



Ingeniería en Sistemas Computacionales

Bases de Datos No Relacionales

MongoDB Lab 1 – APIs, Datos y MQL

Marco Ricardo Cordero Hernández

Tlaquepaque, Jal., 2 de marzo de 2023

Una vez que se ha dejado de lado (al menos de momento) a las bases de datos columnares, es momento de conocer y poner en práctica las utilidades de las bases de datos orientadas a documentos. Para ello, se hará uso del sistema NoSQL más infame hasta la fecha: ***MongoDB***.

Esta base proporciona múltiples puntos de acceso para la manipulación masiva de datos, tales como plataformas web, GUI's en clientes nativos, y, como se pudo manejar en desarrollos pasados, a través de bibliotecas y drivers contruidos para diversos lenguajes de programación pensados principalmente para backend (MongoDB, 2023).

En este laboratorio, se abordará el primer punto de entrada para una base orientada a documentos, de hecho, la piedra angular de estas: los mismísimos documentos. Para ello, se tomará como ejemplo didáctico la propuesta de una estructura de lo que pudiera ser una base de libros para su venta en internet, inspirado fuertemente en tiendas en línea. Pero antes, como punto de partida para un desarrollo más substancial que se avecina, se deben revisar una serie de conceptos antes de pasar a lo descrito.

## **REST APIs**

Según IBM (2021), una *API*, o *interfaz de programación de aplicaciones*, es un conjunto de reglas que determinan cómo las aplicaciones o los dispositivos pueden conectarse y comunicarse entre sí. La anteposición del acrónimo *REST* (“Representational State Transfer”) hace referencia a un enfoque arquitectural para el diseño de servicios web popularmente vinculado a HTTP (Ekuan et al., 2022). Una *API REST* es una API que se ajusta a los principios de diseño de REST, por eso es que ocasionalmente se les denomina como *RESTful APIs*.

### **Casos de uso**

Para identificar sistemas distribuidos basados en estas tecnologías, basta con navegar por internet unos cuantos segundos, sin embargo, el uso de las mismas no se limita a microservicios lovecraftianos. Entre los más interesantes ejemplo se encuentran:

- Aplicaciones basadas en la nube
- Manejo de bancos de credenciales
- Plataformas de interacción social
- CRMs (sistemas de gestión de relaciones con clientes)
- Inventarios departamentales

## Operaciones principales

Según MDN (2022), hay 8 tipos de métodos u operaciones que pueden ser realizadas a través de una API RESTful, sin embargo, hay 4 de suma importancia (The Postman Team, 2020):

1. **GET** – Le permite al servidor encontrar los datos que se solicitaron y los retorna hacia el cliente en alguna representación disponible o especificada.
2. **POST** – Indica al servidor que se insertará una entrada nueva a la base de datos; dicho de otra forma, se presenta una entidad nueva ante el recurso especificado.
3. **PUT** – Ejecuta una actualización hacia el recurso solicitado, es decir, actualiza una entrada en la base de datos en todas sus representaciones dada una actualización.
4. **DELETE** – Borra una entrada del recurso solicitado.

## Analogía con bases de datos

Dentro de la mayoría (si no es que todas) las bases de datos, existe el concepto *CRUD*, el cual, por su significado en inglés, hace referencia a las operaciones de creación, lectura, actualización y remoción de datos (Sharif, 2022). Si se pone en contraste y se piensa un poco en la comparación con las operaciones de la sección anterior, se encontrará el equivalente de una función inyectiva en cuanto a equivalencia se refiere:

- CREATE (Creación)  $\leftrightarrow$  POST
- READ (Lectura)  $\leftrightarrow$  GET
- UPDATE (Actualización)  $\leftrightarrow$  PUT
- DELETE (Remoción)  $\leftrightarrow$  DELETE || Relación literal

En múltiples referencias se puede encontrar el funcionamiento de algunos sitios web, en donde una petición distinta a GET hecha directamente en un navegador afectará directamente a la base de datos sobre la cual se almacenan los recursos de dicho sitio. Esto es posible por la sencilla “traducción” o representación de un dominio tecnología en el otro, más bien, el uno en el otro.

## Modelado de libros basado en documentos

Una vez que se ha conocido bastante acerca de interfaces de comunicación y desarrollo, a continuación se presenta un esquema de almacenamiento de objetos a los cuales se les verá como contenedores de información relevante para libros, esto con ayuda de algunos campos relevantes que Google Play Books (s.f.) detalla.

Cabe destacar que el formato JSON no admite comentarios en su estructura per se, por ello se ha añadido un par de llaves adicionales antes de la estructura propuesta:

```
{
  "owner": "IS727272 – Cordero Hernández, Marco Ricardo",
  "bookStructure": {
    "title": "[title]",
    "publishDate": "[publish date]",
    "description": "[description]",
    "authors": ["[author 1]", "[author 2]"],
    "pageCount": 0,
    "genre": "[genre]",
    "language": "[language]",
    "ISBN": "[ISBN]",
    "mature": false,
    "formats": ["[physical format]", "[digital format]"]
  }
}
```

Como se puede apreciar, se cuenta con múltiples campos de cadenas de caracteres o “strings”, números, listas o arreglos, e incluso booleanos. Siempre existe la posibilidad de mejorar, esta no es la excepción, ya que se pudo haber dotado al campo de autores con otros JSON en donde se detallan aún más los mismos, o convertir los campos de género y lenguajes en listas, o incluir más campos como edición o editorial, no obstante, para el alcance de este laboratorio se considera a lo propuesto como suficiente.

