Proyecto Redis

# **MEMORIA OPERATIVA**

Fecha: 19 de Febrero de 2024



**Autor:** Carlos Agustin, Marcos Burgos, Daniel Trif,

Daniel Jimenez, Daniel Valero

**Curso:** CFGS-2ºH-V

**Especialidad:** Desarrollo de Aplicaciones Multiplataforma

**Modulo:** Acceso a Datos

**Indice**

[MEMORIA OPERATIVA 1](#__RefHeading___Toc510_1202039573)

# ***1. Estudio realizado***

Para el desarrollo de este proyecto hemos valorado varios tipos de bases de datos no relaccionales, entre ellas ***MongoDB***, ***Apache Cassandra***, ***CouchDB***, ***Redis*** y ***Neo4j***.

* **MongoDB:** Esta base de datos no relaccional es la mas usada, esta es un sistema que maneja bases de datos orientadas a objetos. [Los datos no se almacenan en las estructuras conocidas de tablas y registros, sino que se guardan como documentos en formato BSON, una](https://saasradar.net/bases-de-datos-no-relacionales/) representacion del conocido JSON. No hemos usado este tipo de base de datos porque ya se a usado en clase y para este proyecto habia que usar tipos de base de datos no usados hasta el momento.
* **Apache Cassandra:** nace de un proyecto de google, varios procesos han transcurrido desde esa época, hasta que en 2010 obtuvo su graduación como proyecto de alto nivel en **ApacheIncubator**.

Es una base de datos de tipo **clave-valor**. **Cassandra** es una **BBDD NoSQL** está diseñada para almacenar cantidades gigantescas de datos y realizar distribuciones a través de varios nodos.

Esto permite que el almacenamiento de datos pueda estar repartido entre diversos servidores sin un solo punto de fallo. La gran mayoría de nosotros tiene una cercanía con esta **base de datos** ya que es una de las herramientas esenciales de **Facebook.**

* **CouchDB:** es una **base de datos NoSQL** que nace con aspiraciones bastante altas. Su principal desarrollador aspiraba a que **CouchDB** se convirtiera en la base de datos más usada en internet. En 2008 el proyecto pasa a formar parte de **ApacheIncoubator**.

La **BBDD CouchDB** llega a nosotros con la intención de facilitar la accesibilidad y compatibilidad web con diferentes tipos de dispositivos.

Su primera versión estable llegó al público en el año 2010. En esta **BBDD** los datos se incluyen en formato **JSON**.

La forma en la que los datos son organizados es según pares de valor clave.

* **Redis:** Otro importante **motor de base de datos** de tipo **clave-valor** es Redis. Esta **BBDD NoSQL** de Código abierto es patrocinada y desarrollada por **RedisLabs**. Su diseño principal está basado en el almacenamiento de **tablas de hashes** aunque no es restrictiva sólo hacia este modelo. También tiene la posibilidad de ser utilizada como una BBDD persistente.
* **Neo4j:** La **base de datos Neo4j** es desarrollada en **software libre** es del tipo **orientada a grafos**, construida con lenguaje **Java**.

Esta **base de datos** tiene un funcionamiento transaccional que almacena en estructura de **grafos**.

Su primera versión fue lanzada en 2010. Las características principales que ofrece **Neo4J** a sus desarrolladores es un **alto desempeño,** si las consultas empiezan a crecer de forma exponencial el rendimiento de esta **BBDD** no disminuye. Es **Ágil, flexible y** altamente escalable. De sus usos y posibilidades de implementación te hablaremos con seguridad más adelante.

Todos y cada uno de estos tipos de base de datos no sql son muy interesantes pero al final nos hemos decantado por Redis ya que tiene grandes ventajas en estos aspectos:

* ***Rendimiento:*** Todos los datos de Redis residen en la memoria, lo que permite un acceso a datos de baja latencia y alto rendimiento. Esto permite soportar una gran cantidad de operaciones y ofrecer tiempos de respuesta más rápidos.
* ***Estructuras de datos flexibles:*** Redis cuenta con una amplia variedad de estructuras de datos para satisfacer los requisitos de tus aplicaciones1. Los tipos de datos de Redis incluyen cadenas, listas, conjuntos, conjuntos ordenados, hashes, mapas de bits y HyperLogLogs.
* ***Alta disponibilidad:*** Redis Enterprise ofrece alta disponibilidad ininterrumpida y completamente transparente para los usuarios, con replicación sin disco, detección instantánea de fallos y conmutación por error en menos de diez segundos.
* ***Escalabilidad:*** Redis Enterprise ofrece un escalado lineal infinito sin imponer una sobrecarga no lineal en una arquitectura de escalado.
* ***Fácil configuración:*** Redis es fácil de configurar y tiene una curva de aprendizaje baja.
* ***Persistencia de datos:*** Redis ofrece la posibilidad de persistir datos en disco para recuperación ante fallas.