UNIVERSIDAD TECNICA FEDERICO SANTA MARIA | DEPARTAMENTO DE INFORMÁTICA

Tarea 3: MongoDB arXiv
INF-239 Bases de Datos

Profesores: Ricardo Salas (ricardo.salas@usm.cl)
Rodrigo Olavarría (rodrigo.olavarria@usm.cl)
Mauricio Figueroa (mauricio.figueroac@usm.cl)
Ayudante Coordinador: Gonzalo Alarcón(gonzalo.alarconc@usm.cl)
mayo, 2025

1 Objetivo

Investigar y aplicar los conceptos básicos sobre el gestor de base de datos NoSQL MongoDB, utilizando herramientas como Docker para desplegar los servicios, MongoDB Query Language (MQL) y Python.

2 Especificaciones y reglas

El desarrollo de esta tarea debe cumplir las siguientes especificaciones, de lo contrario existiría un descuento en la nota final:

- Se debe ejecutar todos los procedimientos necesarios para implementar la Base de Datos MongoDB sobre Docker, bajo una arquitectura distribuida maestro-esclavo, utilizando Python como lenguaje de programación para realizar las operaciones que se requieran en esta tarea.

- La tarea debe ser entregada el jueves 12 de junio hasta las 23:59 horas, cuya respectiva defensa es obligatoria.

2.1 Entregables

La tarea debe ser entregada como un archivo comprimido de la forma T3-ROL1-ROL2.zip y debe contener los siguientes archivos:

1. Archivo deploy-rol1-rol2.yml utilizado para desplegar los servicios sobre Docker con las especificaciones de arquitectura entregadas.
2. Archivo Jupyter Notebook o Google Colab pymongo-rol1-rol2.ipynb, el cuál debe venir ejecutado con todas las evidencias de los resultados documentadas en el mismo notebook, considerando una sección por cada requerimiento solicitado en la tarea.
3. Archivo uarxiv-rol1-rol2.json que contenga los datos actualizados y finales que den cuentas de las operaciones realizadas según los requerimientos de esta tarea.

2.2 Reglas

- El trabajo debe realizarse en parejas; no se aceptarán tareas individuales. Las copias serán evaluadas con nota 0 y se informará a las autoridades correspondientes. Para consultas, utilicen la plataforma oficial AULA.
- En el foro de búsqueda de pareja podrán buscar compañeros quienes no tengan, siendo esta una responsabilidad exclusiva del estudiante. En caso de problemas con su pareja, podrán contactar al profesor explicando su situación. Solo un alumno debe realizar la entrega.
- Si falla la ejecución de algún comando, no se asignará puntaje a este. Existe la posibilidad de que, durante su defensa, asista su profesor y realice preguntas. Es responsabilidad del estudiante inscribir un horario de defensa y estar presente en la fecha y hora elegida.
- Cada grupo tendrá un horario definido para su defensa; en caso de atraso, contarán con un tiempo menor para presentar su trabajo. La información respecto a la defensa se publicará eventualmente en AULA, lo que incluirá detalles sobre los descuentos. Es su obligación estar atentos a esta información y cumplir con lo establecido allí.

3 Descripción del problema

Se requiere diseñar, poblar y consultar una base de datos en **MongoDB** que almacena los datos de artículos científicos y académicos[1] del repositorio arXiv[2], para lo cual se dispone del dataset llamado **arxiv-metadata-oai-snapshot.json**. Esta fuente de datos es un JSON que recopila 2.735.264 documentos con los datos específicos de publicaciones científicas. La estructura de los documentos que almacena la colección de documentos **articles** es la siguiente:


```
_id: ObjectId("682c0ef73cea57cbb35089ac")
id: "0704.0001"
submitter: "Pavel Nadolsky"
authors: "C. Bal\'azs, E. L. Berger, P. M. Nadolsky, C.-P. Yuan"
title: "Calculation of prompt diphoton production cross sections at Tevatron a..."
comments: "37 pages, 15 figures; published version"
journal-ref: "Phys.Rev.D76:013009,2007"
doi: "10.1103/PhysRevD.76.013009"
report-no: "ANL-HEP-PR-07-12"
categories: "hep-ph"
license: null
abstract: "  A fully differential calculation in perturbative quantum chromodynam..."
› versions: Array
update_date: "2008-11-26"
› authors_parsed: Array
pdf_source: "https://arxiv.org/pdf/0704.0001"
```


Figura 1: Estructura de documento artículo

[1]Dataset y explicación de los datos en: https://www.kaggle.com/datasets/Cornell-University/arxiv
[2] https://info.arxiv.org/about/index.html

4 Requisitos

1. Implementar la arquitectura de Cluster con un set de 3 réplicas. Una de las máquinas será el primario y las otras dos serán los secundarios. Esta configuración deberá estar operativa en su máquina local, donde se levantarán las 3 instancias de Docker con cada uno de los servicios de MongoDB con la redundancia de la Base de Datos llamada "arxiv_db" que contendrá la colección de documentos llamada "articles". En la siguiente imagen se muestra la arquitectura requerida.

