

# CouchDB: Investigación e implementación demostrativa de una base de datos NoSQL

Sebastián Hurtado      Diego Linares      Piero Marini

28 de Noviembre del 2019

## 1 Introducción

El objetivo de la presente investigación es entender el funcionamiento de una base de datos NoSQL y realizar una demostración de esta con una colección de datos. Actualmente, la data que resulta del uso en particular de aplicaciones web, redes sociales, etc. se encuentra en un nivel bajo de estructuración. Como resultado de ello, el uso de base de datos relacionales puede no ser el más adecuado para trabajar con este tipo de data [1]. En este caso, exploraremos una de las soluciones NoSQL, que provee soporte ACID y permite trabajar con data no estructurada en formato JSON, CouchDB.

El siguiente artículo se va a encontrar dividido de la siguiente manera: primero se definirá CouchDB y se comparará frente a otras soluciones de este tipo, observando en que casos conviene el uso del primero. A continuación explicaremos más a detalle las características del modelo de datos y la arquitectura de almacenamiento distribuido. Finalmente se hará una demostración de su uso en una colección de datos del servicio web Yelp, y se realizarán conclusiones en base a la misma.

### 1.1 Definición y Propósito

CouchDB es un *cross-platform, open-source*, base de datos de tipo NoSQL. Su arquitectura interna se encuentra diseñada específicamente para la web, y permite trabajar con grandes cantidades de data. Este hace uso de un **Couch Replication Protocol** para sincronizar documentos JSON entre 2 pares a través de un protocolo HTTP. Por ejemplo, los *requests* son realizados a una determinada URL para obtener información con respecto a las bases de datos que se están manejando. Esto permite hacer uso de *CURL* o cualquier otro intérprete para trabajar con los archivos que se encuentran en una base de datos CouchDB.

Las bases de datos de CouchDB almacenan los llamados **documentos**. Estos consisten de un número variable de campos, los cuales pueden ser de diversos tipos, en conjunto con su metadata. Las ediciones por parte del cliente son realizadas a través de una carga, modificación y guardarlos de vuelta. Las

transacciones son de estilo *todo o nada*, por lo cual en la base de datos nunca se van a encontrar documentos parcialmente editados.

Un aspecto que diferencia a CouchDB de otras base de datos NoSQL es el mantenimiento de las propiedades ACID de una transacción. Las actualizaciones de documentos son serializadas, mientras que las lecturas de los mismos se realizan de forma concurrente. Para esta concurrencia es usado un *Multi-Version Concurrency Control*, en el cual cada usuario ve una etapa constante de la base de datos de inicio a fin. La estructura de datos utilizada para organizar los datos es un B-Tree. Las actualizaciones ocurren hasta el final de una transacción, dejando la base de datos en un estado consistente.

El propósito de CouchDB, es el de trabajar con data no estructurada, particularmente para la web. Hacen uso del paradigma *map-reduce*, replicación fácil y una *straightforward RESTful API* [2]. Como resultado de ello resulta útil para un ambiente de producción web. Además de las previas razones para ello, se le suma el echo de que toma de la arquitectura de la web (en palabras del desarrollador de Django *“It’s built of the web”*). Otras ventajas particulares son su alta escalabilidad y la ya mencionada facilidad de replicación.

## **1.2 Cuadro comparativo con otras BD NoSQL**

## **1.3 Escenarios de uso**

# **2 Características**

## **2.1 Modelo de datos**

### **2.1.1 Inserciones**

### **2.1.2 Actualizaciones**

### **2.1.3 Búsqueda**

### **2.1.4 Indexación**

## **2.2 Arquitectura de almacenamiento distribuido**

### **2.2.1 Fragmentación, Asignación, Replicación**

### **2.2.2 Procesamiento de consultas distribuidas**

# **3 Implementación Demostrativa**

## **3.1 Carga de una colección de datos**

## **3.2 Consulta de datos de forma distribuida**

# **4 Conclusiones**

## Referencias

- [1] Sangeeta Gupta et al. Efficient query analysis and performance evaluation of the nosql data store for bigdata. *Proceedings of the First International Conference on Computational Intelligence and Informatics. Advances in Intelligent Systems and Computing*, 507:549–558, 2016.
- [2] Reuven M. Lerner. At the forge: Couchdb. *Linux Journal*, 195:8, 2010.