



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

PERSISTÊNCIA POLIGLOTA

JOSÉ FRANCISCO CAMPOS LIMONGI

Orientador: Evandrino Barros
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG

BELO HORIZONTE
JULHO DE 2014

JOSÉ FRANCISCO CAMPOS LIMONGI

PERSISTÊNCIA POLIGLOTA

Modelo canônico de trabalho monográfico acadêmico
em conformidade com as normas ABNT apresentado à
comunidade de usuários L^AT_EX.

Orientador: Evandrino Barros
 Centro Federal de Educação Tecnológica
 de Minas Gerais – CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
BELO HORIZONTE
JULHO DE 2014

JOSÉ FRANCISCO CAMPOS LIMONGI

PERSISTÊNCIA POLIGLOTA

Modelo canônico de trabalho monográfico acadêmico
em conformidade com as normas ABNT apresentado à
comunidade de usuários \LaTeX .

Trabalho aprovado. Belo Horizonte, 24 de novembro de 2014

Evandrino Barros
Orientador

Professor
Convidado 1

Professor
Convidado 2

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
CURSO DE ENGENHARIA DE COMPUTAÇÃO
BELO HORIZONTE
JULHO DE 2014

Espaço reservado para dedicatória. Inserir
seu texto aqui...

Agradecimentos

Inserir seu texto aqui... (esta página é opcional)

“Seja realista, exija o impossível.” (Roland Castro)

Resumo

Síntese do trabalho em texto cursivo contendo um único parágrafo. O resumo é a apresentação clara, concisa e seletiva do trabalho. No resumo deve-se incluir, preferencialmente, nesta ordem: brevíssima introdução ao assunto do trabalho de pesquisa (qualificando-o quanto à sua natureza), o que será feito no trabalho (objetivos), como ele será desenvolvido (metodologia), quais serão os principais resultados e conclusões esperadas, bem como qual será o seu valor no contexto acadêmico. Para o projeto de dissertação sugere-se que o resumo contenha até 200 palavras.

Palavras-chave: latex. abntex. modelo. (Entre 3 a 6 palavras ou termos, separados por ponto, descritores do trabalho. As palavras-chaves são Utilizadas para indexação.

Abstract

Translation of the abstract into english, possibly adapting or slightly changing the text in order to adjust it to the grammar of english educated.

Keywords: latex. abntex. template.

Lista de Figuras

Lista de Tabelas

Lista de Quadros

Lista de Algoritmos

Lista de Abreviaturas e Siglas

ABNT	Associação Brasileira de Normas Técnicas
DECOM	Departamento de Computação

Lista de Símbolos

Γ	Letra grega Gama
λ	Comprimento de ondada
\in	Pertence

Sumário

1 – Introdução	1
1.1 Motivação	2
2 – Trabalhos Relacionados	3
2.1 Citações livres	4
2.2 Citações literais	4
2.3 Informações sobre as referências utilizadas	6
3 – Fundamentação Teórica	7
3.1 Banco de Dados	7
3.2 Gêneros de Banco de dados	8
3.2.1 Banco de dados Relacional	9
3.2.2 Banco de dados não relacional	9
3.2.2.1 MongoDB	10
3.2.2.2 Redis	10
3.3 Persistência Poliglota	10
3.4 Implementação	10
3.5 Implementação Monoglota	10
3.6 Implementação Poliglota	10
4 – Metodologia	11
4.1 Implementação	11
4.2 Implementação Monoglota	11
4.3 Implementação Poliglota	11
4.4 Coleta de dados	11
5 – Análise de Resultados	12
5.1 Situação atual	12
5.2 Análise dos dados coletados	12
6 – Conclusão	13
6.1 Trabalhos futuros	13
Apêndices	14
APÊNDICE A–Nome do Apêndice	15

APÊNDICE B–Nome do Apêndice	16
 Anexos	 17
ANEXO A–Nome do Anexo	18
ANEXO B–Nome do Anexo	19

1 Introdução

A necessidade de persistência de dados sempre esteve presente na computação. A medida que os sistemas evoluíram a complexidade da forma que os dados eram armazenados aumentou significativamente. Com isso houve a necessidade da criação de um sistema computadorizado de manutenção de registros (??), o banco de dados.

