proceso, es utilizada como insumo para la recolección de tweets, que se realiza mediante la aplicación Tweets Harvester, desarrollada en el marco de este proyecto. Esta aplicación web realiza la recolección de tweets en forma automática según determinados parámetros que funcionan como filtros, que se aplican en el momento de la captura. Los filtros pueden ser palabras clave, hashtags o nombres de usuarios específicos de la red social.

4.1. Requerimientos

Para la construcción de la aplicación se plantearon los siguientes requerimientos funcionales y no funcionales:

- La arquitectura debe ser capaz de capturar, procesar, limpiar y almacenar 150 tweets por segundo.

```r
#install.packages('rvest')
#install.packages('jsonlite')
library(rvest)
library(jsonlite)

date <- format(Sys.time(), "%d%m%Y")
news_paper <- 'clarin'
categories <- c(politica='politica',
                deportes='deportes',
                economia='economia',
                tecnologia='tecnologia',
                sociedad='sociedad',
                espectaculos='espectaculos')
base_url <- 'https://www.clarin.com/'
news <- list()

for(category in categories){

  # Establezco el nombre del archivo para guardar las noticias
  file_name <- paste(news_paper, '_', names(which(categories == category)), '_', date, '.json', sep = '')

  # Descargo la página principal de la categoría en la sección "Últimas Noticias" para
  # asegurarme de que sean del día
  webpage <- read_html(paste(base_url, 'ultimas-noticias/', category, sep = ''))

  # Tomo el bloque donde estan las noticias propiamente dichas
  html_news <- html_nodes(webpage,'.col-lg-12')%>%
    html_nodes('.list-format')%>%
    html_node('a')

  # Extraigo los links de cada noticia
  news_urls <- html_attr(html_news, 'href')

  # Por cada url
  for(url in news_urls){
```

Figura 3. Extracto del código para web scraping desarollado en R

- La plataforma debe ser escalable y tolerante a fallas, como también presentar características de alta disponibilidad.
  - Escalabilidad. Se refiere tanto a la capacidad de incrementar fácilmente la cantidad de trabajo a realizar por el sistema, como a la posibilidad de incluir fácilmente nuevos nodos.
  - Tolerancia a fallas y alta disponibilidad. Expresa la posibilidad de poder continuar con el procesamiento sin interrupción, incluso ante fallas que puedan suceder en el funcionamiento de las tareas, tales como problemas de conexión, capacidad de la red, errores de lectura de los tweets, capacidad de procesamiento, entre otros.
- Seguridad. Brindar acceso a la gestión de los procesos de captura de tweets solo a usuarios autorizados.
- Interfaz gráfica accesible vía dispositivos móviles o desktop, sin que esto afecte la experiencia de usuario.
- Se debe poder ejecutar la arquitectura propuesta tanto sobre hardware tipo commodity (ej. computadora de escritorio) como en un servidor en producción.
- Los componentes de la arquitectura deben ser libres de licenciamiento.
- Las herramientas a utilizar deben poder conectarse fácilmente entre sí, además de proveer simplicidad de configuración para dar soporte a la comunicación entre ellas.

4.2. Arquitectura de la aplicación y herramientas

A partir de los requerimientos planteados se realizó un primer diseño de la solución y un proceso de investigación para la posterior elección de las distintas herramientas a utilizar. Luego de la selección de dichas herramientas se diseñó la arquitectura de la aplicación

