

1. Breve histórico, características e versões

CouchDB (Couch é um acrônimo para Cluster of Unreliable Commodity Hardware) é um projeto criado em abril de 2005 por Damien Katz, ex-desenvolvedor da IBM. Damien Katz o definiu como “sistema de armazenamento de bancos de dados para objetos de larga escala”. Seu objetivo é tornar o CouchDB um banco de dados para a Internet, voltado à aplicações web. Ele manteve o projeto por quase dois anos, então liberou como código-aberto sob a GNU General Public License. Em fevereiro de 2008, se tornou um projeto da Incubadora de projetos Apache e sua licença mudou para a Licença Apache. Alguns meses depois, se tornou um projeto de alto nível. Isso levou à primeira versão estável, liberada em julho de 2010. A atual versão é a 2.3.1, lançada em Março de 2019.

CouchDB é um banco de dados não-relacional que abrange a web e utiliza documentos JSON para armazenar dados. Os documentos são acessados pelo navegador web, via HTTP e além disso, as consultas e modificações nos documentos são feitas por JavaScript. CouchDB é utilizado na web moderna e em aplicativos mobile.

Diferente de um banco de dados relacional, o CouchDB não armazena os dados e relacionamentos em tabelas. Cada banco de dados é uma coleção de documentos independentes, e cada documento mantém seus próprios dados e esquemas.

O CouchDB armazena os dados como documentos com um ou mais pares de chave/valor, podendo ser números, string, datas e arrays. Esses documentos apresentam uma identificação única, não sendo necessário a criação de esquemas. Dito isso, um campo multi-valorado em um banco relacional exige duas tabelas, mas no CouchDB isso pode ser facilmente contornado por um array.

2. Linguagem de implementação e suporte do sistema

CouchDB foi implementado em Erlang, essa desenvolvida pela Ericsson para suportar aplicações distribuídas e tolerantes a falhas a serem executadas em um ambiente de tempo real e ininterrupto.

Um servidor CouchDB opera em 6 Sistemas operacionais diferentes (Android, BSD, Linux, OS X, Solaris, Windows), suporta 17 linguagens de programação diferentes, dentre elas: C, C#, Java, JavaScript, Lua, PHP, Python,

Ruby. Esse hospeda bancos de dados, que armazenam documentos. Os documentos são indexados em árvores B pelo seu nome (DocID) e um ID de Sequência.

3. Suporte a características de banco de dados relacionais

Por se tratar de uma banco de dados orientado a documentos, não possui uma interface SQL.

As operações de leitura do CouchDB usam um modelo MVCC (Multi-Version Concurrency Control), em que cada cliente vê uma captura instantânea consistente do banco de dados do início ao fim da operação de leitura. Isso significa que o CouchDB pode garantir semântica transacional por documento. Quando os documentos do CouchDB são atualizados, todos os dados e índices associados são liberados para o disco e a confirmação transacional sempre deixa o banco de dados em um estado completamente consistente. As confirmações ocorrem em duas etapas:

- Todos os dados do documento e as atualizações de índice associadas são liberados de forma síncrona no disco.
- O cabeçalho atualizado do banco de dados é gravado em dois pedaços consecutivos idênticos para formar os primeiros 4k do arquivo e, em seguida, liberado de forma síncrona no disco.

O CouchDB também foi projetado para lidar com diferentes tipos de tráfego possuindo suporte para solicitações concorrentes. Por exemplo, se um site está passando por um aumento repentino no tráfego, o CouchDB absorve muitas solicitações simultâneas sem cair.

Tratando-se de tuning, com até dezenas de milhares de documentos, você geralmente encontrará o CouchDB com bom desempenho, independentemente de como você escreve seu código. Depois de começar a entrar nos milhões de documentos, você precisa ter muito mais cuidado.

Os administradores do CouchDB estão definidos no arquivo de configuração, já para os usuários regulares o CouchDB possui um banco de dados de autenticação especial, chamado `_users` por padrão, que armazena todos os usuários registrados como documentos JSON.