O modelo relacional foi um dos primeiros gêneros de banco de dados, sua estrutura são tabelas de duas dimensões com linhas e colunas. Os dados armazenados são tipados podendo variar a quantidade de tipos de acordo com o banco utilizado. Para interagir com esse gênero de banco é necessário realizar consultas com a linguagem SQL. Alguns exemplos de banco de dados relacional são MySQL, SQLite, Oracle e PostgreSQL.

Durante anos, o banco de dados relacional tinha sido considerado a melhor opção para os problemas de pequena e grande escalabilidade, porém surgiram novas soluções com novas alternativas de estruturas, replicação simples, alta disponibilidade e novos métodos de consultas (??). Essas opções são conhecidas como NoSQL ou banco de dados não-relacional

Existem diversos gêneros de banco de dados não-relacional, entre eles chave-valor, orientado à documento, orientado à coluna e orientado à nó. Com o surgimento dessas novas soluções, o questionamento sobre qual banco de dados é melhor para resolver certo tipo de problema, veio à tona. A partir disso, o conhecimento e compreensão sobre os bancos de dados em geral se torna necessário para realizar uma boa escolha.

Entendendo que cada banco se destaca em determinados tipos de problema, é nítido perceber que sistemas que trabalham com mais de um banco de dados podem oferecer um melhor desempenho, dando origem à persistência poliglota.

Este trabalho consiste na comparação de dois sistemas, o primeiro um que utilizará apenas um banco de dados e o segundo que utilizará dois bancos de dados ou persistência poliglota. Os bancos escolhidos para fazer esses sistemas foram o MongoDB e o Redis. O autor escolheu esses bancos de dados por ter experiência na linguagem *Ruby on Rails* que oferece excelentes bibliotecas para esses bancos. O intuito desse trabalho é comprovar que o uso da persistência poliglota melhora o desempenho da aplicação.

MongoDB é um banco de dados do gênero orientado à documento e foi desenhado para ser gigante, como o próprio nome é uma derivação da palavra inglesa *humongous* que significa gigantesco. Além disso, tem soluções interessantes para evitar

leitura suja e as consultas são realizadas na linguagem JavaScript. A diferença desse gênero é que são armazenados documentos compostos de um identificador único e um conjunto de valores de tipos e estruturas aninhadas, chamadas de BSON, uma estrutura parecida com JSON. Esse gênero é bem flexível, pois não tem *schema*, ou seja, não existe tabelas. A organização dos dados é feito por documentos, ao criar a arquitetura do sistema temos que identificar se as entidades criadas são expressivas como um documento (??). Nesse trabalho iremos utilizar o banco MongoDB nos dois sistemas que serão criados. O MongoDB está sendo utilizado em grandes empresas, como Cisco, eBay, Codecademy, Microsoft, New York Time, Craigslist, The Guardian e outras referencia site mongo??.

O segundo banco que iremos utilizar se chama Remote Dictionary Server ou Redis do gênero chave-valor. Esse tipo de armazenamento é mais simples, como próprio nome indica, é armazenado um valor para determinada chave. Essa escolha foi feita devido ao cache que esse sistema realiza antes de efetivar a operação no disco. Esse cache tem um ganho muito alto em performance, porém poderá ocorrer perda de dados, caso ocorra uma falha de hardware (??). A forma de como estruturar esse banco é muito parecida com um tipo estruturado chamado *hash* que são implementadas em algumas linguagens de computação, como Java e Ruby. Esse banco será utilizado no segundo sistema a ser desenvolvido. O Redis está sendo utilizado no Twitter, Github, Craigslist e outros referencia site REDIS.

1.1 Motivação

A persistência poliglota é uma alternativa para melhorar o desempenho de uma aplicação. Utilizando-a conseguimos adaptar cada tipo de problema com um gênero de banco. Em uma mesma aplicação podemos ter um conjunto dados que devem estar sempre disponíveis e em outra parte da aplicação um conjunto de dados que devem ser consistentes. Então podemos separar os dados dessa aplicação em bancos com gêneros diferentes, cada um irá armazenar os dados, no qual o seu gênero se destaca.

2 Fundamentação Teórica

Para entendermos o porquê de utilizar mais de um banco de dados em uma mesma aplicação, temos que entender o que é um banco de dados, quais gêneros existem e para que tipo de problema cada gênero se destaca.

2.1 Banco de Dados

Banco de dados é um sistema computadorizado de manutenção de registros, análogo à um armário de arquivamento eletrônico. Podemos entender como um repositório para manter a coleção de arquivos de dados computadorizados (??, p.3). ??, p.4) define banco de dados como uma coleção de dados relacionados e que dados são fatos com um significado implícito. Porém a definição de ??, p.4) é muito abrangente, logo ele aponta três propriedades implícitas para restringir a definição de banco de dados.

A primeira propriedade é que um banco de dados deve representar alguns aspectos do mundo real, chamado de *universo de discurso* (UoD). As alterações que ocorrem nesse universo são refletidas em um banco de dados.

A segunda propriedade define que o banco de dados é uma coleção lógica e coerente de dados com algum significado inerente, ou seja, uma coleção de dados randômicos não pode ser considerado um banco de dados.

A terceira propriedade afirma que banco de dados é projetado, construído e povoado com dados, atendendo a uma proposta específica. Além disso, possui um grupo de usuários definido e algumas aplicações preconcebidas, de acordo com o interesse desse grupo.

Os bancos de dados tem contribuído para o aumento do uso do computador (??, p.3) e podemos afirmar que eles apresentam um papel crucial em quase todas as áreas em que os computadores são utilizados. Devido a essa importância o estudo sobre banco de dados é extremamente necessário para os profissionais da computação.

Antes da existência dos bancos de dados a aplicação devia gerenciar e processar arquivos para manter os dados persistidos. Para justificar o uso do banco de dados, ??, p.7) cita quatro características: natureza autodescritiva, abstração de dados, suporte para as múltiplas visões de dados e por fim compartilhamento de dados e processamento de transações de multiusuários.

O banco de dados, relacional, é autodescritivo devido ao catálogo que permite identificar a estrutura dos arquivos, formato e tipo de dados. Já ao utilizar o processamento tradicional dos arquivos essas definições de estrutura estarão na própria

aplicação. Isso dificulta outro programa utilizar a mesma base de dados.

Como identificado por ??, p.7) a abstração de dados não é feita no processamento tradicional de arquivos. A aplicação define a estrutura de dados. Suponha que tenhamos diversos programas utilizando o mesmo arquivo para armazenar uma coleção de dados. Se um desses programas precisar de acrescentar algum campo novo, todos os outros programas que acessam esse arquivo, devem modificados para contemplar o novo campo adicionado. Já quando utilizamos banco de dados, a alteração da estrutura dos dados pode não influenciar no funcionamento dos outros programas.

Em relação a característica, suporte para múltiplas visões dos dados, ??, p.7) afirma que quando é utilizado o banco de dados, é possível ter diferentes visões sobre os dados, fazendo o cruzamento das informações. Com a abordagem de processamento de arquivo tradicional isso não é usual.

A última característica comparada por ??, p.7) é o compartilhamento de dados e o processamento de transação multiusuários, essa característica é essencial para que várias aplicações possam acessar e alterar os dados. Porém o sistema de gerenciamento do banco de dados deve ter implementado um controle de concorrência para garantir a atomicidade das transações.

??) não cita a existência de bancos de dados sem catálogo, chamados de *schemaless*. Apesar de não ter a declaração do tipo de estruturas de dados contidas no banco, os bancos de dados *schemaless* faz a abstração dos dados da mesma maneira que os bancos de dados tradicionais, tem suporte para múltipla visões e multiusuários.

Como revelado acima a utilização do banco de dados, facilita o desenvolvimento das aplicações, faz a abstração entre aplicação e dados e além disso faz o controle de concorrência. Após verificarmos que o uso de banco de dados é imprescindível, nos deparamos com uma outra dificuldade, qual banco de dados utilizar. Os bancos de dados, chamados de NoSQL, chamou a atenção da comunidade científica, depois da publicação de dois artigos ??) e ??)

2.2 Gêneros de Banco de dados

Durante anos, o banco de dados relacional tinha sido considerado a melhor opção para a maioria dos problemas sendo de pequena ou grande escalabilidade. O aumento do volume de dados fizeram com que os especialistas buscassem novas soluções que permitissem o armazenamento distribuído dos bancos de dados e que fossem mais eficazes e simples que o relacional (??).

Com isso surgiram novos gêneros de banco de dados que foram denominados de NoSQL. Carlo Strozzi foi o primeiro a utilizar o nome NoSQL, mas não no sentido

que a palavra tem hoje. Strozzi denominou um banco de dados relacional, open source de NoSQL, pois não usava SQL como linguagem de consulta. O nome, era de uma conferência, realizada em São Francisco nos Estados Unidos em Junho de 2009. Johan Oskarsson que organizou essa conferência escolheu esse nome porque queria que fosse uma boa hashtag no Twitter: pequeno, memorável e tivesse poucos resultados no Google. Isso facilitaria os interessados a encontrar a conferência. Apesar de o termo não significar explicitamente o que são esses bancos de dados atendeu bem a intenção de Oskarsson. Os bancos chamados de NoSQL tem em comum: não usar o modelo relacional, funciona bem em clusters, são open source e não tem catálogos (*schemaless*) (??).

2.2.1 Banco de dados Relacional

O modelo relacional é o mais comum atualmente, esse gênero armazena os dados em tabelas de duas dimensões, linhas e colunas. A interação com esse banco é feito por um SGBD (Sistema de Gerenciamento de Banco de Dados) que utiliza o SQL como linguagem de consulta de dados. Os dados armazenados são valores tipados e podem ser numéricos, texto, data e outros tipos, que são configurados e forçados pelo sistema. A grande vantagem do modelo relacional é a facilidade que pode ser feito as consultas. É possível fazer relações entre tabelas, que se transformam em outras tabelas mais complexas. MySQL, H2, HSQLDB, SQLite e PostgreSQL são alguns exemplos de banco de dados relacional (??). A representação do modelo relacional, normalmente é feito com o diagrama de entidade relacionamento, como pode ser visto na figura 1 e a forma que os valores ficam armazenados nas tabelas, pode ser visto na figura 2.

O gênero relacional funciona muito bem para diversas aplicações, pois é bem flexível em relação as consultas, permite concorrência, transações e pode ser integrado com várias aplicações. Porém há uma desvantagem que causa frustração em muitos desenvolvedores, chamada de Impedância de correspondência ou *Impedance Mismatch*. Isso ocorre, pois nem sempre o tipo do campo no banco de dados irá corresponder com o tipo esperado da linguagem utilizada, então é necessário criar uma forma de associação entre o tipo da variável da linguagem com o tipo do valor da tabela. Outra desvantagem é que esse gênero não aceita valores multivalorados, diferenciando a aplicação ainda mais do modelo relacional.

2.2.2 Banco de dados não relacional

Os bancos de dados não relacional ou NoSQL, foram construídos para suprir a necessidade de se trabalhar com grande quantidade de dados e em clusters. Também podemos dizer que o resultado mais importante do crescimento do NoSQL é a persistência poliglota (??).

