# Práctico de CoutchDB

### ¿Qué es CouchDB?

Apache CouchDB es una [base de datos NoSQL](https://aprenderbigdata.com/bases-de-datos-nosql/) open source orientada a documentos que nos permite gestionar de forma distribuida una gran cantidad de datos no estructurados. La primera versión de CouchDB se publicó en el año 2005 y el software está implementado en el lenguaje de programación Erlang.

Si hay una palabra para describir CouchDB, es relax. Es el título del logo oficial de CouchDB y también se muestra en la consola al ejecutar el software. Aprender esta tecnología y sus conceptos básicos debería resultar natural para la mayoría de las personas que han hecho algún desarrollo web y además es fácil de explicar a personas no técnicas.

CouchDB usa varios formatos y protocolos para almacenar, transmitir y procesar datos. El formato principal para el almacenamiento de datos es [**JSON**](https://aprenderbigdata.com/json/). También, usa [**JavaScript**](https://aprenderbigdata.com/cursos-javascript-udemy/) como lenguaje de consulta y [**MapReduce**](https://aprenderbigdata.com/hadoop-mapreduce/) para el procesamiento. Además, utiliza el protocolo HTTP para las comunicaciones.

A diferencia de las bases de datos relacionales, CouchDB no admite el almacenamiento en tablas sino que trata cada base de datos como una colección de documentos independientes.

CouchDB también está diseñado para manejar el tráfico variable. Por ejemplo, si un sitio web experimenta un peak repentino de tráfico, CouchDB generalmente absorberá una gran cantidad de peticiones concurrentes sin caerse. Puede tardar un poco más de tiempo para cada petición, pero todas se responden. Cuando el peak haya pasado, CouchDB volverá a funcionar con la velocidad habitual.

### Arquitectura de CouchDB

CouchDB garantiza la escalabilidad con su arquitectura y replicación multimaster para distribuir los datos globalmente. Un clúster de varios nodos guarda todos los datos de forma redundante, para que siempre estén disponibles cuando sean necesarios.

CouchDB puede desplegarse con un único nodo o bien en modo clúster. La mayoría de los proyectos pueden comenzar con una sola instancia de CouchDB. Los proyectos más exigentes pueden actualizarse cuando sea necesario a un despliegue en modo clúster. Este modo clúster permite ejecutar un único servicio de base de datos en cualquier número de servidores o de máquinas virtuales. El clúster mejora la capacidad y dota al sistema de alta disponibilidad.

### Esquema de Arquitectura en CouchDB

Diagrama

Descripción generada automáticamente

### Replicación de Datos

El exclusivo protocolo de replicación de CouchDB es la base de toda una nueva generación de aplicaciones «Offline First» para aplicaciones móviles y otros entornos con infraestructuras de red difíciles.

La función principal de la replicación es sincronizar dos o más bases de datos. Para ello, debe sincronizar de forma fiable las bases de datos entre varias máquinas para el almacenamiento redundante de datos. También distribuir los datos en un clúster de instancias que comparten las peticiones que llegan al clúster equilibrando la carga. Por último, debe distribuir los datos entre ubicaciones físicamente distantes.

### Almacenamiento de Documentos

Un despliegue de CouchDB aloja bases de datos que almacenan documentos. Cada documento tiene un nombre único en la base de datos y se proporciona una API HTTP REST para leer, actualizar y eliminar los documentos.

Los documentos son la unidad de datos en CouchDB y están formados por cualquier número de campos. Los documentos también incluyen metadatos que son mantenidos por el sistema. Los campos de los documentos tienen un nombre único y pueden contener valores de distintos tipos, como por ejemplo texto, números o booleanos. No hay un límite para el tamaño del texto o del número de elementos.

El modelo de actualización de documentos de CouchDB es optimista y sin bloqueos. Las ediciones de documentos se realizan mediante aplicaciones cliente que cargan los documentos, aplican los cambios y los guardan de nuevo en la base de datos. Si otro cliente que edita el mismo documento guarda sus cambios primero, el cliente recibe un error de conflicto de edición al guardar. Para resolver el conflicto de actualización, se puede abrir la última versión del documento, volver a aplicar los cambios e intentar la actualización de nuevo.

