72.45 - Proyecto Final

Instituto Tecnológico de Buenos Aires

NewsAPI-AR: Un servicio para análisis y visualización de noticias de Argentina

---

Integrantes: Oliver Balfour, Marcos Ariel Atar

Legajos: 55177, 57352

Tutor: Lic. Diego Ariel Aizemberg

Fecha: 2022-05

Resumen: *Implementación de una API que pone a disposición información de noticias de los medios más importantes de Argentina, estas noticias se enriquecen usando un algoritmo de reconocimiento de entidades nombradas.*

Palabras claves: *Reconocimiento de entidades nombradas, Procesamiento de lenguajes naturales, Visualización de la información, Aprendizaje automático.*

Abstract: *Implementation of an API that makes available news information from the most important media in Argentina, these news are enriched using a named-entity recognition algorithm.*

Keywords: *Named-entity recognition, Natural language processing, Information visualization, Machine Learning.*

Índice

1. Introducción

El objetivo del proyecto es desarrollar un servicio web gratuito que pone a disposición información acerca de noticias de Argentina. Para ello, se creó una aplicación que interpreta archivos RSS de noticias de los medios más importantes y almacena en una base de datos la información más relevante.

Las noticias almacenadas son enriquecidas utilizando un servicio de reconocimiento de entidades nombradas y se encuentran disponibles a través de una API para realizar consultas.

Se crearon una serie de visualizaciones que representan posibles usos de la API para el análisis de datos de noticias y medios.

2. Origen del proyecto

El proyecto esta basado en dos proyectos finales anteriores desarrollados por alumnos del Instituto Tecnológico de Buenos Aires: XDATA[1] [1] y NERd API[2] [2].

El primer proyecto, XDATA, consiste en un portal web, focalizado en la visualización de noticias. El mismo incluye una interfaz para filtrar, agrupar y exponer gráficamente la información de las noticias para su posterior análisis.

En el segundo proyecto se desarrolló una API que interactúa con su propio modelo entrenado de procesamiento natural del lenguaje. El mismo cuenta con una herramienta para el reconocimiento de entidades en un texto dado como parámetro.

El principal concepto en torno al cual se desarrolló el proyecto fue mejorar el proyecto XDATA ampliando sus funcionalidades para que incluya el reconocimiento de entidades utilizando el modelo entrenado de la NERd API y realizar visualizaciones con las entidades reconocidas en los titulares de las noticias.

A lo largo del informe veremos como trabajamos en base a ese concepto hasta desarrollar nuestra propia API, similar a la utilizada en el proyecto XDATA pero utilizando diferentes tecnologías e implementando nuevas rutas (*endpoints*) que incluyen el reconocimiento de entidades en los titulares utilizando la NERd API.

[1] http://pf2.it.itba.edu.ar
[2] https://nerd.it.itba.edu.ar

3. Estado del arte

Desde el lanzamiento de su versión beta en 2002, Google News[3] ha servido como fuente de gran cantidad de proyectos de análisis y visualización de noticias. Tal es el caso de newsmap[4], una aplicación que utiliza las noticias de Google News para generar un treemap. Cada temática de noticias es representada por un color distinto, la relevancia de la noticia esta representada por su tamaño y que tan reciente es se representa con la intensidad del color. NewsMap.JS[5] es una versión mas moderna inspirada en este sitio que ofrece una mayor compatibilidad con navegadores actuales.

A finales del 2020, el equipo conformado por Francisco Hanna, Hernan Rocha, Diego Fernandez Lecler y Carlos Ku, lanzan una beta de Jornalia[6] con accesos limitados, es una API JSON que recolecta y pone a disposición noticias de más de 50 medios de Argentina.

En el plano internacional, hay dos proyectos de recolección de datos de noticias que se destacan: News API[7] y GNews[8]. Ambos surgen como alternativa al cierre de Google News API (una API que permitia hacer consultas sobre las noticias de Google News en formato JSON) y ambas son pagas al igual que Jornalia. En el caso de News API, los costos llegan a 1.749 US$ por mes.

También cabe destacar que hay un proyecto llamado Stanford Cable TV News Analyzer[9] de la Universidad de Stanford en colaboración con el programa de becas John S. Knight con el objetivo de analizar las noticias emitidas por medios de televisión por cable. Los datos se obtienen del Archivo de Internet de noticias de TV[10] que tiene grabaciones de los principales canales de noticias desde el 2010 hasta la actualidad.

Medios de gran jerarquía, como *The Guardian* cuentan con un servicio gratuito limitado y otro pago, en este caso llamado *Open Platform*[11], para acceder a sus noticias utilizando una API.

Finalmente, es notable mencionar los servicios que provee Croma Ai[12], una empresa que cuenta con modelos de Machine Learning para medios en español. Entre sus soluciones se puede utilizar una API basada en JSON para buscar recomendaciones de noticias similares o para buscar entidades dentro de una noticia.

[3] https://news.google.com
[4] http://newsmap.jp
[5] https://newsmap-js.herokuapp.com
[6] https://docs.jornalia.net/
[7] https://newsapi.org
[8] https://gnews.io
[9] https://tvnews.stanford.edu
[10] https://archive.org/details/tv
[11] https://open-platform.theguardian.com
[12] https://croma.ai/

4. Requerimientos funcionales

Nuestro proyecto tenía como objetivo principal implementar los mismos *endpoints* del proyecto XDATA[13][1] agregando el uso de la NERd API[14][2] para reconocer entidades en los títulos de las noticias, implementando nuevos *endpoints* que pongan a disposición esta nueva funcionalidad.

Esta nueva implementación se desarrolló en *Javascript* utilizando el entorno de ejecución de *Node.js* y como base de datos *PostgreSQL*.

5. Investigación

Para el desarrollo del servicio web se realizó una serie de pruebas sobre distintas herramientas y funcionalidades que podrían implementarse para añadir valor al modelo del proyecto [1] que luego fueron descartadas y reemplazadas por el modelo actual.

5.1. GraphQL

Se consideró la opción de reemplazar la REST API por GraphQL[15]. Esta opera a modo de *query language*, donde se realizan las consultas a la base de datos a través de un solo *endpoint*.

Esta herramienta desarrollada por los ingenieros de *Facebook* tiene dos grandes ventajas. La primera es evitar el *overfetching*, es decir que el cliente descarga más información que la que se necesita para la aplicación. En segundo lugar, evitar el *underfetching*, esto significa evitar que un *endpoint* específico no provea suficiente información como la que uno necesita, por lo que el cliente tendrá que realizar consultas adicionales para solicitar todo lo que necesita.

GraphQL fue implementado e integrado con la base de datos pero se lo descarto para conservar las rutas del proyecto anterior con una REST API implementada con *Express.js*.

[13] http://pf2.it.itba.edu.ar
[14] https://nerd.it.itba.edu.ar
[15] https://graphql.org

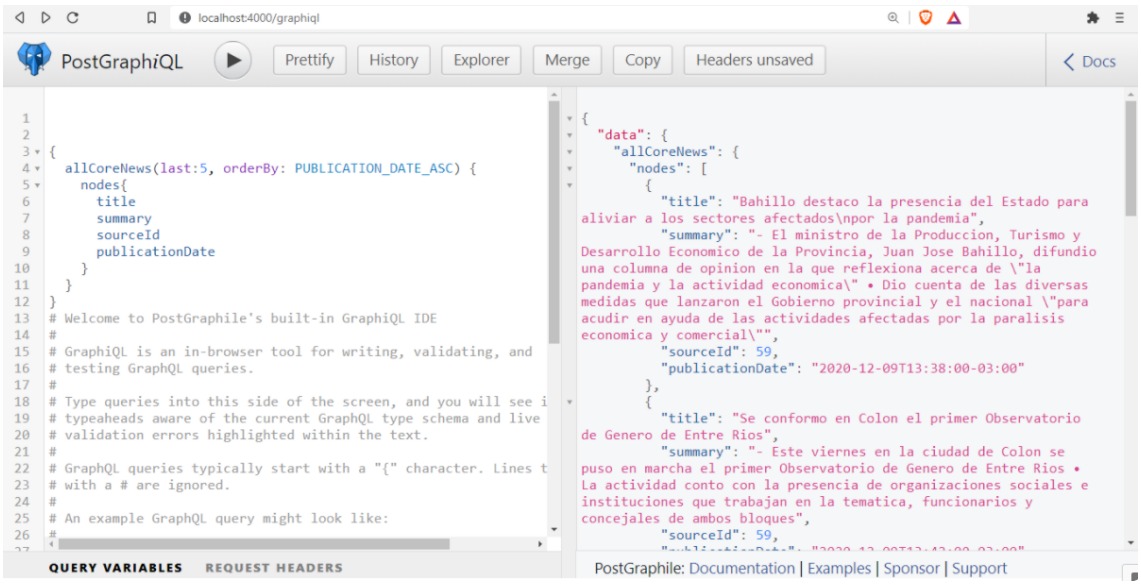

Fig. 1: Interfaz gráfica para realizar las consultas al servidor de GraphQL de medios y noticias.

5.2. Web Scraping

Otra prestación que se consideró agregar fue utilizar *web scraping* para obtener todo el artículo y luego reconocer las entidades sobre la totalidad de la noticia en lugar de solamente sobre el titular, que viene incluido en el RSS. Se realizaron pruebas utilizando la librería Cheerio[16] para hacer este proceso en los sitios de noticias, donde se ingresaba a las direcciones URL obtenidas en el RSS y en base al medio se consideraba una estructura distinta de como podía extraerse los datos del artículo.

