



# **Programción Concurrente**

**0341403**

## **Implementación de Programación Concurrente en Bases de Datos NoSQL**

Por

Jorge Alejandro Cadrecha Del Rey

139846

# DECLARACIÓN

Yo, Jorge Alejandro Cadrecha Del Rey, confirmo que el trabajo presentado en este informe es mío. Si la información procede de otras fuentes, confirmo que así se indica en el informe.

---

Jorge Alejandro

# CONTENTS

<b>Declaración</b>	<b>2</b>
<b>1 Introducción</b>	<b>4</b>
1.1 Enunciado del ejercicio .....	4
1.2 Requisitos Técnicos.....	4
<b>2 Código</b>	<b>6</b>
2.1 Estrutura .....	6
2.2 Modificaciones .....	6
<b>3 Inicialización</b>	<b>7</b>
3.1 Comandos.....	7
3.2 Implementaciones .....	7
<b>4 Conclusión</b>	<b>9</b>
4.1 Conclusion .....	9

# **1. INTRODUCCIÓN**

## **1.1. ENUNCIADO DEL EJERCICIO**

En esta práctica, nuestro objetivo será obtener una comprensión práctica de cómo aplicar los principios de la programación concurrente en un entorno de base de datos NoSQL. Implementaremos diferentes estrategias de programación concurrente y evaluaremos su rendimiento al manejar grandes volúmenes de datos.

En la era actual de Big Data, las organizaciones generan y consumen cantidades enormes de información diariamente. Con tanta información disponible, las bases de datos tradicionales SQL pueden resultar ineficientes para manejar volúmenes de datos tan masivos y estructuras de datos complejas. Por lo tanto, las bases de datos NoSQL se han convertido en una elección popular debido a su escalabilidad y flexibilidad. Sin embargo, la eficiencia de estas bases de datos puede mejorarse aún más con el uso de la programación concurrente.

## **1.2. REQUISITOS TÉCNICOS**

1. Configuración inicial: Inicia configurando una base de datos NoSQL (puedes elegir entre una base de datos clave-valor, orientada a documentos, orientada a columnas o una base de datos de grafos).

2. Modelado de datos: Modela un conjunto de datos para tu base de datos NoSQL elegida. Los datos deben ser lo suficientemente complejos para requerir el uso de programación concurrente (por ejemplo, un número significativo de registros o una estructura de datos compleja).

3. Implementación de programación concurrente: Implementa la programación concurrente para realizar operaciones en tu base de datos. Esto puede implicar el uso de múltiples hilos o procesos para manejar las operaciones de lectura y escritura, la

implementación de bloqueos y otras estrategias de control de concurrencia, o el uso de bibliotecas de programación concurrente de alto nivel.

4. Pruebas y evaluación de rendimiento: Realiza pruebas de rendimiento en tu implementación. ¿Cómo se compara el rendimiento de tu implementación con y sin programación concurrente? Considera el tiempo de ejecución, el uso de recursos y cualquier otro factor de rendimiento relevante.

5. Informe de la práctica: Escribe un informe que documente tu enfoque, los desafíos que encontraste y cómo los resolviste, los resultados de tus pruebas de rendimiento y cualquier hallazgo o conclusión interesante que hayas obtenido de la práctica.

## **2. CÓDIGO**

### **2.1. ESTRUCTURA**

El código está estructurado en varios paquetes que contienen las clases correspondientes. Dentro del proyecto, se pueden identificar los paquetes "config", "controller", "domain", "model", "repos", "rest", "service" y "util". Además, se encuentra la clase "BaseDatosNOSQLApplication", que es la que debe ejecutarse para iniciar el programa.

### **2.2. MODIFICACIONES**

En el proyecto, fue necesario modificar dos archivos clave para establecer la conexión con la base de datos. El primero que ajusté para su correcto funcionamiento fue el archivo "application.yml", encargado de la configuración en aplicaciones Java con Spring Boot. Este archivo se utiliza para definir propiedades de la aplicación, como configuraciones de bases de datos y perfiles de entorno, simplificando la personalización y gestión de la aplicación sin necesidad de código adicional.

### 3. INICIALIZACIÓN

Para iniciar el proyecto, lo primero que necesitamos es un entorno de desarrollo que admita la programación en Java. Posteriormente, debemos asegurarnos de tener instaladas las aplicaciones clave: MongoDB compass, encargada de la gestión de la base de datos; Docker, responsable del despliegue de la aplicación en contenedores de software. Finalmente, es necesario contar con la aplicación de Node.js para ejecutar los comandos de instalación de paquetes y para inicializar el servidor de desarrollo (DevServer).

Además, tuve que crearme un usuario de MongoDB, realizar altas e inicializar un clúster. Luego, busqué en Kaggle una base de datos creada, obtuve el archivo .csv y lo añadí a mi base de datos de MongoDB para que tenga datos registrados.

Por último, utilizamos la plataforma Vercel para generar una página web con acceso, permitiendo ver dicha base de datos desde otro dispositivo.

#### 3.1. COMANDOS

Han sido necesarios los comandos:

- `npm install`
- `npm run devserver`

#### 3.2. IMPLEMENTACIONES

Para el funcionamiento del proyecto fueron necesarias ciertas dependencias en el `pom.xml`:

- `spring-boot-starter-web`
- `spring-boot-starter-validation`

- `spring-boot-starter-data-jpa`
- `postgresql`
- `spring-boot-starter-thymeleaf`
- `thymeleaf-layout-dialect`
- `springdoc-openapi-starter-webmvc-ui`
- `jakarta.persistence-api`
- `lombok`
- `spring-boot-devtools`
- `spring-boot-docker-compose`
- `spring-boot-starter-test`



## **4. CONCLUSIÓN**

### **4.1. CONCLUSION**

En resumen, he adquirido conocimientos sobre la generación de código necesario para desarrollar una aplicación de en línea utilizando NoSQL en mongoDB Atlas. Además, he aprendido los pasos necesarios para la inicialización del proyecto, lo cual incluye la investigación necesaria para su desarrollo. Durante este proceso, también he adquirido habilidades en el uso de aplicaciones como MongoDB compass y repaso de Docker para garantizar una correcta puesta en marcha del sistema.