Las actualizaciones de un solo documento (añadir, editar, eliminar) son transaccionales: o tienen éxito por completo o fallan por completo. La base de datos nunca contiene documentos parcialmente guardados o editados.

### Instalación de CouchDB

Vamos a instalar CouchDB mediante Docker , para esto utilizaremos Docker-compose:

|  |
| --- |
| version: '3'  services:  couchserver:  image: couchdb  restart: always  ports:  - "5984:5984"  environment:  - COUCHDB\_USER=admin  - COUCHDB\_PASSWORD=couchdb  volumes:  - ./dbdata:/opt/couchdb/data |

Luego escribimos en la consola (ya sea en visual studio code o en la terminal del sistema operativo):

**docker-compose up -d**

A continuación deberíamos ver los contenedores corriendo:

**Docker ps**



Luego vamos a comprobar el puerto 5984, para esto escribimos en el navegador

**http://localhost:5984**

Texto

Descripción generada automáticamente

Luego revisaremos la página web de administración de couchDB.

<http://localhost:5984/_utils/#login>

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Entramos y luego crearemos una nueva base de datos “udec”

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Luego tendremos nuestra db lista para la carga de datos JSON. Para esto copiamos el archivo movies.json en la carpeta en donde creamos el archivo Docker-compose.yml

Vamos a cargar el dataset de películas en formato json, para esto vamos a entrar al contenedor en modo interactivo.

**docker exec -it docker-cdb-couchserver-1 bash**

Paso siguiente vamos a cargar los datos de movies.json con el siguiente comando:

**curl -X POST -d @/opt/couchdb/data/movies.json -H "Content-Type: application/json"** [**http://admin:YOURPASSWORD@localhost:5984/udec/\_bulk\_docs**](http://admin:YOURPASSWORD@localhost:5984/udec/_bulk_docs)

Veremos cargada nuestra BD

Gráfico

Descripción generada automáticamente con confianza baja

Una de las ventajas de CouchDB es que permite ser consultada a través de REST API, sin la necesidad de un driver o conector especial. Para hacer las consultas a una REST API, es recomendable ocupar POSTMAN.( https://www.postman.com/downloads/)

Cabe recordar que estamos tras el sistema de autenticación de couchDB, por lo que debemos configurar la autenticación, para eso vamos a la pestaña Auth y configuramos la autenticación en Basic Auth. El verbo debe estar configurado como GET .

En primer lugar vamos a listar todos los documentos de la base de datos udec, ocupando la siguiente URL:

<http://localhost:5984/udec/_all_docs>

Luego presionamos “Send”:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora cambiamos la URL a:

<http://localhost:5984/udec/1e34f2a626d12abaf649d1c0ac0006e4>

Y ejecutamos ejecutamos:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Ahora vamos a confeccionar una consulta, sin embargo , debemos definir un índice en la base udec, para eso vamos la pestaña:

Texto

Descripción generada automáticamente con confianza media

Y creamos el índice por año, de esta manera nos permitirá realizar consultas más complejas.

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Volvemos a postman y escribimos en la url:

<http://localhost:5984/udec/_find>

En el cuerpo de la petición escribimos:

{

    "selector": {

        "year": {"$gt": 2019}

    },

    "fields": ["\_id", "\_rev", "year", "title"],

    "sort": [{"year": "asc"}],

    "limit": 20,

    "skip": 0,

    "execution\_stats": **true**

}

Lo que nos permite consultar por el índice, todas las películas que cuyo año sea mayor que 2019, los campos \_id, \_rev, year y title, ordenado ascendentemente, que limite los datos a 20 resultados, comenzando de la posición 0 (importante para una posible paginación),y que además muestre las estadísticas de ejecución.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como bonus track , POSTMAN tiene un ícono en la exquina supererio derecha:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Que se llama code, lo presionamos y veremos una lista de lenguajes con el código correspondiente que es capaz de hacer la llamada a CouchDB.

Nota: Es posible que algunos códigos no funcionen al 100%, son de referencia.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente