

Desenvolvimento de uma interface WEB com PHP para gerenciamento de banco de dados CouchDB

Kassiano Matteussi, Elton Minetto

Universidade Comunitária da Região de Chapecó - Unochapecó

ACEA – Área de Ciências Exatas e Ambientais

Av. Senador Atílio Fontana 591-E – CEP 89809-000 – Chapecó – SC – Brasil

{kassianojm, elm}@unochapeco.edu.br

Abstract. *Web development has grown exponentially along with the presence of databases in this environment. The relational model, very famous in desktop development, has been adapted to operate in a web environment. However, the rapid growth and data flow on the Internet have brought some problems that did not occur in the old environment, including flexibility, performance, scalability and compatibility, factors that are essential for web applications.*

Recently, a new model for data storage with focus on systems hosted on the Internet has been created, which became known as document-oriented model (DOM), presenting itself as a solution to some of the major deficiencies presented in relational models.

Among this pretext the article behind a study of a practical approach to the implementation of a prototype web interface with PHP for managing the database CouchDB.

Resumo. *Pode-se observar que o desenvolvimento web vem crescendo exponencialmente em conjunto com a presença de bancos de dados neste ambiente. O modelo relacional, muito famoso no desenvolvimento desktop, foi adaptado para atuar no ambiente web. No entanto, o rápido crescimento e fluxo de dados na internet trouxeram alguns problemas que não ocorriam no antigo ambiente, como por exemplo, flexibilidade, desempenho, escalabilidade e compatibilidade, fatores estes que são essenciais para aplicações na web.*

Recentemente, foi criado um novo modelo para armazenamento de dados com foco em sistemas para internet, que ficou conhecido como modelo orientado a documentos. Este por sua vez, apresenta-se como solução para algumas das principais deficiências apresentadas nos modelos relacionais.

Diante deste pretexto, o artigo trás um estudo de uma abordagem prática para a implementação de um protótipo de interface web com PHP para gerenciamento da base de dados CouchDB.

1. Introdução

Pode-se observar que o desenvolvimento Web vem crescendo exponencialmente, devido a sua versatilidade e facilidade de uso. Na mesma linha crescem os bancos de dados neste ambiente. O modelo relacional, muito famoso no desenvolvimento desktop, foi adaptado para atuar no ambiente Web, porém o rápido crescimento e fluxo de dados na internet trouxeram alguns problemas que não ocorriam no antigo ambiente, como por exemplo, flexibilidade, desempenho, escalabilidade e compatibilidade, fatores estes que são essenciais para aplicações na Web modernas.

Bancos relacionais possuem boa escalabilidade, mas quando se chega a 100% de uso de um servidor, é necessário o balanceamento de carga para vários servidores, assim batendo de frente com a complexidade do modelo com seu potencial de escala. E quando essa necessidade de balanceamento de carga se torna muito grande, acaba-se prejudicando a viabilidade nas grandes plataformas de desenvolvimento.

O modelo de banco de dados orientado a documento surgiu para contribuir especialmente com o desenvolvimento de sistemas para web, pois combina um modelo de armazenamento de documentos de uma forma intuitiva com um poderoso mecanismo de consulta e facilidades para escalar. Este modelo oferece um novo método para armazenar dados, o que é referido como um modelo de banco de dados orientado a documentos livre de esquema. Ao invés de haver um armazenamento de dados altamente estruturado de um modelo relacional, o CouchDB armazena dados de maneira semi-estruturada.

Sendo assim, percebe-se que o mercado carece de uma ferramenta para auxiliar o gerenciamento destes bancos de dados. Desta forma, no presente artigo será apresentado o estudo para o desenvolvimento de uma ferramenta Web com PHP para o gerenciamento de banco de dados CouchDB com o foco em auxiliar na aprendizagem de comunidades acadêmicas e afins, além de fornecer uma ferramenta para contribuir com o trabalho de DBAs no que se refere a gerenciamento de diversas bases de dados orientadas a documentos.

2. Modelo Orientado a Documento

Os bancos de dados neste modelo compreendem uma série de documentos, cada um contendo uma série de campos e valores. Cada documento é independente um do outro e não usam nenhuma restrição quanto a sua estrutura durante a criação. Em vez de um campo de chave primária, cada documento em um banco de dados CouchDB tem um ID único. Esta identificação exclusiva pode ser atribuída pelo usuário ou pelo aplicativo, ou ele pode usar um identificador universalmente exclusivo (UUID); este número aleatório é gerado pelo CouchDB que reduz a chance de duplicar IDs.

