

# Programación concurrente

## Implementación de Programación Concurrente en Bases de Datos NoSQL

By

Pablo Rodriguez

139852

# RESUMEN

Este proyecto aborda la implementación de programación concurrente en bases de datos NoSQL, un enfoque esencial en la era del Big Data. El objetivo es aplicar prácticamente los principios de programación concurrente en entornos de base de datos NoSQL, utilizando herramientas como Maven [1] para la gestión de dependencias y configuración del proyecto. Se seleccionó MongoDB [2], una base de datos NoSQL orientada a documentos, por su flexibilidad y escalabilidad, características clave para manejar grandes volúmenes de datos y estructuras complejas.

# OBJETIVOS

El objetivo principal de esta práctica es obtener una comprensión práctica de cómo aplicar los principios de la programación concurrente en un entorno de base de datos NoSQL.

# CONTENTS

<b>Resumen</b>	<b>1</b>
<b>Objetivos</b>	<b>2</b>
<b>1 Introducción</b>	<b>4</b>
1.1 Configuración inicial.....	4
1.2 Modelado de datos .....	4
<b>2 Materiales y metodos</b>	<b>5</b>
2.1 Conjunto de datos .....	5
2.2 Funcionalidad de Crear, Editar y Borrar Tweets .....	5
2.3 Integración Exitosa con la Base de Datos NoSQL .....	5
2.4 Interfaz del usuario .....	6
2.5 npm y webpack .....	6
<b>3 Resultados</b>	<b>7</b>
<b>Referencias</b>	<b>8</b>

# **1. INTRODUCCIÓN**

## **1.1. CONFIGURACIÓN INICIAL**

Para la base de datos NoSQL, se considerará el uso de MongoDB, orientada a documentos, o Apache Cassandra, orientada a columnas. Se discutirán las ventajas de cada opción y se elegirá la más adecuada en función del conjunto de datos y los requisitos del proyecto.

## **1.2. MODELADO DE DATOS**

Se seleccionará un conjunto de datos de un recurso público que sea lo suficientemente grande para justificar el uso de programación concurrente. Este podría incluir datos de comercio electrónico, redes sociales o registros meteorológicos, entre otros.

## **2. MATERIALES Y METODOS**

### **2.1. CONJUNTO DE DATOS**

Para las pruebas, se seleccionó un conjunto de datos de un recurso público como lo es Kaggle. Se describirá el proceso de selección del conjunto de datos, asegurando que su tamaño y complejidad fueran adecuados para el uso de programación concurrente.

### **2.2. FUNCIONALIDAD DE CREAR, EDITAR Y BORRAR TWEETS**

Se desarrollaron con éxito funcionalidades CRUD para los tweets en la base de datos. Estas operaciones se optimizaron mediante técnicas de programación concurrente, lo que resultó en una mejora significativa en la eficiencia y la capacidad de respuesta del sistema. La implementación permitió manipulaciones de datos en tiempo real, incluyendo la creación, actualización y eliminación de tweets, destacando la agilidad y la robustez del sistema desarrollado.

### **2.3. INTEGRACIÓN EXITOSA CON LA BASE DE DATOS NOSQL**

La implementación de la base de datos NoSQL con el conjunto de datos de Kaggle resultó exitosa. Los tweets relacionados con el mundial se importaron y almacenaron eficientemente, demostrando la capacidad de la base de datos para manejar grandes volúmenes de datos en tiempo real. A pesar de los desafíos iniciales relacionados con la estructura de datos y la optimización del rendimiento, se lograron adaptar las estrategias de programación concurrente para asegurar una integración fluida y eficiente.

## **2.4. INTERFAZ DEL USUARIO**

Para el desarrollo de la interfaz de usuario de la aplicación web, se utilizó Thymeleaf [3] , un motor de plantillas moderno para aplicaciones web en Java.

## **2.5. NPM Y WEBPACK**

La gestión de dependencias del proyecto se llevó a cabo utilizando npm [4], el gestor de paquetes por defecto para Node.js. Para la gestión y compilación de los activos de la aplicación, se utilizó Webpack [5], una herramienta moderna para el empaquetado de módulos JavaScript, así como Bootstrap [6] se utilizó para el diseño frontend, proporcionando una amplia gama de componentes responsivos y estilos predefinidos.

### **3. RESULTADOS**

Los resultados obtenidos del proyecto proporcionan evidencia convincente del impacto positivo de la programación concurrente en la eficiencia de las bases de datos NoSQL. Las técnicas implementadas mejoraron notablemente el rendimiento y la capacidad de respuesta del sistema, contribuyendo a una experiencia de usuario final enriquecida. Este proyecto sirve como un valioso caso de estudio sobre la aplicación efectiva de la programación concurrente en el manejo de Big Data, ofreciendo insights y prácticas que podrían ser útiles en futuras investigaciones y desarrollos en este campo.



# BIBLIOGRAPHY

- [1] Apache Software Foundation, Maven Documentation.

URL <https://maven.apache.org/guides/index.html>

- [2] Pivotal Software, Inc., Spring Data MongoDB Reference Documentation.

URL <https://docs.spring.io/spring-data/mongodb/docs/current/reference/html/>

- [3] The Thymeleaf Team, Thymeleaf Documentation.

URL <https://www.thymeleaf.org/documentation.html>

- [4] npm, Inc., npm Documentation.

URL <https://docs.npmjs.com/>

- [5] Webpack concepts.

URL <https://webpack.js.org/concepts/>

- [6] Bootstrap, Bootstrap Documentation.

URL <https://getbootstrap.com/docs/5.3/getting-started/introduction/>