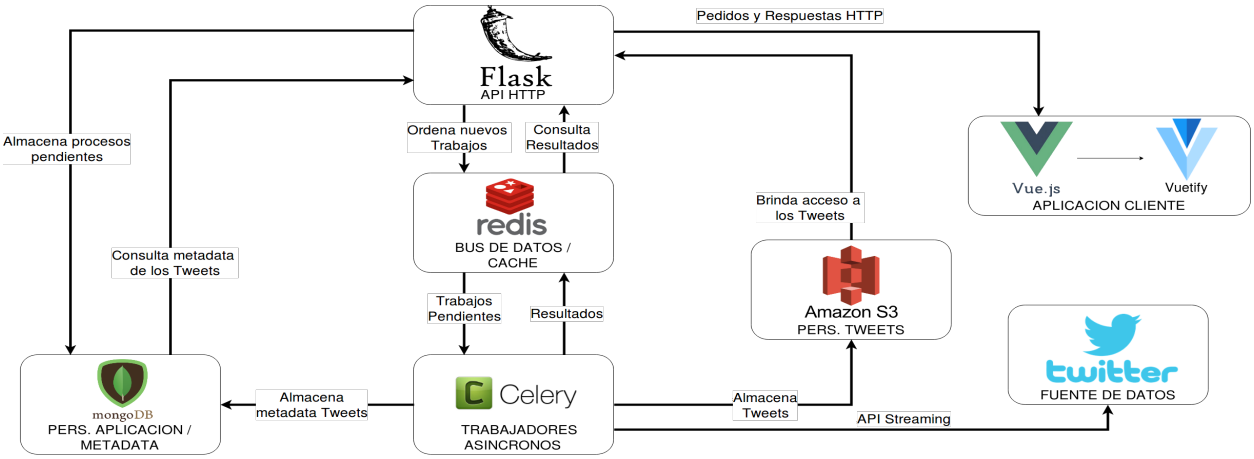

Figura 4. Arquitectura de la aplicación Tweets Harvester

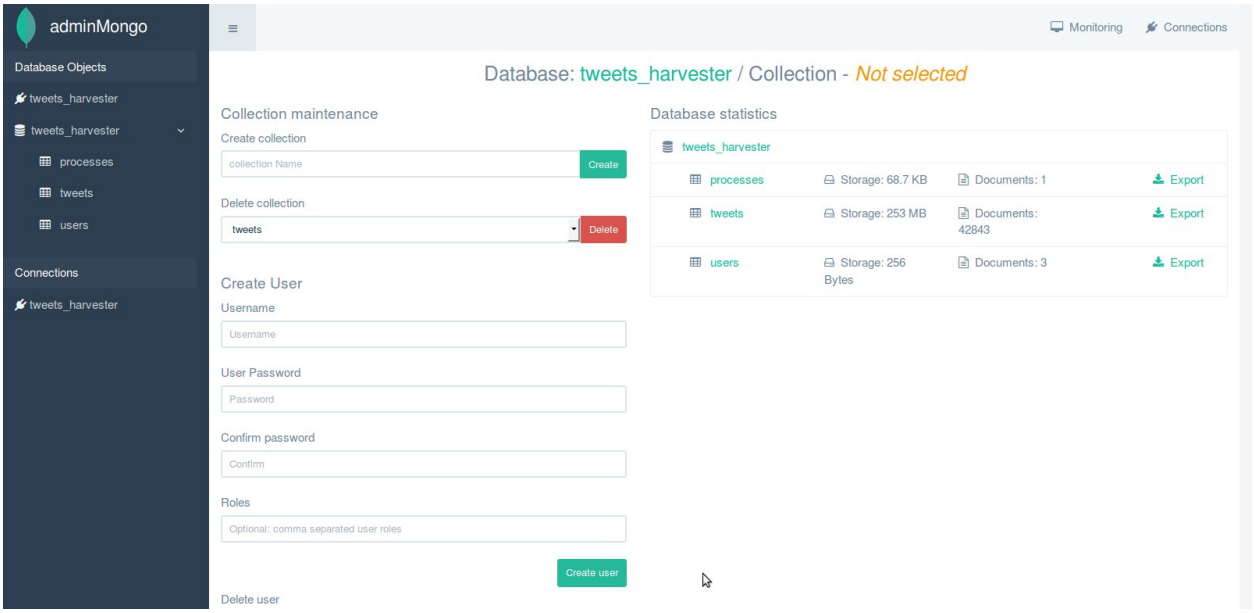

Figura 5. Interfaz de administración de la base de datos Mongo.

en la que se especificaron los módulos, sus interacciones y las herramientas a utilizar (Figura 4).

La arquitectura propuesta sigue el modelo cliente-servidor. El cliente consiste en una WebApp accesible desde un dispositivo móvil o navegador web que interactúa con una API, que permite listar y visualizar el estado actual del proceso de recolección de tweets, y configurar, pausar o poner en marcha el mismo. En el cliente se utiliza el framework de JavaScript Vue.js [20] en conjunto con el framework de estilos por componentes Vuetify [21].

Este cliente se comunica con un servidor compuesto por una API desarrollada en Python [22], utilizando el micro-framework Flask [23], que funciona como interlocutor de los diferentes servicios utilizados. Para la persistencia de los datos de procesos se utiliza MongoDB [24], para la persistencia de los tweets capturados, Amazon Web Service 3 [25], y el gestor de tareas Celery [26], para ejecutar las tareas de recolección, procesado y almacenamiento en segundo plano. En la comunicación con Celery se utiliza el motor de base de datos Redis [27] para mantener una cola de mensajes, con el objetivo de iniciar o detener las tareas. Estos servicios se despliegan en containers Docker [28], junto con interfaces de usuario, para su mejor monitoreo y control.

4.3. Implementación de la aplicación

Tomando como base la arquitectura diseñada, las herramientas seleccionadas y los requisitos planteados, se realizó la implementación de la aplicación. Para cumplir con el requisito de capturarlos en tiempo real y de capturar gran volumen de datos, se hizo uso de la Search API que ofrece Twitter [29]

para tal fin. Para el desarrollo del código de la aplicación se utilizó el IDE Visual Studio Code [30], y para llevarlo a cabo de forma conjunta y teniendo control de versiones se utilizó la plataforma GitHub [31]. Teniendo en cuenta la necesidad de manejar grandes volúmenes de datos de entrada y posteriormente procesarlos, se diseñaron las tareas de captura, procesamiento y limpieza, y almacenamiento de forma separadas para que se ejecuten de forma independiente. El trabajo realizado involucró las siguientes actividades:

- Diseño y construcción de los entornos de desarrollo y producción reproducibles utilizando Docker [28]. Esto implicó la configuración de cada una de las herramientas y bases de datos a utilizar. En la Figura 5 se observa la interfaz de administración de la base de datos Mongo, donde se pueden ver los usuarios, procesos y tweets almacenados.
- Diseño y desarrollo de una API que interactúe con una aplicación web, recibiendo peticiones y devolviendo los datos y, a su vez, con la API de Twitter [29] para la recolección de los tweets (Figura 6).
- Diseño y desarrollo una WebApp accesible desde un dispositivo móvil o navegador web que interactúe con la API antes definida, que permita listar y visualizar el estado actual del proceso de recolección de tweets y configurar, pausar o poner en marcha el mismo.

Se decidió que hasta que la aplicación se encuentre en producción no se utilizará el servicio de Amazon Web Service 3 para la persistencia de los tweets capturados. Debido a esto, la base de datos Mongo DB se utiliza, tanto para la persistencia de los datos de procesos y usuarios, como para la persistencia de los tweets (ver Figura 5).

5. Metodología

En la gestión de las actividades del proyecto se utilizó una propuesta metodológica híbrida de desarrollo propio, pensada para proyectos de ciencia de datos, que se encuentra en proceso de validación. Esta propuesta metodológica surge a raíz de la necesidad de agilizar las metodologías creadas para la gestión de este tipo de proyectos. En [32] se argumenta que las metodologías tradicionales no se han adaptado a las exigencias que presenta la gestión de proyectos de ciencia de datos, ya que se centran únicamente en la mejora de las técnicas de extracción y análisis de datos. La propuesta utilizada en este proyecto se basa en dos metodologías de gestión de proyectos:

- La primera de ellas es CRISP-DM (Cross-Industry Standard Process for Data Mining) [33]: una metodología de enfoque tradicional especialmente desarrollada para proyectos de Minería de Datos, actualmente considerada como un estándar en el mundo empresarial. Propone un conjunto de actividades que hay que llevar a cabo en el desarrollo de un proyecto de minería de datos y cada una de las actividades se divide distintas tareas.
- La segunda metodología es una propuesta para la gestión de proyectos TIC diseñada y probada en otro contexto [34]. Es un enfoque híbrido, ya que surge a partir de la fusión de dos guías de buenas prácticas en gestión de proyectos (PMBOK [35], de enfoque tradicional, y ATERN [36], de enfoque ágil) con las metodologías de gestión de proyectos PRINCE2 [37], de enfoque tradicional, y SCRUM [38] y APM [39] de enfoque ágil.

En consecuencia, la metodología propuesta se divide en tres fases: definición y planificación, ejecución y control y, evaluación final y cierre, basados en el ciclo de vida de PMI [35]. El ciclo de vida de la metodología propuesta para la gestión de proyectos de ciencias datos se muestra en la Figura 7. La misma complementa los lineamientos tradicionales propuestos por CRISP-DM con los ágiles, siguiendo fuertemente los pasos de la metodología ágil SCRUM. Esta propuesta de enfoque híbrido que integra las fortalezas de metodologías tradicionales y ágiles, resulta atractiva para proyectos de minería de datos, dado que: permite optimizar el tiempo y recursos utilizados, no requiere formación específica por su sencillez, sólo se genera la documentación necesaria y la información requerida para asegurar una comunicación eficiente y no incurrir en los mismos errores a partir de las lecciones aprendidas.

6. Conclusiones y Trabajo futuro

En este artículo se presenta un proyecto que trabaja sobre la Teoría de la Fijación de Agenda. La estrategia elegida para abordar dicha teoría consiste en indagar

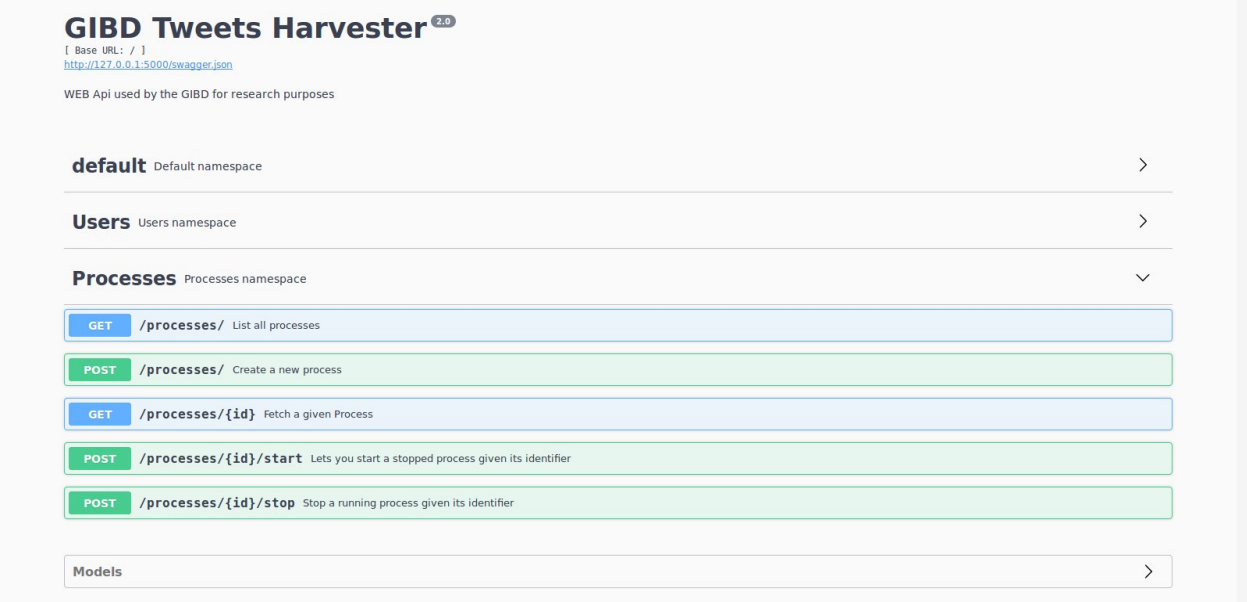

Figura 6. Web API desarrollada.