Las pruebas se realizaron en Observable[17] considerando tres medios (La Nación, Clarín e Infobae). Se concluyó no agregar esta mejora al proyecto debido a que las estructuras eran muy heterogéneas entre sí y bastante cambiantes, lo que dificultaba la mantenibilidad de la herramienta. Además, se considera que dentro del titular hay información suficiente para obtener entidades relevantes para clasificar la noticia.

[16] https://cheerio.js.org
[17] https://observablehq.com/@obalfour/rss-parser

Fig. 2: Entidades extraídas del cuerpo de un artículo utilizando [2].

5.3. Wikidata

Con el fin de obtener un nivel de detalle mayor que acompañe la información del titular y de las entidades detectadas, se realizaron pruebas para conectarse a Wikidata[18] para obtener datos adicionales. Por ejemplo, la NERd API puede reconocer a 'Alberto Fernández' como una persona, pero Wikidata nos podría agregar mas información de contexto[19].

El resultado de estas pruebas mostró que era posible obtener información de algunas entidades pero la complejidad de implementación era alta y sumado a esto, hubo otro inconveniente: el resultado de Wikidata podía ser nulo (por ejemplo, nombres de personas que aún no tienen un artículo en Wikipedia relacionado) u obtener varios resultados dependiendo del contexto (Por ejemplo personajes célebres con el mismo nombre pero profesiones distintas).

[18] https://www.wikidata.org
[19] https://www.wikidata.org/wiki/Q2642828

Fig. 3: Descripción obtenida de Wikidata sobre "Shakira", entidad detectada como persona.

5.4. RSS de Google News

Un aspecto importante en el desarrollo del proyecto fue obtener información de las noticias a través de RSS de medios. Para tener una mejor mantenibilidad se planteó tener una misma estructura para todas las noticias que iban a extraerse utilizando unos RSS que pueden generarse utilizando el portal de Google News, generando una URL especifica para tomar las noticias de cada medio.

Esto era posible, ya que el sitio permitía generar el RSS de una consulta de noticias de un sitio web específico, algo que se descubrió sin tener una documentación oficial al respecto. Se podía utilizar como reemplazo de Google News API si se genera un RSS por cada consulta que antes se hacia a través de la misma.

Si bien en un principio funcionó, los RSS de Google News cambiaron dejando obsoletos su uso tal como se había implementado, para poder seguir teniendo el proyecto funcionando se tuvo que volver a utilizar directamente los RSS de cada medio en particular, teniendo en cuenta las diferencias que pudieran existir entre cada uno.

6. Arquitectura

Para el desarrollo del proyecto se propuso una arquitectura en capas, separando la presentación de la lógica de negocio y la de datos. Los modelos de la base de datos se pensaron teniendo en cuenta los *endpoints* del modelo anterior, considerando que iban a almacenarse y consultarse también las entidades detectadas.

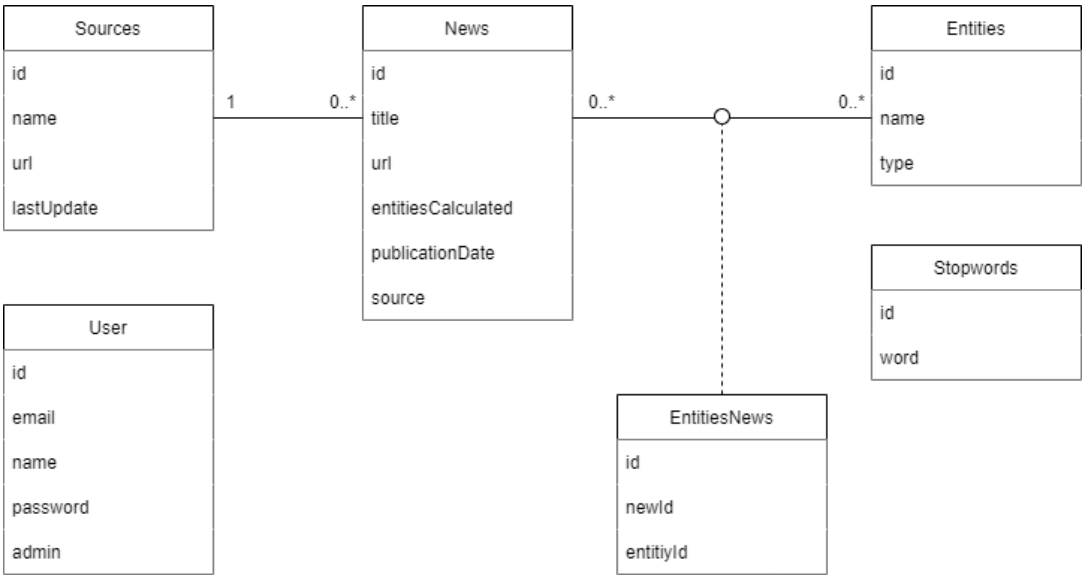

Fig. 4: Diagrama UML para el modelado de las clases del proyecto.