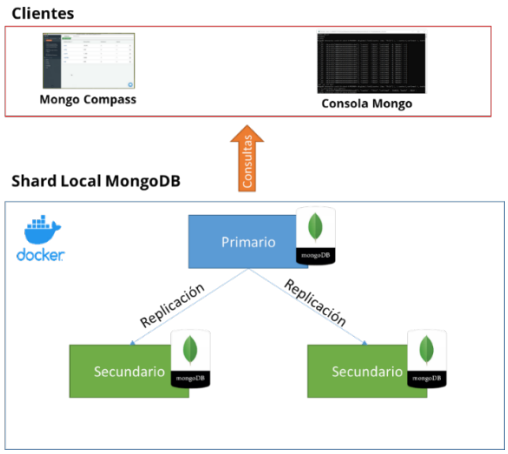

Figura 2: Arquitectura General

2. En base al **dataset arxiv-metadata-oai-snapshot.json** entregado, y luego implementar el código Python para las siguientes consultas requeridas para resolver las preguntas del negocio utilizando el cliente :

   a. Devolver los títulos y fechas de creación de artículos publicados en el año 2025. Mostrar solo esos campos y limitar a los primeros 20 resultados.

   b. Devolver los títulos y los autores de artículos que pertenezcan a las categorías "cs.AI" o "stat.ML" y que tengan al menos tres autores. Mostrar solo esos campos y limitar a los primeros 10 resultados.

   c. Devolver los títulos, las categorías y los enlaces al PDF de artículos que pertenezcan a la categoría "hep-ph" y tengan un DOI asignado. Mostrar solo esos campos y limitar a 15 resultados.

   d. Devolver los títulos, nombres de los autores y la referencia de publicación (journal-ref) de los artículos que tengan un DOI asignado. Mostrar solo esos campos y ordenar los resultados alfabéticamente por título. Limitar a los primeros 20 resultados.

   e. Devolver los títulos y la fecha de la primera versión (versions.created) de los artículos enviados entre los años 2010 y 2015. Mostrar solo esos campos y limitar a los primeros 15 resultados.

   f. Devolver los títulos, comentarios y reportes técnicos (report-no) de artículos que tengan comentarios definidos y no nulos. Mostrar solo esos campos, ordenando por fecha de actualización (update_date) en orden descendente. Limitar a 10 resultados.

3. Una vez realizados los puntos anteriores, realizando operaciones necesarias desde Python, se pide mostrar evidencia objetiva que permita demostrar:
    a. La consistencia de los datos en los 3 nodos replicados, por ejemplo, posterior a la inserción, actualización y/o eliminación de datos, los datos deben permanecer siempre consistentes.
    b. La alta disponibilidad, por ejemplo, bajando el nodo local para que responda alguno de los nodos replicados sin que el usuario se percate del problema.

5 Consideraciones técnicas

Para llevar a cabo esta tarea, se recomienda llevar a cabo las siguientes actividades técnicas:

1. Instalar el software de desarrollo y tecnología para contenedores Docker Desktop y Docker Compose. Usar como referencia este sitio https://docs.docker.com/manuals/.

2. Crear un Replica Set de MongoDB en Docker con docker-compose usando el archivo docker_compose.yml de ejemplo y subir los servicios.

3. Conectarse al Replica Set desde Python usando la biblioteca PyMongo, con la dirección de conexión de ejemplo mongodb://mongo1:30001,mongo2:30002,mongo3:30003/?replicaSet=my-replica-set&readPreference=primary&appname=MongoDB%20Compass&ssl=false. Asegúrese de modificar el archivo de hosts para que los nombres de los servidores sean mapeados a la IP local del host.

4. Descargar el dataset dataset arxiv-metadata-oai-snapshot.json y luego importar los datos con las instrucciones necesarias desde Python (arXiv.ipynb). Puede usar como referencia el siguiente Notebook PyMongo-DB.ipynb.

5. Documentación acerca de MongoDB, su arquitectura y su lenguaje de consultas puede ver en https://www.mongodb.com/docs/manual/.

6 Aclaraciones

1. Las consultas de esta tarea DEBEN realizarse en el foro de aula.

2. Es obligatorio utilizar las tecnologías y ambientes solicitados, existirá descuento en la pauta de no usarse.

3. Cualquier supuesto debe ser especificado de forma explícita en el notebook entregado.

4. Se recomienda iniciar con tiempo esta tarea.