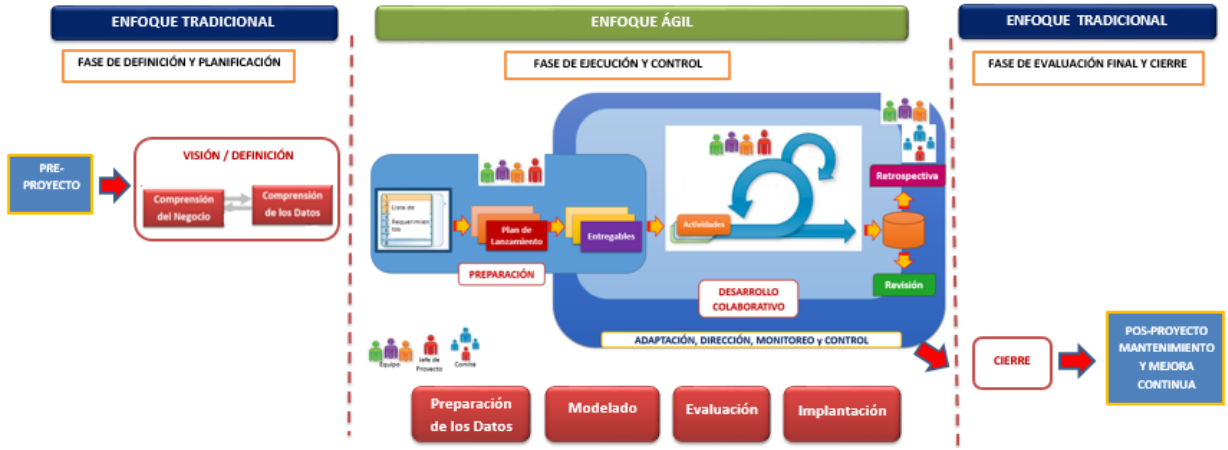

Figura 7. Metodologia empleada en la gestión del proyecto

noticias periodísticas publicadas en periódicos digitales de Argentina, determinar los conceptos o palabras claves presentes en las noticias y posteriormente verificar si dichos conceptos claves se mencionan en las publicaciones de usuarios en redes sociales, durante cuánto tiempo permanecen vigentes los conceptos detectados, qué grupos de usuarios los mencionan, entre otros factores de análisis. En particular se han seleccionado los periódicos Clarín, La Nación e Infobae y la red social Twitter. El proyecto consta de varias etapas. Hasta el momento se han podido completar los mecanismos para realizar las tareas de recolección de datos: para la recolección de noticias de periódicos se han diseñado scripts en lenguaje R para aplicar web scrapping sobre los sitios de los periódicos digitales, y para la recolección de tweets mediante la especificación de ciertos parámetros de usuario, se ha desarrollado la aplicación Tweets Harvester. Se ha avanzado en la aplicación de técnicas de minería de textos para la determinación de conceptos clave en los textos no estructurados para poder establecer los filtros de búsqueda en Twitter. Actualmente se está desarrollando la interfaz web de la aplicación. Durante el desarrollo del proyecto se concretó la incorporación de un servidor propio, que será utilizado como plataforma para el almacenamiento de las noticias y