El contenido del modelo de **news** fue pensado teniendo en cuenta los atributos de las noticias que podían extraerse de los RSS. Las **entities** que se extraen son almacenadas teniendo en consideración su tipo (Lugares, Organizaciones, Personas y Misceláneos). Cada noticia tiene asociada un **sources**, que es el medio que publicó el artículo. En la base de datos se almacena todos los atributos del medio para extraer sus noticias diariamente. Por último, vale la pena mencionar que hay un modelo para almacenar **stopwords** (palabra vacías) que serán utilizadas para descartar palabras frecuentes que no aportan información relevante y otro modelo para tener **users** con acceso de escritura.

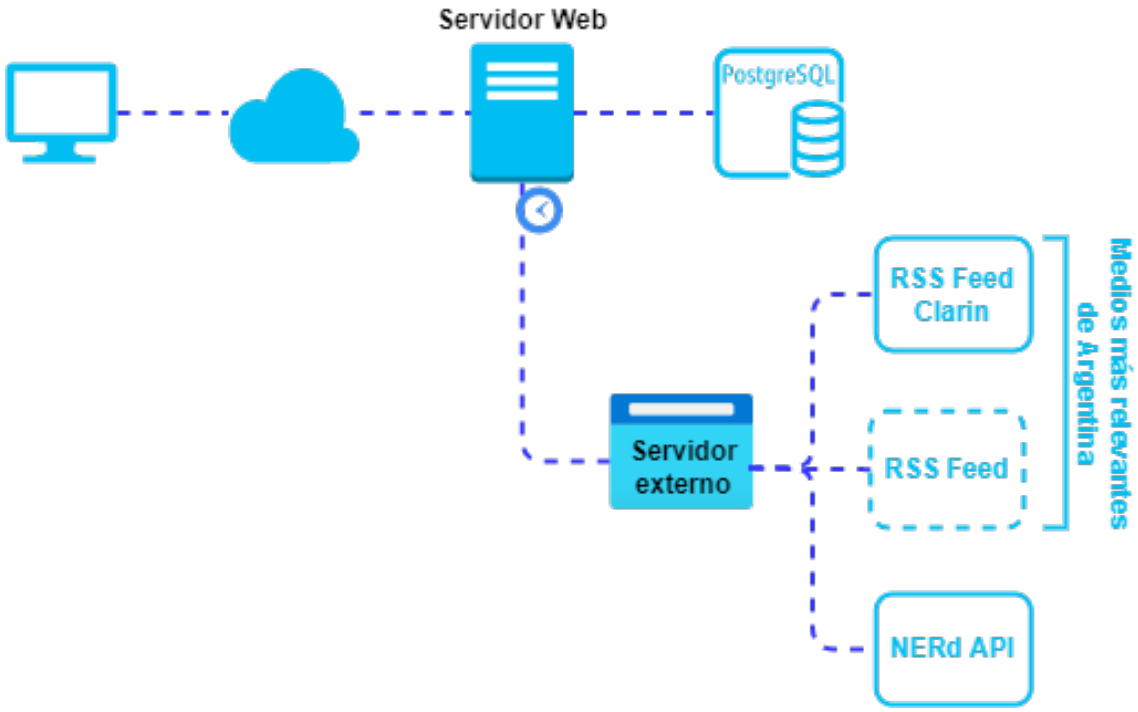

Fig. 5: Diagrama sobre la arquitectura propuesta para el desarrollo del proyecto.

Como muestra la figura 5, el usuario se comunica a través de Internet, a medida que interactúa con la interfaz web se van realizando distintas consultas a la base de datos alojada en un servidor propio. El servidor web, además de atender las consultas sobre la base de datos, realiza solicitudes periódicamente a servidores externos cada hora para extraer información de los RSS de los distintos medios y para consultar las entidades presentes en los titulares.

