



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

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 Trabalhos Relacionados

Este capítulo inclui muitas citações bibliográficas. Os principais itens de bibliografia citados são livros, artigos em conferências, artigos em *journals* e páginas Web. A bibliografia deve seguir o padrão ABNT¹.

A bibliografia é feita no padrão `bibtex`. As referências são colocadas em um arquivo separado. Os elementos de cada item bibliográfico que devem constar na bibliografia são apresentados a seguir.

Para livros, o formato da bibliografia no arquivo fonte é o seguinte:

```
@Book{linked,
  author = {A. L. Barabasi},
  title = {Linked: The New Science of Networks},
  publisher = {Perseus Publishing},
  year = {2002},
}
```

A citação deste livro se faz da seguinte forma `\cite{linked}` e o resultado fica assim (??). Para os artigos em *journals*, veja por exemplo (??), descrito da seguinte forma no arquivo `.bib`:

```
@article{acmsurveys,
  author    = {Deepayan Chakrabarti and Christos Faloutsos},
  title     = {Graph mining: Laws, generators},
  journal   = {ACM Computing Surveys},
  volume    = {38},
  number    = {1},
  year      = {2006},
  pages     = {2-59},
  publisher = {ACM},
  address   = {New York, NY, USA},
}
```

O artigo (??) foi publicado em conferência. Embora às vezes seja difícil distinguir um artigo publicado em *journal* de um artigo publicado em conferência, esta distinção é fundamental. Em caso de dúvida, procure ajuda de seu orientador.

¹ Este não é o endereço oficial da ABNT pois as Normas Técnicas oficiais são pagas e não estão disponíveis na Web.

Veja também duas citações juntas (????) e como citar endereços Web (??). O trabalho realizado para editar as citações no formato correto é compensado por uma bibliografia impecável.

2.1 Citações livres

Citações são trechos transcritos ou informações retiradas das publicações consultadas para a realização do trabalho. As citações são utilizadas no texto com o propósito de esclarecer, completar, embasar ou corroborar as ideias do autor.

Todas as publicações consultadas e efetivamente utilizadas (através de citações) devem ser listadas, obrigatoriamente, nas referências bibliográficas, de forma a preservar os direitos autorais e intelectuais.

Na utilização de citações, normalmente, utiliza-se referências. Para cada tipo de referência presente no texto será apresentado um exemplo do comando utilizado para criá-lo.

Há basicamente dois tipos de citações: citações livres e citações literais.

Nas citações livres, reproduzem-se as ideias e informações de um autor, sem, entretanto, “copiar letra por letra” o texto do autor. Há várias maneiras de se fazer uma citação livre, como mostra os exemplos abaixo.

Por outro lado, ??) defende um princípio de lógica. Para o autor, quando dizemos ...

Além disso, ??) argumenta que ... Observe o detalhe do termo *et al.* que deve ser utilizado quando o trabalho citado possui mais de três autores. Esse recurso é automatizado pelo estilo `abntex2`. Caso não haja desejo em abreviar o nome dos demais autores através do termo *et al.*, deve-se incluir a opção `abnt-no-et-al-label`.

Para evitar uma interrupção na sequência do texto, o que poderia, eventualmente, prejudicar a leitura, pode-se indicar a fonte entre parênteses imediatamente após a citação livre. Porém, neste caso específico, o nome do autor deve vir em caixa alta, seguido do ano da publicação, como no exemplo a seguir.

A física, então, constituiu-se como a prova mínima da efetividade do método científico para descobrir as verdades do universo (????).

2.2 Citações literais

Nas citações literais, reproduzem-se as ideias e informações de um autor, exatamente como este a expressou, ou seja, faz-se uma “cópia letra por letra” do texto do

autor. Há várias maneiras de se fazer uma citação literal, como mostra os exemplos abaixo.

As citações longas (mais de 3 linhas) devem usar um parágrafo específico para ela, na forma de um texto recuado (4 cm da margem esquerda), com tamanho de letra menor do aquela utilizada no texto e espaçamento simples entre as linhas, seguido dos sobrenomes dos autores em caixa alta (separados por ponto e vírgula), ano de publicação e número da página. Veja o exemplo abaixo.