los tweets, ya que soporta los niveles de crecimiento de datos previstos en el proyecto y provee una capacidad de procesamiento adecuada. Posteriormente se deberá desarrollar la etapa de análisis de resultados obtenidos, que busca responder preguntas como: ¿logran instalar los medios periodísticos temáticas en los usuarios de redes sociales? ¿la aparición de determinados tópicos en ambos dominios ocurre en simultáneo? ¿la desaparición de un tema de los periódicos implica que desaparezca de las menciones de usuarios en Twitter?. En el marco de este proyecto se ha utilizado una metodología de enfoque híbrido adaptada para proyectos de ciencias de datos, que ha permitido gestionar los tiempos y actividades del proyecto de una manera eficiente y ha permitido lograr resultados y su correspondiente validación en una tiempo relativamente acotado.

Referencias

[1] E. Castro, A. Iglesias, P. Martínez, and L. Castaño, "Automatic identification of biomedical concepts in spanish-language unstructured clinical texts," in *Proceedings of the 1st ACM International Health Informatics Symposium*, ser. IHI '10. New York, NY, USA: ACM, 2010, pp. 751–757. [Online]. Accesible en: http://doi.acm.org/10.1145/1882992.1883106

[2] S. Abhyankar, D. Demner-Fushman, F. M. Callaghan, and C. J. McDonald, "Combining structured and unstructured data to identify a cohort of icu patients who received dialysis," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 801–807, 2014. [Online]. Accesible en: http://dx.doi.org/10.1136/amiajnl-2013-001915

[3] A. Ittoo, L. M. Nguyen, and A. van den Bosch, "Text analytics in industry: Challenges, desiderata and trends," *Computers in Industry*, vol. 78, pp. 96–107, 2016.

[4] S. Beliga, "Keyword extraction : a review of methods and approaches," 2014.

[5] L. Khan and W. Fan, "Tutorial: Data stream mining and its applications," in *Database Systems for Advanced Applications*, S.-g. Lee, Z. Peng, X. Zhou, Y.-S. Moon, R. Unland, and J. Yoo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 328–329.

[6] A. Ittoo, L. M. Nguyen, and A. van den Bosch, "Text analytics in industry: Challenges, desiderata and trends," *Computers in Industry*, vol. 78, pp. 96–107, 2016. [Online]. Accesible en: https://doi.org/10.1016/j.compind.2015.12.001

[7] M. Ben-Dov and R. Feldman, "Text mining and information extraction," in *Data Mining and Knowledge Discovery Handbook*. Springer, 2005, pp. 801–831.

[8] M. Allahyari, S. A. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "A brief survey of text mining: Classification, clustering and extraction techniques," *CoRR*, vol. abs/1707.02919, 2017. [Online]. Accesible en: http://arxiv.org/abs/1707.02919

[9] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 39.

[10] S. Ceri, A. Bozzon, M. Brambilla, E. D. Valle, P. Fraternali, and S. Quarteroni, *Web Information Retrieval*. Springer Publishing Company, Incorporated, 2013.

[11] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *CoRR*, vol. abs/1708.05148, 2017. [Online]. Accesible en: http://arxiv.org/abs/1708.05148

[12] M. Moens, *Information Extraction: Algorithms and Prospects in a Retrieval Context*, ser. The Information Retrieval Series. Springer, 2006, vol. 21. [Online]. Accesible en: http://link.springer.com/book/10.1007\%2F978-1-4020-4993-4

[13] B. Kimelfeld, "Database principles in information extraction," in *Proceedings of the 33rd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, ser. PODS '14. New York, NY, USA: ACM, 2014, pp. 156–163. [Online]. Accesible en: http://doi.acm.org/10.1145/2594538.2594563

[14] M. Allahyari, S. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text Summarization Techniques: A Brief Survey," Jul. 2017. [Online]. Accesible en: http://arxiv.org/abs/1707.02268

[15] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002. [Online]. Accesible en: http://doi.acm.org/10.1145/505282.505283