Tratando-se de Backup e Restore, o CouchDB, a abordagem mais simples e fácil para o backup do CouchDB é usar a replicação do CouchDB para outra instalação do CouchDB. Você pode escolher entre réplicas normais (one-shot) ou contínuas, dependendo da sua necessidade. No entanto, você também pode copiar os arquivos .couch reais do diretório de dados do CouchDB (por padrão, data /) a qualquer momento, sem problemas. O formato de armazenamento somente anexado do CouchDB para bancos de dados e índices secundários garante que isso funcione sem problemas.

4. Características de distribuição de dados, replicação, escalabilidade, tolerância a falhas, suporte a big data, etc

O design do CouchDB empresta muito da arquitetura da Web e dos conceitos de recursos, métodos e representações. Ele aumenta isso com maneiras poderosas de consultar, mapear, combinar e filtrar seus dados. Adicione tolerância a falhas, escalabilidade extrema e replicação incremental, e o CouchDB define um ponto ideal para bancos de dados de documentos.

A replicação do CouchDB tem como sua função fundamental sincronizar dois ou mais bancos de dados do CouchDB, podendo ocorrer de duas maneiras: mestre-mestre ou mestre-escravo. Isso pode parecer simples, mas a simplicidade é essencial para permitir que a replicação resolva vários problemas: sincronize de maneira confiável bancos de dados entre várias máquinas para armazenamento de dados redundantes, distribuir dados para um cluster de instâncias do CouchDB que compartilham um subconjunto do número total de solicitações que atingem o cluster (balanceamento de carga), e distribuir dados entre locais fisicamente distantes.

Se você tem um aplicativo em execução ao vivo, o CouchDB se esforça novamente para evitar incomodá-lo. Sua arquitetura interna é tolerante a falhas e as falhas ocorrem em um ambiente controlado e são tratadas normalmente. Problemas únicos não são transmitidos em cascata por um sistema de servidor inteiro, mas ficam isolados em solicitações únicas.

O CouchDB possui suporte a Big Data, com até dezenas de milhares de documentos, você geralmente encontrará o CouchDB com bom desempenho, independentemente de como você escreve seu código. Depois de começar a entrar nos milhões de documentos, você precisa ter muito mais cuidado.