Segundo Moraes (2009), a “base de dados orientados a documento é diferente. Não existe hierarquia de dados, somente uma coleção de documentos que pode conter virtualmente qualquer espécie de dados”. Os documentos não precisam ser necessariamente do mesmo tamanho, pois alguns documentos podem conter detalhes de campos que outros não precisam armazenar. “Em outras palavras, você não está condicionado a um esquema de banco de dados”.

Diferente dos bancos de dados relacionais, o banco orientado a documentos não grava dados em tabelas em linhas com campos uniformes. Neste banco de “dados” são inseridos documentos com determinadas características, a cada domínio são inseridos vários campos e esses campos por sua vez podem possuir vários elementos.

2.1. Características

De acordo com Bain (2009), podem ser descritas algumas características deste modelo de banco de dados. Podem-se criar bancos diferentes com vários esquemas.

- Itens são definidos por chaves e a cada item podem ser adicionados vários atributos;
- Em algumas implementações os atributos são todos de um tipo, em outras assumem inteiros, listas e string, entre outros;

Mais algumas características, agora sobre acesso aos dados:

- os dados são criados, atualizados, e recuperados usando métodos de chamada por API¹;

¹ **API**, Sigla de Application Program Interface. É o conjunto de recursos que permitem criar uma interface com um sistema operacional

- toda a aplicação e os dados que a integram a lógica estão contidos no código da aplicação.

3. Apache CouchDB

CouchDB foi criado em meados de 2005 por Damien Katz, escrito originalmente em C++ e traduzido para Erlang² em 2008. Este combina um modelo intuitivo de armazenagem em documentos com um poderoso sistema de pesquisa acessível via RESTful³ e HTTP/JSON Api⁴.

Lennon (*apud* OLIVEIRA, 2009, p.19) observa que o termo ‘Couch’ é um acrônimo para “Cluster of Unreliable Commodity Hardware” (Conjunto de Hardware Não-Confíáveis), “refletindo o objetivo do banco de dados ser extremamente escalável, oferecendo alta disponibilidade e confiabilidade, mesmo quando executando em um hardware que é tipicamente suscetível a falhas”.

O couchDB combina um modelo intuitivo de armazenagem em documentos com um poderoso sistema de pesquisa acessível via RESTful⁵ e (HTTP/JSON Api)⁶, dentre suas funcionalidades está a replicação bidirecional com detecção de falhas e resolução de conflitos; sua base de dados é baseada em princípios MapReduce⁷ e em vez de armazenar dados em linhas e colunas, o CouchDB gerencia uma coleção de documentos, os chamados “JSON documents”.

Este ainda utiliza um modelo de views baseado em JavaScript para a geração de agregações e filtros, provendo resultados de consultas a partir destes documentos semi-estruturados. Ao invés de armazenar dados em linhas e colunas, o CouchDB gerencia uma coleção de documentos, os chamados “JSON documents”. O mesmo usa um método de visualização (Design Documents) com funções definidas que agregam dados e filtros para serem computados em paralelo.

² **Erlang** é uma linguagem de programação desenvolvida pela Ericsson para suportar aplicações distribuídas.

³ **RESTful** é uma técnica de Engenharia de Software para sistemas hipermídia distribuídos para WWW

⁴ **HTTP/JSON Api** são protocolos de transferência/comunicação de dados, unindo funcionalidades do protocolo HTTP com funcionalidades Javascript

⁵ **RESTful** é uma técnica de Engenharia de Software para sistemas hipermídia distribuídos para WWW.

⁶ **HTTP/JSON Api** são protocolos de transferência/comunicação de dados, unindo funcionalidades do protocolo HTTP com funcionalidades Javascript.

⁷ **MapReduce** é Originalmente feito para reduzir funções e reconstruí-las em uma forma menor e funcional, utilizando mapfunctions que são funções Javascript.

Dentre suas funcionalidades estão à replicação bidirecional com detecção de falhas e resolução de conflitos. Essa base de dados é baseada em princípios de redução de código o chamado MapReduce⁸.

3.1. Características Apache CouchDB

Para Anderson (2009), as principais características do CouchDB são:

- (i) atomicidade;
- consistência;
- isolamento;
- durabilidade;
- confiabilidade;
- tolerante a falhas;
- suporte a replicação.

3.2. Interface de Administração Futon

Após uma análise da interface existente para o gerenciamento do banco de dados CouchDB versão 0.10, chegou-se a conclusão sobre algumas desvantagens apresentadas pela mesma.

