# Guerra de Titanes

Facundo Torterola
*Facultad de Ingeniería, Universidad de la República*
Montevideo, Uruguay
facundo.torterola@fing.edu.uy

Santiago Berruti
*Facultad de Ingeniería, Universidad de la República*
Montevideo, Uruguay
santiago.berruti@fing.edu.uy

3 de julio de 2022

## Resumen

Esta investigación consiste en la extracción y recolección de datos del sitio de noticias Montevideo Portal, con el fin de crear dos bases de datos no relacionales para su futura comparación. Las bases de datos son creadas en Neo4J y en MongoDB.

## 1. Introducción

El siguiente informe documenta el proyecto final de la asignatura Bases de Datos no Relacionales. En este proyecto se han creado dos bases de datos no relacionales, una de documentos y una de grafos, con datos extraídos de un portal de noticias uruguayo. Entre los datos principales se encuentran noticias, comentarios, usuarios y temas. El objetivo del proyecto es estudiar el comportamiento del sitio de noticias y documentar ventajas y desventajas de los dos tipos de bases de datos utilizadas.

## 2. Trabajos Relacionados

En una investigación[1] hecha en la Universidad Tecnológica De Lublin, se compara una base de datos relacional, una de documentos y una de grafos sobre el mismo conjunto de datos. Para la base de datos de documentos se usa MongoDB y para la de grafos se utiliza CassandraDB. Se realizan comparaciones en tiempo de ejecución de las consultas.

## 3. Desarrollo del proyecto

### 3.1. Obtención de datos

Para el desarrollo de la investigación fue necesario extraer datos del sitio web de Montevideo Portal. Para lograr esta tarea se necesita extraer datos de diferentes páginas dentro del portal. Entre ellas encontramos:

- Noticias

[1] `https://www.researchgate.net/publication/314639479_Comparison_of_Relational_Document_and_Graph_Databases_in_the_Context_of_the_Web_Application_Development`

- Perfiles de usuario
- Tags

Cabe destacar que Montevideo Portal cuenta con diferentes dominios donde publican sus noticias, siendo básicamente la misma web cambiando solo algunos detalles estéticos. Entre ellas encontramos:

- **montevideo.com.uy**: La página principal. Cuenta con noticias de política, policiales y temas de interés general.
- **futbol.uy**: Es donde se publica información relacionada con el mundo del deporte.
- **pantallazo.com.uy**: Se cuentan noticias del mundo del espectáculo.
- **airbag.uy**: Cuenta con noticias relacionada con los autos y automovilismo.

Todas estas páginas tienen una estructura casi idéntica y los métodos utilizados para extraer datos sirven en todas. Además, los usuarios son compartidos entre las diferentes plataformas. Para tener una mayor variedad de datos y aprovechándose de estas similitudes, se extrajeron datos de todos los dominios mencionados anteriormente. El *scraper* se realizó en Python y se utilizaron algunas bibliotecas y protocolos para facilitar la extracción de datos. Entre ellas encontramos:

- **Scrapy**: es un marco de trabajo de *scraping* y *crawling* de código abierto, escrito en Python
- **Selenium**: Es una biblioteca en Python que nos permite emular el comportamiento de un navegador web.
- **JSON**: Utilizamos JSON para guardar los datos extraídos.
- **Pandas**: Es una biblioteca utilizada para la manipulación de datos en los *jupyter notebook* para mostrar el resultado de las consultas

Para la recolección de noticias primero se obtuvo una lista de URLs que apuntaban a estas. Para este primer punto, a falta de un lugar en el sitio web donde apareciera una lista con todas las noticias, se aprovechó del sistema de recomendación de lecturas del portal. Cada vez que se entra a la página principal o a alguna noticia, existe una sección con recomendaciones de lectura que llevan a otras noticias. Para aprovecharse de esto se configuró un programa con la herramienta Scrapy[2] la cual recorrería estas recomendaciones identificándolas por el formato de estos elementos dentro del sitio web utilizando diferentes etiquetas de HTML y CSS, y por último guardando las *URLs* de las noticias.
Cabe mencionar que muchas veces aparecen las mismas recomendaciones en dos noticias diferentes, por lo que luego de recorrer alrededor de cinco mil noticias, se tuvieron que purgar las URLs extraídas para eliminar duplicados, lo que dejó un número final de 4260.
Una vez obtenida una lista de noticias, se procedió a ejecutar el programa de Selenium[3]. Se decidió el uso de esta herramienta ya que los comentario con más de tres reacciones negativas son ocultados y para acceder a ellos se debe interactuar presionando un botón para poder ver el contenido del comentario. Este comportamiento invalidaba el uso de otras librerías que obtienen el HTML estático como es el caso de *BeautifulSoup*.
Como fue mencionado anteriormente, estos datos fueron guardados de forma bastante cruda en archivos JSON. Luego de esta primera recolección, la información de las noticias fue utilizada para obtener una lista de URLs de perfiles de usuarios y otra de Tags.

Luego se volvió a reproducir este proceso solo que esta vez con dos programas distintos uno para obtener la información del usuario y otro para obtener la información de los tags.
Los números finales de datos recolectados son:

- 4260 noticias
- 3148 perfiles de usuario
- 2100 tags

[4]

[2] `https://scrapy.org/`
[3] `https://www.selenium.dev/`
[4] `https://gitlab.fing.edu.uy/facundo.torterola/BDNR_web_scraper`

## 3.2. Modelado de Datos

### 3.2.1. Base de documentos

Para realizar la modelado de una base de datos basada en documentos se utilizo la herramienta MongoDB. El modelado de esta consiste en 3 colecciones, Noticias, Tags y Usuarios.

**Noticias** cuentan con la siguiente estructura:

- id: identificador único usado por Montevideo Portal
- titulo
- descripción
- fecha
- tiempo de lectura
- URL
- tags: temas vinculados con la noticia, tiene información que relaciona las colecciones de tags y noticias.
- comentarios: una entidad embebida que relaciona el usuario con la noticia.

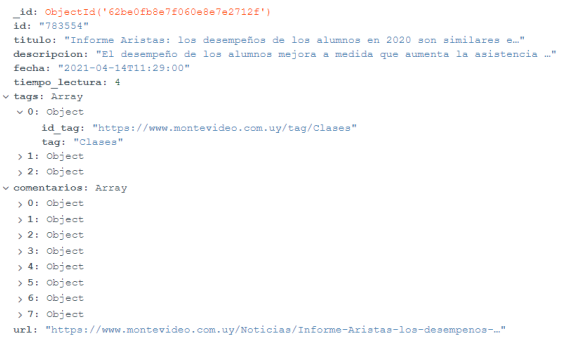


Figura 1: Modelo de noticias en Mongo

**Tags**:

- id: URL del Tag
- título

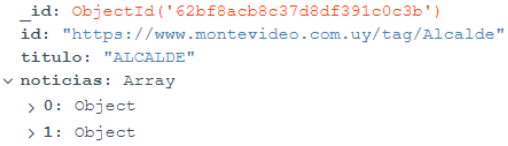


Figura 2: Modelo de tags en Mongo

- noticias: noticias del tag, tiene información que relaciona las colecciones de tags y noticias.

**Usuarios**:

- id: URL del perfil del usuario
- nick: nombre desplegado a otros usuarios
- sexo: género del usuario
- registro: fecha de registro en la plataforma
- seguidores: usuarios que siguen al usuario
- seguidos: usuarios al que el usuario sigue
- comentarios: comentarios que realizó el usuario en noticias, es una entidad embebida que relaciona el usuario con la noticia[5].

Figura 3: Modelo de usuario en Mongo

### 3.2.2. Base de grafos

El modelo de base de grafos es similar. Cuenta con tres nodos, Usuarios, Noticias y Tags. También cuenta con tres relaciones las cuales son Sigue, Tema y Comenta. Los nodos cuentan con las misma información que se detallo en la parte anterior.

[5]Se limitó a los últimos 500 comentarios

En cuanto a las relaciones conectan los siguientes nodos.
La relación Sigue, conecta a dos usuarios y no tiene información extra en la relación.

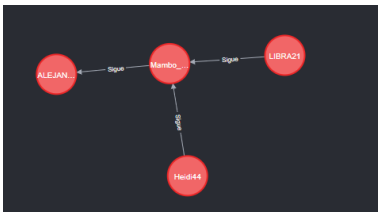


Figura 4: Modelo de relación

En la imagen [4] se puede notar que el usuario *mambo* tiene dos seguidores y sigue solamente a al usuario *Alejandro*.
El nodo Tema, une una Noticia con un Tag.

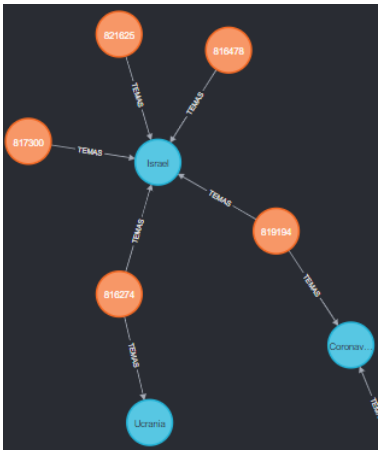


Figura 5: Modelo de relación TEMAS

En la imagen[5] podemos apreciar la noticia 819194 toca el tema *Israel* y *Ucrania*.
La relación Comenta, conecta los nodos Usuario y Noticia. Tiene la siguiente información:

- texto
- fecha
- me gusta: saldo absoluto de reacciones hechas por otros usuarios

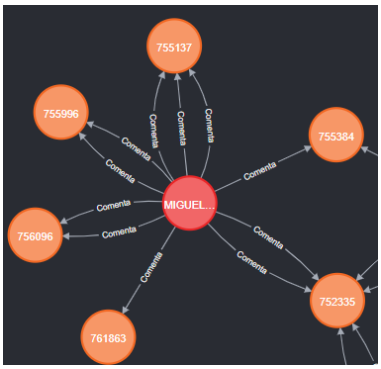


Figura 6: Modelo de relación Comenta

## 3.3. Carga de datos

Al haber extraído los datos desde Internet y guardados directamente en formato JSON. La subida de datos a la base en MongoDB es instantánea y se puede realizar directamente desde interfaces.

Los resultados no fueron los mismos para la base de datos en Neo4J. En esta base se presenta un nivel de complejidad más alto y esto es reflejado en varios aspectos como la subida de datos. En este caso no se pueden importar datos directamente y se necesita realizar consultas para lograr reflejar el modelo planteado.

Además, para realizar estas consultas se necesita de tiempo de procesamiento, y dependiendo la cantidad de datos el tiempo puede ser elevado.

Los resultados vistos en este caso fueron que para subir:

- 27739 noticias
- 7601 usuario
- 1918 tags
- 3241 relaciones Sigue
- 54000 comentarios

El tiempo requerido para subir estos datos utilizando un ambiente con 16GB de memoria RAM y un procesador AMD Ryzen 7 fue de 30 horas.

# 4. Experimentación

Se decidieron realizar las siguientes consultas[6] sobre ambas base de datos para comparar los rendimientos.