5. Linguagem(ns) de definição, manipulação e consulta de dados

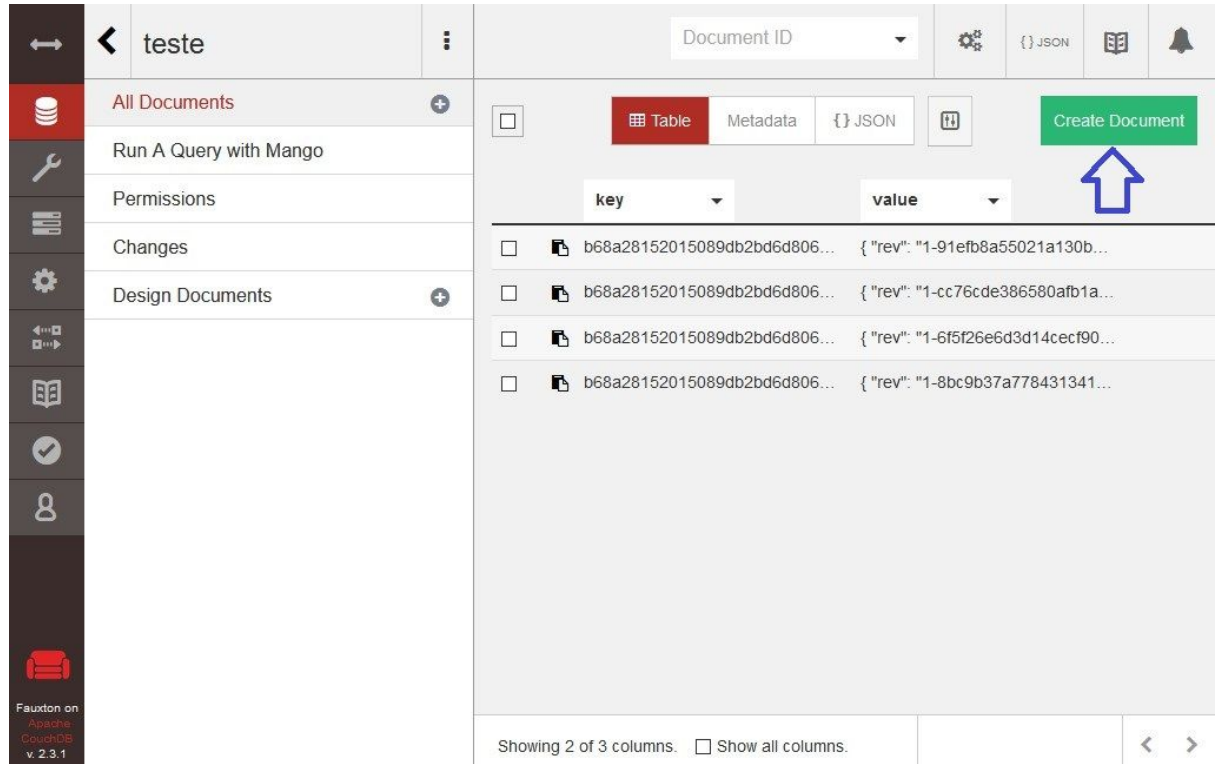
Usando Node, Express, e Nano, temos:

- Para realizarmos a função de criar um documento, usamos a função `db.insert()`
- Para realizarmos a função de atualizar um documento, usamos a função `db.insert(docname)`
- Para realizarmos a função de apagar um documento, usamos a função `db.destroy(docname)`
- Para realizarmos a função de consultar um documento, podemos usar as funções `db.get(docname)`, para exibir um documento específico ou `emit(key, value)` para buscar um conjunto de documentos

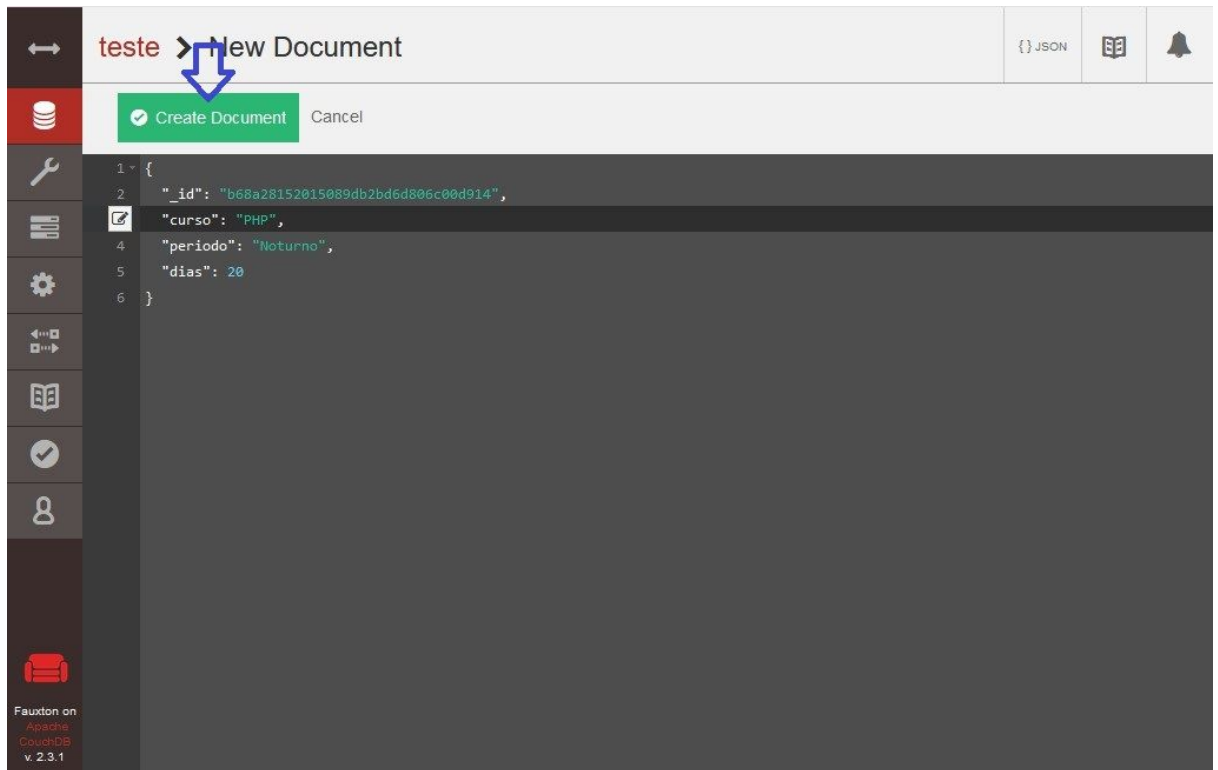
Usando a interface do banco(Restful), temos:

- Para realizarmos a função de criar um documento:

Basta entrarmos em um banco já criado e pressionar “Create Document”



Após isso inserir as informações do documento no espaço escuro dentro das chaves(correspondente ao JSON do documento), assim como no exemplo abaixo, clique em “Create Document”:



- Para realizarmos a função de atualizar um documento:

Selecione no documento que deseja atualizar.

Podemos assim ver que toda a funcionalidade do couchDB se dá através de comandos de excelente usabilidade, tendo pouco contato com codificação em si.

The screenshot shows the Apache CouchDB interface. On the left is a sidebar with navigation icons. The main area displays a list of documents in the 'teste' database. The 'Table' view is selected, showing columns for 'key' and 'value'. A document with ID 'b68a28152015089db2bd6d806c00378f' is highlighted with a blue box, and a blue arrow points upwards towards it. The document's value is a JSON object: `{ "rev": "1-8bc9b37a778431341..." }`.

Após serem feitas as alterações no JSON, clique em “Save Changes”

The screenshot shows the Apache CouchDB interface with the 'Save Changes' dialog open for the document 'b68a28152015089db2bd6d806c00378f'. The dialog has a 'Save Changes' button (highlighted with a green box) and a 'Cancel' button. Below the dialog, the document's JSON content is displayed in a code editor:

```

1 {
2   "_id": "b68a28152015089db2bd6d806c00378f",
3   "_rev": "1-91efb8a55021a130b7e9ef2f1131a914",
4   "curso": "Blockchain",
5   "periodo": "Manhã",
6   "dias": 10
7 }

```

- Para realizar a função de apagar um documento

Selecione no documento que deseja apagar

The screenshot shows the Apache CouchDB interface. On the left is a sidebar with navigation icons. The main area displays a list of documents under the 'teste' database. The document with ID 'b68a28152015089db2bd6d806c00378f' is highlighted with a blue box. A blue arrow points to this document. The interface includes a 'Document ID' dropdown, a 'Create Document' button, and a table view showing document keys and values.

key	value
b68a28152015089db2bd6d806...	{ "rev": "1-91efb8a55021a130b..." }
b68a28152015089db2bd6d806...	{ "rev": "1-cc76cde386580afb1a..." }
b68a28152015089db2bd6d806...	{ "rev": "1-6f5f26e6d3d14cecf90..." }
b68a28152015089db2bd6d806...	{ "rev": "1-8bc9b37a778431341..." }