7. Casos de uso del proyecto

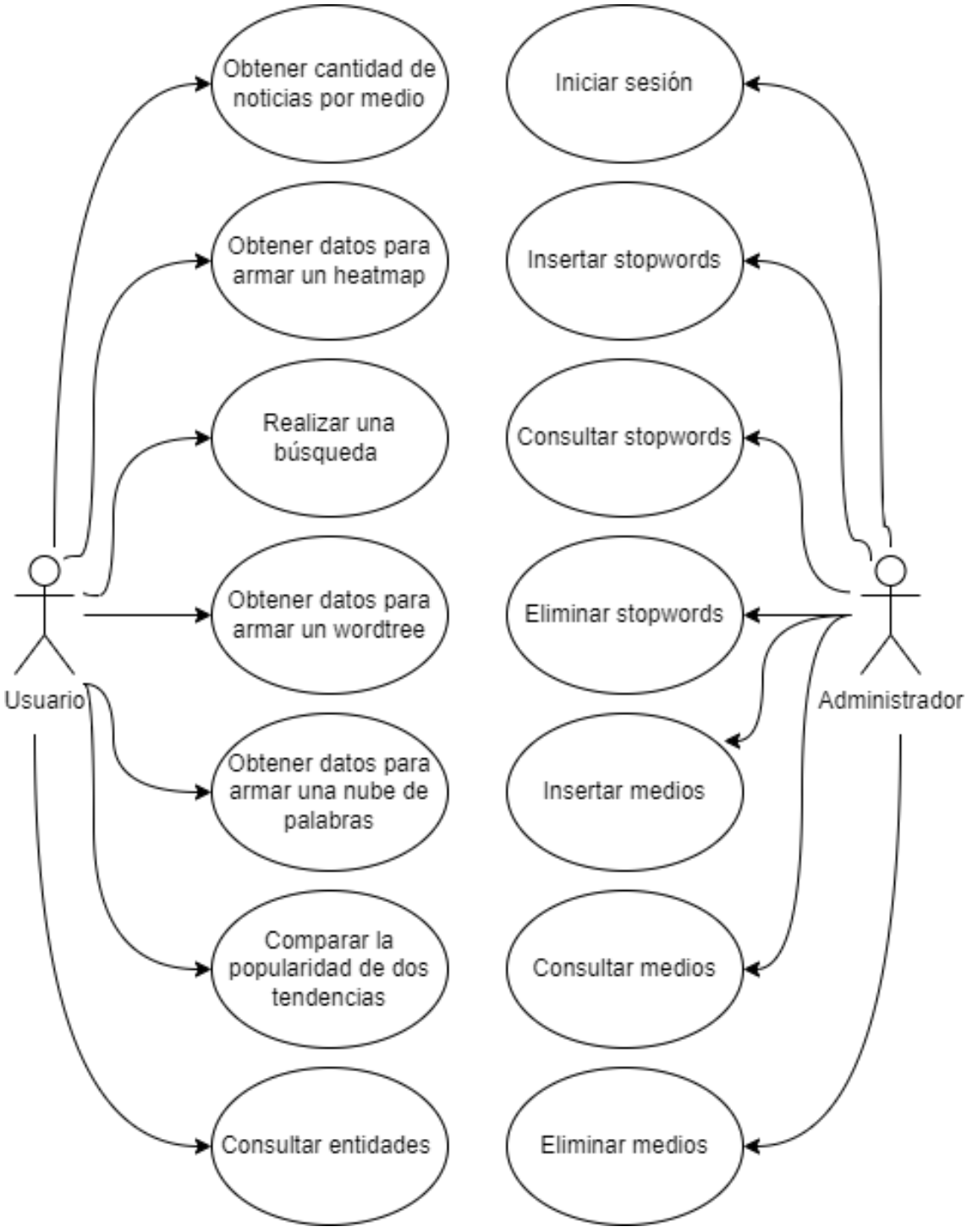

Fig. 6: Diagrama de casos de uso del proyecto.

El diseño de la API se pensó de manera tal que un usuario pueda realizar acciones para obtener información de la base de datos que le permita realizar análisis sobre los medios de Argentina.

Entre las acciones que pueden llevarse a cabo, se incluye poder realizar búsquedas de noticias en la base de datos, obtener datos para obtener distintos tipos de visualizaciones, comparación de la popularidad de tendencias, consultar la cantidad de noticias registradas en la base y las entidades detectadas en los titulares.

De la misma manera, un usuario administrador debería poder iniciar sesión, de modo que le permita tener acceso a funciones acciones adicionales. Una vez autenticado, un administrador debería poder insertar, consultar y eliminar medios o stopwords.

Las stopwords sirven para que no se tengan en cuenta palabras comunes (como por ejemplo pronombres) a la hora de armar la nube de palabras. Ademas, los medios sirven como fuente de donde obtener noticias para la API.

8. Documentación de la API

La API esta documentada y se puede probar en el Swagger del proyecto[20].

Las rutas principales para acceder a la información almacenada en la base de datos son las siguientes:

GET

- **nube-de-palabras:** genera un conjunto de palabras y para cada palabra cuenta su frecuencia absoluta.
- **cantidad-de-noticias:** devuelve la cantidad de noticias publicadas en una fecha.
- **heatmap:** genera un conjunto de medios por fecha y para cada medio cuenta la cantidad de noticias publicadas.
- **tendencias:** dado un conjunto de palabras y un rango de fechas, para cada palabra cuenta su frecuencia relativa al máximo en dicho periodo.
- **busqueda:** devuelve el resultado de una consulta personalizada.
- **wordtree:** genera un conjunto de titulares.
- **medios:** genera un conjunto con todos los medios que se encuentran registrados para consultar.
- **stopwords:** genera un conjunto con todas las palabras registradas como stopwords.

[20] https://newsapi.it.itba.edu.ar/api/v1/swagger/

- **entidades:** genera un conjunto con todas las entidades encontradas.

POST

- **login:** recibe las credenciales para el acceso a la aplicación como usuario.
- **insert-stopword:** recibe la stopword a registrar en la aplicación.
- **insert-medio:** recibe el medio a registrar en la aplicación.

DELETE

- **stopword:** elimina una stopword registrada.
- **medio:** elimina un medio registrado.

9. Visualizaciones

Con el objetivo de que un usuario final pueda observar los datos para un análisis detallado sobre los medios y noticias de Argentina se creó una colección de visualizaciones[21] que representan la información de la **NewsAPI-AR** de una forma dinámica, amigable e interactiva.

9.1. Entidades

Devuelve todas las entidades encontradas en un rango de fechas, cada color representa un tipo diferente de entidad (Localizaciones, Personas, Organizaciones y Misceláneos). Si no se especifica rango de fechas, devuelve las entidades del día.

Fig. 7: Nube de palabras con las entidades del 15 de Febrero de 2022.

[21] https://observablehq.com/collection/@obalfour/itba-news-api

9.2. Tendencias

Visualiza la tendencia de las palabras elegidas en un gráfico de líneas por fecha, relativas a un rango de días indicado.

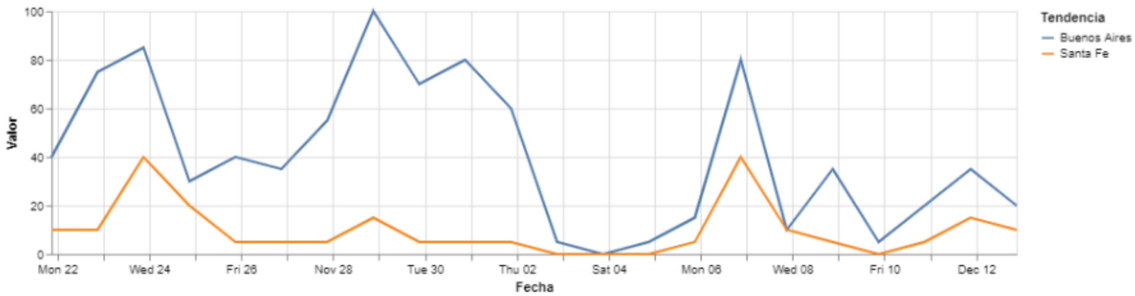

Fig. 8: Gráfico de líneas comparando las tendencias “Buenos Aires” y “Santa Fe”.

9.3. Nube de palabras

Devuelve las palabras con más apariciones en los títulos de las noticias. Se puede seleccionar la cantidad de palabras, el tamaño de la tipografía y la fecha referentes a la visualización.

Fig. 9: Nube de palabras para el 15 de Febrero de 2022.

9.4. Wordtree

Devuelve los títulos de las noticias en la base de datos que contengan la palabra clave que se ingresa.

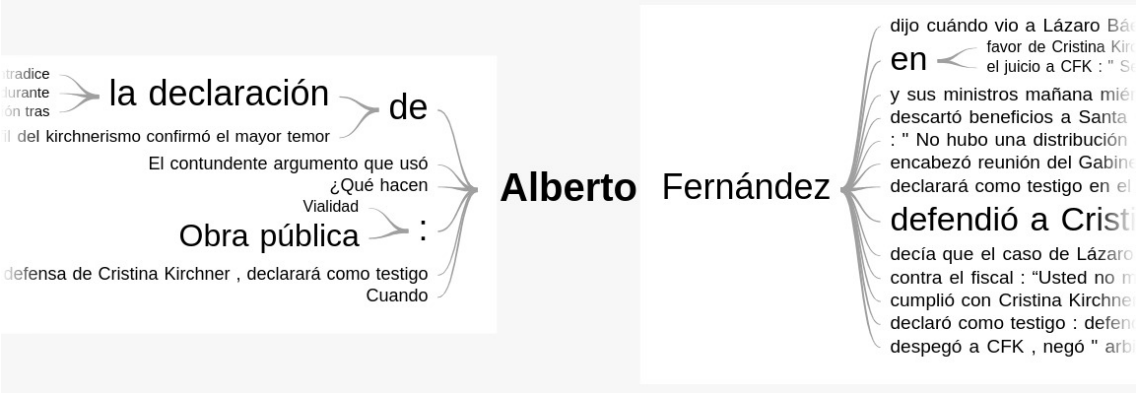

Fig. 10: Wordtree para la palabra clave "Alberto" para el 15 de Febrero de 2022.

9.5. Búsqueda

Muestra una tabla con todos los links a las noticias que coinciden con los parámetros de búsqueda ingresados.

Titulo
Video: lo encontró robando y lo obligó con un machete a que muestre su cara a una cámara de seguridad
El FMI trata las sobretasas que cobra por deuda excesiva, pero nadie cree que las vaya a eliminar
Se filtro el mensaje que Antonio Laje le envió a su equipo de trabajo tras el escándalo
El BCRA acelera el ritmo de devaluación diario, pero no logra detener la caída de las reservas
El complicado método para descargar y actualizar la app Cuidar por el pase sanitario que instauró el Gobierno
La frase de Joan Manuel Serrat que provocó la emoción de Gerardo Rozín
El INE se ajustó al presupuesto para 2022: la Revocación de Mandato costará 3,830 millones de pesos
Sneed ausente en duelo de Chiefs tras asesinato de hermano
Muerte de Vicente Fernández minuto a minuto: Hijos del charro montan la Guardia de honor junto a su féretro

Fig. 11: Ejemplo de una tabla con el resultado de búsqueda.

9.6. Noticias por fecha y medio

Heatmap que visualiza la cantidad de noticias recolectadas en la API por cada medio disponible.

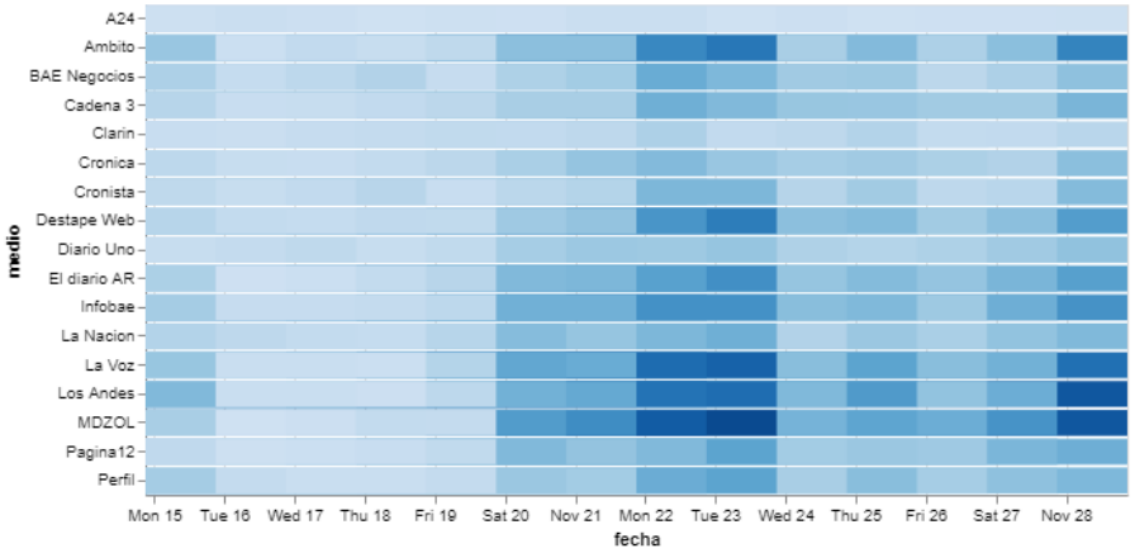

Fig. 12: Visualización de la cantidad de noticias recolectadas por medio para las últimas dos semanas de noviembre 2021.

9.7. Primicia

Con el fin de presentar en el D3 Meetup en Español[22] un ejemplo práctico del uso de la API , se realizó la siguiente visualización que muestra cómo se propagan las noticias en los medios argentinos. Dado un tema y un intervalo temporal, se visualiza quién arranca con la noticia y quién se copia acerca de un tema específico.