## MQL

Para culminar con este laboratorio, se hará uso de *MQL* (MongoDB Query Language) a través de un contenedor de Docker con *mongosh*, el cual ha sido creado previamente.

```
marcordero@mrch-ubuntu:~$ sudo docker start mongo_marco
mongo_marco
marcordero@mrch-ubuntu:~$ sudo docker exec -it mongo_marco mongosh
```

La lista de pasos utilizada fue la siguiente:

```
// Crear o cambiar a base de datos del laboratorio
use lab_1

// Asegurar creación de colección
db.books.insertMany([{}])
db.books.deleteMany({})

// Insertar datos en la colección
db.books.insertMany([
  {
    "title": "Diary of a Wimpy Kid",
    "publishDate": "04-01-2007",
```

```

        "description": "Boys don't keep diaries-or do they?",
        "authors": ["Jeff Kinney"],
        "pageCount": 224,
        "genre": "humour",
        "language": "english",
        "ISBN": "1419741853",
        "mature": false,
        "formats": ["paperback cover", "hardcover", "audiobook", "e-book"]
    },
    {
        "title": "Calculus and Analytic Geometry (9th Edition)",
        "publishDate": "08-14-1995",
        "description": "Textbook presents a modern view of calculus enhanced by the use
of technology.",
        "authors": ["George B. Thomas", "Ross L. Finney"],
        "pageCount": 1264,
        "genre": "BDSM",
        "language": "english",
        "ISBN": "0201531747",
        "mature": true,
        "formats": ["paperback cover", "hardcover"]
    },
    {
        "title": "The Mixers Manual: The Cocktail Bible for Serious Drinkers",
        "publishDate": "11-04-2014",
        "description": "Mix drinks like a pro with this indispensable collection of 150
of the world's finest cocktails",
        "authors": ["Dan Jones"],
        "pageCount": 128,
        "genre": "cook",
        "language": "english",
        "ISBN": "1742707742",
        "mature": true,
        "formats": ["hardcover"]
    },
    {
        "title": "Akira Yoshizawa, Japan's Greatest Origami Master: Featuring over 60
Models and 1000 Diagrams by the Master",
        "publishDate": "10-11-2016",
        "description": "This collection of projects by the 'father of modern origami'
contains detailed instructions for 60 of the master's original works.",
        "authors": ["Akira Yoshizawa", "Kazuo Hamada", "Kiyo Yoshizawa", "Robert J.
Lang"],
        "pageCount": 192,
        "genre": "handcrafting",
        "language": "english",
        "ISBN": "4805313935",
        "mature": false,
        "formats": ["hardcover", "e-book"]
    },
}
])

// Actualizar varios libros
db.books.updateMany(
    {"mature": true},
    {"$set: {"ageRestriction": 18}}
)

// Borrar un libro
db.books.deleteOne({ "ISBN": "1419741853" })

```

```
lab_1> // Actualizar colección

lab_1> db.books.updateMany(
...     {"mature": true},
...     {$set: {"ageRestriction": 18}}
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

```
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63fdc9b9e6f98b6903c2fe0f"),
    '1': ObjectId("63fdc9b9e6f98b6903c2fe10"),
    '2': ObjectId("63fdc9b9e6f98b6903c2fe11"),
    '3': ObjectId("63fdc9b9e6f98b6903c2fe12")
  }
}
```

```
lab_1> db.books.deleteOne({ "ISBN": "1419741853" })
{ acknowledged: true, deletedCount: 1 }
```

Con esto se concluye el primer laboratorio de MongoDB, a la espera del siguiente reto del curso y con un mejor entendimiento entre transacciones de este tipo de bases.

## Referencias bibliográficas

MongoDB. (2023). *MongoDB: The Developer Data Platform / MongoDB*. Recuperado el 28 de febrero del 2023 de <https://www.mongodb.com/>.

IBM Cloud Learn Hub (2021). *API REST*. Recuperado el 28 de febrero del 2023 de <https://www.ibm.com/mx-es/cloud/learn/rest-apis>.

Martine, K., Alberts, M., Sherer, T., Dahan, U., Kistler, M., Ferreira, N., ..., Cheung, F. (2022). *RESTful web API design*. Recuperado el 28 de febrero del 2023 de <https://learn.microsoft.com/en-us/azure/architecture/best-practices/api-design>.

MDN Contributors. (2022). *HTTP request methods*. Recuperado el 28 de febrero del 2023 de <https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods>.

The Postman Team. (2020). *What Is a REST API? Examples, Uses, and Challenges*. Recuperado el 28 de febrero del 2023 de <https://blog.postman.com/rest-api-examples/>.

Sharif, A. (2022). *WHAT IS CRUD?*. Recuperado el 28 de febrero del 2023 de <https://www.crowdstrike.com/cybersecurity-101/observability/crud/>.

Google Play Books. (s.f.). *Book metadata & information*. Recuperado el 28 de febrero del 2023 de <https://support.google.com/books/partner/answer/3237055?hl=en>.