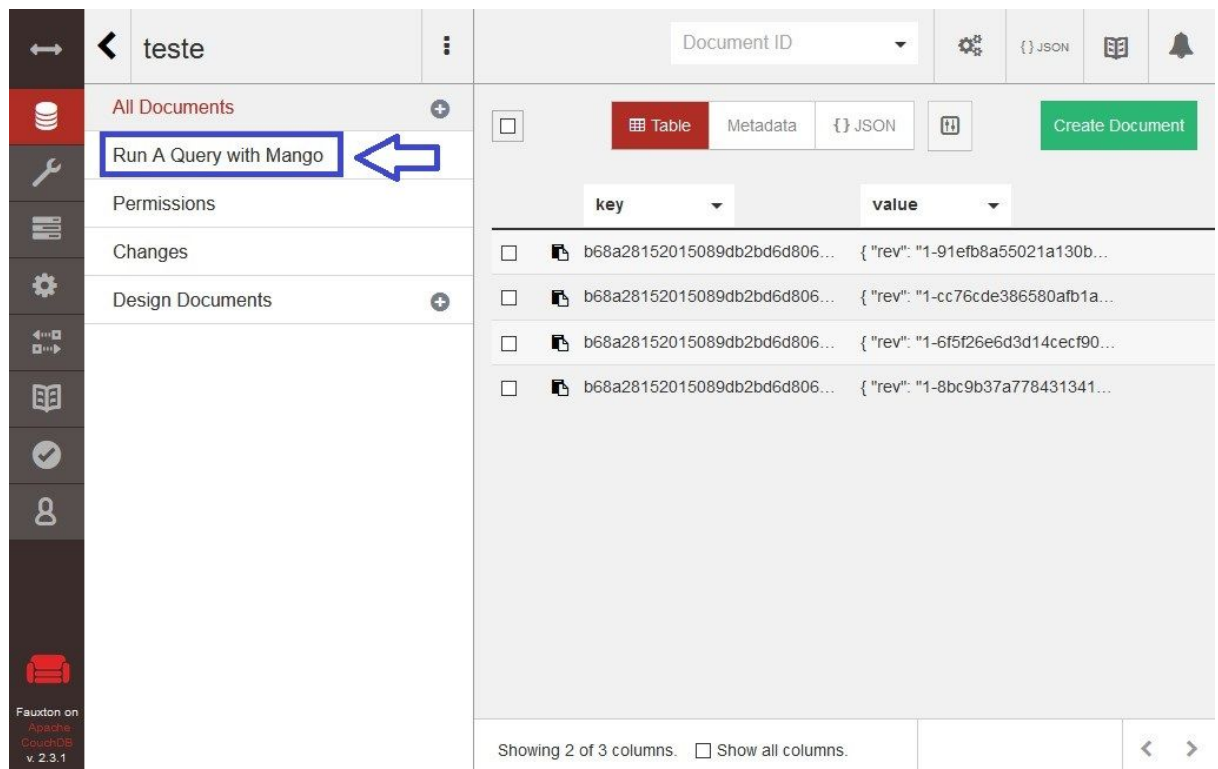
Após abri-lo clique em “Delete”

The screenshot shows the Apache CouchDB interface with the document details for 'b68a28152015089db2bd6d806c00378f' displayed. The document content is shown in a code editor. The 'Delete' button is highlighted with a blue arrow. The interface includes a 'Save Changes' button, an 'Upload Attachment' button, a 'Clone Document' button, and a 'Delete' button.

```
1 {
2   "_id": "b68a28152015089db2bd6d806c00378f",
3   "_rev": "1-91efb8a55021a130b7e9ef2f1131a914",
4   "curso": "Blockchain",
5   "periodo": "Manhã",
6   "dias": 10
7 }
```

- Para realizar a função de consultar um documento:

Após iniciar o banco, clique na opção "Run A Query with Mango"