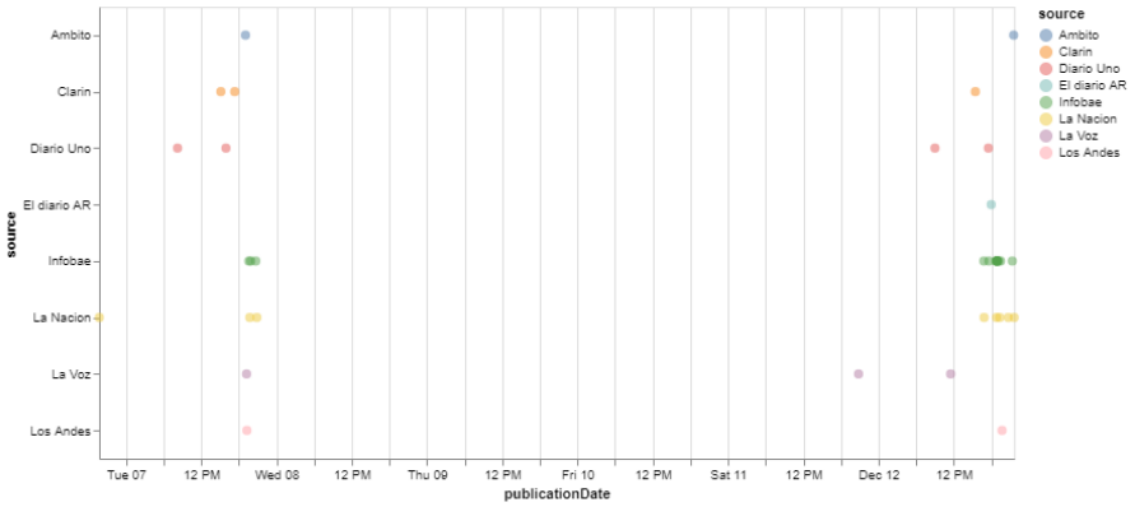

Fig. 13: Gráfico de puntos que ilustra que medios son los que arrancan la propagación de noticias acerca de la transferencia de Messi al PSG.

[22]https://observablehq.com/@john-guerra/agenda-del-d3-meetup-en-espanol

9.8. Cantidad de noticias

Permite visualizar la cantidad de noticias por fecha.

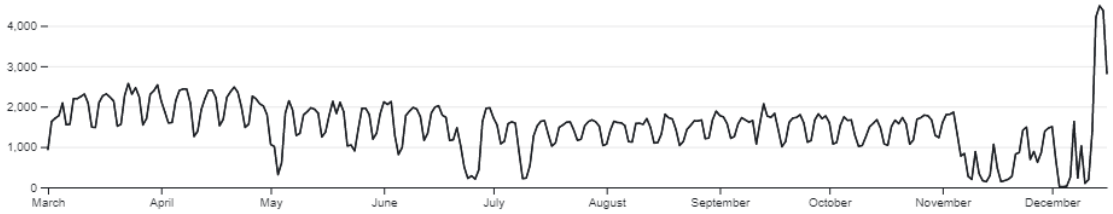

Fig. 14: Gráfico de líneas con la cantidad de noticias desde marzo a diciembre del 2021.

10. Trabajo futuro

Hoy en día el proyecto está obteniendo datos de 18 medios, sería una buena oportunidad de mejora aumentar esta cantidad con el fin de tener una visión más amplia de las noticias que se consumen a diario en Argentina. Además, el proyecto podría seguir escalando y recolectar noticias en todos los medios de noticias *online* de habla hispana, no solo de nuestro país.

Otro punto interesante para seguir trabajando, es el desarrollo del reconocimiento de entidades. Es posible seguir entrenando el modelo de [2] con el objetivo de lograr una mayor precisión a la hora de procesar, localizar y clasificar las entidades. Esta práctica podría ser constante ya que el lenguaje cambia continuamente y el algoritmo entrenado debería actuar de la manera más precisa posible.

Además, hay diversos medios que publican noticias de forma automática como por ejemplo "Previsión meteorológica: El tiempo hoy en *Santa Fe*: *13 de diciembre*". En este sentido, un cambio que añadiría un gran valor al proyecto sería reconocer este tipo de patrones, filtrarlos o directamente no almacenarlos en la base de datos.

Finalmente, el proyecto está pensado para ser integrado con un *frontend* que permita interactuar con la API de forma dinámica, directa y práctica. Con visualizaciones que permitan observar la información de una manera simple e interactiva, que esté al alcance de todo el público que quiera investigar sobre noticias y medios.

11. Conclusiones

En el contexto actual, es trascendental el lugar que ocupan los medios de comunicación en nuestra sociedad. Las noticias se comparten y se viralizan en períodos de tiempo sorprendentemente muy cortos. Este tema ha tenido especial repercusión en el último tiempo debido al caso de las *fake news*, donde todo el mundo pudo notar el creciente impacto que

ejercen las noticias en la vida de todas las personas.

El análisis de noticias es una herramienta de extraordinario valor para la investigación, permitiendo una visión más amplia y clara sobre los acontecimientos que impactan en nuestra sociedad. El proyecto facilita el acceso a datos de noticias y permite desarrollar un estudio con mayor detalle sobre los medios de Argentina.

Asimismo, el reconocimiento de entidades es una característica que agregó un valor significativo, ya que agrega un contexto a los artículos y permite una clasificación del contenido de los titulares. Se puede tener una mejor visión para desarrollar análisis más profundos.

Referencias

[1] Pablo Fernández y Andrea Lata. *XDATA*. https://ri.itba.edu.ar/handle/123456789/942. Instituto Tecnológico de Buenos Aires. CABA, Argentina, 2017.

[2] Juan Pablo Orsay y Horacio Miguel Gómez. *NERd: anotador eficiente de modelos estadísticos para el reconocimiento de entidades nombradas*. https://ri.itba.edu.ar/handle/123456789/1879. Instituto Tecnológico de Buenos Aires. CABA, Argentina, 2019.