[16] C. H. A.-q. Miranda and J. GuzmÃ¡n, "A Review of Sentiment Analysis in Spanish," *Tecciencia*, vol. 12, pp. 35 – 48, 06 2017. [Online]. Accesible en: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1909-36672017000100035&nrm=iso

[17] M. J. Garciarena Ucelay, M. P. Villegas, L. Cagnina, and M. L. Errecalde, "Cross domain author profiling task in spanish language: an experimental study," *Journal of Computer Science and Technology* , vol. 15, pp. 122 – 128, 11 2015. [Online]. Accesible en: http://sedici.unlp.edu.ar/handle/10915/50079

[18] M. P. Villegas, M. J. Garciarena Ucelay, M. L. Errecalde, and L. C. Cagnina, "A Spanish text corpus for the author profiling task," *Proc. del XX Congreso Argentino de Ciencias de la Computación* , 10 2014. [Online]. Accesible en: http://sedici.unlp.edu.ar/handle/10915/42290

[19] R. Leon, A. Honrado, R. O'Donnell, and D. Sinclair, "A word stemming algorithm for the spanish language." in *7th International Symposium on String Processing and Information Retrieval (SPIRE)*, String Processing and Information Retrieval (SPIRE). Los Alamitos, CA, USA: IEEE Computer Society, 2000, pp. 139–145.

[20] Vue.js. Framework javascript orientado a componentes. https://vuejs.org/.

[21] Vuetify. Framework orientado a componentes basado en el estándar de material design. https://vuetifyjs.com/en/.

[22] Python. Lenguaje de desarrollo. https://www.python.org/.

[23] Flask. Micro framework para python. http://flask.pocoo.org/.

[24] MongoDB. Gestor de base de datos nosql. https://www.mongodb.com/.

[25] Amazon. Servicio de almacenamiento en la nube. https://aws.amazon.com/es/s3/.

[26] Celery. Servicio que brinda colas de tareas distribuídas. http://www.celeryproject.org/.

[27] Redis. Motor de base de datos y broker de mensajes. https://redis.io/.

[28] Docker. Tecnología de containerización. https://www.docker.com/.

[29] Twitter. Api para la extracción de contenido de twitter. https://developer.twitter.com/.

[30] V. S. Code. Entorno de desarrollo integrado. https://code.visualstudio.com/.

[31] GitHub. Plataforma de control de versiones. https://github.com/.

[32] J. S. Saltz, "The need for new processes, methodologies and tools to support big data teams and improve big data project effectiveness," in *2015 IEEE International Conference on Big Data (Big Data)*, Oct 2015, pp. 2066–2071.

[33] G. Piatetsky, "Crisp-dm, still the top methodology for analytics, data mining, or data science projects." KDnuggets, Oct 2014. [Online]. Accesible en: https://goo.gl/WMfCfB

[34] A. M. Cristaldo P, Ballejos L, "Un enfoque híbrido de gestión de proyectos de tics en el sector público," 2015. [Online]. Accesible en: http://www.sedici.unlp.edu.ar/handle/10915/52408

[35] in *PMBOK Guide. A Guide to the Project Management Body of Knowledge. 5 Ed.* NewtownSquare, Pennsylvania, USA: Project Management Institute, 2013.

[36] DSDM. Dsdm atern handbook (2008). http://www.dsdm.org/dig-deeper/book/dsdm-atern-handbook.

[37] A. Böhm, *Application of PRINCE2 and the Impact on Project Management*.

[38] J. Schwaber, K.; Sutherland, *The Scrum Guide, the Definitive Guide to scrum: The Rules of the Game*, 2011. [Online]. Accesible en: http://www.scrum.org/Portals/0/Documents/ScrumGuides/Scrum_Guide.pdf

[39] J. Highsmith and J. Highsmith, *Agile Project Management: Creating Innovative Products*, ser. Agile software development series. Addison-Wesley, 2010. [Online]. Accesible en: https://books.google.com.ar/books?id=BcWPmAEACAAJ