Ao abrir, abaixo do "Mango Query" você verá uma caixa, onde poderá realizar as consultas. A sintaxe de consulta é composta por um "selector": entre as chaves principais, e após isso outro par de chaves contendo dentro dele a restrição da consulta. Estruturada a consulta, é só clicar em "Run Query".

teste > Mango Query

Query history

Mango Query ?

```
1 {
2   "selector": {
3     "curso": "Blockchain"
4   }
5 }
```

Run Query Explain manage indexes

Create Document

No Documents Found

Showing 0 documents. Documents per page: 20

No caso da consulta acima, este irá retornar todos os documentos onde curso será igual a Blockchain.

teste > Mango Query

Query history

Mango Query ?

```
1 {
2   "selector": {
3     "curso": "Blockchain"
4   }
5 }
```

Run Query Explain manage indexes

Executed in 27 ms

Table {} JSON Create Document

	_id	curso	dias	periodo
<input type="checkbox"/>	b68a281...	Blockchain	10	Manhã

Showing 4 of 5 columns. ☐ Show all columns.

6. Limitações conhecidas do sistema, inclusive limites físicos

Podemos citar como limitações do CouchDB:

- Troca de disco versus velocidade: os bancos de dados podem se tornar enormes em comparação com outros SGBDs (embora haja funcionalidade de compactação).
- As exibições temporárias em grandes conjuntos de dados são muito lentas.
- A replicação de bancos de dados grandes pode falhar.
- Mapear/reduzir paradigmas podem requerer repensar, já que dão trabalho.
- Os dados estão em JSON, o que significa que os documentos são muito grandes.

7. Principais usuários e exemplos de aplicações

Por ser um banco de dados orientado a documentos é mais adequado do que um banco de dados relacional tradicional para, por exemplo, wikis, blogs e sistemas de gerenciamento de documentos.

O CouchDB é particularmente adequado para a web moderna e o desenvolvimento de aplicativos móveis. É até possível servir aplicativos web diretamente do CouchDB.

Atualmente o CouchDB possui três tipos de serviços prestados, sendo eles hospedagem, suporte e consultoria. Como exemplo de empresas que utilizam os serviços temos a IBM/Cloudant, que utiliza o como seu banco de dados não relacional como serviços de hospedagem, assim como o Smileupps. No caso de serviços de suporte, temos a Neighbourhoodie Software e a XavaMedia WebServices. Por fim, na questão de consultoria, além das últimas duas mencionadas, temos a Red Comet Labs.

Referências:

<https://docs.couchdb.org/en/stable/intro/index.html>

<https://db-engines.com/en/system/CouchDB>

<https://pt.slideshare.net/hugosouza/apachapache-couch-db>

<https://pt.wikipedia.org/wiki/CouchDB>

<https://www.ibm.com/developerworks/br/library/os-couchdb/index.html>

[https://www.ibm.com/cloud/learn/couchdb#targetText=Apache%20CouchDB%20\(CouchDB\)%20is%20an.in%20JSON%2Dbased%20document%20formats.&targetText=As%20an%20open%20source%20project.use%20and%20embracing%20the%20web.](https://www.ibm.com/cloud/learn/couchdb#targetText=Apache%20CouchDB%20(CouchDB)%20is%20an.in%20JSON%2Dbased%20document%20formats.&targetText=As%20an%20open%20source%20project.use%20and%20embracing%20the%20web.)

<https://www.devinstincts.com/2019/03/24/couchdb-crud-node-express-nano/>

<http://guide.couchdb.org/draft/cookbook.html>

<https://docs.couchdb.org/en/2.2.0/api/database/find.html#selector-syntax>

<https://dzone.com/articles/couchdb-rest-api-for-document-crud-operations-exam>

<https://cwiki.apache.org/confluence/display/COUCHDB/Professional+Services>

<https://www.ibm.com/cloud/cloudant>

<https://www.smileupps.com/>

<https://neighbourhood.ie/couchdb-support/>

<https://xavamedia.nl/>

<https://www.redcometlabs.com/>