Uma das principais diferenças desse tipo de banco sobre o relacional é que permite agregação

2.2.2.1 MongoDB

Falar do mongo, basear na documentação para explicar como funciona as relações e citar alguns estudos de caso do site

2.2.2.2 Redis

Falar do redis, basear no NoSQL e SDSW, mostrar alguns estudos de casos que estão no site

2.3 Persistência Poliglota

Utilizar o NoSQL como base e fazer a comparação com o artigo de paradigma de programação

3 Metodologia

Este trabalho será realizado em três etapas.

A primeira etapa consiste do estudo dos bancos de dados escolhidos, buscando entender melhor como os dados são estruturados. Ainda nessa etapa, será feito uma pesquisa bibliográfica sobre sistemas que utilizam persistência poliglota. Essa pesquisa será fundamental para a próxima etapa, pois irá direcionar a criação da aplicação para que ressalte as diferenças de um sistema com persistência poliglota.

A segunda etapa consiste da criação de duas aplicações semelhantes ao Twitter, porém uma utilizará apenas um banco de dados e a outra utilizará dois bancos de dados. Essa aplicação terá as funcionalidades básicas do Twitter.

A terceira e última etapa é a comparação da performance das operações sobre os dados dessas aplicações. Logo, consiste na medição de tempo gasto para inserção, leitura, atualização e exclusão dos dados.

Para a realização desse trabalho será necessário de um computador com os servidores dos bancos de dados instalados.

3.1 Implementação

Explicar a aplicação, como foi feito e apresentar os casos de uso

3.2 Implementação Monoglota

Apresentar os diagrama de classe da parte persistida e da aplicação como um todo, falar do framework MongoId, explicar a forma com que a associacao dos models

3.3 Implementação Poliglota

Apresentar os diagrama de classe da parte persistida e da aplicação como um todo, falar do framework do Redis, explicar onde foi usado um banco e onde foi usado outro.

3.4 Coleta de dados

Explicar como foi coletado os dados

4 Análise de Resultados

Inserir seu texto aqui...

4.1 Situação atual

Inserir seu texto aqui...

4.2 Análise dos dados coletados

Inserir seu texto aqui...

5 Conclusão

Espera-se que o uso do estilo de formatação LATEX adequado às Normas para Elaboração de Trabalhos Acadêmicos do CEFET-MG (`abntex2-cefetmg.cls`) facilite a escrita de documentos no âmbito desta instituição e aumente a produtividade de seus autores. Para usuários iniciantes em LATEX, além da bibliografia especializada já citada, existe ainda uma série de recursos (??) e fontes de informação (???) disponíveis na Internet.

Recomenda-se o editor de textos Kile como ferramenta de composição de documentos em LATEX para usuários Linux. Para usuários Windows recomenda-se o editor TEXnicCenter (??). O LATEX normalmente já faz parte da maioria das distribuições Linux, mas no sistema operacional Windows é necessário instalar o software MiKTeX (??).

Além disso, recomenda-se o uso de um gerenciador de referências como o JabRef (??) ou Mendeley (??) para a catalogação bibliográfica em um arquivo BIBTEX, de forma a facilitar citações através do comando `\cite{}` e outros comandos correlatos do pacote ABNTEX. A lista de referências deste documento foi gerada automaticamente pelo software LATEX + BIBTEX a partir do arquivo `refbase.bib`, que por sua vez foi composto com o gerenciador de referências JabRef.

5.1 Trabalhos futuros

Inserir seu texto aqui...

Apêndices

APÊNDICE A – Nome do Apêndice

Inserir seu texto aqui...

APÊNDICE B – Nome do Apêndice

Inserir seu texto aqui...

Anexos

ANEXO A – Nome do Anexo

Inserir seu texto aqui...

ANEXO B – Nome do Anexo

Inserir seu texto aqui...