Desse modo, opera-se uma ruptura decisiva entre a reflexividade filosófica, isto é a possibilidade do sujeito de pensar e de refletir, e a objetividade científica. Encontramo-nos num ponto em que o conhecimento científico está sem consciência. Sem consciência moral, sem consciência reflexiva e também subjetiva. Cada vez mais o desenvolvimento extraordinário do conhecimento científico vai tornar menos praticável a própria possibilidade de reflexão do sujeito sobre a sua pesquisa (??, p. 28).

Para se criar o efeito demonstrado na citação anterior, deve-se utilizar o comando:

```
\begin{citacao}  
  <citacao>  
\end{citacao}
```

Opcionalmente, pode-se referenciar os autores no corpo de texto (neste caso seus nomes devem vir em minúsculas), e em seguida colocar a citação literal, em um novo parágrafo recuado. Note que pode após a citação literal não mais aparece o nome dos autores, visto que já se encontra no texto. Veja o exemplo seguinte.

??, p. 33), ao fazerem as suas críticas à ciência, explicitam uma ideia coletiva:

Mas o curioso é que o conhecimento científico que descobriu os meios realmente extraordinários para, por exemplo, ver aquilo que se passa no nosso sol, para tentar conceber a estrutura das estrelas extremamente distantes, e até mesmo para tentar pesar o universo, o que é algo de extrema utilidade, o conhecimento científico que multiplicou seus meios de observação e de concepção do universo, dos objetos, está completamente cego, se quiser considerar-se apenas a si próprio!

As citações curtas (menos de 3 linhas) devem ser inseridas diretamente no texto (entre aspas), seguida do nome do autor (em caixa alta), ano e página, como no exemplo a seguir.

Então significa apenas que “assumo que não posso fazer referência a entidades independentes de mim para construir meu explicar” (??, p. 35).

O conhecimento de ??, p. 35) aponta que isto significa apenas que “assumo que não posso fazer referência a entidades independentes de mim para construir meu explicar”.

Finalmente, e isto vale para citações curtas ou longas, caso seja necessário inserir, no meio de uma citação uma palavra ou frase curta de sua autoria, que sirva para clarear ou completar a frase do autor citado, isto deve ser feito colocando a citação entre aspas. O comentário deverá ser inserido sem aspas. Ou seja, todo texto da citação deverá ficar envolvido por aspas. O exemplo abaixo apresenta o resultado esperado.

Significa apenas que “assumo que não posso fazer referência a entidades” objetivas no sentido tradicional “independentes de mim para construir meu explicar” ??, p. 35).

2.3 Informações sobre as referências utilizadas

Nesta seção serão apresentadas os comandos necessários para a criação das referências utilizadas anteriormente. As informações serão apresentadas da seguinte maneira:

- ??)
`\citeonline{maturana:2003}`
- ??)
`\citeonline{teste:2004}`
- (??, p. 28)
`\cite[p.~28]{morinmoigne:2000}`
- ??, p. 33)
`\citeonline[p.~33]{morinmoigne:2000}`
- (??, p. 35)
`\cite[p.~35]{maturana:2003}`
- ??, p. 35)
`\citeonline[p.~35]{maturana:2003}`
- (????)
`\cite{teste:2004,maturana:2003}`

3 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.

3.1 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 ??)

3.2 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*) (??).

3.2.1 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.

3.2.2 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

3.2.2.1 MongoDB

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

3.2.2.2 Redis

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

3.3 Persistência Poliglota

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

3.4 Implementação

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

3.5 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 associação dos models

3.6 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.

4 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.

4.1 Implementação

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

4.2 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

4.3 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.

4.4 Coleta de dados

Explicar como foi coletado os dados

5 Análise de Resultados

Inserir seu texto aqui...

5.1 Situação atual

Inserir seu texto aqui...

5.2 Análise dos dados coletados

Inserir seu texto aqui...

6 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.

6.1 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...