Consulta 1: Noticia con más comentarios.
Consulta 2: Comentarios con mas me gustas.
Consulta 3: Usuario con más comentarios
Consulta 4: Usuarios que comentaron en la misma noticia que otro usuario X.

| Consulta | Mongo | Neo |
|---|---|---|
| Consulta 1 | 2.5s | 9.5s |
| Consulta 2 | 1.5s | 17.5s |
| Consulta 3 | 0.5s | 11.9s |
| Consulta 4 | 0.7s | 10.2s |

# 5. Resultados

Para esta siguiente sección se introduce el concepto de usuario *activo* (o interactivo). Este es un usuario que ha comentado una noticia en 2022. Notamos que hay usuarios que pueden leer las noticias y nunca comentar, pero este dato no puede ser extraído por los medios utilizados y no se toma en cuenta en los siguientes resultados.
El concepto de *polémico*, refiere a aquel comentario o noticia que genera un número elevado de interacciones.
El concepto de *dependencia* entre dos temas refiere a la cantidad de de apariciones de un tema junto a otro tema.

- La cantidad de usuarios activos en la plataforma es de 1300.
- El usuario activo más antiguo de la plataforma fue registrado en 1998.
- Los temas que tienen más comentarios son Peñarol, Frente Amplio, Nacional, Lacalle Pou y Manini Ríos.
- Los usuarios activos más populares dentro de la plataforma por número de seguidores tienen 50, 43, y 38 seguidores respectivamente.
- Entre algunos casos interesantes de dependencia encontramos los pares Ucrania y Rusia con un porcentaje de dependencia de 38 %. Sin embargo, el nivel de dependencia de Rusia con Ucrania es de 29 %.
- El tema con más noticias es Coronavirus con un total de 4164 noticias. Si se toma en cuenta que la primera noticia es del 10 de Marzo de 2020 y la última del 30 de Junio de 2022, se publicaron un promedio de 4.94 noticias sobre el Coronavirus por día.

Otro dato interesante es que el usuario *gallinolmitomano*, tiene múltiples comentarios con más de 10000 me gustas, un saldo absoluto de 58239 me gustas y un promedio de 1021 me gustas por comentario. Cabe destacar que el segundo usuario con más me gustas lo sigue con un total de 3587.
También el usuario *JIME N0* tuvo múltiples comentarios con más de 7000 no me gustas.

# 6. Conclusiones

Montevideo Portal es un sitio con excelentes condiciones para ser estudiado, dada su falta de restricciones de acceso, dado que no se necesita usuario registrado ni hay límites de acceso.

Montevideo Portal es un sitio que no se liga a ningún tema específico, pero se puede ver que los temas más recurrentes de su público fueron el Coronavirus, luego el fútbol y luego la política.

Se puede intuir que su sistema de me gustas cuenta con algún problema de seguridad dada las anomalías en algunos de sus comentarios y usuarios.

Neo4J cuenta con un mayor tiempo de procesamiento inicial de los datos, pero da ventajas a nivel de las consultas posibles sin elevar la complejidad de escritura de las consultas.

Los resultados muestran que las consultas hechas en MongoDB tuvieron respuestas más rápidas que las consultas hechas en Neo4J. Cabe destacar que para la

[6] `https://gitlab.fing.edu.uy/facundo.torterola/BDNR_web_scraper/-/tree/master/consultas`

base de datos de documentos se utilizó Mongo Atlas, es decir que se utilizó una base en la nube mientras que las consultas en Neo4J se utilizó un ambiente local. Por lo que las comparaciones de tiempo no son totalmente justas.

Para usuarios nuevos en ambas plataformas, se debe tener en cuenta que si se cuenta con poco tiempo, es recomendable analizar de antemano que consultas se quieren realizar y pensar si son posibles con MongoDB, ya que el tiempo de subida de los datos a Neo4J puede ser una gran desventaja.

Para trabajos futuros se recomienda expandir la información obtenida. Existe otro tipo de relación en Montevideo Portal que son las etiquetas en comentarios. Un usuario puede etiquetar a otro usuario en un comentario, esa relación fue excluida en esta investigación pero puede ser de relevancia para otros informes. También sería bueno realizar un estudio de comunidades basado en la base de datos de grafos y algún método como el de Lovaina.