- Parte de configurações não funciona muito bem, trava e algumas configurações simplesmente não funcionam na interface é necessário editar arquivos no servidor;
- Controle de usuários é precário, você somente pode adicionar usuários para administradores ou criar usuários com acesso à visualização;
- Gerenciador não funciona no Windows;
- Gerenciador não pode controlar mais que um banco, ou seja, cada servidor deve ter sua interface para controle;
- Todos os bancos criados são visualizados juntos, não a controle por usuários mesmo que administradores.

⁸ **MapReduce**, originalmente feito para reduzir funções e reconstruí-las em uma forma menor e funcional, utilizando mapfunctions que são funções Javascript.

Vendo estas deficiências, se torna válido o estudo de uma nova interface que venha a contribuir com o banco de dados CouchDB em desempenho, agilidade e segurança.

4. Desenvolvimento do protótipo

Para o desenvolvimento do projeto, a linguagem PHP foi selecionada devido às inúmeras vantagens compostas pela mesma, tais como, sua vasta documentação, facilidade de uso e aprendizagem, bem como sua imensa biblioteca de funções.

4.1. PHP

Segundo o Niederauer (2004, p.19), PHP significa (um acrônimo⁹ recursivo para "PHP: Hypertext Preprocessor"). É uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico. “Hoje mais de 10 milhões de sites no mundo inteiro utilizam PHP”.

O PHP teve início por Rasmus Ledorf, que resolveu criar alguns scripts simples para integração com seu site pessoal no ano de 1994. Ledorf queria na verdade somente dar um pouco de vida aos sites estáticos existentes naquele momento, mas ele não imaginava que estava nascendo ali uma das mais poderosas ferramentas para Web, o PHP.

4.2 Usos da Web Service no presente trabalho

No presente trabalho o uso de Web Service se torna primordial para o funcionamento da base de dados uma vez que todas as manipulações são feitas através de chamadas HTTP, usando RESTful/JSON Api.

A Web Service reúne padrões que asseguram a interoperabilidade; estes podem ser acessados via protocolos padrões da Web, tais como: HTTP, HTTPS, entre outros. Os mesmos

⁹ Um **acrônimo** ou sigla é um agrupamento das iniciais de várias palavras. Acrônimos recursivos são acrônimos onde a expansão inclui o próprio termo, como na definição de funções recursivas. PHP: Hypertext Pre-Processor (Originalmente, PHP significava Personal HomePage).

podem ser aplicados a diversas plataformas de integração e suportam tanto aplicações ponto a ponto (estruturas mais simples) quanto aplicações distribuídas (estruturas mais complexas).

Figueiredo (2008, p.5) salienta que “os Web Services são por vezes denominados de serviços de aplicações, sendo serviços que foram disponibilizados por servidores Web para utilizadores Web ou para programas ligados à Web, sendo os seus fornecedores denominados, na generalidade, de Application Service Providers”.

4.3 Características Web Service

Algumas características que podem ser observadas em um Web Service:

- uso intenso de XML;
- grande apoio da indústria, tais como: IBM, Microsoft;
- baseado em padrões abertos, tais como: XML, HTTP, SOAP, WSDL e UDDI;
- uso de URIs¹⁰ para identificação.

4.4 RESTful

O termo REST (Representational State Transfer) vem da dissertação de doutorado de Roy Fielding, publicada em 2000.

Para Fielding (*apud* OLIVEIRA, 2009, p.51), REST é “um conjunto de princípios arquiteturais que quando aplicadas como um todo enfatiza a escalabilidade da interação entre componentes para reduzir a latência de interação, garantir segurança e encapsular sistemas legados”.

Fielding também cunhou o termo RESTful e os classificou com os seguintes princípios:

- cliente servidor;
- apoiar sistemas de cache;
- estado nulo¹¹;
- sistema em camadas, ou seja, suportar escalabilidade;

¹⁰ **URI** - Uniform Resource Identifier é uma cadeia de caracteres compacta usada para identificar ou denominar um recurso na Internet.

¹¹ **Estado nulo** - cada requisição de um cliente ao servidor deve conter todas as informações necessárias para entender a requisição;

- stateless¹².
- sistema uniforme composto por 4 diretivas padronizadas, GET, PUT, POST, DELETE.

4.5 Características RESTful

Algumas características do RESTful:

- atribuir identificação as unidades de recurso – URI;
- utilização de métodos padronizados, GET, PUT, POST, DELETE;
- apresenta recursos com múltiplas representações ou seja como os dados podem ser tratados text/json, text/xml;
- comunicação sem controle de estado, esta define que não deve haver dados da sessão do cliente armazenados no servidor.

4.6 Implementação do protótipo

O protótipo tem por fim gerenciar bancos de dados CouchDB, bem como criar o controle para acesso por usuário sobre estes bancos, além de suprir algumas deficiências encontradas em sua interface de administração original, o Futon.

4.6.1 Preparação do ambiente

Para a preparação do ambiente foram utilizados os seguintes equipamentos:

- 1 Computador Pentium 4, 3.4 ,com sistema operacional Ubuntu 9.10;
- 1 Notebook Acer Aspire 5920 Core 2 Duo, com sistema operacional Windows Xp Pro.

Também foram usados os seguintes softwares:

- software Geany 0.18, este usado para programação PHP, Java Script e edição de arquivos;
- software Xampp 1.7.3, neste software foram utilizados as ferramentas PhpMyAdmin e o Servidor Apache;

¹² **Stateless** - Comunicação sem controle de estado;

- software DBDesigner, este usado para modelagem relacional do protótipo;
- software Apache CouchDB 0.10, este instalado no microcomputador Ubuntu 9.10 com objetivos de testes para o protótipo da interface.

4.6.2 Pré-requisitos para o funcionamento do protótipo

É necessário observar alguns itens para o pleno funcionamento da interface, antes de usá-la. São eles:

- o firewall do servidor deve estar desativado, ou deve conter regras para liberação de leitura e escrita para os clientes;
- o servidor deve conter o Apache CouchDB 0.10 ou superior instalado para funcionamento, uma vez que o protótipo foi construído para uso nestas versões;
- é necessário que o bind_adress do servidor CouchDB esteja setado com o ip “0.0.0.0”, isso fará com que o banco de dados libere acessos a conexões externas;
- é necessário que as duas pontas estejam conectadas minimamente a uma rede;
- a porta deve estar configurada como 5984/tcp no momento de cadastros de novos bancos;

4.6.3 Interface

Abaixo uma das telas correspondentes a interface produzida.

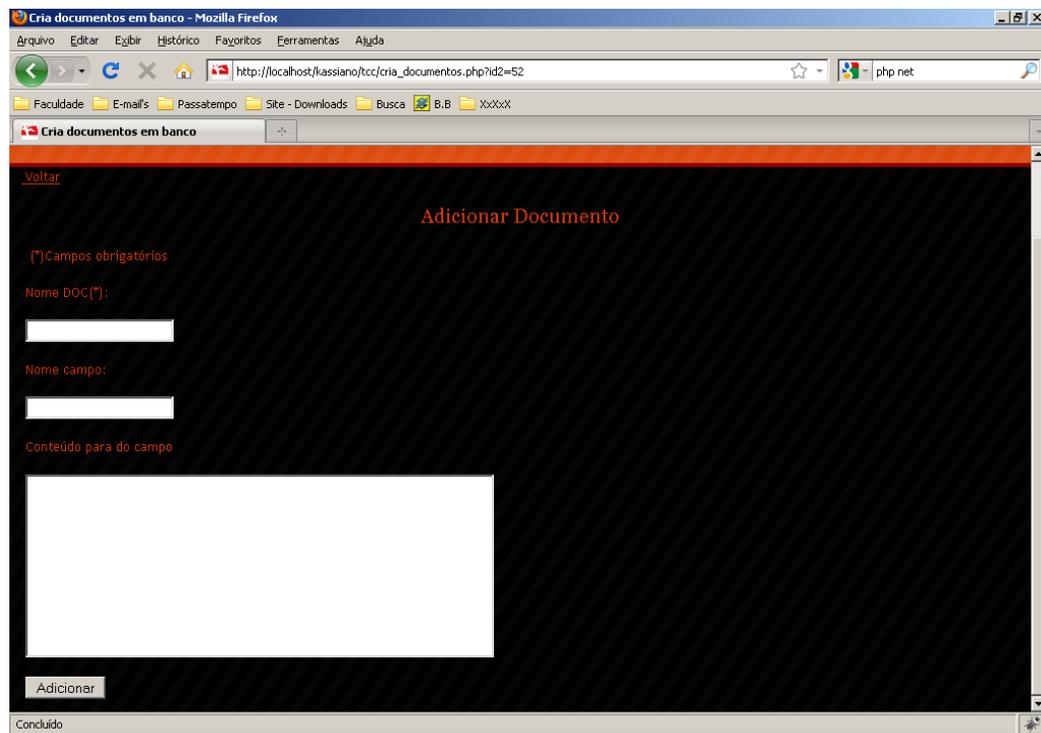


Figura 1: Tela para adicionar documento e atributos.

Nesta Figura 1, o administrador pode cadastrar um documento passando um nome para o documento, um atributo e um valor; as validações são feitas com javascript, codificação pode ser vista na continuidade

```
$curl = curl_init();//Inicia a seção
$aux_conect = "http://".$sip_v.":".$porta_v."/".$bde_v."/".$documento["_id"];
// Remove a chave _id do documento, necessário quando se vai adicionar mais um
//atributo ao documento, caso isso não seja feito o banco retorna um erro de
//conflito
$documento["_id"] = false;
$documento = array_filter($documento);
curl_setopt($curl, CURLOPT_URL, $url);
//Define o tipo do conteúdo das mensagens a serem criadas, no caso text/json
curl_setopt($curl, CURLOPT_HTTPHEADER, array("Content-Type: text/json"));
curl_setopt($curl, CURLOPT_CUSTOMREQUEST, "PUT");
```

```

curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
//Definindo a entrada dos dados ao documento
curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($documento));
$retorno = curl_exec($curl);
curl_close($curl);
$retorno = json_decode($retorno);
//Quando um documento é criado com sucesso ele retorna a mensagem {ok ,true}
//sendo assim faço a verificação abaixo
if(isset($retorno->ok) && $retorno->ok == "true") {
    //retorna mensagem
}

```

Quadro 1: Criando documentos e atributos.

O Quadro 1 corresponde à codificação da criação de documentos e atributos.

4.6.4 Vantagens da interface produzida

Abaixo algumas vantagens correspondentes a nova interface produzida:

- gerenciamento multiplataforma;
- gerenciamento distribuído;
- controle de acesso;
- segurança;
- flexibilidade;
- controle de usuários;
- controle sob os dados com maior facilidade;
- controle de níveis de acesso sobre os bancos aos usuários;
- com adaptações no código pode ser usada para gerenciar outros bancos de dados SGBD NoSQL;

5. Conclusões

O presente trabalho teve como objetivo realizar um protótipo de interface Web com PHP para gerenciamento de banco de dados CouchDB. Mesmo com algumas dificuldades quanto à documentação e suas funcionalidades durante o desenvolvimento do protótipo, pode-se dizer que ao final se obteve êxito.

O CouchDB é um ótimo banco de dados. É um novo paradigma que está crescendo rápido, graças a sua consistência. Sua documentação melhora gradativamente junto com seus grupos de discussões e sites com conteúdos e informações sobre as suas funcionalidades.

No presente momento, em que foi produzido este protótipo, pode-se observar que o esquema de funcionamento RESTful é uma ótima abordagem em comparação com o modelo tradicional Client/Server. Esta abordagem se torna válida devido à grande facilidade de se trabalhar com este modelo de Web Services, no desenvolvimento de aplicações para Web.

O PHP teve papel fundamental no desenrolar do processo, pois combina poderosos mecanismos de programação aliados a sua facilidade de uso e aprendizagem, outro sim se pode observar a evolução obtida pelo acadêmico em técnicas de programação no decorrer do desenvolvimento da interface.

Com este protótipo, o controle da aplicação CouchDB fica muito mais fácil, pois combina controle de usuários com gerência sobre os bancos e suas informações, ficando muito semelhante a ferramentas disponíveis para os SGBDR.

Referências

ANDERSON, Chirs. 2009. **Apache CouchDB: The definitive Guide**. Disponível em: <http://couchdb.apache.org/index.htm> Acessado em 05/06/2009.

BAIN, Tony. **Is the Relational Database Doomed?**, 2009. Disponível em: http://www.readwriteWeb.com/archives/is_the_relational_database_doomed.php. Acessado em: 03/06/2009.

CHANDLER, Christopher. **CouchDB in Action**: Greenwich, Manning, 2009.

FIGUEUREDO, Fernando. **XML E Web Service**. 2008. Disponível em: http://www.ticua.net/Apontamentos/TRABALHO1_FINAL_ASES_E_REIS.pdf. Acessado em 11/06/2010.

LENNON, Joe. *Begining CouchDB*, USA: Apress, 2009.

MORAES, Leonardo. **Introdução ao CouchDB**. Disponível em:
<http://leonardomoraes.com.br/blog/>. Acessado em 06/06/2009

NIEDERAUER, Juliano. **PHP 5**. São Paulo: Novatec Editora LTDA, 2005.

_____. **Desenvolvimento Websites com PHP**: Aprenda a criar Websites dinâmicos e interativos com PHP e bancos de dados. São Paulo: Novatec, 2004.

OLIVEIRA, Leonardo Eloy. **Estado da arte de banco de dados orientados a documento, 2009**. Monografia (Conclusão do Curso de Graduação em Ciências Tecnológicas) – Universidade de fortaleza – UNIFOR, Ceára, 2009.